Effective Methods of Semantic Analysis in Spatial Contexts

Raimundo F. Dos Santos Jr.

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science and Applications

Chang-Tien Lu, Chair
Naren Ramakrishnan
Ing-Ray Chen
Jason Xuan
Grace Hui Yang

May 19, 2014
Falls Church, Virginia

Keywords: Ontologies, Spatial Data, Information Retrieval, Hierarchical Structures

Effective Methods of Semantic Analysis in Spatial Contexts

ABSTRACT

Raimundo F. Dos Santos Jr.

With the growing spread of spatial data, exploratory analysis has gained a considerable amount of attention. Particularly in the fields of *Information Retrieval* and *Data Mining*, the integration of data points helps uncover interesting patterns not always visible to the naked eye. Social networks often link entities that share places and activities; marketing tools target users based on behavior and preferences; and medical technology combines symptoms to categorize diseases. Many of the current approaches in this field of research depend on semantic analysis, which is good for inferencing and decision making.

From a functional point of view, objects can be investigated from a spatial and temporal perspectives. The former attempts to verify how proximity makes the objects related; the latter adds a measure of coherence by enforcing time ordering. This type of spatio-temporal reasoning examines several aspects of semantic analysis and their characteristics: shared relationships among objects, matches versus mismatches of values, distances among parents and children, and brute-force comparison of attributes. Most of these approaches suffer from the pitfalls of disparate data, often missing true relationships, failing to deal with inexact vocabularies, ignoring missing values, and poorly handling multiple attributes. In addition, the vast majority does not consider the spatio-temporal aspects of the data.

This research studies semantic techniques of data analysis in spatial contexts. The proposed solutions represent different methods on how to relate spatial entities or sequences of entities. They are able to identify relationships that are not explicitly written down. Major contributions of this research include (1) a framework that computes a numerical entity similarity, denoted a *semantic footprint*, composed of spatial, dimensional, and ontological facets; (2) a semantic approach that translates categorical data into a numerical score, which permits ranking and ordering; (3) an extensive study of GML as a representative spatial structure of how semantic analysis methods are influenced by its approaches to storage, querying, and parsing; (4) a method to find spatial regions of high entity density based on a clustering coefficient; (5) a ranking strategy based on connectivity strength which differentiates important relationships from less relevant ones; (6) a distance measure between entity sequences that quantifies the most related streams of information; (7) three distance-based measures (one probabilistic, one based on spatial influence, and one that is spatio-logical) that quantifies the interactions among entities and events; (8) a spatio-temporal method to compute the coherence of a data sequence.

Acknowledgments

First and foremost, I would like to thank my advisor, Dr. Chang-Tien Lu, for his support during several years of Ph.D. study. I consider myself fortunate to be under his guidance from whom I learned the necessary skills to conduct rigorous research. From Dr. Lu, I have also acquired a strong sense of responsibility to be a serious scientist, and resiliency to seek the truth.

I am very thankful to all the committee members. Without their guidance, I would not have been able to complete my dissertation. Thanks to Dr. Ramakrishnan for offering his deep expertise in writing and publishing. Thanks to Dr. Yang and Dr. Xuan for providing insightful comments and valuable suggestions from my preliminary proposal to the final defense. Their knowledge of research pointed me in alternative directions that I might not recognize otherwise. Special thanks goes to Dr. Chen, who made himself promptly available not only for research questions, but also for correct policies and regulations.

I would like to thank the unwavering support of two dear colleagues. Dr. Feng Chen provided me continuous support for over five years and helped me understand valuable techniques that became part of my thesis. Dr. Arnold Boedihardjo worked closely with me at the U.S. Army Corps of Engineers, and was fundamental in showing me how to perform technically-sound research that can solve real-world problems.

I would like to express appreciation to my friends in the Spatial Data Management Laboratory: Kevin Lu, Sumit Shah, Manu Shukla, Bingsheng Wang, Haili Dong, Kaiqun Fu, Ting Hua, and Liang Zhao. Over many years of seminars, their precious comments fostered new thoughts that helped shape my thesis. In the course of my Ph.D. study, they made the journey enjoyable with many happy memories.

Finally, I would like to dedicate a special thanks to my mother, sister, and brothers. Their words of encouragement and support helped me endure challenging times, and gave me strength to successfully complete the program.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

As the use of spatial information increases, understanding the similarities and relationships among data points has become an important, but challenging task. Law enforcement agencies may want to link crime events, while environmentalists may want to find common sources of pollution. The implications are far reaching (e.g., legal, financial, cultural), driving exploratory analysis to look beyond traditional syntactic methods of analysis. The next stage has been to adapt spatio-temporal models into semantic techniques able to assert more robust understanding of the underlying data.

In order to study a group of entities and their relationships described in context (e.g., in a news story), a prevailing approach is to select a set of semantic concepts and apply them to the data of interest. In general, semantic analysis uses these concepts in an attempt to uncover some underlying knowledge of the entities, a goal that is shared by traditional syntactic approaches. In contrast to the syntactic view, however, semantic analysis reaches beyond the mechanical aspects of the data, aiming to elicit knowledge not explicitly spelled out. This differentiation helps to ensure that results are not simply copied and displayed, but rather, inferred and linked. For example, in the context of crime data, *auto theft* can be related to many other events such as *armed robberies* even when no immediate connections among them are apparent. When combined with aspects of space and time, semantic analysis can possibly better identify these connections and infer the type of knowledge that makes their interactions truly relevant in the real world.

1.1 Motivation

In a broad sense, data analysis comes in two flavors: *quantitative* and *qualitative* techniques. Quantitative approaches walk through numerical reasoning to explain facts. As such, they benefit significantly from descriptions that are explicitly typed in unambiguous format. Frequencies of objects, variances throughout the dataset, and averages of important attributes are common examples. Moreover, quantitative approaches are also effective in estimating values when hard numbers cannot be determined. Probabilistic methods and fuzzy approaches are two examples. Because they

are strongly tied to **content**, quantitative methods tend to be syntactic by nature.

In qualitative approaches, the focus is not necessarily what the data values are, but rather how they relate to one another. Some examples are ontological structures where elements can be part of other elements, or spatial grids that describe a neighborhood. Because qualitative methods focus on **structure** and **relationships**, they tend to lean more semantic.

In this proposal, both quantitative and qualitative data analyses are described, one often in conjunction with the other. The reason to combine them is two-fold: first, in real-life applications, the division between quantitative and qualitative can be blurred to a point that neither one may be deemed more semantic or syntactic than the other; second, the two approaches are complementary to one another, contributing knowledge exclusive to their respective domains. Using this as a motivation, both structure and content are leveraged as sources of information.

Structure and content must be considered in the realm of spatial entities, which possess a **location** element along with a **time** of observation, and one or more non-spatial **attributes**. Further, **relationships** among entities can be given explicitly or may be derived implicitly. In general, these four concepts (i.e., locations, time, attributes, and relationships) form the cornerstone of spatial data analysis. More specifically, they are analyzed under many lights: how they compare to one another; in what sense they are linked; what their relationship types are; how similar their attributes may be. As a consequence, proper data analysis demands careful knowledge of the underlying **context**. For instance, in web marketing, it is important to relate online purchases to a user's click behavior. Or in medical data, where it may be useful to understand how disease spreads under changing weather events.

While very useful, both semantic and syntactic analysis are hindered by some of the same challenges seen in traditional *Information Retrieval*: ambiguous meanings, contradictory terms, disparate vocabularies, and missing or noisy data, among others. These challenges impact one's ability to understand how spatial entities are tied to one another, i.e, the ability to effectively measure **similarity**, **likeness**, and **relatedness**, while working around the issues. Some concrete examples that explore entity analysis include: a graph-based outlier detection that recursively eliminates entities in a spatial context [65]; a framework that links entities based on relationships in a connected network [50]; and an incident detection system in Internet traffic [14].

In summary, data analysis in spatial contexts represents a crucial step in a wide range of real-life applications. Ultimately, its goal is to extract knowledge from both spatial and non-spatial components while maximizing the effectiveness of semantic and syntactic approaches. Employing an improper representation can result in meaningless outcomes and promote severely erroneous decision making.

1.2 Research Issues

Traditional data analysis has relied on **syntactic** information (i.e., data types and formats) across heterogeneous data sources [37]. As data exchange among disparate systems grows, however, purely syntactic approaches to data analysis have proven inefficient and often erroneous. The underlying reason is that, at an abstract level, information processing has evolved into the **semantic** realm. It must now encompass not only mechanical parts, but also the true meaning of entities, their attributes, and relationships. Incorporating *semantics* brings about several challenging issues:

❶ **Numerical applicability**. Measuring similarity based on categorical (i.e., non-numerical) attributes. Ontological data are grouped in levels for which distance or ordering between any two elements is often arbitrary. For example, there is no explicit distance between pollutants whose main constituents are based on *carbon*, *lead*, and *nickel*. Assigning similarity among them is non-intuitive;

❷ **Massive numbers of entities**. Entities are commonly described in terms of people, organizations, events, and objects. In many datasets, they come as millions of instances, which complicate handling and reasoning. As a consequence, there exists a strong need for methods that can lower the number of investigated entities.

❸ **Explosive number of potential connections**. Entities may be connected for many different reasons: because of colocation, due to sharing of attributes, related by the same document, etc. No single heuristic establishes what a good connection should be. For this reason, ranking is needed based on what one would consider an important connection versus an irrelevant one.

❹ **Lack of semantic similarity measures for entity sequences**. In many applications, individual entities are of little value. Their true expressiveness appears when they are linked into meaningful sequences that form a story. This linkage necessitates measures of similarity for the entire story, for which little has been proposed in spatio-temporal settings.

The above issues add uncertainty into the process of semantic analysis that would normally be simpler in a solely syntactic approach. The objective of this report is to investigate and present techniques to address parts of the above challenges. The contributions are listed below.

1.3 Contributions

Data analysis in spatial contexts requires the examination of information from various perspectives. The ontological space relays structural information, class distances, and entity relationships. The dimensional space provides attributes for comparison and similarity. A hybrid of ontological and dimensional spaces allows categorization via numerical analysis. And not unrelated, but

complementary to the above, other factors such as temporal considerations, multi-attribute and multi-resolution data must also be accounted for. In line with these items, the major proposed research contributions are as follows:

A Spatial similarity in multidimensional spaces. Spatial entities are often rich in dimensions that can be incorporated in analytical tasks. This task targets those attributes deemed important for a given application, and applies them in determining entity relationships.

- **A1 Devise a dimensional similarity based on shared attributes**. When an entity is described with several dimensions, some of those dimensions can often be observed in other entities as well. This fact underscores the need to include shared attributes in the computation of dimensional similarity, which is introduced in this research.

- **A2 Resolve spatial affinity**. For many entities, spatial location may not be known. In such cases, an entity may take on the location of its parent according to the spatial hierarchy. This report introduces the concepts of *locally-fit*, *locally-displaced*, and *globally-displaced* entities to address missing locations throughout the dataset.

- **A3 Establish a pairwise semantic footprint**. Spatial, dimensional, and ontological affinities are merged into a single measure, denoted a *semantic footprint*. Because each part originates from a different aspect of the data space (i.e., spatial location, shared dimensions, and ontological hierarchy), it becomes a powerful tool able to establish relatedness among spatial entities.

- **A4 Analyze the affinity of the dataspace**. A *semantic footprint* between two entities is influenced by their spatial proximity as well as dimensional and ontological affinities. This research introduces the concepts of *dilution*, *hardness*, *concentration*, and *concession*, to indicate when the *semantic footprint* receives most of its value from the ontological or dimensional aspects of the dataspace.

B Spatial similarity in categorical domains. Relating entities based on their categorical attributes can be challenging due to lack of ordering. This task investigates how categorical values can be related to one another so that relationships between entities can be built at the attribute level.

- **B1 Utilize co-occurrence frequency based on a spatial region**. Because ontological data often provide no natural means of similarity, entities whose ontological values appear frequently can be deemed more related than infrequent ones. This work applies *Pair Correlation Function* to compute a measure of *ontological similarity* based on attribute values and spatial distance.

- **B2 Model ontological similarity based on segments**. Categorical data are viewed from two perspectives: spatial location and hierarchical classification. The former establishes spatial

segments of entity pairs that share a common physical distance. The latter generates ontological segments of entity pairs that share common attribute values. The algorithm in this research combines these segments to generate a numerical *ontological distance* between any pair of attributes.

- **B3** **Provide a comprehensive view of spatial hierarchies**. *Semantic entity similarity* is applicable to any data sources that reside in the spatial (i.e., has location) and ontological spaces (i.e., has categories). To support the proposed approaches, however, this study provides a review of *Geography Markup Language* (*GML*) as a representative example of a spatial hierarchy. A detailed report describes issues that arise with the use of GML related to querying, parsing, and storing spatial data, and relating them to semantic and syntactic similarity measures.

- **B4** **Perform extensive experiments to validate the proposed approaches**. Each method has been targeted and analyzed for its effectiveness. Both synthetic and real datasets are utilized, while existing methods have been compared to the proposed techniques in this research.

C **Spatial similarity in graph networks**. Spatial analysis of entities are commonly viewed as a connected graph. Graphing in this context tends to allow stronger semantic understanding of the data for better reasoning. While the previous tasks investigated individual entities (or pairs), this part aims at reasoning over sequences of entities that can be traversed in graphs.

- **C1** **Limit the data space to regions of high entity density**. Datasets come in massive numbers of entities. This research devises a method that finds regions of high entity clustering, constraining the analysis to only the entities located in those regions. The practical effect is that it lowers the number of entities from the millions to a few thousands that are more manageable to deal with.

- **C2** **Differentiate entity relationships**. Entities interact with one another in many different ways. Some of these interactions are important while others may be irrelevant. Segregating them is important to save resources and to increase coherence. This study proposes a method for entity connectivity that looks only into a few strong relationships as opposed to dozens (or even hundreds) of uninformative ones.

- **C3** **Relate groups of entities into meaningful stories**. Information about single entities only relay limited knowledge about its environment. However, when many entities are combined in sequences, they can tell a more complete story. This work proposes four methods that establish similarity and relatedness among sequences of entities. These measures, which are numerical, can then be applied to common analytical tasks, such as clustering and classification

D **Spatial similarity in sequential data streams**. Entity analysis can reveal meaningful information when they are viewed in connected sequences. Because many sequences can often be

generated from spatial data, methods must be provided to compare and rank them. Once ranked, they can be used across many analytical tools.

- **D1** **Establish similarity between data streams**. This section devises a method that calculates the similarity between two data streams using *Dynamic Time Warping*. We extend it with spatial distance and temporal proximity that is able to find alignment between any sequences even when entities are missing.
- **D2** **Devise analysis methods for inferencing and association**. Carefully designed entity sequences can be used in many analytical tasks. This study shows how they can be applied to forecasting, association, and inferencing in social networks.

E **Establish similarity in coherent paths**. While generating entity sequences can be challenging, generating them in such a way that is intelligible to the human mind is even more elusive. This section generates well-described entity sequences, even though they are not explicitly spelled out in the underlying data.

- **E1** **Devise similarity based on semantic signatures**. Before coherence can be proposed, a level of connectivity strength between any two entities must be calculated. For this purpose, we propose a novel function, namely semantic signatures, which combines relationship types, spatial distances, and temporal differences into a single score. This score represents the binding strength between two entities, and is used in the design of semantic coherence described next.
- **E2** **Devise semantic coherence**. A coherent data stream is one that focuses on few topics of interest, often limited to a small k number. This limiting of data helps in the design of storylines that are intuitive in specific domains of application. We propose two measures of semantic coherence: (1) lengthwise, which computes a score based on the number of entities related to a theme of interest; (2) distancewise, which calculates a score based on the spatial distances between entities of a storyline.

The combination of the five above modules (A, B, C, D, and E) amount to a comprehensive view of entities as an application of data analysis. These approaches are designed to cover elements that are semantic (i.e., has an extended meaning) and provide syntactic typing (i.e., has explicit characteristics). In each, spatial, dimensional, and ontological spaces are investigated, both from structure and content perspectives. They are often combined into hybrid solutions and are covered throughout this research.

1.4 Proposal Organization

The remainder of this work is organized as follows. Chapter 2 first provides background information applicable to this study. Further, it develops both semantic and syntactic methods of entity

analysis utilizing spatial, dimensional, and ontological aspects of the data. Chapter 3 describes a method based on *Pair Correlation Function* and entity segmentation to generate a numerical similarity measure for categorical data. In Chapter 4, a comprehensive view of *GML* is given as an example of a spatial hierarchy and its implications to both semantic and syntactic entity analysis. Chapter 5 presents a method that combines spatial proximity and entity connectivity to find streams of information called storylines. These storylines are further explored in Chapter 6, where they are compared with one another with four different similarity measures. Chapter 7 is complementary to the previous discussions as it proposes not only an alternative method of storyline generation, but also a method to compute semantic coherence for storylines. Future directions and a summary of our contributions are given in Chapter 8.

Chapter 2

Spatial Similarity in Multidimensional Spaces

2.1 Introduction

Geographic feature expansion is a common task in Geographic Information Systems (GIS). Identifying and integrating geographic features is a challenging task since many of their spatial and non-spatial properties are described in different sources. In this study, the expansion problem is tackled by defining semantic footprints as a measure of similarity among features. Furthermore, three quantifiers of semantic similarity are proposed: spatial, dimensional, and ontological affinity. These measures are shown to dilute, concentrate, harden, or concede the feature space, and provide useful insights into the semantic relationships of the spatial entities. Experiments demonstrate the effectiveness of this approach in semantically associating the most related spatial features.

2.2 Background and Motivation

Geospatial web services as well as Geographic Information Systems (GIS) commonly exchange data for a multitude of application domains from real estate to marketing. For these systems, one major challenge has been interoperability: the capacity for understanding different data sources in spite of syntactic and semantic differences in language. Several organizations have attempted to mitigate this problem with standardized specifications. The *Open Geospatial Consortium* (OGC), for instance, has proposed a set of frameworks in an attempt to bring uniformity to spatial data processing [96]. In general, these frameworks use standard grammars such as *Extensible Markup Language* (XML) for data transport. Google and Yahoo! often use KML (*Keyhole Markup Language*) in their mapping APIs. Government agencies often use *Geography Markup Language* (GML) for data exchange [82]. One advantage of XML is its hierarchical structure which helps

define relationships among entities. As a consequence, it also lends itself well to object orientation that is so prevalent in modern computing.

	Data Source 1	Data Source 2	
1	<gml:coordinates>	<gml:coordinates>	1
2	-56.3159,	-56.3101,	2
3	52.5168	52.5199	3
4	</gml:coordinates>	</gml:coordinates>	4
5	</gml:Point>	</gml:Point>	5
6	</ogr:geometryProperty>	</ogr:geometryProperty>	6
7	<ogr:building>	<ogr:building>	7
8	<ogr:AREA>	<ogr:AREA>	8
9	5.000	3.932	9
10	</ogr:AREA>	</ogr:AREA>	10
11	<ogr:PERIMETER>	<ogr:PERIMETER>	11
12	25.010	22.882	12
13	</ogr:PERIMETER>	</ogr:PERIMETER>	13
14	<ogr:NAME>	<ogr:NAME>	14
15	Leon Dept of Housing	Hope Apartments	15
16	</ogr:NAME>	</ogr:NAME>	16
17	<ont:living space/>	<ont:apartment/>	17
18	<ogr:LAT>	<ogr:LAT>	18
19	543831	523300	19
	</ogr:LAT>	</ogr:LAT>	
	<ogr:LONG>	<ogr:LONG>	
	56100	52449	

Figure 2.1: Example of GML Data Sources

Consider the two GML examples depicted in Figure 2.1: Data Source 1 describes a *geometryProperty* named *Leon Dept of Housing*, whereas Data Source 2 describes another geometric object called *Hope Apartments*. What is the relationship between these two geographic features/objects? A quick look at their attributes provides some hints: they are within close proximity of each other (lines 1-3), both are urban structures (line 6), and one object occupies similar but less area than the other (lines 7-9). Based on these observations, the following possibilities arise: (1)*Hope Apartments* is part of the *Leon Dept of Housing*; (2) They are indeed the same since *Leon Dept of Housing* was renamed *Hope Apartments* and moved across the street from its original location into a smaller facility; (3) They are two independent facilities that are coincidentally co-located. Without further contextual considerations, only domain experts can make a complete and necessary determination of the nature of the relationship between these two geographic features.

The discussion above illustrates the challenges in reasoning on disparate data sets. Work in this field of research proposes a wide variety of approaches to handle data disparity: value comparisons, word distances, disambiguation, look-ups on gazetteers, and others. While some of these approaches have been successful to some extent, they often introduce a high level of complexity in semantic processing. Our work aims to reduce this complexity by proposing a semantic framework which exploits spatial relationships built into the geographic features. The framework will help elicit hidden and useful semantic information about the geographic features and their neighbors. Our goal is not only to determine possible matches, but also to determine whether geographic features can be deemed complementary (or irrelevant) to one another. We would like to determine if *Leon Dept of Housing* and *Hope Apartments* are the same building or just similar facilities. We

are also interested in measuring their physical proximity and then combine their associated descriptions so that a higher authority (i.e., the domain expert) may make a final decision based on his/her own constraints.

We propose a method of **semantic footprints** based on three relational concepts: the **spatial affinity** within the data space; the **dimensional affinity** within the XML hierarchy; and the **ontological similarity** based on the feature's class label. In addition, we describe an approach that utilizes the above measures to associate and link disparate geographic features. Because the number of geographic features is potentially large, we devise the concepts of **dilution**, **hardness**, **concentration**, and **concession** as a means to efficiently and effectively perform semantic analysis on the data. These concepts provide criteria to evaluate the ongoing progress of our analysis and help answer the following questions: are geographic features/objects being found in close proximity to the initial geographic feature query? If so, do these geographic features add sufficient relevant information to the initial geographic feature query? If the user is initially seeking only k number of features, then are the current ones sufficiently relevant or should the process continue to search for others that may be more relevant?

Our motivation relates to tools and technologies that rely on hierarchically semi-structured data (e.g., XML, GML, and KML), have strong syntactic capabilities, but lack semantic support for data processing, and can exploit semantic footprints as an auxiliary tool to enhance semantic alignment.

2.3 Related Works

Early research on spatial entities is related to the works of GIS. With the support of organizations such as the OGC, standards have been established for the management of geographic features [96] using common communication protocols (e.g., HTTP) and XML-based encodings (e.g., GML). With the advent of geospatial portals (e.g., Google Maps, Yahoo! Maps), geographic features have taken on increased popularity. Traditionally, geographic feature matching and expansion have been primarily utilized in spatial indexing methods for database systems. The use of spatial indices is abundant in this area as exemplified in [28, 8, 30]. However, our work does not focus on spatial indices but rather emphasize on the development of an approach that will enhance the extraction, processing, and analysis of semantic information in spatial data. Other aspects such as data quality and composability of grammars are described in [131, 39]. Current literature in semantic information processing can be classified into one of the following categories:

Schema Matching: Rahm et al. proposed the decomposition of complex schemas into simpler sets [107, 106]. Doan et al. used a set of semantic mappings to learn other mappings using machine learning techniques [36]. Islam et al. proposed a method to determine the semantic similarity of words and another for word segmentation [55]. Schema matching becomes challenging when many schemas are involved. In addition, it often only works with textual elements which makes spatial processing inefficient and/or impractical. We depart from the above works by considering the spatial characteristics of objects, which is not in the scope of any of the aforementioned works.

Object Consolidation: The difficulty of combining objects described in different sources is addressed by Beeri et al [10]. They extend the one-sided nearest neighbor join into mutually nearest neighbors. As described by Bleiholder et al., data fusion can also be performed at a query language level [13]. Instead of relying on schema information, objects are considered for their attribute values rather than attribute types. Seghal et al. proposed entity resolution primarily as a function of locations [119]. The spatial component is deemed similar when their distance meets a certain threshold. We differ from these approaches by extending our work beyond object fusion and propose methods to evaluate semantic relationships within the attribute and ontological spaces. An example output of our method includes determination of geographic features that are complementary within an application domain. **Ensemble Reasoning**: This class of techniques combines

Table 2.1: Summary of Semantic Information Processing Approaches

Class	Name	Primary Focus	Goal	General Spatial Applicability
Schema Matching	Rahm [107, 106] Doan [36]	Logical Structure	Feature Matching	Low
Object Consolidation	Beeri [10] Bleiholder [13]	Attribute Values	Feature Matching	Medium
Ensemble Reasoning	Fazzinga [40] Leitao [72] Bernstein [12]	Structure, Attributes, Types	Feature Matching & Likeness	Medium
Ensemble Reasoning	***Semantic Footprints***	Spatial Structure	Feature Matching, Likeness & Complement	High

characteristics of both schema matching and object consolidation to provide semantic analysis. They tend to be more effective in applications in which prior knowledge of the schemas is available. Fazzinga et al. proposed a query language to combine partial answers from different sources on the basis of limited knowledge about the local schemas in XML documents [40]. Leitao et al. proposed a method to detect duplicate objects in XML data using Bayesian networks [72]. A schema matching approach, Protoplasm, is an aggregation of several existing methods to reconcile named entities [12]. Unlike our proposed framework, these studies do not consider the spatial component of an object and rely primarily on non-spatial textual content.

Table 2.1 provides a summarized view of the literature in semantic feature analysis. The last row gives a snapshot of how our work differs from existing approaches. Our proposed framework is unique in several ways. **First**, we take a qualitative view of feature expansion by avoiding explicit comparisons on data values. **Second**, we extend the notion of spatial co-location to include the most semantically relevant nearby features which are not necessarily the closest in geographic space. For example, if a source describes several buildings and water bodies, nearby houses are possibly more relevant to a query originating from a house than a water body. **Third**, our framework is oriented towards data sources of similar application domains. As an illustration, consider the marketing realm. In its context, nearby stores and malls would most likely provide more relevant information than, for instance, weather data. We propose spatial proximity, dimensional affinity, and ontological similarity to improve the efficiency of our semantic analysis by limiting

the number of geographic features or objects under consideration.

2.4 Problem Definition of Spatial Feature Expansion

The nomenclature below formalizes the spatial feature expansion problem. Given:

- Set $D = \{D_1, ..., D_i, ..., D_n\}$ where d_i is a semi-structured hierarchical data source (e.g., *GML* file);[
- Geographic feature set $f_{geo}(d_i) = \{g_1, ..., g_j, ..., g_m\}$ where g_j's are all the geographic features or objects of data source d_i and $m = |d_i|$ is the number of geographic features in d_i.
- Set $G = U_{i...n} f_{geo}(d_i)$. The set G is the union of all geographic features in all data sources $d_1 ... d_n$.
- Attribute set $f_{att}(gj) = \{a_1, ..., a_k, ..., a_q\}$ where the a_k's are all element/attribute types of the geographic feature g_j.

Objectives:

I) From a starting geographic feature g_s (initial query), find the set $G_{close}(g_s) = \{g_j | g_j \in G$ and $dualAff(g_s, g_j) \geq \xi_{close}\}$ where $dualAff$ is a measure of the degree of spatial closeness and ξ_{close} is a user-defined threshold.

II) From a starting geographic feature g_s, find the set $G_{dim}(g_s) = \{g_j | g_j \in G_{close}(g_s)$ and $dimAff(g_s, g_j) \geq \xi_{dim}\}$ where G_{dim} is a measure of attribute similarity and ξ_{dim} is a threshold based on the ranking order of $dimAff(g_s, g_j)$.

III) From a starting geographic feature g_s, find the set $G_{ont}(g_s) = \{g_j | g_j \in G_{close}(g_s)$ and $ontAff(g_s, g_j) \geq \xi ont\}$ where G_{ont} is a measure of ontological similarity and ξont is a threshold based on the ranking order of $ontAff(g_s, g_j)$.

IV) From a starting geographic feature g_s, find an ordered set $G_{final}(g_s) = \{g_j | g_j \in G_{close}(g_s)$ and $(i < j \rightarrow Sem\phi(g_s, g_i) \geq Sem\phi(g_s, g_j)\}$ where $Sem\phi$ is a measure of similarity based on $dimAff$ and $ontAff$.

Concept of Semantic Footprints: Hierarchical structures encapsulate a rich set of relationships not always visible to the naked eye. Names do not always match, locations are ambiguous, and characteristics may range wildly. These differences arise because data is affected by many factors, such as external noise, human subjectivity, and uncalibrated measuring tools. While some systems attempt to match features by introspecting their properties [22], we avoid exhaustive attribute comparisons as they tend to increase computational complexity when many geographic features are present. To establish an efficient and effective representation of semantic relationships, we define *semantic footprints* and their components in the subsections below.

2.4.1 Spatial Affinity within the Data Space

Geographic features are commonly described in terms of their locations and hence we give our first definition for describing spatial closeness:

✣ *Definition 1*: Geographic feature g_i is said to be **locally-fit** (LF) in data source d_i if its minimum bounding rectangle (MBR) is explicitly provided in the data source.

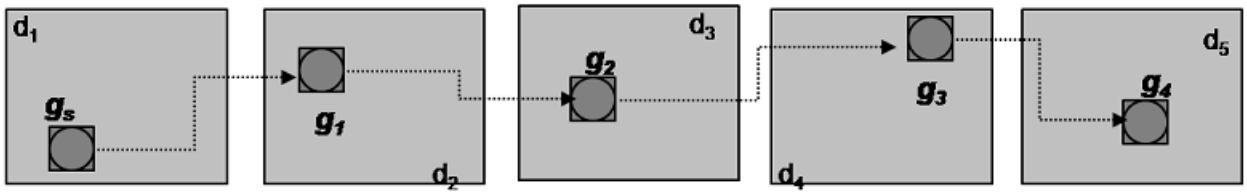


Figure 2.2: A set of 5 *locally-fit* features in 5 data sets

Figure 2.2 shows five *locally-fit* geographic features $g_s, ..., g_4$ residing in data sources $d_1, ..., d_5$, respectively. We investigate whether g_s, the starting query feature, has any spatial significance to $g_2, ..., g_4$. We give the spatial significance, namely **dual affinity**, by:

$$DualAff(g_i, g_j) = 1 - \frac{Dist(g_i, g_j) - MinDist(g_i, g_j)}{MaxDist(g_i, g_j) - MinDist(g_i, g_j)} \tag{2.1}$$

Assuming that the geographic features g_i and g_j share a common coordinate system, Equation 2.1 defines dual affinity as the degree of spatial closeness between the features. The *Dist* function can be generalized to any appropriate spatial distance. For example, we may consider geodesic distances for latitudinal and longitudinal coordinates. Other distances such as Euclidean or Manhattan distances can also be used. Furthermore, the choice of locations of spatial extents can be approximated by its centroid, which is an acceptable approach in many types of application. For example, $Dist(g_i, g_j)$ may use the centroids of g_i's and g_j's MBRs as their representative locations. The functions *MinDist* and *MaxDist* represent the shortest and longest possible distances between two geographic features respectively.

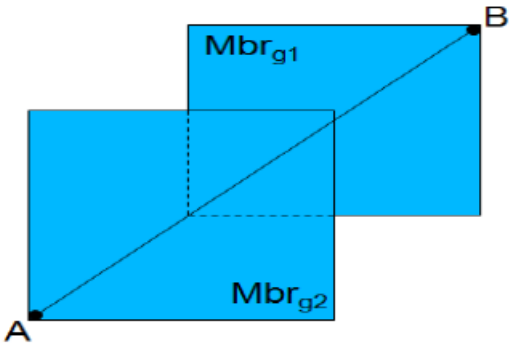


Figure 2.3: *MinDist* and *MaxDist* for two MBRs

In Figure 2.3, for instance, the geographic features are described by their MBRs. Therefore, the *MaxDist* between any two objects is the length of the segment *AB* and *MinDist* is zero since the MBRs overlap. From a spatial point of view, two features have maximal spatial affinity when their locations are the same, i.e., $DualAff = 1$. Hence, to achieve *Objective I*, $G_{close}(g_s)$ can be determined by collecting all features whose *DualAff* is higher than a given ξ_{close}. We build upon *DualAff* to define the spatial footprint of a geographic feature:

✣ *Definition 2*: The *footprint* ϕ of a geographic feature g_s is given by the set of all attributes of all geographic features in $G_{close}(g_s)$:

$$\phi(g_s) = \cup_{i=1...|G_{close}(g_s)|}(f_{att}(g_i)) \tag{2.2}$$

where $g_i \in G_{close}(g_s)$. The *footprint* represents the maximal collection of attribute types within the set of $G_{close}(g_s)$. This maximal set imposes a bound on the computational complexity of the proceeding semantic operations.

2.4.2 Dimensional Affinity in the Data Space

One attractive aspect of XML is its ability to define class relation in a hierarchical fashion. This idea gives rise to dimensional affinity and applies to all geographic features, whether they are *locally-fit* or do not have an explicit location. In these cases, we observe the dimensions of the feature (its attributes/elements), while relying on the location of its parent. In Figures 2.4 and 2.5, the five features (the circles) are within some **MBR not of their own**, indicated by the encompassing squares covering an area larger than the features themselves. In Figure 2.4, only the location of the parent is available (i.e., *fig:locally-displaced* feature), and Figure 2.5 has no location but the bounds of the data set (i.e., *globally-displaced*). While these two cases do not have an explicit location, they can still be useful to establish a *semantic footprint*. Figure 2.6 represents a geographic feature for which no spatial location is available. This case would require an external tool for location resolution (e.g., gazetteer), which is outside of the scope of this study. Dimensional affinity gives the ability to measure how similar two geographic features are in relation to their elements and attributes.

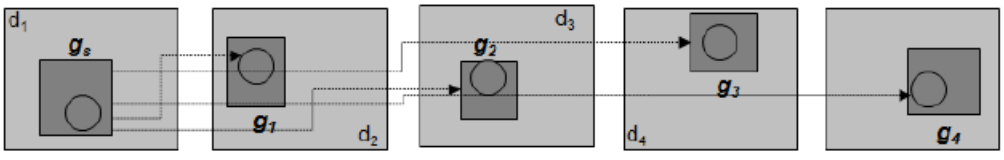


Figure 2.4: A set of 5 *locally-displaced* features in 5 data sets

We define dimensional affinity as follows:

$$DimAff(g_s, g_k) = \frac{|(f_{att}(g_s) \bigcap f_{att}(g_k))|}{\phi(g_s)} \tag{2.3}$$

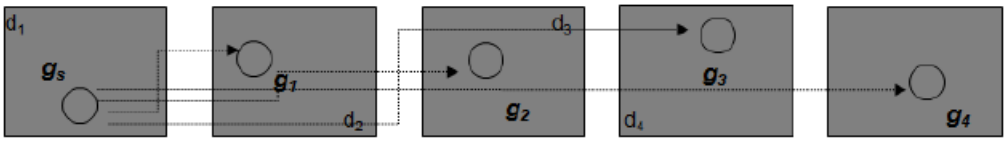


Figure 2.5: A set of 5 *globally-displaced* features in 5 data sets

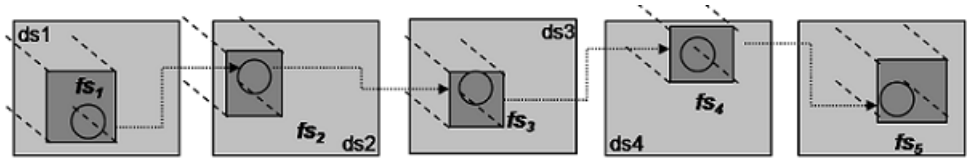


Figure 2.6: A set of 5 *globally-null* features in 5 data sets

where g_s,$g_k \in G_{close}(g_s)$. *DimAff* gives the ratio of common attributes between two geographic features, g_s and g_k, in relation to its total number of attributes, i.e., its *footprint*. Hence, the dimensional affinity is dependent upon the spatial proximity of features in $G_{close}(g_s)$ and what attribute types they share in common. If *Leon* and *Stellar* together have 22 attributes, but only 5 in common, then $DimAff(Leon,Stellar) = \frac{5}{22} = 0.23$ and if the ξ_{dim} is met, the geographic features can later be utilized in the analysis of the complete semantic footprint. *Objective II* is then achieved by forming $G_{dim}(g_s)$ as the sorted set of all geographic features with dimensional affinity $\geq \xi_{dim}$.

Ontological Class Affinity: Ontologies represent a classification scheme to group similar objects and are commonly used in a wide range of fields, from medicine to the data sciences [79, 53]. Given this as a motivation, we show a method to compute the hierarchical ontological distance among features as the third component of our semantic footprint. We define the class distance between two nodes in a common hierarchical ontology as follows [4]:

$$ClassDist(g_s, g_k) = d(g_s, root) + d(g_k, root) - 2 \times d(LCA(g_s, g_k), root) \tag{2.4}$$

where $d(m,n)$ is the edge length between the classes of m and n, *root* denotes the root of the categorical hierarchy, and $LCA(g_s,g_k)$ is the *Lowest Common Ancestor* defined as the farthest node from the root that is the most immediate ancestor of both g_s and g_k.

From the class distance measure above, we define the ontological class affinity *OntAff* as follows:

✣ *Definition 3*: The ontological class affinity $OntAff(g_s, g_k)$ is the degree of similarity between the classes of g_s and g_k from a common hierarchical ontology:

$$OntAff(g_s, g_k) = \frac{1}{1 + ClassDist(g_s, g_k)} \tag{2.5}$$

Hence, if geographic features g_s and g_k are of the same class, $OntAff(g_s,g_k) = 1$. For example, if *Leon* is classified as an apartment and *Stellar* is a house, assuming these two classes are two

hops apart in the ontology, then their $OntAff = \frac{1}{1+2} = 0.33$. *Objective III* can then be achieved by creating $G_{ont}(g_s)$ as the sorted set of all geographic features with ontological class affinity $\geq \xi_{ont}$.

Combining the measures of $OntAff$ and $DimAff$, we propose semantic footprint $Sem\phi$ as a total measure of the semantic similarity between two geographic features of $G_{close}(g_s)$. Formally, semantic footprint $Sem\phi$ is defined as follows:

✣ *Definition 4*: The semantic footprint between two geographic features g_s and g_k is given by:

$$Sem\phi(g_s, g_k) = \frac{DimAff(g_s, g_k) + OntAff(g_s, g_k)}{2} \tag{2.6}$$

Because $OntAff$ and $DimAff$ apply to elements of G_{close}, $Sem\phi$ inherits the spatial similarity constraint (via $DualAff$) of the geographic features. Hence, $Sem\phi$ provides a similarity measure between geographic features based on spatial, dimensional, and ontological affinities. From our example in Figure 2.1, the semantic footprint between *Leon* and *Stellar* is $Sem\phi(Leon, Stellar) = (0.23 + 0.33)/2 = 0.28$. Equation 2.6 helps us achieve *Objective 4* by establishing a ranking criterion for $G_{final}(g_s)$ as the set of all geographic features starting from g_s.

One goal of this study is to maintain the total number of threshold parameters to a minimum under the assumption that spatial, dimensional, and ontological affinities are jointly independent. Our framework minimally maintains only one threshold for each of the components of the semantic footprint (*DualAff*, *DimAff*, and *OntAff*). Although we assume joint independence among these components, existence of correlations does not affect the effectiveness of our semantic measures. In fact, potential correlations between these components can be discovered and further explored via our proposed semantic analysis process discussed in Section 2.4.2.

Complexity Analysis: This section provides an analysis of the costs for computing the terminal set of geographic features in $G_{final}(g_s)$ for a given geographic feature query g_s. The total cost for generating the set $G_{final}(g_s)$ is:

$$Cost(G_{final}(g_s)) = Cost(G_{close}(g_s)) + Cost(G_{dim}(g_s)) + Cost(G_{ont}(g_s)) \tag{2.7}$$

Assuming that no spatial indexing has been applied to the geographic feature set G, the cost for generating $G_{close}(g_s)$ is:

$$Cost(G_{close}(g_s)) = |G| \times DistCalc\_Cost = O(|G|) \tag{2.8}$$

where *DistCalc_Cost* is the cost of calculating the distance between two features. The distance calculation is a constant time operation.

To obtain $G_{dim}(g_s)$, the footprint is generated and the set intersect operation is performed between g_s and all other geographic features in $G_{close}(g_s)$. The set intersect operation is implemented using a hash table which gives a linear time cost. The total cost for computing the set $G_{dim}(g_s)$ is thus:

$$\begin{aligned} Cost(G_{dim}(g_s)) = \textstyle\sum_{i=1..|G_{close}(g_s)|}(|f_{att}(g_s)| + |f_{att}(g_i)|) = \\ O(|G_{close}(g_s)| \times Max_{i=1..|G_{close}(g_s)|}(|f_{att}(g_i)|)) = \\ O(|G_{close}(g_s)| \times |\phi(g_s)|) \end{aligned} \tag{2.9}$$

where $\phi(g_s)$ is the *footprint* of g_s. The set $G_{ont}(g_s)$ is obtained by performing ontological class distance calculations between g_s and all other geographic features in $G_{close}(g_s)$. A lookup table of the class IDs which link to the class nodes in the ontology allows for O(1) search time for a given geographic feature class. Once the pair of nodes is found in the ontology graph, the *Lowest Common Ancestor* (LCA) can be determined in time linear to the ontology level size by traversing to the root node and obtaining the longest common node sequence between the two geographic feature classes. The following provides the total cost of generating $G_{ont}(g_s)$:

$$Cost(G_{ont}(g_s)) = O(Ont_{ls}) \tag{2.10}$$

where Ont_{ls} is the level size of the ontology. Hence, the total cost of generating $G_{final}(g_s)$ is:

$$Cost(G_{final}(g_s)) = O(|G|) + O(|G_close(g_s)| \times |\phi(g_s)|) + O(Ont_{ls}) \tag{2.11}$$

Progressive Dilution, Hardness, Concentration, and Concession: Traversing data sources in search of related features is an ongoing process for which no halting point is clearly defined. Using the concepts of our approach, we present a systematic method to evaluate the progression of the relevant features from a starting geographic feature g_s as more geographic features $g_1...g_m$ become available for processing. The goal is to observe the changes in semantic footprint as more geographic features are analyzed, and determine to which extent *DimAff* and *OntAff* are contributing to the semantic footprint $Sem\phi$. For this purpose, we present four definitions also referred to as *density sets*:

✣ *Definition 5*: The set $G_{dilution}(g_s) = \{gj|gj \in G_{close}(g_s)$ and $DimAff(g_s, g_j) \leq t_{dim}$ and $Sem\phi(g_s, g_j) \geq \xi_{sem}\}$, where ξsem is a user-defined threshold for high semantic footprint and t_{dim} is a user-defined threshold that establishes a low level for dimensional affinity.

Dilution is the set of features with high semantic footprint, but low dimensional affinity. It is indicative of features that do not share many attributes in common. In such cases, a high $Sem\phi$ is mostly dependent on $OntAff$, the second component of the semantic measure.

✣ *Definition 6*: The set $G_{hardness}(g_s) = \{g_j|g_j \in G_{close}(g_s)$ and $OntAff(g_s, g_j) \leq t_{ont}$ and $Sem\phi(gs, gj) \geq \xi_{sem}\}$, where ξ_{sem} is a user-defined threshold for high semantic footprint and t_{dim} is a user-defined threshold that establishes a low level for ontological affinity.

Hardness defines a set of features with high semantic footprint, but low ontological affinity. When the features are not similarly-typed (i.e., far in the ontological classification), a high $Sem\phi$ must

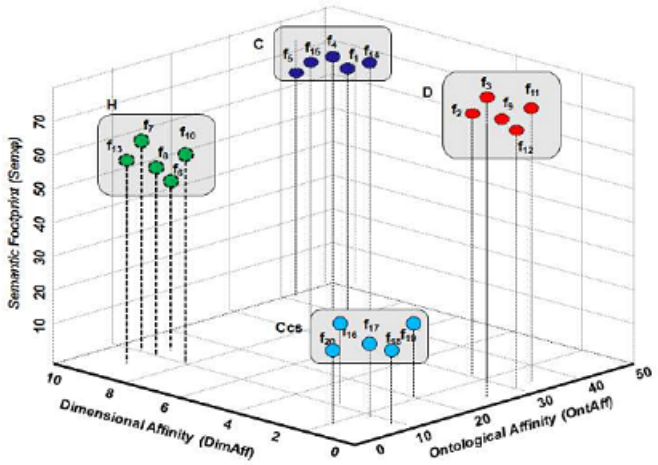


Figure 2.7: A Hypothetical Snapshot of Dilution, Hardness, Concentration, and Concession

rely primarily on $DimAff$.

✣ *Definition 7*: The set $G_{concentration}(g_s) = \{g_j | g_j \in G_{close}(g_s)$ and $DimAff(g_s, g_j) > t_{dim}$ and $OntAff(g_s, g_j) > t_{ont}$ and $Sem\phi(g_s, g_j) \geq \xi_{sem}\}$, where ξ_{sem} is a user-defined threshold for high semantic footprint and t_{dim}, t_{ont} are thresholds for minimum values of dimensional and ontological affinities respectively.

Concentration is the set of features that yield a high semantic footprint from both a high number of shared attributes and close ontological proximity. It balances a mix of geographic features that are not only similar in attribute commonality, but also similar in attribute types.

✣ *Definition 8*: The set $G_{concession}(g_s) = \{g_j | g_j \in G_{close}(g_s)$ and $g_j \notin (G_{concentration}(g_s) \bigcup G_{dilution}(g_s) \bigcup G_{hardness}(g_s$

Concession is the set of features that cannot be classified as any of the types in *Definitions 5-7*. Practically, they represent geographic features with low affinity in general, both dimensional, ontological, and as a consequence, have a low semantic footprint.

Figure 2.7 illustrates the progression graph of a hypothetical geographic feature traversal. The *H-set* shows an area of *hardness* composed of five features with high semantic footprint, but low ontological affinity. *Dilution* can be seen at the *D-set* where dimensional affinity is low. In this case, the high semantic footprint can be explained from the high ontological affinity. The concentration set C shows features with both high dimensional and ontological affinity, whereas all other cases fall under the *concession Ccs-set*. A *concentration* set (C) is possibly a richer source of information that can enhance the starting geographic feature more so than D or H.

Thresholds t_{ont} , t_{dim}, and ξ_{sem} can be manipulated to accommodate application requirements. For instance, if dimensional affinity (i.e., common attributes) is more desirable than type matching

(i.e., ontological proximity), the application should explore a *hardness set* (and vice-versa for a dilution set). When both factors are important, a *concentration set* provides a more suitable mix. It is also possible to provide an initial and automatic determination of t_{ont} , t_{dim}, and ξ_{sem} by using the centroid of the semantic footprints of the geographic features in G_{final}. The automatically generated thresholds can serve as the starting point for which further adjustments can be made as the analysis progresses. The thresholds t_{ont} , t_{dim}, and ξ_{sem} can be obtained as follows for a given starting geographic feature query gs:

$$
\begin{aligned}
t_{ont} &= \frac{\sum_{i=1..|G_{final}(g_s)|} OntAff(g_s,g_i)}{|G_{final}(g_s)|} \\
t_{dim} &= \frac{\sum_{i=1..|G_{final}(g_s)|} DimAff(g_s,g_i)}{|G_{final}(g_s)|} \\
\xi_{sem} &= \frac{\sum_{i=1..|G_{final}(g_s)|} Sem\phi(g_s,g_i)}{|G_{final}(g_s)|}
\end{aligned}
\tag{2.12}
$$

Similarly, the medoid of the semantic footprints can also be used in lieu of the centroid. Employing the medoid can provide a more robust threshold set as it less sensitive to any outliers that may exist in G_{final}.

Algorithm 1: Identifying Dilution, Hardness, Concentration, and Concession Sets

inputs : G_s,G_{close}, ξ_{sem}, t_{dim}, t_{ont}
output: $G_{dilution}(g_s)$, $G_{hardness}(g_s)$, $G_{concentration}(g_s)$, $G_{concession}(g_s)$

1: **Using** g_s **and** g_i **in** G_{close} where $i \in \{1..n\}$
2: **foreach** (g_i) **do**
3: calculate $DimAff(g_s,g_i)$ /*(Equation 2.3) */ ;
4: calculate $OntAff(g_s,g_i)$ /*(Equation 2.5) */ ;
5: Sem$\phi(g_s,g_i)$ = $DimAff(g_s,g_i)$ + $OntAff(g_s,g_i)$;
6: **if** $(DimAff(g_s,g_i) \leq t_{dim}$ && $sem\phi(g_s,g_i) > \xi_{sem})$ **then**
7: add $g_i \longrightarrow G_{dilution}(g_s)$;
8: **else if** $(OntAff(g_s,g_i) \leq t_{ont}$ && $sem\phi(g_s,g_i) > \xi_{sem})$ **then**
9: add $g_i \longrightarrow G_{hardness}(g_s)$;
10: **else if** $(DimAff(g_s,g_i) > t_{dim}$ && $OntAff(g_s,g_i) > t_{ont}$ && $Sem\phi(g_s,g_i) > \xi_{sem})$ **then**
11: add $g_i \longrightarrow G_{concentration}(g_s)$;
12: **else**
13: add $g_i \longrightarrow G_{concession}(g_s)$;
14: **end**
15: **end**
16: **output** $G_{dilution}(g_s)$, $G_{hardness}(g_s)$, $G_{concentration}(g_s)$, $G_{concession}(g_s)$;

Algorithm 1 shows a method that uses *Definitions* 5,6,7, and 8. First, the semantic components are calculated in Lines 3 and 4, and combined as the total semantic footprint in Line 5. Lines 6-12 apply simple logic to determine if the current geographic feature falls under *dilution*, *hardness*, *concentration*, or *concession*. Each feature is stored into its appropriate set for later examination.

2.5 Experiments

Given a starting geographic feature, our goal is to find other related features within one or more data sources. Our datasets are composed of features of the cities of Frankfurt, Leverkusen, and Konigswinter [29]. For the ontology, we used NASA's SWEET [94], which we extended with urban structure concepts of *home*, *apartment*, *hotel*, *building*, *warehouse*, and *construction*.

Our first step is to extract features from the first available data source and calculate their semantic footprint (*DualAff*, *DimAff*, *OntAff*). Subsequently, regions of *dilution*, *hardness*, *concentration*, and *concession* can be identified, allowing their respective sets to be populated according to Algorithm 1. In terms of measurement, we are interested in: **(a)** obtaining $G_{final}(g_s)$ when different parameters are considered; **(b)** identifying sets of *dilution*, *hardness*, *concentration*, and *concession* related to the starting geographic feature.

Table 2.2: Evaluation Queries

gs = Geb537	$\lvert f_{att}(g_s)\rvert$ i.e., Attribute Count (g_s)	$\lvert f_{att}(g_s) \cap f_{att}(g_i)\rvert$ i.e., Shared Attribute Count Range (g_i)	***ClassDist***
Query I	30	min=5, max=24	min=0, max=25
Query II	30	10	min=1, max=29
Query III	30	18	min=10, max=38

Table 2.2 summarizes three representative queries selected from the experiments. We desire to find features located within ξ_{close} = 100 Km of the starting geographic feature (g_s=Geb537) that are considered 'most related" in terms of their semantic footprint. The features in this data set have anywhere from 12 to 40 attributes (or elements) and have a variation of labels in the ontology (e.g., house, apartment, construction, warehouse, etc...).

High Overall Semantic Footprint: *Query I* sets the starting geographic feature at **Geb537** with 30 total attributes, and labeled as a 'house". For the target features, the number of shared attributes varies considerably from 5 to 30. The ontological distance varies from zero hops (i.e., *ClassDist*) for one feature and all the way to 25 for others.

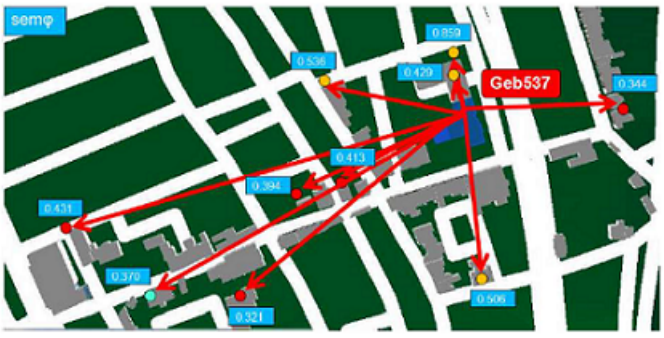


Figure 2.8: Top 10 Highest Semantic Footprint Features related to Geb537

Figure 2.8 gives a visual representation of the top 10 elements in G_{final}(Geb537) with arrows pointing in the direction of the 10 geographic features and labels for the semantic footprint values. Interestingly, the most related geographic features are not necessarily the closest ones. In fact, Figure 2.8 shows that even though Geb537 is surrounded by nearby buildings, its footprint is composed of several farther away buildings. Figure 2.9 shows all geographic features as indicate by the id field of Table 3.

High Dimensional Affinity (*DimAff*) *Query II* targets a more regular data set. We keep the same geographic starting point considering 20 total attributes. Of those, 10 are shared across all features. This configuration has the effect of setting an equal dimensional affinity across the data set (not shown). The ontological distance, however, can be fairly large. Elements are as close as one hop apart in the ontological hierarchy, and as far as 29 hops away. Figure 2.8 shows the top 10 most related elements, most of which have high dimensional affinity. In this scenario, the ontological affinity provides at best a low contribution to the semantic footprint.

High Ontological Affinity (*OntAff*) Still using Geb537 as g_s , *Query III* operates on features

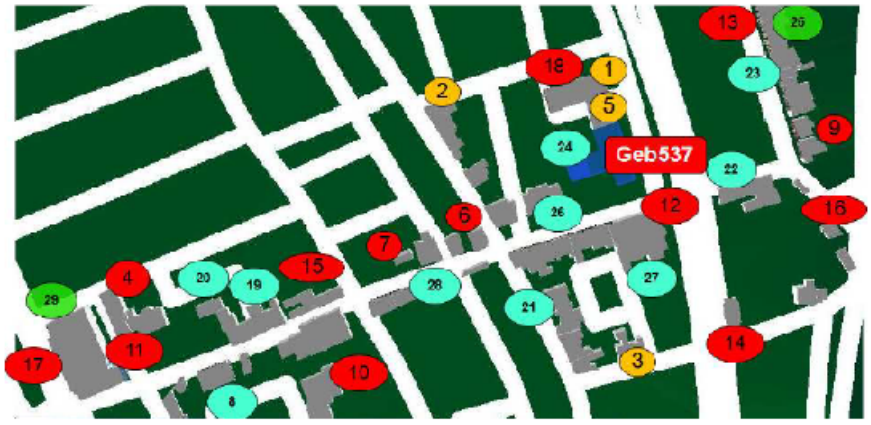


Figure 2.9: Features Related to Geb537 According to Table 3

that share many attributes (i.e., high dimensional affinity on 18 shared attributes). The ontological distance, in addition, is low for most elements, varying from 10 to 38 hops. While ontological affinity is very low, the semantic footprint remains somewhat constant at ~ 0.6 since dimensional affinity is the same across the data set. Since all features are described with similar attributes, it can be inferred that such data set most likely originated from the same provider using the same geographic standards. This is a real-world scenario, albeit possibly less common than *Query I*, where GIS often deal with a high variety of data descriptions from disparate sources.

Dilution, Hardness, Concentration, and Concession Sets Using Algorithm 1, we generate Table 2.4 to list how variations in *DimAff* and *OntAff* create sets of *dilution*, *hardness*, *concentration*, and *concession*. We set both t_{dim} and t_{ont} at 0.3 to designate our minimum cutoff requirements for dimensional and ontological affinity. If the domain expert has a strict demand for both attribute

Table 2.3: Data Results for *Query I*

g_s=Geb537 $f_{att}(g_s)$ = 30								
g_i	id	$\|fatt(g_i)\|$	$\|fatt(g_s) \cap fatt(g_i)\|$	$dmAff(g_s,g_i)$	$ClassDist(g_s,g_i)$	$ontAff(g_s,g_i)$	$Sem\varphi(g_s,g_i)$	Dilution (D), Hardness (H), Concentration (C), Concession (Ccs)
Geb855	1	25	23	0.719	0	1.000	0.859	C
Geb521	2	25	20	0.571	1	0.500	0.536	C
Geb592	3	35	22	0.512	1	0.500	0.506	C
Geb600	4	40	30	0.750	8	0.111	0.431	H
Geb597	5	34	22	0.524	2	0.333	0.429	C
Geb579	6	40	30	0.750	12	0.077	0.413	H
Geb645	7	40	30	0.750	25	0.038	0.394	H
Geb653	8	27	11	0.239	1	0.500	0.370	D
Geb593	9	40	24	0.522	5	0.167	0.344	H
Geb545	10	33	21	0.500	6	0.143	0.321	H
Geb877	11	37	22	0.489	6	0.143	0.316	H
Geb557	12	30	18	0.429	4	0.200	0.314	H
Geb504	13	38	23	0.511	8	0.111	0.311	H
Geb559	14	32	20	0.476	6	0.143	0.310	H
Geb595	15	39	23	0.500	8	0.111	0.306	H
Geb874	16	29	17	0.405	4	0.200	0.302	H
Geb889	17	36	22	0.500	9	0.100	0.300	H
Geb589	18	31	19	0.452	6	0.143	0.298	H
Geb875	19	26	11	0.244	2	0.333	0.289	D
Geb560	20	23	10	0.233	2	0.333	0.283	D
Geb514	21	10	7	0.212	2	0.333	0.273	D
Geb562	22	20	8	0.190	2	0.333	0.262	D
Geb540	23	22	8	0.182	2	0.333	0.258	D
Geb516	24	14	6	0.158	2	0.333	0.246	D
Geb865	25	28	13	0.289	4	0.200	0.244	Ccs
Geb532	26	16	6	0.150	2	0.333	0.242	D
Geb550	27	18	6	0.143	2	0.333	0.238	D
Geb522	28	12	5	0.135	2	0.333	0.234	D
Geb561	29	24	11	0.256	4	0.200	0.228	Ccs

and type similarity, Table 2.4 identifies four features in $G_{concentration}(Geb537)$ that are comprised of those characteristics. The 10 features in $G_{dilution}(Geb537)$ group elements with high ontological/low dimensional affinity, whereas the 12 features in $G_{hardness}(Geb537)$ provide the converse. Figure 2.10 gives a plot of the geographic features in Table 2.3 (only a subset of the geographic features are shown). The three cases above underscore the importance of exploratory tasks in semantic data analysis. Understanding how features compare with and complement one another promote good information extraction and knowledge discovery.

Discussion: From a mathematical perspective, semantic footprint is a measure of similarity be-

Table 2.4: Feature Sets in $G_{dim}(g_s)$ and $G_{ont}(g_s)$

$t_{dim} = 0.3, t_{ont} = 0.3$	$G_{concentration}(gs)$	$G_{dilution}(gs)$	$G_{hardness}(gs)$	$G_{concession}(gs)$
Query I	Geb855 Geb521 Geb592 Geb597	Geb653,Geb875 Geb560,Geb574 Geb562,Geb540 Geb516,Geb532 Geb550,Geb522	Geb600, Geb579 Geb645, Geb593 Geb545, Geb877 Geb857, Geb504 Geb559, Geb874 Geb889, Geb589	Geb865 Geb561

tween two geographic features. But in practice, we would like to understand its qualitative aspect,

i.e., how similar the features are or how related they may be according to their natural characteristics. Looking closer at *Query I* and according to Geb537's semantic footprint, its most related element is Geb855: they share many attributes (Table 2.3 row 1) in addition to being the same type of feature in the ontology ("houses"). For example, their shared attributes include *appearance*, *rgbTexture*, *image*, *ambientIntensity*, and *diffuseColor*, among others. Other geographic features in Table 2.3 lack some of those attributes, such as *image* and *texture*, which are not populated consistently. This scenario depicts an ideal case where semantic footprint is high from both a dimensional and an ontological perspective. As the number of shared elements decreases, so does the dimensional affinity values. Rows 2-5 still maintain a high semantic footprint due to the fairly high dimensional affinity. Row 7 (Geb645) finds a feature much farther in the ontological space ($ClassDist = 25$), causing the semantic footprint to drop as compared to the previous 5. These results force the semantic footprint to fluctuate as expected and demonstrate that semantic footprint is as an effective measure of relatedness.

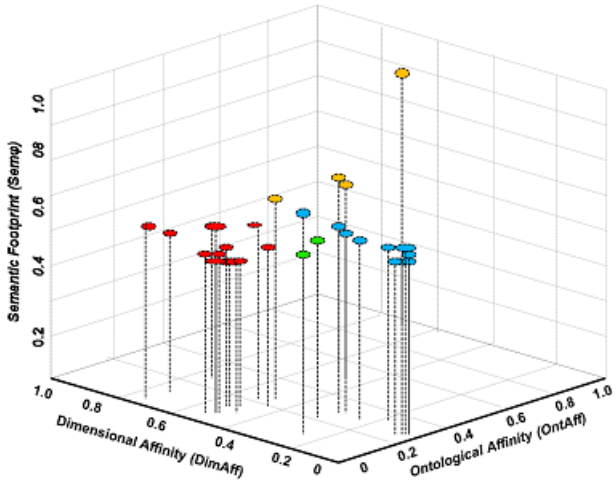


Figure 2.10: Sets of Concentration, Dilution, Hardness, and Concession

For geographic features with far-apart types, the behavior of the semantic footprint can have a different connotation. For instance, looking into Geb537 and Geb645, the ontology indicates they are 25 hops apart. The traversal path goes through "house → private residence → living Space → ...,... → construction → building → private → warehouse". The framework punishes the relationship between these two elements as possibly "unrelated" due to the different nature between house and warehouse. In spite of that, the semantic footprint is still kept high to reward their high number of shared attributes. The implication of this behavior reflects possible real-life scenarios whether the domain expert is looking for a house-house or a house-warehouse correlation. The semantic footprint is flexible enough to allow these adjustments to occur without dismissing one or the other as unrelated.

In terms of density sets, the framework provides interesting insights. First, geographic features originating in the same data set tend to be highly concentrated, i.e., their semantic footprint is fairly balanced from both an attribute and ontology perspective. While this is not exactly surprising,

variations in application domain often give rise to diluted and hardened sets regardless of whether the data sources are the same or different, but from the same provider. We observed this behavior after processing geographic features (buildings in general) from Koenigswinter and Leverkusen. Some of the data sources come in different levels of detail which are hard to compare due to the differences in attributes, but are common in CityGML format. In addition, attempts to relate applications of different domains (e.g., marketing and health) may easily yield concession sets, where the semantic footprint suffers significantly from a lack of common attributes and the fact that the ontology may not always be the same for each source. In our study, we do not propose ontology merging or disambiguation, as it is outside of our scope. However, our framework still operates correctly by placing a lower premium on geographic features for which no common ontology is applied.

2.6 Conclusion

In this study, we approach spatial data analysis from an exploratory perspective. Our work proposes *semantic footprints* as a framework for geographic feature expansion based on three concepts: spatial, dimensional, and ontological affinity. Together these concepts reason over attributes and types to uncover the most related geographic features to a starting point. In addition, they show the dilution, concentration, hardness, and concession of the feature space. Experiments on real data sets demonstrate how semantic footprints provide useful insight into data sources and the adequacy of ontological techniques for spatial applications.

Chapter 3

Spatial Similarity in Categorical Domains

Ontological structures provide a rich hierarchy of concepts and relationships that are helpful in exploratory analysis. Ontologies, however, are often categorical, which introduces ambiguity, and makes numerical analysis difficult. Adding to the problem is the fact that as the number of ontological concepts increases so does computational complexity for a variety of analytical tasks. In this study, we propose both spatial and ontological co-occurrence as a means to derive similarity among categorical values. More specifically, we devise a method that combines entity location as well as categorical frequency into a numerical measure of similarity for any pair of categorical values. In addition, we show how different ontological levels can hide or uncover information content while influencing the number of processed categorical values. We provide an illustrative case study and experiments that demonstrate the effectiveness of our approach.

3.1 Introduction

An ontology is commonly defined as a graph structure that models connectivity among concepts and relationships in a real-world domain [99]. One of its strengths is knowledge sharing, in which field experts agree a priori on standard concepts and linkage among them. The *International Classification of Diseases* (*ICD*), is such an example [141]. Provided by the *World Health Organization* (WHO), *ICD* is a hierarchical structure that assigns codes to diseases and differentiates them at nested levels. In Figure 3.1, Codes [I00 - I99.9], for instance, cover *Diseases of the circulatory system*, which are further reclassified into subcategories: *Acute rheumatic fever (I00-I02.9)*, *Cerebrovascular diseases*(I60-I69.99), and others.

Ontologies are designed such that, at shallow levels, concepts are general in purpose. As traversal moves to deeper levels, concepts become increasingly specific. Thus in Figure 3.1, while Level 1 (L1) has a general node for *Diseases of the circulatory system*, Level 2 (L2) decomposes it into several possibilities, such as *Ischemic heart diseases*. Other levels exist, though not shown.

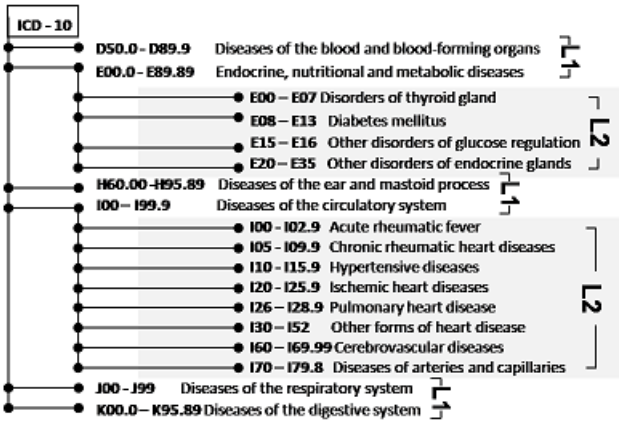


Figure 3.1: International Classification of Diseases (ICD) - partial view.

Ontological concepts are quite often categorical or nominal by nature, making it difficult to establish a similarity measure. Take for instance Figure 3.2, which depicts a few occurrences of heart disease in the region around Washington, D.C. In between *Reston* and *Tyson's Corner*, there is one occurrence of *hypertensive* ($\otimes$) heart disease along with one occurrence of *ischemic* ($\bigtriangleup$) heart disease. Spatially speaking, their distance is just a few miles apart. Ontologically, however, it is neither apparent nor intuitive whether *hypertensive* is closer to *ischemic* than to other diseases, such as *pulmonary* ($\square$), or vice-versa. The hierarchy of Figure 3.1 shows that these elements reside in the same level (L2), but their positioning relative to one another is at best arbitrary. Without a proper numerical similarity, the ontological space becomes challenging to reason over, which is a requirement in many analytical techniques. Certain classification approaches rely on numerical similarity to work properly [140], while other data transformation techniques strictly require numerical values, as is the case in *PCA*-based multidimensional reduction [74] and certain clustering techniques [3]. Determining similarity as a condition of identity and relatedness among spatial entities is normally approached from a *Euclidean*[1] perspective. Distance and other quantitative measures (e.g., a person's age or water temperature) are ideal as comparative norms since they provide an inherent basis for differentiation and ordering.

When combined with spatial properties, a categorical similarity measure may be devised quantitatively according to data point frequency. In this manner, an "ontological similarity between diseases" can be achieved. This idea makes two assumptions: distances between each pair of entities are available (or can be computed) and each entity is annotated with a categorical label. Indeed, this is an ideal situation for ontologies that reside both on the spatial domain (i.e., has location) and on the non-metric space (i.e., has categorical data). It is also considered a *local* approach since it relies on the spatial distribution of data points within a region. In addition, it lends itself well to specific contexts(e.g., *blood diseases*), but can be easily extended as a general purpose approach (e.g., *diseases*).

The number of levels in an ontology (i.e., depth) is countably infinite. While there's no limit on

[1]Other distances are also applicable, but are outside the scope of this document.

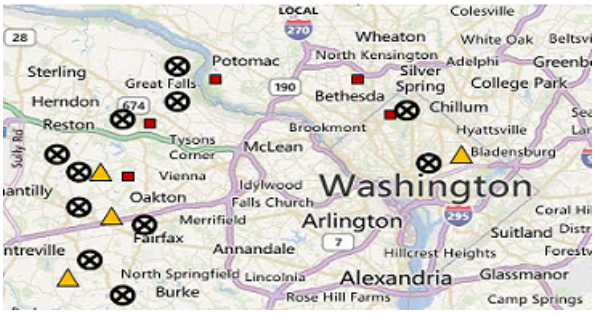


Figure 3.2: Hypothetical heart disease occurrences in Northern Virginia.

depth, there are certainly trade-offs between information accuracy and computational complexity at different levels. Figure 3.1, for instance, has $n = 6$ diseases at L1. Making a hypothetical calculation between one disease and all remaining others at that level would require $(n - 1) = 5$ operations. Comparing all to all would require $\frac{n(n-1)}{2} = 15$ operations, assuming symmetry among diseases (i.e, AB is equal to BA). At L2, with $n = 12$, 1-to-all would need 11, while all-to-all would require 66 operations.

This example underscores the fact that, at deeper ontological levels, computation costs tend to rise. The good news, on the other hand, is that at deeper levels, information is richer: extracting L1 data only yields general types of diseases, while L2 provides more specific knowledge. For a given application, then, what level strikes the right balance between cost and information content? While certain queries may just ask for *metabolic diseases*(L1), others may demand the same, but related to *Diabetes mellitus*(L2). We use this trade-off as a motivation when calculating an ontological similarity. The major contributions of this study are as follows:

1. **A method to generate a quantitave similarity measure for categorical data points**. We extend the concept of *Pair Correlation Function*(*PCF*) [109] as *ontological similarities* which measure the frequency of co-occurrence of a pair of categories at specific spatial distances. In our set up, we are interested in any pairs of entities whose distance fall within a given range, and use their category values and frequencies to determine similarity.

2. **Reasoning over ontological levels**. Determine when working at deeper levels impacts the tradeoff between information content and the number of categories to be processed. A case study is presented, which provides insight into some of those implications.

In section 3.2, we compare existing literature to our work and note where they deviate or target different goals. Section 3.3 provides background information referenced throughout the remainder of this chapter. The foundation for the categorical similarity measure, which includes identification of spatially-proximal entities, segmentation, and merging are given in Subsection 3.4. Section 3.4.3 binds our proposed approach in an algorithm, explains its phases, and provides an illustrative case study along with the computational complexity. We discuss our experiments and provide insight into the results in Section 3.5, and conclude in Section 3.6.

3.2 Related Works

The notion of categorical similarity was applied on taxonomies in the early biological sciences (Gregor Mendel, 1822-1884 [88]). In modern times, similarity based on categorical data can be seen as one of two general types: *entity level*, where the entities themselves are compared based on the number of common characteristics; and *attribute level*, where divergence is established among attribute values, but not necessarily among their entities. Therefore, if two persons share a few disease symptons, one could say the two entities are similar. At the *attribute level*, however, the similarity is simply between symptons, and no assumption is extended to the similarity between the two persons.

Entity Level: These approaches are more related to the idea of concept merging. Consider Table 3.1, where (v_m, v_n) denotes two attribute values. Bouquet *et al.* propose a similar approach to *single hopping*, where the distance between two entities is determined by their shortest path to each other [17], as a function of their distance to the root over the distance of their *Least Common Ancestor* (LCA) to the root. A variation is given by Leacock *et al.* [70], where the shortest path length is scaled by the depth D of the ontological tree. Haase *et al.* consider the length of the shortest path between two concepts and the depth of the tree, adjusting them with parameters α and β respectively for differentiated weighing [47]. In practical terms, they ignore correlation among entities since they do not consider what groups of categories are prevalent. Rather, only paths among individual entities are examined. As will be shown in Section 3.4, we depart from these approaches by looking not only where they occur, but also how commonly they appear in the dataset.

Attribute Level: One of the simplest approaches is the *overlap* measure, in which a value of 1 is assigned when two attributes match, and 0 otherwise. This approach gives every match and mismatch the same importance, and thus may not work well for many applications due to its oversimplified measure. There have been other more robust approaches to the problem.

Inverse Occurrence Frequency (IOF) is related to *term-frequency* and *inverse document frequency* (*TF-IDF*) [21]. However, it operates on categorical data, not documents. The highest similarity occurs when v_m and v_n appear only once. When v_m and v_n are the only two values, and each occurs half the time, then the lowest similarity is observed. This approach also favors small categorical sets. Our approach favors entities that co-occur frequently.

Goodall examines the probability that a particular value is observed for a pair of objects in a random sample [44]. Infrequent values are given more importance, which is more applicable to outlier detection, and less to entity relevance. Our approach also investigates frequencies, but rather, we claim higher frequency as a better indicator of importance. Under Lin [80], the similarity value is rewarded on frequent matches, and punished on infrequent mismatches. A drawback is that in many datasets mismatches are dominant, and thus, may have an adverse effect on the overall similarity. For this reason, we do not take mismatches into consideration.

Table 3.1: Similarity Measures

Measure on Entity ⓔ or Attribute ⓐ	spatial
ⓐ $overlap = \begin{cases} 1 \; if \; v_m = v_n \\ 0 \; otherwise \end{cases}$	✘
ⓐ $Goodall = \begin{cases} 1 - \sum_{m=1}^{\lvert V \rvert} Fr(v_m)Fr(v_m - 1) \; if \; v_m = v_n \\ 0 \; otherwise \end{cases}$	✘
ⓐ $Lin = \begin{cases} 2 \times log(Fr(v_m)Fr(v_m - 1)) \; if \; v_m = v_n \\ 2 \times log(Fr(v_m)Fr(v_m - 1) + Fr(v_n)Fr(v_n - 1)) \\ otherwise \end{cases}$	✘
ⓐ $Eskin = \begin{cases} 1 \; if \; v_m = v_n \\ \frac{Fr(v_m)}{Fr(v_m)+2} - \frac{Fr(v_n)}{Fr(v_n)+2} \quad otherwise \end{cases}$	✘
ⓐ $IOF = \begin{cases} 1 \; if \; v_m = v_n \\ \frac{1}{1+log(Fr(v_m))\times log(Fr(v_n))} \quad otherwise \end{cases}$	✘
ⓔ $Haase = \begin{cases} e^{-\alpha l \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}}} \; if \; v_m \neq v_n \\ 1 \quad otherwise \end{cases}$	✘
ⓔ $Bouquet = \begin{cases} \frac{d(v_m,root)+d(v_n,root)}{2\times d(LCA(v_m,v_n),root)} \end{cases}$	✘
ⓔ $Leacock = \begin{cases} max[-log(\frac{d(v_m,v_n)}{2D})] \end{cases}$	✘
ⓐ $O\sigma^*_{(v_m,v_n)}$ ⟶ see Subsection 3.4.3	✔

$d(v_m, root)$: distance from v_m to tree root
$LCA(v_m, v_n)$: Least Common Ancenstor of v_m and v_n

Eskin *et al.* take a somewhat different direction by looking at the number of values a particular category can take [38]. When the values match, the similarity is maximal. Otherwise, it decreases based on the number of possible values. This approach has better usage in small category sets, but is less than ideal in wide categories. While real-world applications do limit the number of categories to a certain extent, our approach leaves that restriction to the domain expert, not the algorithm.

An additional aspect that differentiates our approach is the use of spatial co-location. In the above methods, the relation between entities is independent of physical proximity. Our motivation is that, even though similarity is influenced by frequency, nearby entities should be more influential than distant ones. We address this problem by calculating both spatial and ontological similarities, while imposing on each a weighing scheme, as one of our contributions. Our approach is the last item in Table 3.1.

3.3 Preliminary Concepts

Entities in a high-dimensional space are often qualified by attributes of different natures: discrete, continuous, categorical (or nominal), interval, ordinal, among others. In our approach, a similarity

measure δ defined by a shared attribute a_k between entities e_p and e_q must meet the following requirements:

1. $\delta^{a_k}(e_p,e_q) > 0$ iff $e_p \neq e_q$ (*positive definiteness*)
2. $\delta^{a_k}(e_p,e_p) = 0$ (*equality*)
3. $\delta^{a_k}(e_p,e_q) = \delta^{a_k}(e_q,e_p)$ (*symmetry*)

Positive definiteness maintains that for any given distance[2] function, two entities can only lie apart from each other if they are separate objects. Further, two separate objects can have zero distance (i.e., same location). Item #2 complements that thought by stating that an object can never be distant from itself. Symmetry establishes that the distances are the same regardless of origin and destination (i.e., going from A to B is the same as going from B to A).

These definitions lend themselves to practical use under certain assumptions: values are numerical and allow ordering to be established. In addition, data points (entities, elements, objects) are well defined in unambiguous format: two objects are either the same exact thing or completely separate elements. The effect of the above rules have significant impact on data reasoning as it simplifies the computation of similarity measures via numerical analysis. In our earlier example, we can easily determine that a heart-disease patient at 239 mg/dL cholesterol is more similar to a 220 mg/dL person than to a 189 mg/dL person.

Commonly, the 3 above requirements are not applicable to categorical analysis. In the same manner that the *curse of dimensionality* causes metric data to become sparse in high dimensions [11], categorical data dilute the meaning of similarity in a complex ontological space. Take for instance the attributes *side-effects* = {*anxiety, nausea, tiredness, swelling, coughing*} and *risk-factors* = {*gender, heredity, smoking, nutrition*}. If persons pe_1 and pe_2 suffer anxiety, while persons pe_2 and pe_3 have the same nutritional diets, which 2 persons are closer, the ones that share *side-effects* or the ones that share *risk-factors*? For these 2 categorical attributes, items #1 and #2 (see the requirements above) do not hold since there's no distance within or between *side-effects* and *risk-factors*. However, we do know they are separate concepts, and should be treated likewise. As such, this study constrains the discussion to a single categorical attribute with many possible values. Item #3 is important because it allows bidirectional processing of attributes, and lowers overall processing when comparing elements. The following definitions will be used going forward:

I) D is a multi-dimensional dataset.

II) $E = \{e_1,e_1,...,e_j\}$ is a set of entities.

III) δ:E x $E \Longrightarrow \mathbb{R}$ is a spatial distance function.

[2]Distance and similarity have an inverse relationship, their converse used interchangeably in this research.

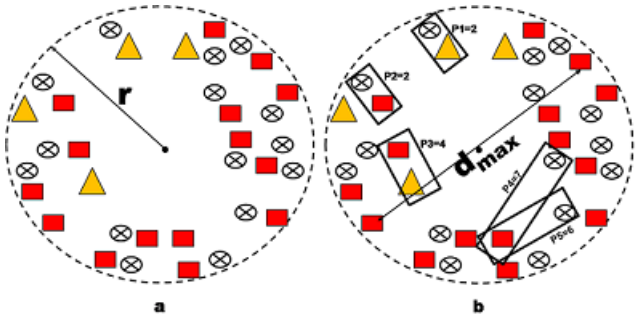


Figure 3.3: (a) A hypothetical region of radius r. (b) 5 pairs of entities segmented by distance.

IV) The region R w.r.t. radius r, denoted R_{r}, constrains the set of entities E such that $\forall k, z \in$ {1,...,n}, z≠k, the spatial distance $\delta(e_k,e_z) \leq 2r$.

V) a_k is one attribute of e_k in E.

VI) $V = \{v_1,v_2,...,v_j\}$ is a set of categorical values of a_k in A.

VII) $Fr(v_k)$ is the frequency of value v_k in the entire dataset D.

VIII) The spatial segment SS^q w.r.t. R_r is comprised of all pairs of entities whose distance is equal to or greater than $q \times c$ and less than $(q + 1) \times c$. q is the segment index and c is a user-defined length of each segment.

IX) The ontological segment $OS_{(v_m,v_n)}$ w.r.t. R_r, is comprised of all pairs of entities whose attribute values are (v_m, v_n) or (v_n, v_m) (i.e., symmetrical).

Objective:

1. Devise an *Ontological Similarity* $O\sigma^*_{(v_m,v_n)}$ between pairs of categories based on co-occurrence frequency and spatial distance.

3.4 Estimating a Local Categorical Similarity

In this section, we describe a method to estimate the similarity between two categorical labels based on co-occurrence frequency and the Euclidean distance between pairs of entities. Other distance types can be substituted.

Given an entity e_k in a dataset of |E| entities, a_i is a single-valued random attribute of e_k assuming one of the values in V. The probability of observing $a_i = v_j$ is the frequency of v_j in the dataset:

$$Fr\left(v_j\right) = P\left[a_i = v_j\right] = \frac{|E|^{v_j}}{|E|} \tag{3.1}$$

where $|E|^{v_j}$ is the number of entities with a v_j attribute and $|E|$ is the total number of entities in $\mathbb{R}_r$.

Figure 3.3(a) has a total of 34 entities within radius r. Each entity is represented by its disease category, i.e., □ or ⊗ or ⧋. There are 15 □, 15 ⊗, and 4 ⧋. According to Equation 3.1, $Fr(\square) = \frac{15}{34}$, $Fr(\otimes) = \frac{15}{34}$, and $Fr(\bigtriangleup) = \frac{4}{34}$.

In order to group pairs by distance, we must define two values: the number of segments we would like to work with; and a uniform length of each segment. These values are application-specific. Refer to Figure 3.4 as an example. For instance, we can create 5 segments, each one being 2.5 miles wide. Each segment has an index $q \in \{1..n\}$ whose greatest value denotes the total number of segments created. The length of each segment corresponds to parameter c in Subsection 3.3 *definition VIII*.

We use each segment as follows: within region R_r, all pairs of entities located less than 2.5 miles apart are allocated to Segment 1. Segment 2 encompasses all pairs of entities between 2.5 miles and less than 5 miles, and so forth. Figure 3.3(b) shows 5 pairs of entities (p_1 through p_5) at different distances. Pair p_1 is composed of two entities with categorical values (⊗ ⧋), which are located 2 distance units from each other. Pair p_2 has the same distance, but with different categories. Both would go to the same segment since they have the same distance. Pair $p3$ would go to segment 2, and pairs $p4$ and $p5$ would go to segment 3.

The number of segments can be set randomly or according to application need, since no single approach can be shown always appropriate under different spatial distributions and varying frequencies. A feasible method, however, is to find it based on d_{max}, the distance between the two farthest entities in R_r, as shown in Figure 3.3(b). Dividing it by a user-defined value λ, produces segment length c:

$$c = \frac{d_{max}}{\lambda} \tag{3.2}$$

Therefore, if the 2 farthest entities are 7.5 miles apart from each other, and λ=3, we obtain 3 segments, which can be set for distance ranges [0-2.5),[2.5,5.0), and [5.0,7.5). Adjusting λ from lower to higher values influences the granularity from coarser to more fine-grained.

3.4.1 Spatial Pair Segmentation

The steps explained earlier create a separation of entities in space. Our motivation is that nearby entities tend to be more similar than distant ones, as is commonly accepted in geographic systems. Based on spatial distance, each pair of entities is allocated to a unique spatial segment:

$$SS^q = \bigcup \{e_j, e_k\} \tag{3.3}$$

such that: $q \times c \leq \delta(e_j, e_k) < (q+1) \times c$ as in Definition VIII. In Figure 3.4, there are 3 segments: SS^1 and SS^3 have 2 pairs of entities each, while SS^2 has 1 pair, based on Figure 3.3(b). For simplicity, this example only shows 5 pairs, though all others would need to be accounted for, as well.

3.4.2 Segment Merging

Having in hand all spatial segments, we must now create separate ontological segments, as described in *Defintion IX*. This is simply a matter of allocating all pairs of entities with the same categories to the same segment. In Figure 3.3(b), for example, we can allocate $p1$ to Segment 1 and $p3$ to Segment 3. As for $p2$, $p4$ and $p5$, they go into segment 2 since they have the same pairs of attributes. The partial results are shown in Figure 3.5.

The next step is to merge the spatial and ontological segments. This is accomplished by comparing the q^{th} SS^q segment against each of the $OS_{(v_m,v_n)}$ segments, and combining the common pairs of attributes:

$$\overline{Seg}^{SS^q}_{OS_{(v_m,v_n)}} = SS^q \bigcap OS_{(v_m,v_n)} \tag{3.4}$$

We denote them as *merged segments*. To paraphrase, we break down each spatial segment based on ontological segment. For example, intersecting SS^1 of Figure 3.4 with $OS_{(\otimes,\triangle)}$ of Figure 3.5 generates only 1 common pair, which is $\overline{Seg}^{SS^1}_{OS_{(\otimes,\triangle)}} = \{P_1\}$. Likewise, $\overline{Seg}^{SS^1}_{OS_{(\otimes,\square)}} = \{P_2\}$.

3.4.3 Ontological Similarity Computation

By looking at each merged segment, we first define the Segmented Correlation Factor *(SCF)*:

$$SCF^q_{(v_m,v_n)} = \frac{|\overline{seg}^{SS^q}_{OS_{(v_m,v_n)}}|}{|\overline{seg}^q_*|} \tag{3.5}$$

SCF computes the probability of observing a given pair of categories among all different pairs of categories in all merged segments with the same index q (denoted by $\overline{seg}^q_*$). In practice, it implies the frequency of entities that share the same attributes within a certain spatial distance. From the previous section, where $\overline{Seg}^{SS^1}_{OS_{(\otimes,\triangle)}} = \{P_1\}$ and $\overline{Seg}^{SS^1}_{OS_{(\otimes,\square)}} = \{P_2\}$, then we can calculate $SCF^1_{(\otimes,\triangle)} = \frac{1}{2}$. Similarly, $SCF^1_{(\otimes,\square)} = \frac{1}{2}$. It is possible to observe pairs of categories that are very frequent, whereas others are very rare. This leads us to define *Ontological Similarity*:

$$O\sigma^q_{(v_m,v_n)} = \frac{SCF^q_{(v_m,v_n)}}{Fr(v_m).Fr(v_n)} \tag{3.6}$$

For all purposes, the *Ontological Similarity* is just the normalized version of the *SCF* calculation. By taking into account the frequencies of each category calculated earlier in this section, it prevents extremely common categories from dominating all others. Therefore, $O\sigma^1_{(\otimes,\triangle)} = \frac{\frac{1}{2}}{\frac{15}{34}\cdot\frac{4}{34}} = 9.70$. Because each segment has an ontological similarity per pair of categories, we end up with a matrix of values as shown in Figure 3.6. Since our goal is to obtain a single measure between categories, we consolidate the various ontological similarities as follows:

$$O\sigma^*_{(v_m,v_n)} = \sum_{i=1}^{\lambda} \frac{O\sigma^i_{(v_m,v_n)}}{i} \tag{3.7}$$

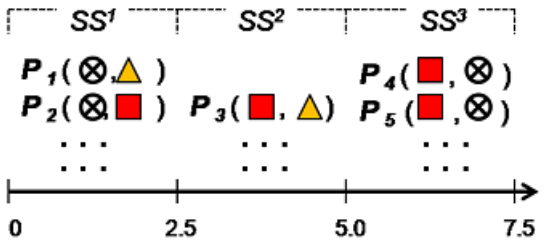


Figure 3.4: Segmentation based on spatial distance

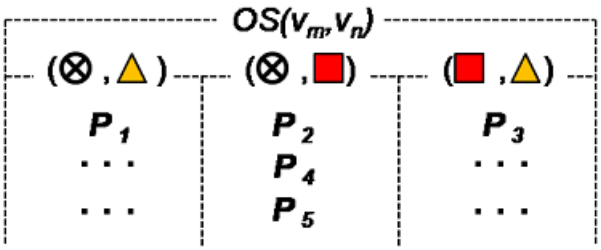


Figure 3.5: Ontological segmentation

Equation 3.7 adds all ontological similarities of a particular pair into one value. Note that it also divides each one by its segment number i. Lower segment numbers have spatially closer entities, and therefore contribute more to the overall value. As the segment numbers increase, the overall contribution decreases. In Figure 3.6, we then have $O\sigma^*_{(\square,\triangle)} = \frac{6.20}{1} + \frac{2.19}{2} + \frac{2.77}{3} = 8.21$. Likewise, $O\sigma^*_{(\square,\otimes)} = 6.11$ and $O\sigma^*_{(\triangle,\otimes)} = 14.8$. Based on spatial co-location and frequency of attributes, these ontological similarities allows us to infer: $\triangle$ is more similar to $\otimes$ than to $\square$. In addition, $(\square, \otimes)$ is the least similar of the 3 pairs.

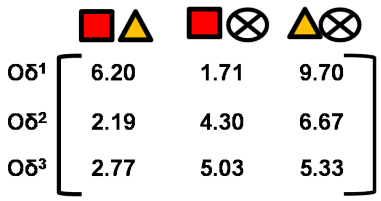


Figure 3.6: Hypothetical matrix of ontological distances

Application of Ontological Similarities In a multi-dimensional data set, a common problem is how to properly select attributes that are mostly relevant to a given task. Dimensionality reduction, which often operates on numerical data, can be applied to identify the most discriminative attributes. When data is categorical, however, most dimension reduction techniques cannot be applied. In these cases, ontological similarities can fill in that gap by performing a dimensional reduction of "sorts" on categorical data.

As seen earlier, the calculation of ontological similarities (Equation 3.7, denoted $O\sigma$ from now on) is based on pairs of attributes with many possible values. A set of persons in a city, for example, may suffer from different types of *heart disease*, for which an $O\sigma$ can be calculated. How does one

Table 3.2: Ontological similarity components ($O\sigma$)

Component	Affected by
➳ Spatial Segments	① segment length
	② d_{max}
	③ distance among entities
➳ Ontological Segments	④ ontology level
	⑤ # of categories in level
	⑥ category frequency

know if *heart disease* is indeed a good attribute of the data? Simply put, the greater the *Ontological Similarity*, the more significant that attribute is in the context of that application. One possibility then is to discard merged segments with pair values whose $O\sigma$ falls below a user-defined threshold t. This has the effect of performing dimension reduction at the segment level, without actually reducing categories or eliminating entities from the analysis.

In many tasks, however, removing certain attributes from the analysis may not be an option. Even if *heart disease* is not a "good" attribute (i.e., low *Ontological Similarity* for certain pair values), for instance, it may still be needed for other information, such as their incidence per area. So, instead of discarding it, we are forced to identify conditions in which *heart disease* would indeed yield a greater value of $O\sigma$.

As seen in Table 3.2, $O\sigma$ is dependent on both spatial and ontological segments. In turn, the number of spatial segments depends on the desired length of each segment, d_{max}, and the distribution of distances among entities. As for ontological segments, they are affected by the ontological level being worked, the number of categories at that level, and the frequency of each category in the dataset. In general, if most entities are uniformly distributed in space at similar distances and have similar frequencies, few spatial and ontological segments are generated. Such conditions influence the various $O\sigma$'s to mirror one another, making the attribute of little discriminative value.

It then begs the question: how can the factors above be manipulated to induce larger values of $O\sigma$? The length of each segment (①) is arbitrary, and no single value can be easily justified to be better than another arbitrary value. Changing d_{max} and category frequencies (② and ⑥) may be impractical because it would imply an increase in the radius of the region so to include more data points. Neither distance among entities (③), a spatial property, nor number of categories (⑤), an ontological property, can be added to or subtracted from without tainting the characteristics of the dataset.

The answer then lies in ④, which denotes working at different levels of the ontology. Consider Figure 3.7, where an ontology is depicted in three nested levels, and spatial distances (*dist*) are available between any pairs of entities. In Level 1, there are 5 individual values, comprising 10 possible ontological segments. Level 2, with 15 individual values, yields 105 ontological segments. Clearly, the ontological segmentation of Level 2 is more granular than Level 1. Level 3 yields even more ontological segments than the previous two. Our approach is the following: when one level does not yield a high enough amount of $O\sigma$, a different level should be examined, with a preference for deeper levels. While deeper levels are not guaranteed to find more significant

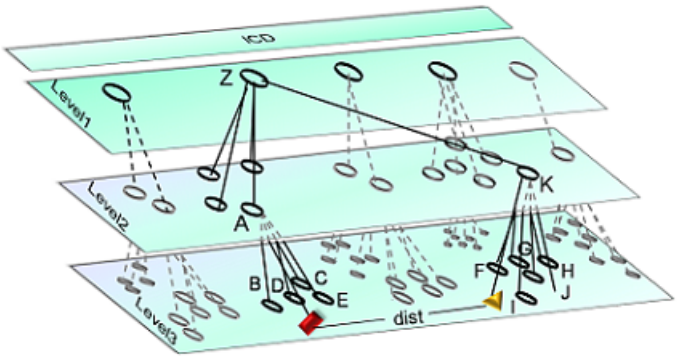


Figure 3.7: Nested ontological levels.

ontological similarities, it is a reasonable assumption. The motivation is simple: at shallow levels, many data points are condensed into the same category, helping eliminate differences in entities. When those same data points are observed at deeper levels, they may be broken down into several subcategories, permitting them to be differentiated. To illustrate this point, Table 3.3 shows the

Table 3.3: Case Study: Farmers Markets

Level	Products (records)	sp segs	ont segs	$O\sigma$
①	(A) Animal (160)	5	3	(A)(B): 5.47
	(B) Plant (449)			(A)(C): 9.12
	(C) Non-food (74)			(B)(C): 0.15
②	(A1) Dairy (54)	5	15	(A1)(A2): 8.07
	(A2) Meat (101)			(A1)(A3): 3.91
	(A3) Other (5)			(A1)(B1): 1.55
	(B1) Land (397)			(A1)(B2): 4.18
	(B2) Water (52)			(A1)(C1): 7.30
	(C1) Non-Food (74)			(A2)(A3): 0.80
				(A2)(B1): 1.10
				(A2)(B2): 9.44
				(A2)(C1): 17.40
				(A3)(B1): 12.15
				(A3)(B2): 3.65
				(A3)(C1): 0.12
				(B1)(B2): 0.76
				(B1)(C1): 2.12
				(B2)(C1): 4.81

results of a case study we have performed. We are interested in observing how the ontological similarities behave in light of different ontological levels. Our dataset is comprised of 683 farmers markets in the state of California, available from the U.S. Dept. of Agriculture [2]. Each market has spatial information (latitude and longitude) and is annotated with categories of products. At Level 1, markets are labeled as one of {*animal*, *plant*, or *non-food*} based on the types of products they sell the most. Level 2 breaks them down further. For example, the 160 animal product markets of Level 1 are categorized in Level 2 as 54 *dairy* + 101 *meat* + 5 *other* markets. We fix the number of spatial segments to 5 (sp segs) and consider the number of ontological segments for each level (ont segs). At each level, Table 3.3 also shows the $O\sigma$ for each pair of categories (e.g., *animal* and

plant = 5.47). We infer the following from the above case study:

➳ Deeper ontological levels may provide an opportunity for higher ontological similarities, even though not guaranteed. The above example shows that pair (A)(C) has $O\sigma$ = 9.12 in Level 1, while one of their subcombinations at Level 2, (A2)(C1), is 17.40.

➳ The number of ontological segments (*ont segs*) does not predict better ontological similarities. It is possible to have a third level with a significant higher number of categories than the others, which still yields very low ontological similarities.

➳ Working at deeper levels of the ontology tends to increase computational complexity. In this case, one possible optimization is to remove segments with low ontological similarities, such as (A3)(C1) or (B1)(B2).

➳ High or low frequency of a pair of categories on its own does not necessarily imply higher or lower ontological similarity. In the table, for instance, *Meat* (101 markets) and *Land* (397 markets) have high frequencies, but low $O\sigma$ (1.10). It indicates that some other factor contributed to keeping the value low, such as the fact that most of those markets are spatially far away from one another.

➳ There are 5 pairs that contain a (C1) attribute at Level 2. Three of them have very low ontological similarity. Those pairs are good candidates to be ignored in future analyses.

Algorithm 2 puts together our approach for ontological similarities in 5 phases. As inputs, it expects a set of entities for which one of its attributes is annotated with a categorical value in a set V, and λ, the maximum number of spatial segments to be considered.

Phase I generates several components needed throughout the algorithm. First, matrix *sp_d_m* stores the spatial distance between every pair of entities in the set E (Lines 1-5). Because our data is spatially symmetric (i.e., distance from A to B equals distance from B to A) only the upper diagonal of the matrix needs to be populated. The frequencies of each category are gathered in Lines 6-8, each of which is stored in its own variable. The next pre-processing step is to obtain from the spatial distance matrix the largest distance between any two entities (Line 9). In combination with λ, it helps determine the length c of each segment that we will work with, and also the index q, to identify each generated segment.

Segmentation is done in **Phase II**, where the first step is to examine the spatial distance matrix. For spatial segmentation (Line 12), each pair of entities whose distance is in the same range is allocated to the same segment q. For ontological segmentation (Line 13), all pairs of entities that share the same categories are also stored in the same segment, separate from the spatial segments.

Phase III is simply a matter of matching pairs of entities: when one pair in a spatial segment is also found in an ontological segment (Line 17), the algorithm creates a merged segment to store it (Line 18). A new index is created to keep track of the merged segments (Line 16).

Algorithm 2: Computing Ontological Similarities

inputs : set of entities E, λ, set of attribute values V
output: a list of ontological distances

{**Phase I: pre-processing steps**}

1: **for** $i = 1$ **to** $|E|$ **do**
2: | **for** $j = i + 1$ **to** $|E|$ **do**
3: | | $sp\_d\_m(i, j) = \delta(e_i, e_j)$ /*build spatial distance matrix*/;
4: | **end**
5: **end**
6: **foreach** *(v_i) in V* **do**
7: | **set** $Fr(v_i) = \frac{|E^{v_i}|}{|E|}$ /*compute frequency of each category*/ ;
8: **end**

{define number and length of segments}

9: **set** $d\_max = max\{sp\_d\_m\}$;
10: **set** $c = d\_max/\lambda$; $q = 1$ **to** λ ;

{**Phase II: allocate entity pairs to spatial and ontological segments**}

11: **foreach** *(e_i, e_j) in sp_d_m* **do**
12: | $SS^q \longleftarrow sp\_d\_m(i, j)$ /*apply *definition VIII* in *Section 3.3* */;
13: | $OS_{(v_m,v_n)} \longleftarrow sp\_d\_m(i, j)$ /*apply *definition IX* in *Section 3.3* */;
14: **end**

{**Phase III: merge spatial and ontological segments**}

15: **foreach** *(e_i, e_j) in SS^q* **do**
16: | set $x = 1$ /*create a new index*/;
17: | **if** *$(e_i, e_j) \subset OS_{(v_m,v_n)}$* **then**
18: | | $\overline{Seg}^{SS^x}_{OS_{(v_m,v_n)}} \longleftarrow (e_i, e_j)$;
19: | | x++;
20: | **end**
21: **end**

{**Phase IV: compute ontological similarities**}

22: **foreach** $\overline{Seg}^{SS^x}_{OS_{(v_m,v_n)}}$ **do**
23: | compute $SCF^x_{(v_m,v_n)}$ using Eq. 3.5 ;
24: | compute $O\sigma^x_{(v_m,v_n)}$ using Eq. 3.6 ;
25: | $map((v_m, v_n, x), O\sigma^x_{(v_m,v_n)})$ /*store values in a map*/;
26: **end**

{**Phase V: combine ontological similarities**}

27: **foreach** *(v_m, v_n, x) || (v_n, v_m, x) in map* **do**
28: | $O\sigma^*_{(v_m,v_n)}$ =+ $O\sigma^x_{(v_m,v_n)}$ /*as in Eq. 3.7 */;
29: | $List \leftarrow O\sigma^*_{(v_m,v_n)}$
30: **end**
31: **output** *List*;

With the merged segments of the previous step, **Phase IV** computes the *Segmented Correlation Factor* for each segment and category pair (Line 23). These values, along with the frequencies of each category from Phase I, are subsequently used in the calculation of the ontological similarities (Line 24). Since each pair of categories gets an ontological similarity value per segment, a map is used as a separate data structure to temporarily store those values (Line 25).

The map is used in **Phase V** as follows: for all pairs of segments with equal (or symmetrical) attribute values, their corresponding ontological similarities for all segments x are summed (Line

28). This allows the algorithm to finally output a list of all ontological similarities (Line 31).

Computational Complexity The path of calculations leading to the Ontological Similarities goes through different stages:

In **Phase I**, there are two pre-processing steps to transform a raw dataset into usable data points. Creating the spatial distance matrix requires computing the distance between every pair of attributes at $O(|E|^2)$ run time. Calculating frequencies of each individual category is in the order of $O(|E|)$. Phase I then reduces to $O(|E|^2)$.

Phase II first performs spatial segmentation. Since pairs of entities are examined, this operation runs at $O(|E|^2)$. The ontological segmentation looks at pairs of attributes at $O(|V|^2)$. Combining them gives a complexity of $O(|E|^2 + |V|^2)$ for Phase II.

In **Phase III** spatial and ontological segments are merged at a cost of $O(|E|^2 \times |V|^2)$. The computation of Ontological Similarities is done in **Phase IV** where the *Segmented Correlation Factors* are first calculated with a run time of $O(V^2)$. Finally, in **Phase V**, all Ontological Similarities are combined into a single value, taking $O(V^2)$.

Thus, the total computational complexity of the algorithm is $O(E^2) + O(|E|^2 + |V|^2) + O(|E|^2 \times |V|^2) + O(V^2) + O(V^2)$, which corresponds to $O(|E|^2 \times |V|^2)$, and represents the worst case scenario. It does not take into account certain optimizations that should be implemented, such as indexing both spatial and ontological pairs of segments for faster processing.

3.5 Experiments

To gauge the effectiveness of our proposed method, several evaluations have been performed. The process of converting a set of categorical values into a numerical similarity is influenced by various factors. Therefore, our goal is three-fold: compare our proposed approach against the state-of-the-art methods summarized in the *Related Works*; observe how these factors impact our computations; and draw conclusions of their implications to real-world applications. Our dataset (*Dawn*) is provided by the *U.S Dept. of Health and Human Services* [1] and has the characteristics outlined in Table 3.4. The entire data is composed of 14 metropolitan areas throughout the United States. In our experiments, however, we limit our region of observation to the 5 locations of the table. The *Dawn* dataset records emergency room events in the 5 metro areas in the year 2009: those are individuals who suffered a drug reaction due to one of several factors, such as illicit drug use, accidental ingestion, suicide attempts, and others. Each event is annotated with its location, and has a *drug name* attribute from an ontological hierarchy of four levels. Level 1 has 22 categories (e.g., *methenamine*, *dapsone*, *acyclovir*), Level 2 has 170 categories (e.g., *oxacyllin*, *carbamazepine*), Level 3 has 195 categories (e.g., *insulin*, *coumarin*), and Level 4 has 981 categories (e.g., *sulfonamides*, *penicillins*).

In terms of spatial segmentation, we pre-processed the data in ranges of 240 miles (lat-lon dis-

Table 3.4: Dataset Characteristics

Total # of entities	380,126
Description	Each entity represents a visit to a hospital's emergency room due to a drug condition, such as allergic reaction or overdose, whether illegal or not.
# categories per level	(L1): 22 (L2): 195 (L3): 170 (L4): 990
Locations	cities of NY, Boston, Minneapolis, Chicago, and Detroit.
Spatial segments in miles	① [0-240)→ (Bos-NY),(Chi-Det) ② [240-480)→ (Chi-Minn),(NY-Det) ③ [480-720)→ (Det-Minn),(Bos-Det),(NY-Chi) ④ [720-960)→ (Bos-Chi) ⑤ [960-1200)→ (NY-Minn),(Bos-Minn)

tance, not driving distance). This gave us 5 segments that include, for instance, areas less than 240 miles apart, such as NY-Boston and Chicago-Detroit, in Segment 1. Metro areas farther than 240 miles, but less than 480 miles fall in Segment 2 (e.g., Chicago-Minneapolis and NY-Detroit) and so forth. We did vary that segmentation in some of our queries, but note when doing so. In the next subsections, we explain the results of applying Algorithm 2 to the above dataset using different parameters. Initially, we ran the pre-processing steps of **Phase I** of the algorithm ahead of time, to obtain the *spatial distance matrix* and the frequencies of each categorical value in the dataset.

3.5.1 Effect of Varying the Ontological Levels

The density of entities in each city is very high. New York alone has 58,645 entities while Boston accounts for 39,526. We take a gradual approach and use the data for each city in increments of 1% up to 5% selected randomly, which yields the range of entities between 1,939 and 9,698. Spatial segmentation (*sp segs*) is kept at the original number of 5 and Levels 1-3 are used for the ontological segmentation (*ont segs*) which yields 231 segments (segments with only 1 occurrence are not considered), as shown in table below.

entities[min-max]	sp segs	ont segs	ontological level
1,939-9,698	5	231	1

We compared our approach against those of *Leacock* (LE), *Goodall* (G), and *Lin* (LI). We are interested in how well our *Ontological Similarity* can differentiate categorical values in comparison to each approach. In real terms, the best similarity measure is the one capable of maintaining consistent behavior: high(low) frequencies and close(far) spatial proximity yield high(low) similarity values across the dataset. This not only helps identify related entities, but also serves as a predictive tool of entity behavior. For this purpose, we plot the number of entities (n) against the ontological similarity (Oσ) of *merged segment* 1 and most popular category pair of each run.

For the other approaches that do not use segments, we compute their similarity directly using their respective formulas. In the plots, we have also normalized the similarities from zero to one since our measure can have much higher values than some of the other approaches. We used a standard *min-max* normalization approach.

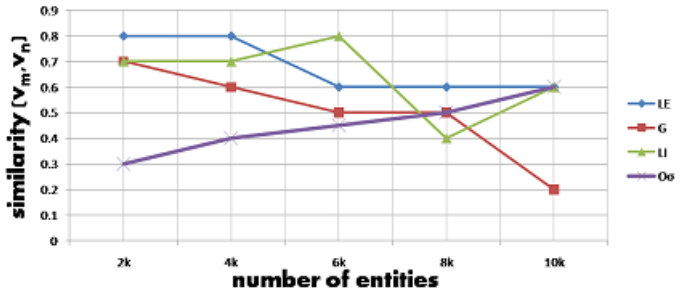


Figure 3.8: Number of Entities vs. Ontological Similarity - Level 1

Figure 3.8 shows the similarity between (*warfarin*,*ibuprofen*), which came out to be the most frequent drugs in Segment 1. The graph reveals an inverse trend between our approach ($O\sigma$) and the others: while *LE*, *G*, and *LI* display a high initial similarity, they tend to fall as the number of entities increases. By contrast, ours starts low and ends higher. The parameters for this run works at a shallow level of the ontology (L1), where the other approaches are fairly efficient. We note, however, that our approach is more robust because it is the only one with a consistently predictive behavior: $O\sigma$ always increases when the number of entities increases. *LI* has a steep drop after 6K entities, while both *G* and *LI* are impacted after 4K. For the next run below, we move to ontological Level 2, where the number of categories increases to 195 and the parameters are as follows:

entities[min-max]	sp segs	ont segs	ontological level
1,939-9,698	5	2,313	2

Figure 3.9 shows a different result from the previous plot. Our approach tends to take advantage of a higher number of categories, and thus more ontological segments. And despite such increases, our ontological similarity remains fairly stable, or at least does not show drastic changes. *LI* is particularly susceptible to ups and downs as the number of entities change because it punishes the similarity when it finds mismatches, which we observe after the 8K mark. Our approach is agnostic to mismatches, and thus behaves more consistently as more entities are added, displaying $O\sigma$ between 0.5 and 0.7. Because we avoid such fallacies, our approach makes a stronger case as a meaningful similarity. On the next run, we visit Level 3, where the number of categories substantially increases. After cleaning up the very infrequent pairs of categories, we end up with the following parameters:

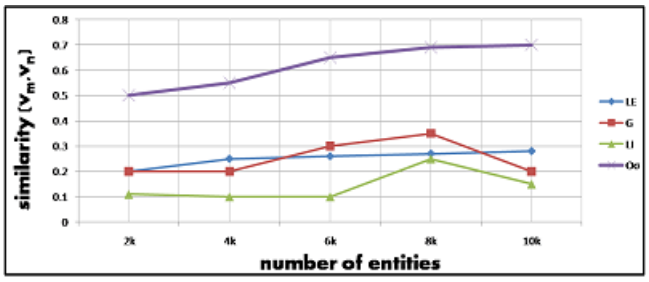


Figure 3.9: Number of Entities vs. Ontological Similarity - Level 2

entities[min-max]	sp segs	ont segs	ontological level
1,939-9,698	5	8,000	3

This setup is particularly interesting because Level 3 of the *Dawn* dataset is extremely large, as shown by the 8,000 ontological segments it generates. Figure 3.10 shows the result. LE and LI behave more poorly than G and $O\sigma$ because their similarity trends fluctuate across a wide range of entity pairs. G and $O\sigma$, on the other hand, remain stable throughout the process, with G slightly outperforming our approach in the 2k-4K entity range. Ideally, we would like our approach to at least remain stable. The reason is that G has encountered a certain number of fairly infrequent values from which it benefits. For example, with 9K entities, we observed approximately 1,100 segments whose attribute pair occur no more than 5 times. In our approach, this fact helps decrease the $O\sigma$, while under *Goodall* it helps increase the similarity.

3.5.2 Effect of Removing Infrequent Categories

As noted earlier, the calculation of *Ontological Similarities* can be computationally costly when the ontology is very large. Therefore, it would be beneficial to find pairs of categories with low $O\sigma$ and avoid including their corresponding merged segments in the analysis. For this purpose, it is necessary to introspect our dataset and find out how many merged segments can be eliminated whenever $O\sigma$ lies below a given threshold t. We set a range of thresholds from 0.1 to 0.4, with 3 spatial segments (Bos-NY, Chi-Det, and NY-Det), and use Level 4 of the ontology that generates 32,000 ontological segments, as below.

entities	sp segs	ont segs[min-max]	ont level	t[min-max]
1,939	3	32,000	4	0.1-0.4

Figure 3.11 depicts the number of merged segments whose $O\sigma$ falls below threshold t. The relationship between these two factors is almost linear, and serves to confirm that more category pairs have lower ontological similarity when the threshold is higher. While this may seem obvious, such trend cannot always be predicted. If entities are observed to have a very frequent number of cate-

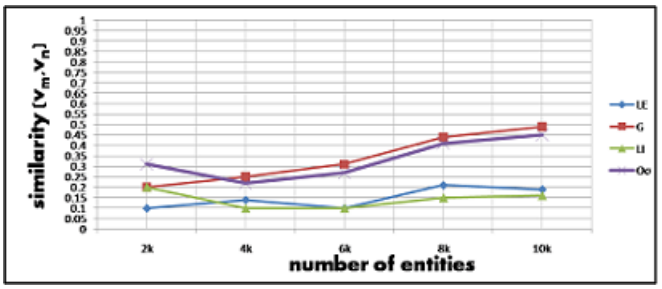


Figure 3.10: Number of Entities vs. Ontological Similarity - Level 3

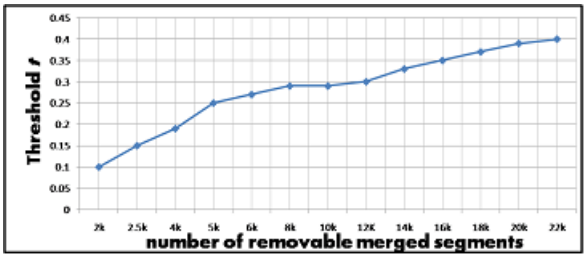


Figure 3.11: Number of Removable Merged Segments vs. threshold - Level 4

gorical pairs, the $O\sigma$ will tend to spike higher, and possibly beat the threshold most times. Figure 3.11 could then show a decreasing trend or even remain stable. Nevertheless, this graph illustrates one important contribution: Level 4, which in its totality can generate 489K ontological segments, has 22K removable ones. This would save $\frac{990*989}{2} - \frac{968*967}{2} = 21,527$ computations from future analyses.

3.5.3 Practical Implications

The previous section compared our approach with different methods and manipulated various factors to observe the behavior of our proposed approach. However, we seek to understand the practical results of *Ontological Similarities*.

First and foremost, categorical co-occurrence is an integral part of our approach. One would then expect that when pairs of categories occur frequently, their ontological similarity would always be high. In fact, we observe this is not true quite often. In the Det-NY stretch, for example, the combination (*methenamine*,*glimepiride*) is a fairly common emergency room event. The former has 81 events and the latter has 112. Their *Ontolgical Similarity* $O\sigma = 42.20$, however, is fairly low as compared to other pairs with even lower frequencies, but that are located in closer spatial proximity. Two such pairs are (*topotecan*,*iodixanol*) whose $O\sigma$=51.0 and (*cidofovir*,*pilocarpine*) whose $O\sigma$=77.12. These two are located in the Bos-NY area, whose lesser distance helps increase their similarity measures.

The combination (*cocaine*, *miscellaneous agents*) occurs 20,389 times. It appears to be more commonly reported in hospitals of Chicago and NY, followed by Detroit, Boston, and Minneapolis,

Table 3.5: Most Similar Drug Occurences

v_m, v_n	$Fr(v_m)$	$Fr(v_n)$	$O\sigma$
ethanol, *heroin*	61,819	18,621	112.75
cocaine, *miscellaneous agents*	20,389	5,334	110.04
marijuana,*drug unknown*	12,875	8,057	81.01
antineoplastics,*hydrocodone*	5,615	5,237	77.76
inotropic agents,*amoxycillin*	4,527	4,513	71.55

in this order. An interesting fact however, is that *cocaine* is not reported in those cities in conjunction with any other drugs. This leads us to believe that *miscellaneous agents* comprise a large sub-ontology for which we do not have information in our *Dawn* dictionary, and thus cannot infer what other drugs are linked to *cocaine*.

All *merged segments* have corresponding pairs of categories in at least one level of the ontology. However, not all *merged segments* have pairs that are present at all levels. As an example, we find that (*antihistamines*, *methylphenidate*) are two drugs present in Level 1. However, no drug sub-categories exist for them in Levels 2 through 4 (*Dawn* populates them with the value -7). This has a practical implication: working at different ontological levels may not be applicable in many domains whose ontologies provides low coverage at deeper levels.

Lastly, Table 3.5 presents the most similar pairs of drugs along with their frequencies and *Ontological Similiarities* $O\sigma$ for Level 3. It shows, for example, that *ethanol* is ontologically more similar to *heroin* than to any other drugs. Moreover, *marijuana* is highly similar to other drugs for which not data is reported (i.e., *drug unknown*). The high frequencies shown in the table also corroborate our approach that, along with co-occurence and spatial distance, these elements are able to devise a useful *Ontological Similarity* helpful in exploratory analysis.

3.6 Conclusion

Ontologies provide a wealth of information hidden in nested hierarchical levels. One of its limitations, however, is that ontological data is often categorical, making it inappropriate for many analytical tasks that require numerical values. In this work, we proposed *Ontological Similarities*, a numerical measure of categorical values based on attribute frequencies and entity co-occurrences. Our approach considers spatial apects of the data and is able to determine application-specific similarity between any pair of categories. We also compare our work to existing methods, and show where our approach is more efficient. Further, we show how categorical pairs can be eliminated from the analysis to save computing cycles. Our work has been effective in uncovering insightful information hidden in different levels of the underlying data.

Chapter 4

Data Analysis in Geospatial Applications

This chapter presents a study of Geography Markup Language (GML), the issues that arise from using GML for spatial applications, including storage, parsing, and querying. GML is a modeling language developed by the *Open Geospatial Consortium* (OGC) as a medium of uniform geographic data storage and exchange among diverse applications. Many new XML-based languages are being developed as open standards in various areas of application. However, many of them lack direct support for semantic and syntactic data analysis. The goal is to analyze the technologies that affect GML from the perspective of entity similarity, provide insight in how they are related, and point out advantages and drawbacks.

4.1 Introduction

With the increasing popularity of the Internet as a medium for information exchange, there has also arisen the need to develop applications that exchange data seamlessly. eXtensible Markup Language (XML) is an attempt in this direction. As a meta language, other languages can extend XML for use in specific areas of application.

Geography Markup Language (GML) is an XML encoding designed for use with geographic information. This language helps in the storage, exchange, and modeling of geographic information containing both spatial and non-spatial attributes. GML uses the concepts provided in the Abstract Specification of the Open Geospatial Consortium (OGC) for modeling geographic objects, such as geometry, topology, and features. GML data is self-descriptive, serving as a mechanism for information discovery, retrieval and exchange [100].

Example 4.1 illustrates a GML description of a road linking the german cities of *Stuttgart* and *Ludwigsburg*. It shows certain attributes such as an "id" in Line 2 and a "description" in Line 5. The location is given by the "pos" tags in Lines 8-9. This study is concerned with the following questions:

Table 4.1: A GML instance document **exampleRoad.xml**

```
1: <featureMember>
2: <ex:Road gml:id="rd455">
3: <curveProperty>
4: <CompositeCurve srsName="WGS84">
...
5: <description> Highway between Stuttgart and Ludwigsburg</description>
 6: <boundedBy>
7: <Envelope srsName="WGS84">
8: <pos>10 100</pos>
9: <pos>20 150</pos>
10: </Envelope>
11: </boundedBy>
```

1. Can a GML query language relate two similar persons who have traveled recently in this road?
2. What types of storage systems can best accommodate similar entities?
3. What indexing strategy is mostly appropriate to access related entities in GML documents?
4. Which similarity measures are efficient to identify GML entities?

A geospatial application is supported by a database or file system that can handle spatial data types. Spatial objects have both non-spatial attributes, such as *id* and *description*, and spatial attributes, such as location, geometry, and neighborhood properties. The application must provide various functionalities, including input, storage, retrieval, selection, and analysis of the information [110]. Although these features are also provided by traditional applications, they seldom handle spatial information in a uniform format, which may lead to problems in the exchange of data. GML representation of information is unique, the way its information is used can differ, and its meaning can vary according to context.

Using GML for geospatial applications has both advantages and disadvantages. GML documents nest spatial data types, permitting the effective representation of the various components of spatial data. This data has to be stored in such a way as to allow efficient query processing. However, extracting information from GML documents can be challenging due to time constraints and application complexity. The choice of spatial XML DBMS (Database Management System) also plays a role in the data extraction process. Inefficient query processing, especially for large data sets, is often difficult to overcome. Since GML is based on XML, the query languages and other data processing capabilities available to XML can also be used for GML. However, ideally, they should be extended to support the processing of semantic analysis and entity relations.

This study addresses various aspects of using GML for geospatial applications in the context of semantic and syntactic entity analysis. Several approaches to entity similarity are discussed, and some examples are given. The discussion is organized as follows: Section 4.2 discusses GML schemas and storage, relating them to entity similarity measures related to structure and content. GML parsers and query languages are addressed in Section 4.3, applying to them entity similarity measures specific to attribute values. These ideas are summed up in this research's conclusion of

Section 4.4.

4.2 GML Schemas

GML targets both information storage and retrieval in its specifications. In terms of spatial analysis, however, there is no native functionality such that entities can be analyzed in terms of their semantic importance. In other words, while a GML document may describe several buildings, it does not necessarily show how they are related or if they are somehow linked to one another. Therefore, at both the schema and storage levels, semantic analysis is absent.

Spatial data is heavy by nature. Any single map, for instance, may consist of thousands of entities and millions of attributes. As a consequence, GML documents can be (and often are) very large, raising concerns about processing and transport. GML documents were designed to be transferred between systems in a transactional fashion, allowing users to process the incoming data as it streams in. Semantic analysis, therefore, benefits when the schemas are well understood and storage is efficient for each application domain.

4.2.1 Semantic Similarity Issues in GML Schemas

The method of querying GML documents can have a significant impact on the computation of a semantic similarity measure. Many querying variations have been built around the concept of *Lower Common Ancestors* (LCA). Simply put, it tries to find a subtree that contains all terms in a search, and further identify a subset of that subtree with just the ideal answer to the initial query. Below we note some of the implications of LCA to entity similarity.

In a GML document, define $t(v)$ as the tag label of node v. In addition, let $u \sqsubset v$ represent that u is an ancestor of v. Given a keyword query $Q = q_1, q_2, ..., q_m$ and an input document D, we use KL_i to denote the keyword list of q_i, i.e., the list of nodes which directly contain q_i. LCA is commonly defined as follows:

Definition 1. *Given m nodes $n_1, n_2, ...n_m$, v is the LCA of these m nodes if v is an ancestor of node n_i, $i \in \{1...n\}$ and $\nexists u$, $v \sqsubset u$, where $u \sqsubset n_i$, denoted as $v = LCA(n_1, n_2, ...n_m)$.*

To paraphrase, if a query with keyword k_i is issued, then node n_i answers k_i when $v = LCA(n_i)$ and v contains input keyword k_i. For example, consider the document in Figure 4.1, where a user wants to search for the *condition* with the title containing "pulmonary" and one *cause* being "lungs". Node $condition^{11}$ is a LCA of the two input keywords and represents an appropriate answer.

However, there are certain problems that LCA does not address. For instance, consider the query {"pulmonary","heart"}. The nodes $patient^2$, $condition^{11}$, and $condition^8$, shaded in Figure 4.1, are the LCAs. It can be seen that $patient^2$ should not be an answer of this keyword query. The reason

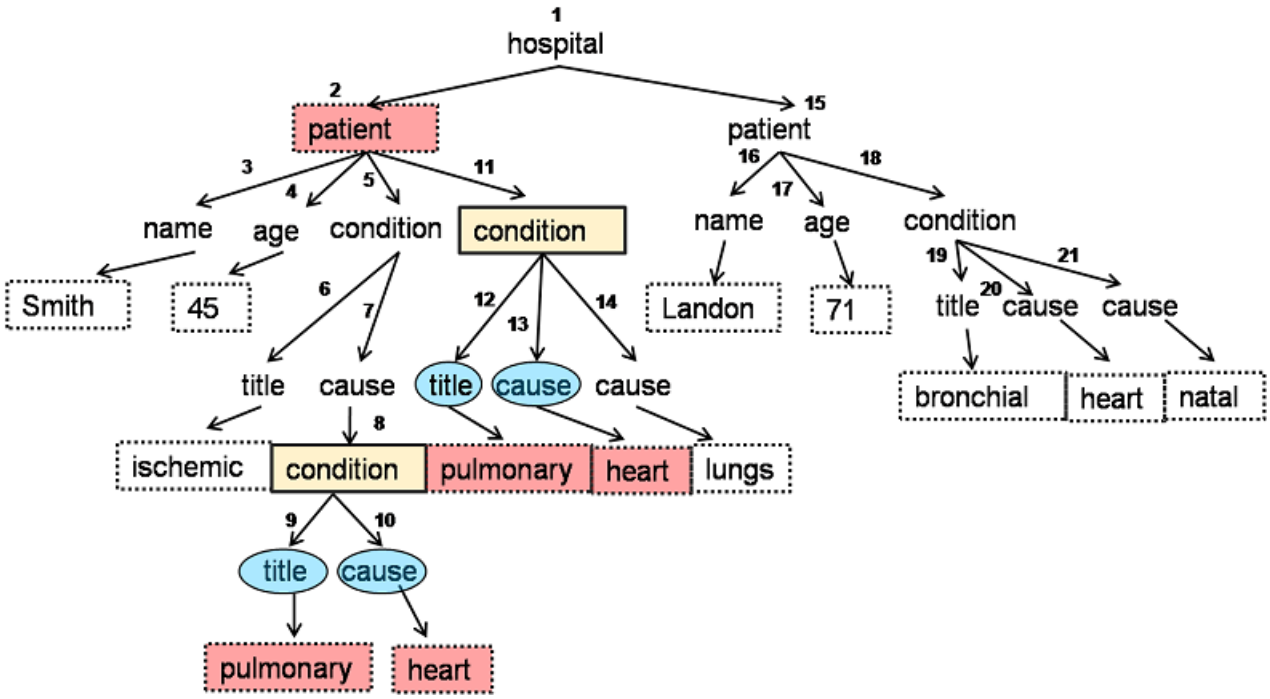


Figure 4.1: False Positives of LCA

is that *title*[9] and *cause*[13] do not belong to the same *condition*. On the contrary, the only two results should be *condition*[11] and *condition*[8]. This is the **first problem**. This false positive problem is addressed by Smallest LCA (SLCA) [146], and defined as follows:

Definition 2. *Given a keyword query* $Q = \{q_1, q_2, ..., q_m\}$ *and a GML document D, the set of SLCAs of K on D is* $SLCA(I_1, I_2, ..., I_m) = \{v | v \in LCA(I_1, I_2, ..., I_m)$ *and* $\nexists u, v \sqsubset u, u \in LCA(I_1, I_2, ..., I_m)\}$.

The important concept behind SLCA is that, if node v possesses all the keywords, its ancestors will be less meaningful than v itself. For this reason, SLCA introduces the concept of the *smallest tree*. In short, the smallest tree contains all the keywords of the query. However, within it, there are no other subtrees which also contain all the keywords. For example, although $patient^2 \in$ LCA of query ("pulmonary","heart") in Figure 4.1, it is not in SLCA, as $condition^{11} \in$ LCA and $patient^2 \sqsubset condition^{11}$. Therefore, SLCA = $\{condition^8, condition^{11}\}$ for this query.

Despite solving the above problem, SLCA still has others problems: it generates false positives (i.e., accepting some irrelevant nodes) and false negatives (i.e., missing correct results). For example, in Figure 4.2, consider query ("pulmonary","heart") issued on the GML document. Both $condition^5$ and $condition^{11}$ should be valid answers. However, $condition^5$ is an ancestor of $condition^{11}$, and thus is eliminated from the SLCA computation according to Definition 2. Therefore, SLCA generates a false negative. This is the **second problem**.

Another representative example of a false positive is depicted in figure 4.3. If the query asks for ("pulmonary","lungs"), then both $patient^2$ and $condition^{19}$ are included in the answer. $patient^2$,

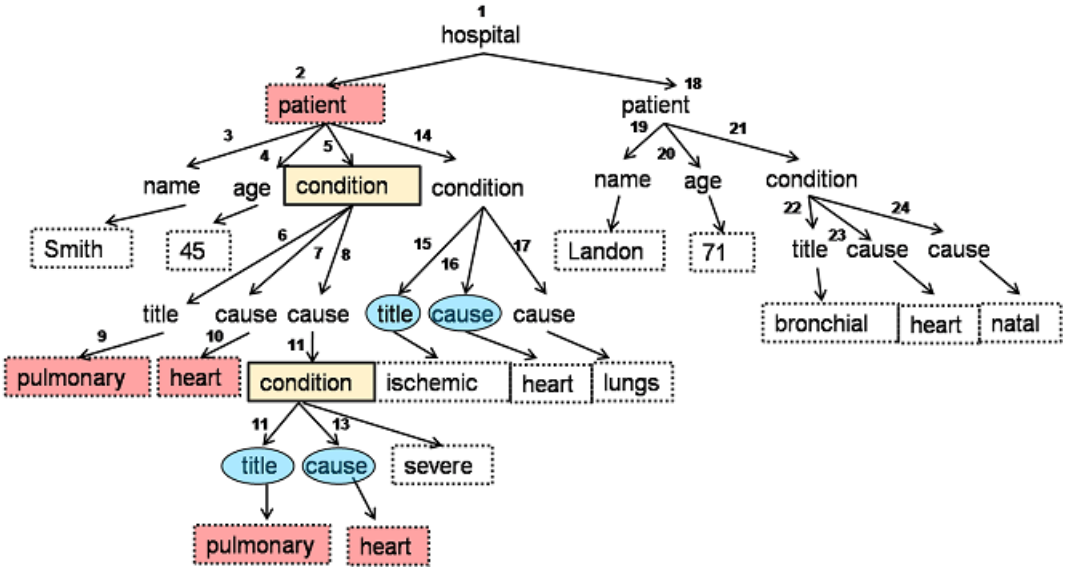


Figure 4.2: False Negatives of LCA

however, should not be included since *title*[6], and *cause*[15] do not come from the same element. The above problems are vital in the design of a valid semantic similarity measure.

Similar to the SLCA approaches described above, this research also makes use of parenthood between entities. This is done in the computation of *Ontological Class Affinity* in Chapter 2 using Formulas 2.4 and 2.5. However, unlike SLCA, this work avoids false positives and false negatives with simple heuristics: (1) when entities reside in the same branch of the ontological tree, they are always considered more similar than anything outside of the branch; (2) intra-branch comparisons favor entities with the shortest possible path; (3) entities are not compared in relation to their subtrees; (4) attribute values are only compared when they belong to the same entities. For example, in the **first problem** of Figure 4.1, *patient*[2] would not be identified as an answer to {“pulmonary”,“heart”} because these attributes are not directly part of *patient*[2], even though they related by parenthood. The **second problem** of Figure 4.2 is avoided because searches do not look for or eliminate common subtrees. Rather, each entity is examined regardless of where they are located, avoiding missing true answers. Lastly, in Figure 4.3, *condition*[5] would be considered more related to *condition*[9] than to *condition*[19] because the former are in the same branch, and intuitively, have more similar types.

GML defines various XML schema types and elements such as features, geometries, and topologies through a hierarchy of GML objects. The GML objects defined in the OGC specification are broken down into several schema documents that cover aspects such as Feature, Geometry, Topology, Value, Coverage, Temporal, Coordinate Reference System, XLink, and StyleDescriptor. However, these schemas do not provide a suitable document for all the instances and only make available the foundation structures that an “application schema” can use. In other words, both schema structure and content should be analyzed. The next subsections describe similarity approaches using GML from both a structure and content perspectives.

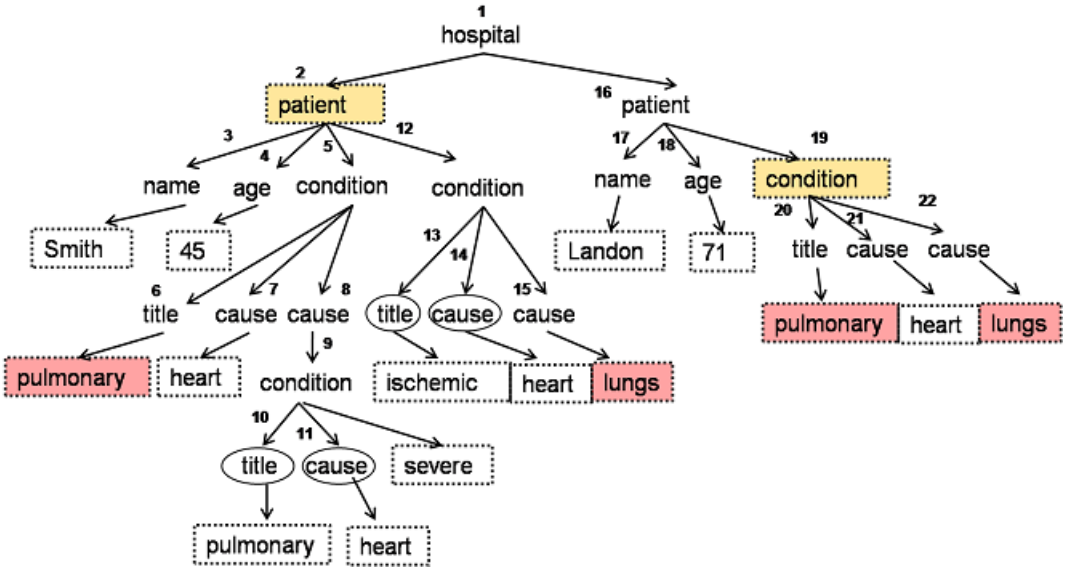


Figure 4.3: False Positives of LCA and SLCA

4.2.2 Semantic Similarity by Structure

In this section, a comparison is made between a structure-based approach to data analysis and the *Ontological Similarity* method hereby proposed in Chapter 3 , and more specifically Formula 3.7. Since GML is simply an ordered tree structure, it can be decomposed into one or more subtrees. If each subtree designates a concept, then comparing subtrees is the equivalent of measuring the similarity between the concepts. For example, in Figure 4.3, there is a subtree rooted at $patient^2$ and another at $patient^{16}$. A common approach to entity similarity in GML documents is to investigate structural components such as paths. The idea is that the more paths of subtrees match, the more they are similar or the more they are related. This is related to the concept of *best affinity* (*ba*). Given two paths p_i and p_j, and a tag $t \in p_i$, the *ba* of t w.r.t. p_j is defined as:

Definition 3. $ba(p_j, t) = \{t' \in p_j | \nexists t'' \in p_j, t'' \neq t', sim(t, t'') > sim(t, t')\}$

In other words, the *best affinity* of a tag_1 in path_1 is the set of tags in path_2 that has the highest similarity with tag_1. This similarity can be computed with many different methods, or it can be looked up in external ontologies, such as WordNet [143].

Let e_i and e_j be two subtrees and p_i=ti_1,ti_2...,ti_n, p_j=tj_1,tj_2...,tj_m be their tag paths in their respective subtrees. A *semantic structural similarity* (SSim) between entities e_i and e_j can be defined as follows [129]:

$$SSim(e_i, e_j) = \frac{1}{n+m}\left(\sum_{t \in p_i}\sum_{t' \in ba(p_j,t)} \frac{TSim(t,t')}{|ba(p_j,t)|} + \sum_{t \in p_j}\sum_{t' \in ba(p_i,t)} \frac{TSim(t,t')}{|ba(p_i,t)|}\right) \tag{4.1}$$

In other words, the structural similarity is derived by comparing the path similarities of lengths m

and n between two entities ($TSim$, defined below), but offset by their respective *best affinities*. In a spatial hierarchy, path-based methods have been used widely to determine the semantic similarity of entities. It should be noted, however, that the depth of the path plays different levels of influence on the similarity. Since deeper paths relay more specific information than shallow ones, the level of specificity must be accounted for in the similarity measure.

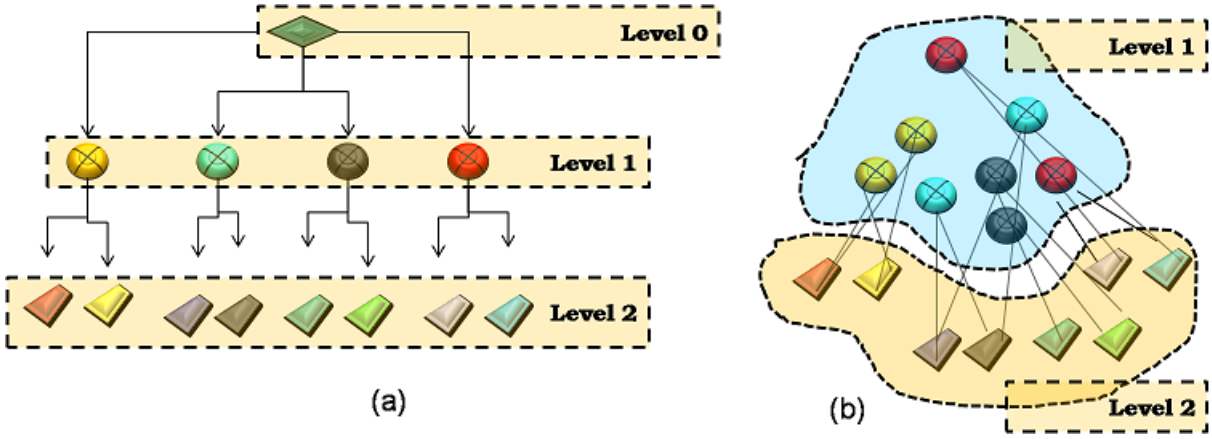


Figure 4.4: (a) Categorical Hierarchy (b) Spatial Hierarchy

One problem with this type of approach has to do with depth. When the entities in question always have the same depth, depth on its own is not a good means of differentiation or comparison. Consider Figure 4.4(a), which shows several entities in a categorical hierarchy of objects. Any comparisons done on *ovals* are irrelevant in terms of depth because all *ovals* appear only in **Level 1**. The same is true for the *polygons* of **Level 2**.

Another important aspect is the frequency with which two entites co-occur in the data space. Given these factors, assume that X is a set of GML entities and t_1,t_2 are two tags. The similarity between the tags is defined as:

$$TSim(t_1,t_2) = \frac{2 \times depth(LCA(t_1,t_2))}{depth(t_1)+depth(t_2)} \times \frac{freq(t_1,t_2,X)}{freq(t_1,X)+freq(t2,X)-freq(t_1,t_2,X)} \tag{4.2}$$

where $depth(t_i)$ is the distance from t_i to the root node of the hierarchy, $freq(t_i,X)$ is the number of GML entities that contain tag t_i, and $freq(t_i,t_j,X)$ is the number of GML entities that have both a tag t_i and t_j. Applying Equation 4.2 into Equation 4.1 gives an overall semantic structural similarity. Another problem with this approach is that when all entities have the same frequency across the dataset (or the frequencies are very similar), frequency becomes irrelevant. Figure 4.4(b) shows that at **Level 1**, each entity has exactly 2 instaces. At **Level 2**, there is one instance for each entity. Equation 4.2 would compute the same number for any pair of tags, which is undesirable. Looking back at Figure 4.3, both the *tag similarity* and *semantic structural similarity* between $condition^9$ and $condition^{12}$ can be performed as shown in table 4.2. It breaks down the components of Equations 4.2 and 4.1 and shows an example calculation.

Table 4.2: Computation of Structural Similarity

Equation 4.2		
LCA($condition^9$,$condition^{12}$) =	$patient^2$	
$2 \times depth(patient^2)$ =	2×1 =	2
$depth(condition^9)$ =	6	
$depth(condition^{12})$ =	3	
$freq(condition^9, X)$ =	3	
$freq(condition^{12}, X)$ =	1	
$freq(condition^9, condition^{12}, X)$ =	1	
$TSim$ =	$\frac{2\times1}{6\times3} \times \frac{1}{3+1-1}$ =	0.037
Equation 4.1		
$n = 2$ $m = 3$		
$TSim(title, title)$ =	1	
$TSim(title, cause)$ =	0.037	
$ba(path, title)$ =	2.77	
$ba(path, cause)$ =	1.90	
$SSim(condition^9, condition^{12})$ =	$\frac{1}{2+3}\left(\frac{1}{2.77} + \frac{0.037}{2.77} + \frac{1}{1.90} + \frac{0.037}{1.90}\right)$ =	0.18

To get around this problem, this research proposes the solution in Chapter 3, in which spatial and ontological segmentation is performed (Subsections 3.4.1 and 3.4.2). Under spatial segmentation, the subtree is not examined. Instead, entities are compared in terms of their spatial distance. In ontological segmentation, any entities of the same kind are compared. The intersection of spatial and ontological segments are used to determine their similarity. The ones with the highest frequency are considered the most similar. For example, if there are many {“red oval”,“yellow oval”} entities within 5 miles of each other, these entities would be deemed more similar than other pairs with less frequency within those 5 miles. This resolves the data distribution problem to manageable. In addition, when a certain segmentation does not provide much differentiation between entities, values can be manipulated for better results. The spatial distance, for instance, can be changed to 3 or 2 miles to attain a higher resolution.

4.2.3 Semantic Similarity by Content

Apart from the structural components of a GML document, another common approach to semantic entity similarity is done around the contents of the file. These methods are not concerned with metadata elements or how deep they reside in the ontological hierarchy. Rather, attribute values are the important factors. In general, values are evaluated under two perspectives: the number of times it appears in a document and their frequency in the entire corpus. This is a traditional *term-frequency inverse document frequency* (*TF-IDF*)-based approach [130]. Given a query term w_j and a set of attribute values u_i in a GML document, TF-IDF is given by Equation 4.3:

$$TF - IDF(w_j, u_i) = freq(w_j, u_i) \times log\left(\frac{N}{n_j}\right) \tag{4.3}$$

where $freq(w_j, u_i)$ represents the number of times w_j is observed in u_i, N is the total number of at-

tribute values in all subtrees, and n_j is the number of attribute values that contain w_j. This approach, while simple and efficient, takes into consideration only the frequency of the term verbatim. For example, a search for "heart" yields only "heart" or possibly one of its substrings. However, it would not produce "cardiac" or another relevant synonym. This is important in ontology-based hierarchical structures since analogous categorical data can be described with many different terms: "house", "residence", "home", etc. One method to address this problem takes into account the rarity of the query keyword in terms of the number of synonyms it may have in a standard ontology such as *WordNet*. The *rarity* of a term w is defined as:

$$R(w) = log\left(\frac{s - index}{|synonyms(w)|}\right) \tag{4.4}$$

where $synonyms(w)$ denotes the set of synonyms belonging to the keyword w and $s - index$ is a constant representing the number of meanings of a word in the *dictionary*, e.g., *WordNet*. The log function favors keywords with low $s - index$. In combination with *TF-IDF*, rarity can be incorporated as an influence measure:

$$Influence(w_j, u_i) = TF - IDF(w_j, u_i) \times R(w_j) \tag{4.5}$$

Equation 4.5 takes into account not only how frequent a keyword is, but also how common it is in terms of synonyms. In a GML structure, any attribute value, e.g. the value of a leaf node, can be associated to a term vector containing the value $Influence(w_j, u_i)$. Given two entities e_i and e_j, their *content similarity* (CSim) can then be computed using *cosine similarity* between the vectors associated with their *influence*:

$$CSim(e_i, e_j) = \frac{\vec{u_i} \times \vec{u_h}}{\vec{||u_i||} \times \vec{||u_j||}} \tag{4.6}$$

Referring to Figure 4.3, if one wishes to calculate the *content similarity* between $patient^2$ and $patient^{16}$, it can be done as follows. Each attribute under these two nodes are first gathered. For each value, first the *TF-IDF* is calculated based on its frequency over the set of attributes (e.g., how often "pulmonary" appears, which in this case is 3). Next, its *rarity* is computed by looking up the number of synonyms of the word "pulmonary", e.g. assumed to be 5. Combining these two previous values yield the *Influence* = $3 \times 5 = 15$ for this keyword. The vector for $patient^2$ = {*Smith*, *45*, *pulmonary*, *heart*, *ischemic*, *heart*, *lungs*, *severe*, *pulmonary*, *heart*}. For $patient^{16}$, the vector is {*Landon*, *71*, *pulmonary*, *heart*, *lungs*}. If any one of the values in these 2 vectors has *Influence* = 15 (in relation to all subtrees), then their *CSim* can be computed using Equation 4.6.

The two previous approaches (*structural similarity* and *content similarity*) are independent measures. One added benefit to these approaches, however, is that they can be combined into a hybrid semantic similarity measure ($HSim$):

$$HSim(e_i, e_j) = \kappa \times SSim(e_i, e_j) + (1 - \kappa) \times CSim(e_i, e_j) \tag{4.7}$$

where κ represents a user-defined weight to allow for an adjustment of each similarity measure. In this manner, *structure* can be made more important than *content* or vice-versa.

Table 4.3: A GML instance document **exampleHouse.xml**

```
1:<?xml version="1.0" encoding="UTF-8"?>
2:<ex:RoadInfrastructure
3: xmlns:ex=http://www.opengis.net/examples
4: xmlns= http://www.opengis.net/gml
5: xmlns:gml=http://www.opengis.net/gml
6: xmlns:xlink=http://www.w3.org/1999/xlink
7: xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
8: xsi:schemaLocation="exampleHouse.xsd">
...
14: <featureMember>
15: <ex:Road gml:id="r1">
16: <curveProperty>
17: <CompositeCurve srsName="EPSG">
...
22: <description> A house in Main St. </description>
23: <boundedBy>
24: <Envelope srsName="ESPG">
25: <lat>31.7700</lat>
26: <lon>-77.9958</lon>
27: </Envelope>
28: </boundedBy>
```

Depending on the requirements of the application domain, designers can create different types of schemas by extending or restricting the features from the GML base schema. This affords great flexibility in using GML to represent a diverse range of spatial objects. In general, when a query focuses on *structure*, $SSim$ is the most appropriate method. In Figure 4.3, for example, if the user is seeking to identify the link **"condition → severe"**, the key element is the *relationship* established between the *patient* and how serious his ailment is. Relationship is a structural component. On the other hand, if the query asks for anything *"severe"*, then this would be a *content* query, in which case $CSim$ is mostly appropriate. Below, some aspects of GML schemas are briefly explained in an effort to emphasize the importance of the structural elements that should be considered in semantic analysis of spatial entities.

There have been initiatives towards the implementation of standard application schemas to specific domains. Brodaric et al., for instance, describe the GeoSciML project as a tailored GML schema used to manage scientifc data suited for geological mapping [19]. GeoSciML illustrates a strong application of how GML can be leveraged to describe features related to the geological domain, such as Earth structures, fossils, material compounds, and their relevant attributes. The similarity approaches on these types of documents are highly dependent on how structurally complex, or how content-rich they are populated.

The OGC Abstract Specification describes a real-world phenomenon in terms of a set of features which may or may not have geometric properties. In the scope of this study, a feature is the equiva-

lent of an entitiy. A spatial feature may be associated with one or more geographic properties, such as location. The feature in Example 4.3 is a *house*. A feature is described by a set of properties, for example, *compositeCurve* and *curveProperty*. An important spatial characteristic is the *Spatial Reference System* (SRS), which is a means of referencing geographic features to a specific surface, such as that of the Earth [69]. The SRS for the *house* is "EPSG", for example. The term *Envelope* describes a region bounded by a pair of positions denoted by its corners. This is the location of the entity and can be promptly stored in a spatial index, such as *R-Tree*. As can be seen, GML documents can be very oriented towards structure in some instances. At other times, they can be extremely descriptive in non-spatial terms. Therefore, semantic entity similarity has the potential to successfully benefit from both approaches.

GML Storage Efficient exchange and storage of GML documents is an important issue. GML is substantially functional and very rich in its hierarchical structure. While this can be useful, a rich document often means a large document. As a consequence, storing GML documents often requires a significant amount of disk space.

In any amount of raw data, however, not all entities may be needed. For instance, a marketing application may seek household information, but may want to ignore medical data. Or possibly, it may only want household information of a certain income range. In this case, a semantic similarity should attempt to identify entities within that range. In other words, the semantic similarity has a direct impact on storage.

Identifying a set of entities to be stored (and by the converse, identifying a set of entities to be eliminated) can be done using any semantic similarity measure, such as the ones already described in this study. Spatial entities, such as persons and locations, can be represented as vectors in terms of their pairwise similarity. Formally, these similarities can be described in an $m \times n$ matrix S, where there are m entities and each entity has n columns:

$$S = \begin{bmatrix} 0 & & & & \\ sim_{(2,1)} & 0 & & & \\ sim_{(3,1)} & sim_{(3,2)} & 0 & & \\ \dots & \dots & \dots & \dots & \\ sim_{(m,1)} & sim_{(m,2)} & \dots & \dots & 0 \end{bmatrix}$$

The value of each $sim(m, n)$ is nonnegative which is close to zero when the entities are highly dissimilar. The highest possible similarity is not bounded, though many approaches limit it to 1. Apart from the semantic similarity measures already presented, there are very well-established measures of distance that can be used as a means of similarity (or dissimilarity). One such case is the generalized **Minkowski Distance**:

$$M = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}} \tag{4.8}$$

where p is a factor that determines the type of distance. Large distances indicate less similarity. When $p = 1$, the **Manhattan Distance** is obtained. when $p = 2$, **Euclidean Distance** is defined. If i=($x_{i1}, x_{i2}, ..., x_{in}$) and j=($x_{j1}, x_{j2}, ..., x_{jn}$) are n-dimensional entities, the *Euclidean Distance* between i and j is given by:

$$E(i,j) = \left[\sum_{i=1}^{n}(x_{ik} - x_{jk})^2\right]^{\frac{1}{2}} \tag{4.9}$$

In case of binary attributes, the distance can be computed using the **Jaccard Coefficient** as follows [48]:

$$J(i,j) = \frac{r+s}{q+r+s} \tag{4.10}$$

where r is the number of attributes equal to 1 for entity i, but that are 0 for entity j; s is the number of attributes that equal zero for entity i, but equal 1 for object j; and q is the number of attributes that equal 1 for both.

The problem with the above approaches is that they operate on the numerical domain. GML documents in many instances are populated with categorical data, which can be translated to a numerical domain.

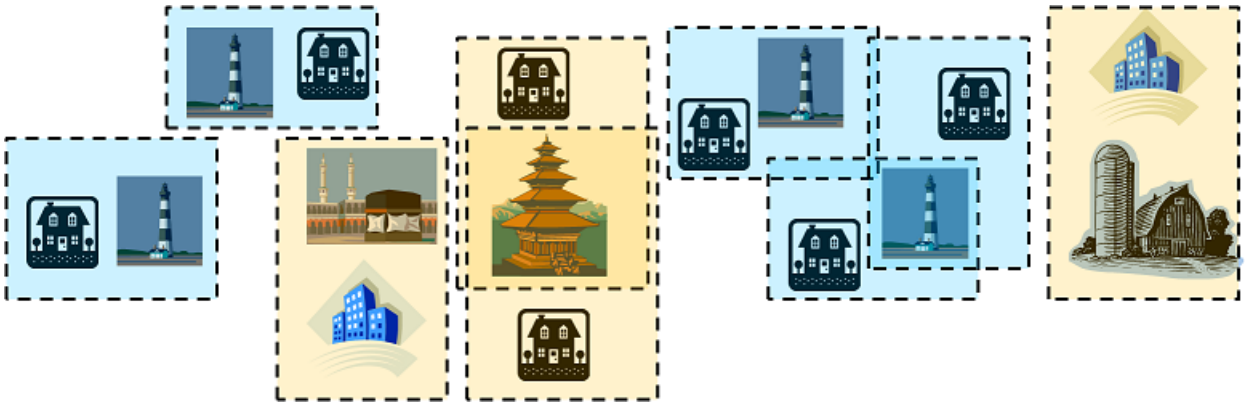


Figure 4.5: A categorical set of buildings

Consider Figure 4.5 which shows a set of several structures with attributes {*house*, *lighthouse*, *buddhist temple*, *highrise*, *muslim temple*,*barn*}. Based on this attribute, none of the above distance measures (Euclidean, Manhattan, Jaccard) are usable since the attribute is not numeric. The approach in this proposal is to perform the translation from categorical to numerical data using a combination of spatial and ontological distribution as explained in Chapter 3. The idea is simple. Looking at the figure, it can be seen that the combination {*house*,*lighthouse*} occurs 5 times, which is more than any other combinations. Therefore, in the context of this application, the attribute *house* is more related to *lighthouse* than to any other attributes. Numerically, if the pair is maximally frequent, their similarity approaches the maximum upper bound, normally 1. This

relatedness is further strengthened by their close proximity, which is numerical, and thus can be calculated with one of the generalized *Minkowski* distances. This has several implications to storage because it allows entities to be stored in many different formats. The most related entities can be stored as a set of objects with numerical relatedness in an object-oriented database. Alternatively, the categorical values may be used for the entities in a traditional relational database. In addition, they can be translated as an XML structure, and stored as such.

There are several alternative approaches that can be deployed for the storage of semistructured data or XML documents [62], and Table 4.4 summarizes some of the most common. In specialized data

Table 4.4: Various approaches to storing GML/XML documents

		Advantages	**Drawbacks**
✪	Object-oriented DBMS	➟ Some support for GML extracts and loads	➟ Complex Model
		➟ Handles complex, inter-related data	➟ Difficult to change schema
		➟ Fewer join operations	➟ Language dependent
✪	Object-relational DBMS	➟Support for abstract data types (ADT)	➟ Difficulty in translating object data to relational data
		➟ Relational and object-oriented features	➟ Comparatively less interoperability than a relational DBs
		➟ Code reuse	➟ Lack of standards
✪	Relational DBMS	➟ Easy searching	➟ Difficult to normalize GML into tables
		➟ Available indexing support	➟ Complex to extract data into GML
		➟ Widespread use	
✪	XML Database	➟ Standard XML APIs and tools	➟ Less SQL support
		➟ Support for SAX and DOM	➟ Newer technology
		➟ Support for XSLT adn XQuery	➟ Less expertise by software developers
✪	GML file on disk	➟ Strong validation	➟ Complex parsing
		➟ Accepted standards	➟ High storage requirements
		➟ Little initial processing	➟ Less interoperability between systems
		➟ Freedom of formats	➟ Lack of standards
		➟ Less strict rules	➟ Little semantic meaning

management systems, such as Rufus [83], Lore [86] and Strudel [41], the models are customized to store and retrieve semi-structured XML data. These systems are ideal for GML in the sense that highly-similar entities share a significant number of characteristics, and thus tend to simplify the underlying data model. Storing of GML data can use one of several approaches: a relational model, an object-oriented model, an object-relational model, a specialized XML database, or full file storage on disk. Storing the whole file represents less overhead, since no heavy processing needs to be performed. However, this approach tends to become space-intensive over time.

In the case of a relational model, the data is mapped into relations, and queries are posted in a semistructured query language, which is then translated to SQL queries. But if the task at hand is to measure similarity among entities, relational data becomes somewhat cumbersome to handle since entities may have to be reconstitued from many tables. In this case, applying a distance formula such as the *Euclidean Distance* or via a *Dissimilarity Matrix* over many relations can be computationally costly.

Using a database designed for semi-structured data seems to be the best approach with respect to scalability and handling of large amounts of entities. The object-oriented approach is suitable for more complex data, as in the case of GML, because entities can be retrieved as a whole, as opposed to a traditional relational model. Nonetheless, tradeoffs do exist. For instance, computing the Euclidean Distance between single attributes is efficient in a relational model. But in an object-oriented model, where the attribute must first be extracted, it adds to the cost of the similarity process. The relational model provides processing advantages due to its availability of ready-made data management tools. However, mapping a GML Application Schema to a relational database tends to result in complex structures (e.g., many tables and relationships), which may degrade system performance. In terms of semantic entity similarity, object-oriented implementations have proven efficient for GML data.

The approaches to designing database schemas for XML documents can be conveniently divided into two categories: *structure-mapping* and *model-mapping* [148]. Under *structure-mapping*, the design of the database schema is based on the DTD (Document Type Descriptor), or GML schema that describes the structure of the GML documents. With the model-mapping approach, a fixed database schema is used to store any GML documents without the assistance of GML schema or DTD. These mappings are performed on element types, attributes, and text.

Table 4.5: GML/XML data models

		Models	**Example Implementations**
✪	Approaches to storing XML documents	➟ XML DBMSs designed specifically to store XML documents ➟ Relational model: represents information as relations (tables) and queries are transformed to SQL ➟ Object oriented model: represents information as objects and their attributes	➟ Lore, Rufus, Strudel, Xhive, Tamino ➟ Oracle, XRel ➟ O_2, object stores
✪	Approaches to mapping XML documents to database schema	➟ Model mapping ➟ Structure mapping	➟ Monet, XParent ➟ LegoDB
✪	Hybrid	➟ Combination RDBMS/XML DB ➟ Object relational	➟ Oracle XDB ➟ JAXB

Corcoles and Gonzalez compare three types of document-storing techniques based on relational databases: LegoDB (*structure-mapping*) [15], Monet [118] and XParent (*modelmapping*) [56]. All three approaches were modified to support spatial objects. The advantage of using relational databases to store GML documents is the availability of robust tools for data processing, such as disaster recovery, management services, concurrency control, and query optimizers. LegoDB works well for both queries involving large numbers of attributes and documents having large amounts of data. If the GML application schema is external to the relational database, there are considerable advantages to a XML DB. Otherwise, its advantages are greatly reduced. Table 4.5 summarizes the various approaches that can be used for storing GML data.

4.3 XML Parsers and Query Languages

A GML document is fully readable by XML parsers, and can be retrieved by standard XML queries. But the choice of parsers brings about different consequences. Application designers must consider carefully which of the existing XML tools will serve their purpose when it comes to semantic entity analysis. This section discusses some of these considerations.

4.3.1 XML Parsers

The W3C XML schema definition language has been used to define the contents of GML. A parser reads a GML document, validates it against the schema, and creates a representation of the document. In fact, XML parsers can be used for parsing GML files since GML is based on XML specifications [93].

Table 4.6: Comparison between DOM and SAX parsers

		DOM	**SAX**
✪	Basic Difference	➟ Presents documents as a tree structure in memory	➟ Presents document as a serialized event stream
✪	Memory Required	➟Relatively high	➟ Significantly less, especially for larger documents
✪	Queries	➟Better for joins	➟ Better for point and range queries
✪	Suitable for	➟Small documents	➟ Large documents
✪	Advantage for GML	➟Random access to data	➟ Handles events that are less taxing on memory resources
✪	Disadvantage for GML	➟In-memory processing makes handling large data sets prohibitive	➟ API implementation complexity

The key question here is how a semantic similarity measure can aid toward a less heavyweight approach to entity parsing. Some available XML parsers are Xerces2 [6], XSV [132] and MSXML [89]. A software application should be able to understand the meaning of each entity in the GML dataset, whether the element refers to a feature, a property of a feature, or a feature collection. The software uses a GML or XML parser to validate the data so that it conforms to the GML schema, and it should understand how the data has been defined in GML according to the specification and the application schema. This knowledge helps the application correctly interpret data. Large datasets often make data processing a challenging task. Any XML parser should be able to read the GML file character by character and then represent the data in a meaningful manner. This is likely to slow the performance of GML for storing and retrieving documents when many entities are present.

There are two standard APIs that are currently used by software applications to parse GML documents: the Document Object Model (DOM) and Simple API for XML (SAX). Table 4.6 summarizes the features currently provided by DOM and SAX. The choice of a DOM or SAX parser

for GML documents depends on the resource usage and efficiency. DOM builds a tree structure as it processes the data, which tends to require a large amount of memory in the case of spatial databases. In contrast, a SAX parser traverses the document sequentially, treating the document as a data stream. This tends to consume fewer resources and hence can be used for larger datasets. However, the SAX parser does not support random access of data, and thus may prove inefficient in the case of large spatial datasets. Various studies have compared the performance of these parsers for GML [124, 137].

XML parsers are very efficient recognizing specific attributes, and thus can be used in conjunction with semantic similarity measures at the attribute level. The **Levenshtein Distance**, for example, is a well-known method of comparing strings [73]. In short, it computes the number of operations necessary to transform string σ_1 into σ_2. Three types of operations are taken into consideration: *insert* a character; *delete* a character; and *replace* a character. Algorithm 3, which computes the

Algorithm 3: Levenshtein Distance

```
   input  : strings s1, s2
   output: Levenshtein Distance L(s1,s2)
 1 set m[0,0]=0 ;
 2 for i ← 1 to |s1| do
 3 |   m[i,0]=i ;
 4 end
 5 for j ← 1 to |s2| do
 6 |   m[0,j]=j ;
 7 end
 8 for i ← 1 to |s1| do
 9 |   for j ← 1 to |s2| do
10 |   |   m[i, j] = min{m[i − 1, j − 1] + 1 if s1[i] = s2[j], +0 otherwise,
11 |   |   m[i − 1, j] + 1,
12 |   |   m[i, j − 1] + 1 } ;
13 |   end
14 end
15 return m[|s1|, |s2|] ;
```

Levenshtein Distance, creates a matrix of size $i \times j$, where i is the size of the first string, and j is the size of the second string. Each entry $[i, j]$ holds the distance needed to transform s_1 up to character i into s_2 up to character j. The bulk of the work is done in Lines 8-14, where the two strings are compared. The idea is simple: the algorithm tries to find the minimum number of operations needed for that transformation, whether the operation is an *insert*, a *delete*, or a *replace*.

Consider the matrix of Figure 4.6, where the *Levenshtein Distance* is computed between s_1="cat" and s_2="cap". Clearly, only the last letter is different, and therefore, distance should be equal to 1. To find out the transformation distance from "CA" to "CAP", $m[2, 3]$ can be looked up. In this case, only the letter "P" must be inserted, and so the distance is 1. To transform "CAP" to "CAT", $m[3, 3]$ shows 1 operation, which corresponds to replacing "P" by "T". *Levenshtein* is a syntactic distance of simple implementation.

However, in terms of entity processing, it does not suit the task appropriately. The first problem is that, in the ontological space, attributes are often described in vastly different ways (e.g., houses, homes, residences,...). Comparing such different formats is not helpful for similarity purposes

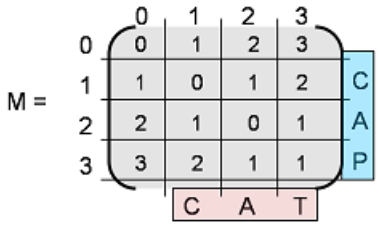


Figure 4.6: Matrix of Levenshtein Distances

because the distances of transformation tend to be high across all entities. Very few similiarities can be identified. In addition, *Levenshtein Distances* tend to consume a significant amount of memory for large strings or pieces of text.

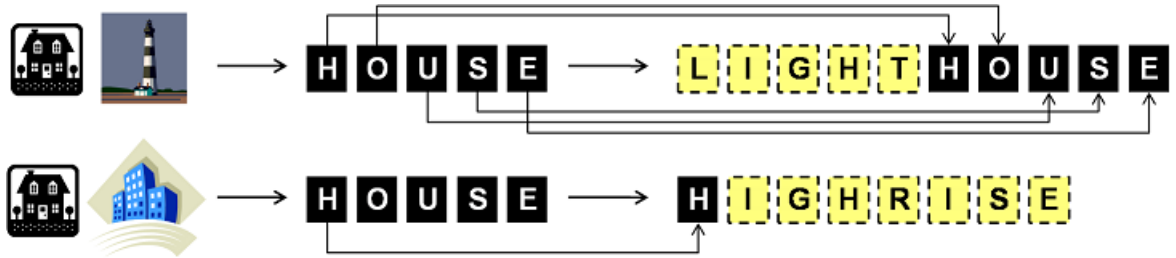


Figure 4.7: String transformation under Levenshtein Distance

Consider Figure 4.7 which depicts two transformations. In the first, to transition from *house* to *lighthouse*, 5 add operations are needed for the substring *light*. The remainder (*house*) requires no processing. This operation is efficient because the strings are already somewhat similar. In the second transformation, to go from *house* to *highrise*, only the first letter (*h*) requires no processing. All other 7 remaining letters must be modified, which is inefficient.

To avoid the above problem, this research introduces the concept of *Dimensional Affinity* to compare attributes, explained in Subsection 2.4.2. In it, entities are similar not by string-based distance, but rather, based on the number of attributes they share, regardless of how they are described. Therefore, if the attribute is *construction*, any entities that are qualified by a *construction*, regardless of *construction* being a *house* or *construction* being a *highrise*, are considered more similar than others that do not have a *construction* of some kind.

The practical implication is the following: applications that utilize DOM parsers often tie up a good amount of memory resources. When the parsing process also includes computing a similarity measure, such as Levenshtein's, the system can get affected negatively by lack of available memory for other needs. The proposed approach alleviates some of that burden by only considering entities with shared attributes. SAX is a better parsing candidate in this scenario, since it demands a significantly lower memory footprint. It can, nevertheless, have other repercussions. While SAX tends to be more efficient for point and range queries, DOM has better performance with join operations. DOM parsers can be less desirable for large GML documents because of the substantial memory usage. SAX parsers, on the other hand, can be inefficient in cases where a query involves

a large number of attributes, though the combination and types of attributes can make a difference. Therefore, a parser ideally should combine the advantages of both DOM and SAX. In the next sections, two more distances are described in the context of query languages.

4.3.2 GML Query Languages

Even the well-known query approaches that work well with XML files do not always give acceptable results when applied to GML documents that contain a combination of numeric, alphanumeric, and spatial data [31]. A GML query language must be flexible enough to support querying and retrieving such data. A candidate recommendation of XML Query (XQuery), a query language for XML, has been published by the World Wide Web Consortium (W3C) [25].

Many query languages have been proposed for querying GML documents [31, 135]. Although GML may utilize the readily available query languages developed for XML, these languages must be extended with spatial operators if they are to be used for GML. A specification of query language for GML based on extending the concept of XML-QL [118] was proposed by Corcoles and Gonzalez [31]. The authors provide a comparison of query languages currently available for XML, namely XQL (XML Query Language), XML-QL, Quilt, XQuery, and Lorel [16]. But how do these languages perform when a query tries to find a pair of entities whose similarity matches a certain threshold or falls within a range?

All query languages are based on an underlying data model that abstracts away from the physical representation of the data. The objects represented in GML are often more complex than those typically encoded in common XML, since geographic objects have both spatial and non-spatial attributes. The data model for a GML query language therefore has to reflect this complexity, and the queries must follow suit. Table 4.7 summarizes some of the pros and cons of using XML versus GML query languages for GML documents. The queries for GML data can be either spatial

Table 4.7: Comparison between XML and GML query languages

		Advantages	**Disadvantages**
✪	XML Query Languages	➟ Full industry support ➟ Less rigid hierarchy than GML query languages ➟ Available in out-of-the-box RDBMS	➟ Must be adapted to include spatial capabilities ➟ Supports alphanumeric types only ➟ May not understand GML fully
✪	GML Query Languages	➟ Standard for geospatial data ➟ Spatial constructs and joins ➟ Existing extensions to XQuery ➟ Ability to link related features ➟ Efficient filtering of desired features	➟ Still in research stages ➟ Dependent on a spatial data model ➟ Less efficient with non-spatial DBMs

or non-spatial. For this reason, a semantic similarity measure must be able to operate on both of those constraints, and still be able to correctly identify related entities. A simple, but efficient method is *Semantic Footprints*, which is proposed in Chapter 2. It is able to evaluate the similarity among entities s_1 and s_2 in the following manner:

1. Obtain the spatial distance between s_1 and s_2.
2. Identify their dimensional affinity. They are all the shared attributes in s_1 and s_2, even if their values do not match.
3. Using the shared attributes, find their ontological class distances: look up the ontological tree to find out how far apart they are.
4. Combine the spatial distance, dimensional affinity, and ontological class distances into one value. This is the final distance.

The *Semantic Footprint* is formally computed as follows (details are given in Chapter 2):

$$SemF(s_1, s_2) = \left[\sum_{i=1}^{n}(s_1 - s_2)^2\right]^{\frac{1}{2}} + \frac{(attributes(s_1) \bigcap attributes(s_2))}{attributes(s_1) + attributes(s_2)} + ClassDist(s_1, s_2) \quad (4.11)$$

Figure 4.8 shows an example calculation between two *construction* entities. Their spatial distance

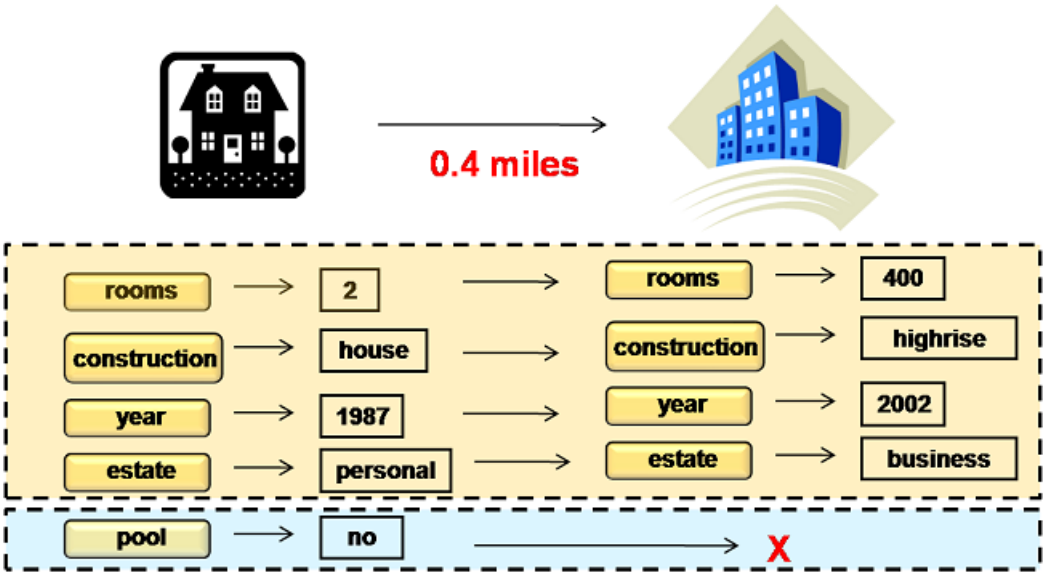


Figure 4.8: Hypothetical description of two entities

is 0.4 miles. There are five distinct attributes out of which they share 4 (*rooms*,*construction*,*year*,*estate*). Therefore, their dimensional affinity is $\frac{4}{5} = 0.8$. Note that the fact that one has 2 rooms and the other has 400 is irrelevant: in this method, values are not directly compared. The fact that a value may be obtained is what makes it relevant. Assuming that construction types, which are *house* and *high-rise*, have a class distance of 0.7, then their total distance can be computed as $0.4 + 0.8 + 0.7 = 1.9$. This measure takes into account both spatial and non-spatial aspects of the data.

XML query language models can be extended with an entity similarity measure such as *Semantic Footprints* to include the spatial query attributes of GML. This approach is not *edit-distance*-based, such as *Levenshtein*'s, but is still well-suited for GML attributes, which tend to be short in terms of description. This takes advantage of the existing XML query processing capabilities and at the

same time provides the additional capabilities required for GML data processing. GML queries differ from XML queries as they tend to involve larger joins over large datasets. In addition, querying spatial data requires more abundant resources than those needed for relatively simple alphanumeric data [110].

XQuery has been designed to meet the requirements of an XML query language, as identified by theW3CXML query working group [25]. Vatsavai extended XQuery as a base for a GML query language due to its more robust functionality than other solutions [135]. It can cope with complex queries involving different types of joins, and serves as the current standard. XQuery also allows extension functions that can include spatial operations such as intersects. Several other approaches for developing query languages have been proposed [9, 31, 135]. However, an important consideration when developing such languages is to embed in them a native form of similarity measure that can compute the amount of "likeness" based on the concept of shared features. *Semantic Footprints* is an effective candidate for such purpose.

It requires several items to be known: spatial distances between entities, shared attributes, and knowledge of any ontologies being used. The *Levenshtein Distance* has a strong filtering characteristic. It tends to favor entities that appear frequently in only a few documets, but which is not very frequent throughout all documents. *Semantic Footprints*, on the other hand, allows filtering at three levels. When utilized in combination with a threshold, the application can be tailored to consider only entities within a certain spatial distance, or a minimum number of shared attributes, or within a certain ontological distance. Since GML commonly replicates many spatial features across the document collection, these measures are helpful in eliminating an excessive number of entities. This has a direct impact in the indexing approach adopted by the application.

Indices contain data storage information that can speed up searches [123]. As mentioned earlier, GML documents can either be stored as is or the data can be stored in a database and converted to GML when required. Existing spatial indexing techniques can be used for storing GML data in databases. Well-established approaches, such as R-Trees, have a hierarchical structure that is suitable for fast retrieval of spatial data, though such an implementation may consume a great deal of computation power. Other variations, such as R* Trees and R+ Trees, insert objects in distinct paths, making them exclusive to a node. Z-Order B-Trees provide access control to objects while permitting concurrency control, though this may impose a high performance cost. Depending on the nature of the spatial data, Quad Trees, may or may not be as suitable for spatial data as the other approaches, though they still offer a fast search method. Table 4.8 lists some of the more common indexing approaches currently available.

It is important to constrain data for efficient query execution, which can be greatly enhanced by implementing special indexing techniques such as the ones listed in Table 4.8. There are two approaches that can be used to search XML documents: searching value and searching structure. These approaches somewhat mirror semantic analysis by content and structure, respectively. Similarly, the indices for XML documents are also divided into two categories: path indices and value indices. Path indices are used for regular path expressions, while value indices are used for locating objects in the XML documents. Spatial indices assume a structure that can support spatial data.

Table 4.8: Comparison of indexing techniques for spatial data

	Indexing Approach	**Advantages**	**Drawbacks**
✪	R-Trees	➟ Efficient spatial data manipulation ➟ Nested multidimensional structure ➟ Nodes map to disk pages for easy access ➟ Balanced tree structure	➟ Needs extra filtering to remove redundant objects ➟ Less efficient for non-interval-shaped objects ➟ More disk access and computation
✪	R*Trees	➟ Minimizes region overlap ➟ Avoid multiple search paths ➟ Uses reinsertion to improve storage use	➟ Large CPU time needed for reinsertion ➟ Needs efficient node-splitting for tree balancing ➟ Less robust filtering
✪	R^+Trees	➟ No object overlap for faster searches ➟ Searches follow single paths	➟ Can disperse data in more than one page ➟ May cause object redundancy ➟ Requires partition of data space to avoid overflow
✪	Z-Order B-Trees	➟ Reorganizes itself after small changes ➟ Robust concurrency control mechanism	➟ Performance cost on insertion and deletions ➟ Less efficient for geospatial applications
✪	Quad-Trees	➟ Quick access and manipulation of objects ➟ Good for recursive image processing ➟ Fast searching	➟ Large space requirements ➟ Less efficient for high-dimensional data

Common approaches such as R-Trees and Quad-Trees may therefore be helpful, but their usage ultimate depends on whether the GML data is being directly deposited into the storage system, or whether it is first parsed and then stored as text or in some other format.

4.4 Conclusion

Geospatial applications can benefit from GML's robust functionality and the technologies that enhance it. When it comes to semantic and syntactic entity analysis, however, several obstacles hinder GML usage. For this reason, we have described several areas of GML applications and some points of concern in this study.

The first consideration is the storage of GML documents, and the different database models that can be used to support GML data. The proper utilization of GML schemas is fundamental in the implementation of geospatial applications, as there are advantages and disadvantages to handling GML data in each of the available formats. Similarity measures based on a combination of structure and content can help alleviate some of the false positives and negatives associated with GML data.

Further, system designers must understand which parsers and query languages are suitable for GML, and which indexing strategies can be applied to GML documents. In these cases, entity similarity approaches based on *TF-IDF*, vector space, and *Semantic Footprints* tend to handle attribute values robustly.

Nevertheless, each one of these approaches impose conditions and obstacles which must be overcome. Data analysis will then be accomplished in its most accurate and meaningful manner.

Chapter 5

Spatial Similarity in Graph Networks

Social media have ushered in alternative modalities to propagate news and developments rapidly. Just as traditional IR matured to modeling storylines from search results, we are now at a point to study how stories organize and evolve in mediums such as *Twitter*, a new frontier for intelligence analysis. This study takes as input *Twitter* feeds and extracts and connects entities into interesting storylines not explicitly stated in the underlying data. First, it proposes a novel method of spatio-temporal analysis on induced concept graphs that models storylines propagating through spatial regions in a time sequence. Second, it designs a method to control search space complexity by providing regions of exploration. And third, it devises *ConceptRank* as a ranking strategy that differentiates strongly-typed connections from weakly-bound ones. Experiments on the Ukraine political crisis and Mexico civil unrest demonstrate storytelling's high application potential, showcasing its use in event summarization and the forecasting of events before they hit the newswire.

5.1 Introduction

Social media, e.g., *Twitter*, have provided us an unprecedented opportunity to observe events unfolding in real-time. The intelligence community has embraced its power, but has an ongoing struggle on how to incorporate its vast resourcefulness. The reason is that the rapid pace at which situations play out on social media necessitates new tools for capturing the spatio-temporal progression of entities (i.e., people, organizations, events, and objects). Take for instance the *Boston Marathon* bombings of April 15, 2013. In the immediate days afterward, law enforcement officers collected a significant number of eyewitness accounts, photo and video footage, and background information on several suspects who were spatially and temporally tagged. What followed was a succession of outcomes: several people were detained near the blast spots; the residence of a Saudi national was searched; MIT police officer S. Collier was killed; the Tsarnaev brothers were identified as two suspects. All these developments could be observed on *Twitter*, but to the best of our knowledge there exists no automated tool that can provide picturesque analyses from tweets.

The underlying problem is one of *storytelling*, the process of connecting entities through their behavior and actions [134]. In this work, unlike other traditional methods, an event is simply treated as a special type of entity that represents actions, such as a "riot" or a "protest". *Information retrieval* and web research have studied this problem, i.e., modeling storylines from search results, and linking documents into stories [67][49][51] (the terms *stories* and *storylines* are used interchangeably). Textual *storytelling* attempts to link disparate entities that are known ahead of time, such as the connections between two individuals. In this study, however, our focus is not traditional text analysis. Rather, we explore spatio-temporal entity analysis, which can fill some of the gaps left by traditional approaches. Our goal is to not only find meaningful connections, but also to derive new stories for which we do not know the endpoint, if one exists. For example, we would be interested in examining the passing of a new law and the reactions it provokes, such as protests in nearby areas. This falls in the field of exploratory analysis where the main focus is discovering new patterns of knowledge that is so pervasive in intelligence analysis. We target spatio-temporal techniques on short, ill-formed text of *Twitter* data for which deriving stories has proven to be a difficult task.

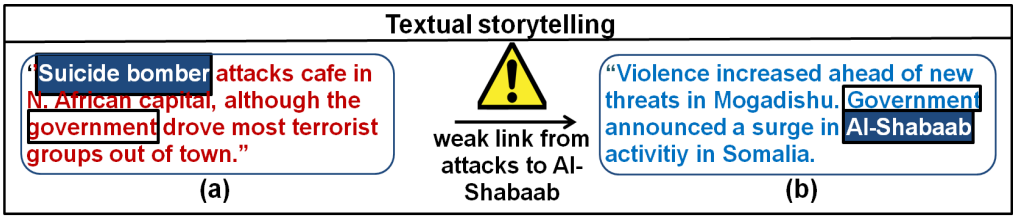


Figure 5.1: Under *textual storytelling*, (a) and (b) represent two partial NY Times articles (2013). The two documents are weakly connected because no patterns other than two "government" entities relate the two documents, making the link between the "suicide bomber attacks" and the "Al-Shabaab" terrorist group of difficult identification.

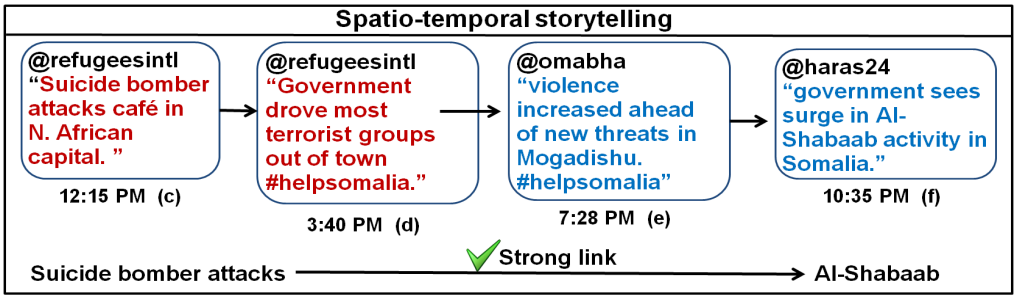


Figure 5.2: Under *spatio-temporal storytelling*, (c)(d)(e) and (f) show four tweets with similar content to the NY Times articles of Fig 5.1. These tweets are strongly connected through the following features: *Twitter* user "@refugeesintl" in (c) and (d), hashtag "#helpsomalia" in (d) and (e), and locations "Mogadishu" and "Somalia" in (e) and (f). Together, the four tweets provide a stronger belief that the suicide bomber attacks are indeed linked to the Al-Shabaab terrorist group.

Textual storytelling has been mostly successful on news articles, blogs, as well as structured databases. In general, it makes one strong assumption: the availability of comprehensive data sources, where textual content is robust and ideas are well presented. In this manner, it is able to perform document analysis using several techniques, some of which include vector-space measures such as *cosine similarity*, natural language processing (NLP) for *parts-of-speech tagging*, and keyword matching, among others.

A common problem with such methods is that inferences may be missed whenever linkage among documents cannot be strongly asserted. Consider the example of Fig. 5.1. In **(a)**, a partial *NY Times* news articles describes a suicide bomber attack in Somalia in 2013, whereas **(b)** tells about a surge in terrorism activity. If the goal is to establish correlation between the suicide bomber in document (a) and the terrorist group Al-Shabaab of document (b), we would first have to link the two documents. Deriving this link is difficult for the following reason: except for government - government, no other terms are shared between the two documents. A simple *cosine similarity* calculation would yield a low score, and the suicide bomber - Al-Shabaab link would most likely be missed due to weak connectivity between the two sources.

The above example illustrates why techniques that apply to textual storytelling tend to perform poorly on social media content, such as *Twitter*, where text lacks proper form and function, and word matching can be challenging. For this reason, social media storytelling demands new techniques that can benefit not only from its textual content, however limited, but also from embedded tweet features. These features come in two flavors: (1) spatio-temporal knowledge of the entities described in text; (2) and intrinsic characteristics of social media represented in the form of metadata. An example of how these features can be helpful is given by Fig. 5.2 **(c)(d)(e)** and **(f)**, which shows four hypothetical tweets modeled after the *NY Times* documents of **(a)** and **(b)**, but written in a more "*Twitter*-like style" (showing the emitting users and some hashtags). Just as in the *NY Times* example, performing *cosine similarity* on any pair of the four tweets would also yield meaningless results, given that very few terms are shared. At closer investigation, however, *Twitter* data allow us to link all four documents through different means. First, tweets **(c)** and **(d)** can be linked because they were issued by the same *Twitter* user (**@refugeesintl**). Second, tweets **(d)** and **(e)** are connected through the *hashtag* **#helpsomalia**, a strong indication that they address the same general topics. To close the gap, tweets **(e)** and **(f)** are connected by location: geocoding Mogadishu and Somalia allows one to determine that the *latitude/longitude* of the former is enclosed in the latter, and thus making them geospatially related. Now that all four tweets are linked, it becomes possible to discover a connection between the desired suicide bomber - Al-Shabaab entities, one attractive aspect of this approach that textual storytelling did not cover.

As simple as this example may be, it shows that tweets can be linked in many ways, such as by users, locations, and hashtags. This study strongly emphasizes the **spatio-temporal** aspect of the data, considering only tweets that have locations and timestamps. Other features, which we explain later, are also available. Five aspects of this approach are noticeable. First, it allows us to create a short storyline that, as concisely as possible, represents the four tweets without replicating them. The storyline that we envision has the following format:

$$\text{suicide bomber} \xrightarrow{attacks} \text{government} \xrightarrow{increased} \text{violence} \xrightarrow{surge} \text{Al-Shabaab}$$

.

It is composed of a sequence of entities identified in the tweets, such as suicide bomber and

government, and relationships, such as $\overset{attacks}{\rightarrow}$ and $\overset{surge}{\rightarrow}$, also from the tweets, which serve to make connections between the entities. The first entity in the sequence (suicide bomber) is the storyline's *entrypoint*, whereas the last one (Al-Shabaab) is the *endpoint*. Note that storylines do not necessarily follow grammar rules since they are meant to capture the semantics of the data stream rather than the syntax of the language. Later sections will explain how to create storylines and discuss other mechanical aspects, such as why some entities are included while others are ignored, and how to use the relationhsips. Second, storylines can be made as elastic as necessary by injecting new tweets in an incremental approach. Third, when represented as a graph, a theoretically-unlimited number of tweets can be collapsed into fewer entities and their corresponding relationships. For example, government or Al-Shabaab may appear thousands of times in the raw dataset, but in this approach, they are only represented once each, minimizing resource usage. In this manner, the number of generated storylines tends to be several orders of magnitude smaller than the number of tweets that generate them; fourth, they enforce time sequencing, which promotes storyline coherence by preserving the order of facts. In Fig. 5.2(c), the storyline begins at 12:15 PM when the "suicide bomber attack" takes place, and ends with Fig. 5.2(e) at 10:35 PM when the "government announces the surge in Al-Shabaab activity.". Fifth, graph structures are more machine-friendly than file systems, allowing efficient searches, spatial operations, and automated data mining.

The importance of location and time: Applying traditional network analysis tools to find and link entities across tweets can lead to 'runaway' stories. Three important problems have to be surmounted. First, to ensure meaningfulness, we must use spatio-temporal coherence as both a desirable aspect of stories and as a way to control computational complexity. It is desirable because entities might be related to one another only under certain circumstances, and modeling spatio-temporal coherence ensures explainable stories. It is a way to control computational complexity because it avoids searching for stories that might not be central to the topic under consideration. For instance, tweets that refer to *suicide bombing* in *South America* are most likely not related to *suicide bombing* cases in *Somalia*. Thus **spatial** is a fundamental consideration. Second, time and space must support the notion of typing to connections. For instance, a suicide bombing $\overset{met-with}{\longrightarrow}$ Al-Shabaab link can potentially be inferred by the intelligence analyst if these entities are both in **proximal areas** and **close in time**. Otherwise, stating that one is related to the other in different places and times is mere speculation. Again such a notion of typing aids in both explainability and scalability objectives. Third, we require algorithms that can operate without specific provision of start and end points as long as entities can be coherently identified **in a location** and **within a timeframe**. The ability to support these dynamic aspects of storylines as they evolve is critical to modeling fast-moving social media streams such as *Twitter*. The goal of this study is to address the above issues and enhance the current state of storytelling. The key contributions are:

1. **Modeling short text over space and time**: This research describes arguably the first algorithm to conduct storytelling without specific endpoints (i.e., without supervision) over short text (tweets), represented as an entity graph, and provides strategies to enforce coherence,

precision, and the influence of spatial entity types on the generated storylines.

2. **Reasoning over spatio-temporal features**: Key to obtaining coherent stories is to identify regions of spatial propagation where related entities cluster. We demonstrate the use of *Ripley's K* function for this purpose and its use in conjunction with temporal propagation where time windows help keep stories succinct and coherent. In combination, they limit the search space from possibly millions down to the thousands of entities.

3. **Devising spatio-temporal storylines based on connectivity**: We provide a parameter-free relevance measure based on *ConceptRank*, which differentiates relationship types, boosts strongly-connected spatial entities, and helps eliminate large numbers of poorly-connected ones. In addition, storylines are found "on the fly", demonstrating our ability to generate lines of exploration that span across space and time.

4. **Performing extensive experiments on social media**: To show the effectiveness of spatio-temporal storytelling on *Twitter* data, this approach is evaluated on current world events that encompass social unrest in Mexico, the political crisis in Ukraine, and the evolution of the *Syrian Civil War*. Included is a comparison of this approach to others based on an event summarization task, and the discussion of a case study related to event forecasting. The experiments also discuss the impact of spatio-temporal parameters and the importance of *Twitter* features to stortyline generation.

Throughout this study, various components needed for storytelling are introduced. Section 5.2 elaborates on existing work, highlighting differences. Sections 5.3 and 5.4 explain the spatio-temporal mechanics of entity discovery, ranking, and storyline generation. Experiments are presented in Section 5.5 and a conclusion is given in Section 5.6.

5.2 Related Works

Storytelling is not a single analytical task with a self-contained purpose. It can be better understood as a framework of intelligence analysis in which various tasks can be accomplished by different means. Very broadly, entities must be extracted, ranked, and connected, in order to make storylines visible. In this sense, storytelling involves a mix of quantitative analysis and semantic reasoning over which the boundaries are flexible. Similarly, the work proposed in this research spans many areas of expertise, from clustering to geographic networks. This research best lines up with the approaches described below.

Storytelling: The phrase 'storytelling' has been introduced in an algorithmic context by *Kumar et al.* [67] who proposed it as a generalization of *redescription mining*. At a high level, *redescription mining* takes as input a set of objects and a collection of subsets defined over those objects with the goal of identifying objects described in two or more different ways. Such objects are interesting

because they may signal shared characteristics and similar behavior, which can be a powerful tool in the context of *storytelling*. One such algorithm is *CARTWheels* [108] which utilizes induced classification trees to model redescriptions along with the *A* Algorithm* for least-cost path traversal. [51] develop this idea to connect two unrelated PubMed documents where connectivity is defined based on a graph structure, using the notions of hammocks (similarity) and cliques (neighborhoods). This work was generalized to entity networks in [50] and specifically targeted for use in intelligence analysis. Their motivation is that current technology lacks better support for entity linkage, explanation of relationships, exploration of user-specified entities, and automated reasoning in general. The tools used in this work include concept lattices as a network where candidate entities are identified with three nearest neighbor approaches (Cover Tree, k-Clique, and NN Approximation). The *Soergel Distance* measures the strength between entities, while *coreferencing* serves to identify entities mentioned in various parts of the text using differing terms. All these works require specific start and endpoints, and link entities according to a desired neighborhood size and distance threshold. In many of these works, edge weight has been based on a variation of term frequency $\times$ inverse-document-frequency (*TF-IDF*). This class of works represent *traditional storytelling* approaches even though neighborhood distances are considered, albeit not from a geospatial perspective.

Connecting the Dots: The primary focus of these works is on document linkage rather than entity connectivity. For this reason, textual reasoning is a strong facet of the targeted methods, which departs from a spatio-temporal view of events. Endpoints must (again) be specified and link strength utilizes the notion of *coherence* across documents, which is proposed by [120]. In this work, stories are modeled as chains of articles, where the appearance of shared words across documents help establish their relatedness. Another important aspect is the determination of *influence* between documents based on the presence of a given word. For this purpose, a bipartite graph is built using documents and words as nodes, where edge strength among them can be obtained by third-party tools or with *TF-IDF* scores. Extending that work, they also propose related methods to generate document summaries, i.e. *Metro Maps*, in [122] and [121], which target scientific literature. Some of the goals are to measure the importance of an article in relation to the corpus, find the probability that two papers originate from the same source, and identify research lines. The basic data structure is also a directed graph, where for each map that has been generated, its *coverage* is calculated using each document as a vector of word features. The *coverage* is then defined for a set of words as *TF-IDF* values, which can be extended to sets of documents. Connectivity between maps is measured by the number of paths that intersect two maps. Overall, *connecting the dots* methods rely heavily on the abundance of robust content such that the aforementioned calculations (coherence, influence, coverage, etc.) can be calculated acceptably. Social media, however, breaks the assumption of robust content, limiting the amount of textual reasoning that can be performed. Thus, *connecting the dots* is less than ideal for environments that utilize *Twitter* data feeds.

Relationship Extraction: One of the most fundamental tasks in storytelling is the extraction of relationships from text. When correctly identified, the relationships that link entities lead to more robust and coherent facts. This effort often involves the introspection of data items at their

most granular levels: documents are usually broken down into paragraphs, which are subsequently decomposed into paragraphs, sentences, and words. Once this pre-processing has come to completion, analysis can take a few different directions, of which three of the most popular are described below:

1. **Kernel-based relationship extraction**: A kernel function K maps a pair of objects x and y to a similarity score $K(x,y)$. Kernel functions obey the properties of *symmetry* and *positiveness*, making them attractive to GIS applications that require metric compliance. One of the most popular families is *support vector machines* (SVM) introduced by Cortes and Vapnik [32], which attempts to find a separation margin between entities based on some of their attributes. This ability to seggregate objects has proven successful in classification tasks, and could be promising for spatio-temporal storytelling. The intuition here is that in many investigative scenarios, certain relationships are only feasible when their composing entities have specific features, and thus classifying them first enforces coherence. SVM kernels are considered linear when a straight line separates objects of similar features from their disparate counterparts. In many datasets, however, linear separation is not possible, giving rising to other separation strategies based on curves or circular shapes [33]. Zelenko proposed a relation extraction method in which the kernel function is based on *parse trees*, ie, a minimal representation of words based on their syntactic grammatical structure [150]. A contrasting approach was proposed by Miller et al. where extraction utilized relation-specific attributes, not simply syntactic parsing [90].

2. **Feature-based relationship extraction**: For this family of approaches, the goal is to construct for each entity a vector in which the features have a strong degree of differentiation. When operated over many training samples, an objective function is learned such that for a given new vector, the function can classify it correctly. Roth and Yih develop a method for recognizing relations and entities in sentences taking mutual dependencies among them into account [112]. Kambhatla devises an extraction method that looks for pairs of mentions based on different features such as entity types and words between mentions [58]. Guodong et al explore phrase chunking in conjunction with lexical and syntactic knowledge to build feature vectors from aspects such as bag of words and words before and after [46]. Feature approaches have been known to perform well in answering granular questions such as which two people were talking within a group.

3. **Knowledge-based relationship extraction**: As with others mentioned previously, this line of research also utilizes linguistic knowledge. Unlike the others, however, it builds patterns based on speech and semantics. Appelt et al, for example, construct submodules such as single-word and word-sequence tokenizers to build such patterns and customize them for different domains [7]. Knowledge-based approaches are popular not only for relationship extraction, but also for the identification of named entities. Callan et al [20] propose a rule-based method, namely *KENE*, where each rule is a sequence of markup tags and punctuation. Strings that are extracted are then looked up in a database of named entities so that matches can be identified. With the identified entities, the task of deriving their relationships becomes

less complex. Other methods, which can also be considered probabilistic, include *Hidden Markov Models* such as in the work of Skounakis [125] that extracts relations from biomedical articles and Lafferty et al [68] that segments and label sequence data with *Conditional Random Fields*.

All of the above methods provide a certain amount of contribution to the relationship extract process, while imposing limitations. Kernel-based methods can be challenging if the data distribution does not allow a minimum amount of differentiation among entities. Otherwise, it can be precise and efficient. Feature-based models are helpful if entity descriptions are robust and many features can be obtained, but otherwise can be computationally costly in the number of comparisons that it makes. Knowledge-based approaches have relied on frequent patterns or sequential data, which may not always be available. In the context of spatio-temporal storytelling, all of these techniques appear fundamentally sound, warranting further investigation that the current literature does not yet provide.

Event Detection and Summarization The goal of storytelling is to find meaningful streams of information that are neither spelled out in text nor apparent to the naked eye. As such, storytelling should not be labeled as an event detection technique or a summarization tool. However, because stortytelling captures the underlying relationships among entities, it can serve broadly to summarize real-world developments or to aid in event forecasting.

In terms of event detection, event expansion and topic trending are two commonly-studied aspects. Event expansion starts with limited bits of information about an event and seeks to expand it using social media data. Topic trending, on the other hand, monitors large volumes of social streams to find the most popular themes of discussion. The work of Sakaki *et al* ([114]) targets the detection of earthquakes in Japan using common classification techniques. Events are defined by the user by selecting keywords. TEDAS [76] describes a system for detecting new events related to crime and natural disasters, and identify their importance. It first crawls tweets, classify them as event-related or not, and stores spatio-temporal information. Users then issue queries that contain location, time, and keywords, which the system uses to retrieve and display related events. The importance of event reporting over *Twitter* is questioned by the work of Petrovic *et. al* [101]. The authors claim that the benefit of tweets comes from increased coverage, not timeliness. They devise a system that clusters both tweets and news articles, and measure their overlap to discover the coverage of one versus the other. Comparisons can then be done on their spread over time. Twevent [75], a different approach, proposes segment-based event identification. Initially, it detects bursty events and clusters them using frequency and content similarity. The similarity between segments is computed using their associated tweets, while Wikipedia is searched to verify which events are realistic or not. In [138], Walther and Kaisser monitor specific locations of high tweeting activity. They further analyze clusters of those tweets, using machine learning to detect if the identified posts during high activity represent real events or not.

Textual summarization has been well studied in IR, using a wide variety of techniques, such as latent semantic analisys and machine learning [43, 34]. Event summarization, as an extension, has

gained strength in recent years due to social networks. *TwitInfo* describes a system that allows users to navigate a repository of tweets, where the system discovers high peaks of twitter activity [85]. In addition, the system allows geolocation and sentiment visualization. A more comprehensive approach to event summarization is detailed in [24]. The authors propose a segment-based approach where summarization takes places within each segment. This technique can take on different variations. The first uses cosine similarity as a straightforward method. The second applies a similar approach, but considers tweets that fall within a specific time window. A third approach uses a Hidden Markov Model (HMM) where each state can be a sub-class of events (e.g., "touchdown" in a football event). An alternative technique also based on time segmentation is given by [87], but with the added assistance of synonym expansion for keywords. For each of these approaches, the output is the set of tweets that best summarizes the events.

Differences: Each of the above approaches have different goals and apply vastly different techniques to accomplish their objectives. As a result, direct comparisons to this proposal must be done carefully. For example, this study does not seek to summarize or detect events as the end goal, but to show them as potential uses. An adequate description for storytelling is to determine how entities are involved in particular events, and show them as a meaningful storyline. This approach relies on a spatio-temporal model in which both geographical proximity and time ordering are favored over textual content. Most of the other approaches, instead, rely on the textual nature of documents. For these reasons, this study does not propose a competing method. Rather, it shows complementary spatial techniques that fill a niche which has remained mostly understudied. The experiments section compares this proposed approach to three methods described in [24] and [87], explaining the differences along the way.

5.3 Spatial Modeling

This section provides a visual representation of the proposed methods and explain the technical aspects of spatio-temporal storytelling. Fig. 5.3 shows the three stages taken: (1) in the preprocessing stage, entities such as people and events, as well as concepts (i.e., relationships), are extracted from *Twitter* data. Combining the extracted entities and their relationships allows a concept graph to be constructed; (2) in the spatio-temporal modeling stage, entities are discovered in regions through which a storyline is most likely to propagate, using the concept graph to further rank those entities, and temporally order them; (3) Storylines are then generated using the highest-ranked entities and their observed relationships. First, the definitions used throughout the remainder of this research are provided. **Definitions**: In the scope of this study, an entity network is a graph $G(E,R)$ where entities $E=\{e_1, ..., e_n\}$ can be linked to one another through relationships $R=\{r_1, ..., r_n\}$ defined by conceptual interactions, and thus called a *concept graph*. Given a set of documents $D=\{d_1, d_2, ..., d_n\}$, the following definitions apply:

Definition 1. *An entity e represents a person, location, organization, event, or object described in at least one document $d_i \in D$. Only entities for which a location and a timestamp can be obtained*

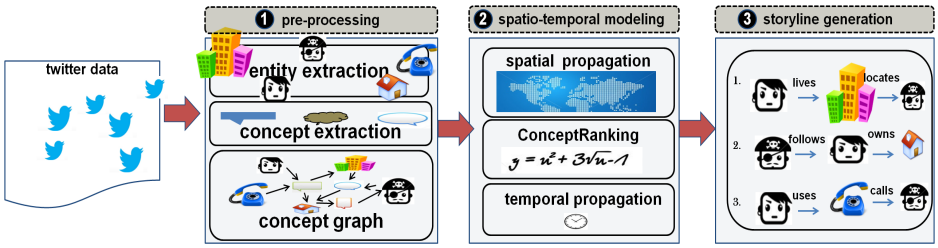


Figure 5.3: Three-step process for spatio-temporal storyline generation using *Twitter* data: (1) In the pre-processing stage, entities and concepts (relationships) are extracted and used to build the concept graph; (2) Under spatio-temporal modeling, spatial propagation first discovers entities in nearby locations. For each entity, *ConceptRanking* determines its relevance in the graph, and the entities are subsequently time-ordered for proper temporal propagation; (3) Storylines are then generated by linking the *top-k* ranked entities in time order.

are considered in this study.

Definition 2. *An event represents a special type of entity denoted by an action. Our previous examples mentioned several events such as an "attack" and an "explosion".*

Definition 3. *A semantic constraint is a user-defined data delimiter similar to a query parameter. For example, if one seeks stories related to "explosion" and other related terms (e.g., "bombing" or "blast"), he/she may use these terms as semantic constraints to guide the storytelling process toward those concepts.*

Definition 4. *A relationship, connection, or link defines a unit of interaction between two entities and is denoted by* $e_i \xrightarrow{interaction} e_j$. *It is deemed explicit if it is extracted from tweet text, such as in* $D.Tsarnaev \overset{talks-to}{\rightarrow} T.Tsarnaev$. *A relationship is implicit if it comes from metadata, as in the Twitter case of "follows". Note that all relationships* $e_i \xrightarrow{interaction} e_j$ *are intended to be directional.*

Definition 5. *An entrypoint is any entity e in the dataset and the point from where the story evolves. It is application-dependent from the perspective of the intelligence analyst. For instance, in the Boston Marathon Bombings scenario, the entrypoint can be the blast site (i.e., a location), an individual seen in the vicinity (i.e., a person), or any other entity of interest. The endpoint is the entity where the story ends.*

Definition 6. *A storyline is a time-ordered sequence of n entities* $\{e_1, ..., e_n\}$ *where consecutive pairs* (e_i, e_j) *are linked by one relationship. The number of entities n is the length of the storyline.*

Twitter Features: In order to capture the importance of entities, both tweet metadata and textual content are used in the following manner:

1. **users** are person entities and the subject and objects of **mentions**, **reply-to**, **following**, and **follower** relationships. They help establish implicit relationships, as defined above in 9.

2. **countries**, **states/provinces**, **cities**, and **addresses** are geocoded and become location entities, both coming from metadata and text. Tweets without location are not considered.

Figure 5.4: Concept graph example. The solid lines between entities represent *explicit* relationships extracted from tweet textual content. The dashed lines denote *implicit* relationships from tweet metadata.

3. **hashtags** implicitly link entities either in the same or across tweets.

4. **created At** (from tweet metadata) and **dates** (when available from tweet text) are both used for temporal analysis. Whenever an entity is extracted from text, a timestamp is associated to it. If the tweet text has an inline timestamp that can be associated to the entity, this timestamp will be used. Otherwise, the timestamp of the tweet metadata is used instead. Dates extracted from text are always given preference, if available.

5. **organizations** are extracted from text (i.e., not metadata).

Fig. 5.4 shows a simple concept graph related to the *Boston Marathon Bombings* where the entities were extracted from several tweets. Solid lines represent explicit relationships, while dashed lines denote implicit ones. We have the following: D. Tsarnaev (D.T.) and T. Tsarnaev (T.T.) are connected through a "talk" relationship, which was extracted from *Twitter* text (not *Twitter* metadata), and is thus defined as explicit. The same is true for the "meets" link between T.T and S. Collier (S.C.), the "works" link from S.C. to MIT, and the "drives" link from D.T. to MIT. The various links to other unknown entities (small triangles) come from *Twitter* metadata ("follows", "following"), and therefore are implicit.

The reason to differentiate the above relationships comes from a simple notion: in entity networks such as *Twitter*, semantic closeness in the form of social interactions is probabilistically correlated to spatio-temporal proximity [45] akin to Tobler's first law of Geography, in which similar things tend to be near one another. Intuitively, this notion has several implications to storytelling:

- **Relationship Typing**: explicit connections are more helpful than implicit ones. Knowing that "D.Tsarnaev spoke to T. Tsarnaev" is more powerful than simply learning that "D.Tsarnaev mentions (in the *Twitter* sense) T.Tsarnaev". This idea is explored in Subsection 5.3.2 about Concept Ranking.

- **Relationship Propagation**: a story can be modeled as a graph of entities and semantic relationships propagating through spatially-close regions in a temporal sequence. Consider Fig. 6.5(a) which depicts several locations related to the *Boston Marathon Bombings*. Most of its developments took place in an 8-day interval (Apr 15-22, 2013) and in proximal areas:

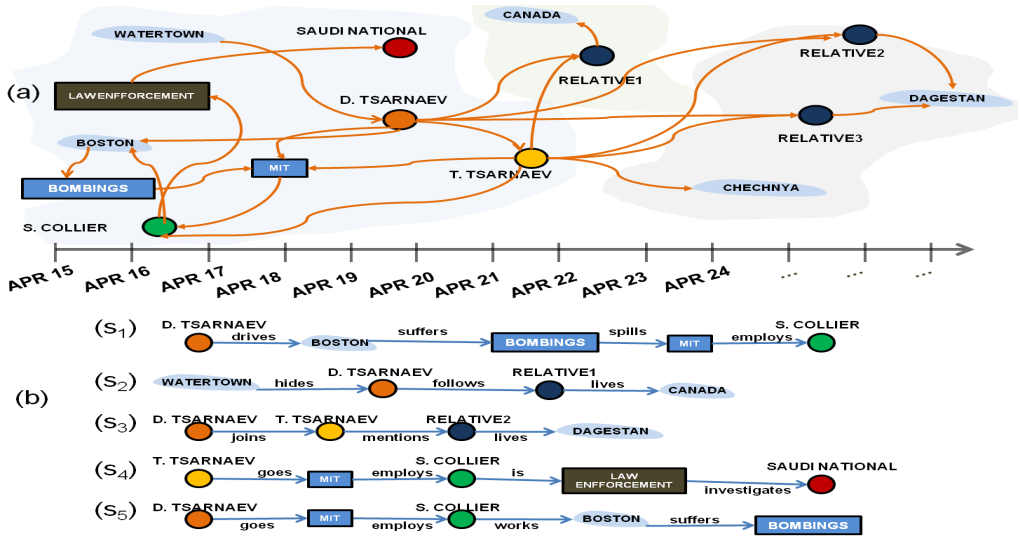


Figure 5.5: Boston Marathon Bombings spatio-temporal sequence. In (a), each shape represents an entity observed in a tweet. The edges denote relationships between the entities. In (b), S_1 through S_5 represent five storylines connecting different entities. The english verbs define their relationships and correspond to the edges of the concept graph in (a).

Boston - MIT Campus - Watertown. Developments in Canada or Chechnya are an evolving part of the story, but do not necessarily play a major role. Based on these ideas, Section 5.3 defines spatio-temporal propagation in order to explore constrained regions of entity connectivity where stories can evolve from.

- **Relationship Boundaries**: stories do not necessarily have endpoints. Entities come and go, relationships develop, and locations vary. In the *Boston Marathon Bombings*, the entry point could be any one of thousands of persons. The end could propagate through Canada, Russia, and other places. This idea is applied in the experiments section to further justify the use of evolving stories.

5.3.1 Spatial Entity Discovery

In the process of telling a story, the entrypoint can be any entity such as a person or event, as in the "bombing" scenario. Given an entrypoint, the goal is to delimit a region where the "most amount of information" can be found, and grow that region until seemingly relevant information becomes sparse. To find this region, several techniques could be explored, but not all of them fit spatio-temporal *storytelling* adequately. One of them would be to perform a simple *Nearest Neighbor*(*NN*) search on the area of study and collect the found entities. *NN* searches, however, are "blind" to the dataspace, i.e, they find entities without relaying information about how they disperse, and thus are not used here. Another alternative method is *Pair Correlation Function* (*PCR*) [109], which divides the data space into spatial segments, allowing each segment to be weighted higher (lower) for closer (farther) entities. Spatio-temporal *storytelling*, however, only

needs nearby regions, thus segmenting them does not serve a useful purpose. *PCR*, therefore, is not an ideal choice. Other possible methods are the variations of partitional clustering, such as *K-means*, which could serve to group related entities before linking them. While feasible, this type of clustering demands several initialization centroids, which *stortytelling* does not provide (in our approach, only one entrypoint is initially given). In addition, this early in the process, performing any type of clustering adds complexity that can be avoided by other approaches. Below, we explain a preferable method.

Consider Fig. 5.6(a) where each point represents a person who tweeted during the *Boston Marathon Bombings* near the blast sites. Circle A designates an area of 1 km around the entrypoint (i.e., blast site) with a high concentration of person entities. If we consider 2 km, as in circle B, the density decreases, while circle C becomes even more sparse. Intuitively, the investigation should focus on the 1 or 2-km radii where most of the information resides. In theory, this is the modeling of a point process (i.e., a collection of persons who sent tweets) in terms of a randomly chosen event E (i.e., bombing) with an estimator distance function for a given density λ, which is given by *Ripley's K-coefficient* $K(r) = \lambda^{-1}E$. Mathematically, $K(r)$ can be stated as:

$$K(r) = \sqrt{\frac{A \sum_{i=1}^{n} \sum_{j=1}^{n} w(i,j)}{\pi n(n-1)}}, i \neq j \tag{5.1}$$

where r is a desired radius originating at a chosen entrypoint, n is the total number of entities in the data space, A is the entire area of study, and $w(i, j)$ represents a weight. $w(i, j) = 1$ if distance(e_i, e_j) $< r$, and 0 otherwise. In effect, $K(r)$ performs a nearest-neighbor search and can be viewed as a clustering coefficient for a desired type of entity (e.g., persons sending tweets) within a limited radius. The coefficient can be evaluated at different scales, such as $r = 1$ km or $r = 1.5$ km. Fig(s) 5.6(b) and (c) show two simple calculations of the *K-coefficient* for 3 persons {P_1,P_2, and P_3} located in a (3 km x 3 km) area A. In 5.6(b), the chosen radius is 1 km. The calculation follows: using each entity P_i as the center of a 1 km circle, count the number of other entities P_j within that radius, adding 1 if their distance is less than the radius, zero otherwise. In that range, P_1 "can see" 2 others (P_2 and P_3), since their respective distances ($dist(P_1, P_2)$ and $dist(P_1, P_3)$) are both less than $r = 1$. Using P_2 as the center of a 1km-radius, P_2 "sees" only P_1. The same is true for P_3, which yields $K(1) = 0.53$. In Fig. 5.6(b), the radius is increased to 1.5 km, and the calculations are repeated, yielding a $K(1.5) = 0.65$.

Comparing the two calculations indicates that the larger radius picked up more points and resulted in more clustering, with the same density. Increasing the radius can potentially find more empty space, which is undesirable. *Ripley's K-coefficient* is an elegant method of discovering related nearby things, but does not tell what a good radius should be or whether lower/higher density is better or worse. *Ripley's* gives us an opportunity to present a set of heuristics that calculates a feasible $K(r)$ in the discussion below.

Finding a feasible *K(r)*: In the previous analysis, one needs a systematic way to determine if the 1-km radius is better than 1.5 km, or vice-versa, to avoid guessing. The region delimited by the radius that yields the highest K-function score is where the storytelling process will initiate. Given

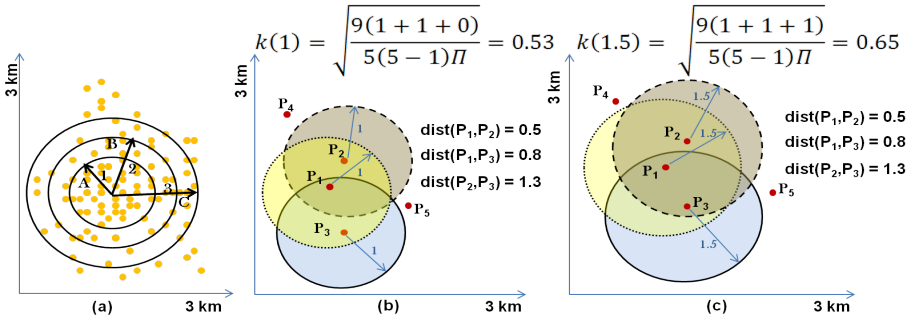


Figure 5.6: Spatial scaling for different radii. (a) Circle A depicts high entity density, becoming more sparse in circles B and C. (b) and (c) shows the calculation of *Ripley's K* function for a 1 and 1.5-km radius respectively.

that a real-world dataset may contain millions of entities, a feasible region is one that includes enough data points, but not all of them. Looking at Fig. 5.6(a), Circle B covers most of the entities in that dataset, which may be excessive for many applications. The problem is that Circle B has a 2-km radius, which corresponds to most of the length of the entire study area of $9km^2$. A better approach in this case can be done according to Algorithm 4, which is explained below.

The first step is to select an initial random radius to work with. Since this ideal initial radius is not known, the algorithm takes a "half-decrement" approach, in which the analyzed radius is cut by half of the length of the dataset iteratively until a reasonable radius is found. This is initiated where the algorithm specifies r_i as 1/2 the length of the data set (r_i in Line 1). This initial radius can be manipulated higher or lower to comply with application needs or when better knowledge of the dataset is known apriori. Using radius r_i from the story's entrypoint, a list of entities is obtained by performing a range query over the spatially-indexed entities in the database L (Line 2). A simple check is then made: if the ratio of retrieved entities (|Ents|) and total number of entities ($|e_A|$) is equal to or greater than a certain threshold, say 10%, then too many entities have been retrieved (Line 3) and they are discarded (Line 5). The algorithm halves the initial radius (Line 6) and tries again (Line 7). Once the calculation hits a point below the threshold, the algorithm has found r_{Limit}, i.e., a radius that covers an adequate number of entities (Line 8).

On its own, r_{Limit} is possibly good enough, but not necessarily the best radius. For example, it is possible that r_{Limit} corresponds to Circle B of Fig. 5.6(a). Ideally, however, it would be better to find Circle A, or even a smaller circle inside of A, as they seem to concentrate most of the entities. The goal, then, is to find the highest clustering coefficient beginning with r_{Limit}, which is stored as $K(r_i)$, through an iterative process, but one which does not exceed threshold T_e. Using r_{Limit}, $K(r_i)$ is computed (Line 9). In successive steps, r_i is incremented by half the value of r_{Limit} and its K is recomputed (Lines 11-17). As soon as $K(r_i)$ stops growing from its previous value or the number of retrieved entities reaches threshold T_e, the process stops. $K(r_i)$ has reached an adequate coefficient for this specific radius, which is output in Line 19. In theory, there is no guarantee the "truly best" radius has been found, but since increments of r_{Limit} become smaller and smaller over many iterations, we hit the law of "diminishing returns" and stop the process for the sake of efficiency. It now can be stated that the storytelling process will include all entities located within

Algorithm 4: Distance Computation

inputs : spatially-indexed entity database L, area A, entity count threshold T_e, number of entities in A $|e_A|$, story entrypoint e
output: radius r_i

1: initialize: i=1; k = i-1; $r_i = \frac{length(A)}{2}$; // set the initial radius as half of the length of the study area (customizable).
2: List {Ents} ← rangeQuery(L, e, r_i) ; // create a list of entities by performing a range query from the radius.
3: **if** $\left(\frac{|Ents|}{|e_A|} \geq T_e\right)$ // compare the list of entities against a desired threshold
4: **then**
5: discard {Ents} ; // if too many entities are found , discard them all.
6: **set** $r_i = \frac{r_i}{2}$; // shorten the radius by half of the previous size.
7: iterate Line 2 ; // and run a new iteration with the new radius
8: **set** $r_{Limit} = r_i$; // save the newly found radius.
9: $K(r_i) = calculateK(r_{Limit})$; // calculate Ripley's K function for the new radius.
10: initialize $K(r_k) = 0$;
11: **while** $\left(K(r_i) > K(r_k)\ and\ \frac{|Ents|}{|e_A|} < T_e\right)$ // run more iterations until K stops increasing and threshold is not met, then output r_i
12: **do**
13: {Ents} ← rangeQuery(L, entrypoint, r_i) ;
14: $K(r_k) = K(r_i)$;
15: **set** $r_i = r_i + \frac{r_{Limit}}{2}$; // as long as $K(r_1)$ keeps increasing, increase r_i by half its previous value.
16: **set** $r_{Limit} = \frac{r_{Limit}}{2}$; // save the new radius temporarily.
17: $K(r_i) = calculateK(r_i)$; // calculate Ripley's K function for the increased radius.
18: **end**
19: **output** r_i;

range r_i of e.

5.3.2 Concept Ranking

In Subsection 5.3.1, r_i is calculated as the radius originating at the entrypoint from where the storyline should propagate. Within that range, many entities can be present, which requires a ranking strategy to determine an order in which entities should be investigated. For this purpose, there are alternative approaches, as in performing textual similarity based on methods such as *cosine similarity* [104] or comparing the values of attributes from each entity [80]. These approaches, however, are efficient on textually-rich sources, but not adequate for *Twitter* data, which are more often than not poorly described. Since this work uses a graph of connected entities as data representation, ranking is proposed as a variation of *PageRank* [18], extended as *ConceptRank*, and explained below.

Given a network of web pages, *PageRank* assigns the highest(lowest) importance to the most(least) referenced page(s), offset by the relevance of the referring page. It is given by:

$$PR_{(p_k)} = \left(\frac{1-\Gamma}{N}\right) + \Gamma \sum_{p \in Links(p_k)} \left(\frac{PR_{(p_i)}}{OL_{(p_i)}}\right) \tag{5.2}$$

where $PR(p_k)$ is the *PageRank* of page p_k, N is the total number of web pages, Γ is a user-defined damping factor in [0..1], $Links(p_k)$ is the set of links to page p_k, and $OL(p_i)$ is the number of outbound links from page p_i. Consider the concept graph of Fig. 5.7, where each node, instead of a web page, is assumed to be a spatially-tagged entity. It can be seen that T.TSARNAEV has

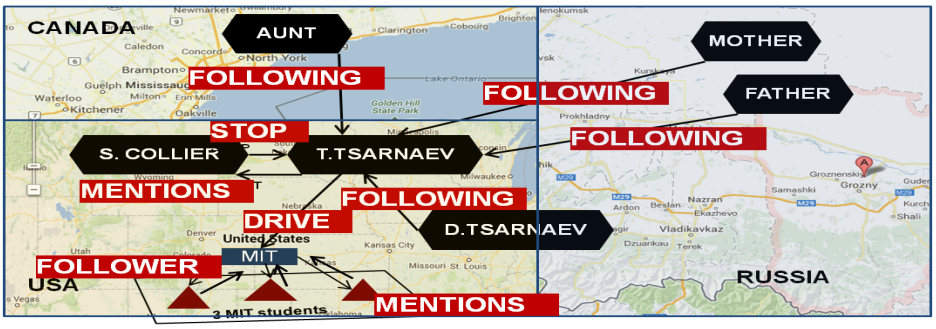


Figure 5.7: Concept Graph with Mixed Relationships. Twitter features such as *following*, *follower*, and *mentions* are considered *implicit relationships*. Others, such as *stop* and *drive* are deemed *explicit*.

the most **inbound links** (5), MIT has four, and S.COLLIER has only one. The other entities have none. Under *PageRank*, the most important entities (i.e., entities with the highest *PageRanks*) would be T.TSARNAEV, MIT, and S.COLLIER since they are the most connected entities.

One notable aspect of *PageRank* is that it does not differentiate relationships. Thus, in Fig. 5.7, "stop" and "drive" have the same influence in the *PageRank* calculation as does "following" or any other relationships. In terms of storytelling, this represents a deficiency because the types of interaction among entities relay strong information and should be accounted for. For example, persons seen around the blast site may hold clues to the bombing. However, students commuting to the MIT Campus from other directions most likely play no role in the bombing. Therefore the types of links influence the story and should be discriminated appropriately.

Given the above discussion, we propose *ConceptRank* not on web pages, but rather on entities, as follows. In a concept graph, the relevance of an entity is determined by a combination of both *implicit* and *explicit* relationships, as stated in *Definition* 9, but differentiated by their respective frequencies. Mathematically, *ConceptRank* is defined as follows:

$$CR_{(e_k)} = \left(\frac{1-\Gamma}{N}\right) + \Gamma \sum_{p \in Links(p_i)} \left(\frac{CR_{(e_i)}}{\psi_{e_i}} + \frac{CR_{(e_i)}}{\Phi_{e_i}}\right) \tag{5.3}$$

where $CR_{(e_k)}$ is the *ConceptRank* of entity e_k, N is the total number of entities in the concept graph, Γ is the same damping factor as before, $Links(p_i)$ is the set of links to page p_i, ψ_{e_i} is the number of explicit outbound relationships of entity e_i, and Φ_{e_i} is the number of implicit outbound relationships of e_i. For all purposes, Φ_{e_i} can be viewed as a *Twitter*-specific parameter obtained from metadata relationships as outlined by the *Twitter features* of Subsection 6.3.1. In real datasets, explicit relationships are less prevalent while implicit relationships tend to abound, making them less useful in a ranking strategy. An illustration follows.

Consider the case in which law enforcement is investigating persons who were **stopped** by a cop, or anybody **driving** to the MIT Campus. The underlined words are the semantic constraints sought on text. The concept graph of Fig. 5.7 depicts a few interactions related to $N = 10$ entities. We set $\Gamma = 0.75$, which can be viewed as the initial *ConceptRank* value that every entity receives regardless of its connections. This parameter can be manipulated. For each entity i, we must first

Table 5.1: ConceptRank Illustration. Network of N=10 entities in Fig. 5.7 with a starting damping factor of Γ=0.75. Entites are ranked from highest to lowest values of *ConceptRank* $CR(e_i)$.

	q=**stop,drive**	N = 10	Γ = 0.75	
i	Entity (e_i)	ψ	Φ	$CR(e_i)$
1	T.TSARNAEV	1 ($\overset{drive}{\rightarrow}$)	1 ($\overset{mentions}{\rightarrow}$)	0.0282
2	MIT	0	3 ($\overset{follower}{\rightarrow}$)	0.0276
3	S.COLLIER	1 ($\overset{stop}{\rightarrow}$)	0	0.0264
4	MIT Students (each)	0	1 ($\overset{mentions}{\rightarrow}$)	0.0250
5	MOTHER, AUNT, FATHER, D.TSARNAEV (each)	0	1 ($\overset{following}{\rightarrow}$)	0.0241

determine its number of implicit (Φ) and explicit (ψ) outbound relationships. S.COLLIER has one outbound relationship ($\overset{stop}{\rightarrow}$), which is explicit since it comes from *Twitter* text (not *Twitter* metadata), and no implicit ones. Thus its $\psi = 1$ and $\Phi = 0$. FATHER has only one outbound relationship ($\overset{mentions}{\rightarrow}$), which comes from *Twitter* metadata, and so is considered implicit. Thus its $\psi = 0$ and $\Phi = 1$. Table 5.1 summarizes the data for all entities, along with their *ConceptRank* (calculations not shown). What the *ConceptRank* values contribute is a ranked list such that the most relevant entities and their relationships can be weaved into a storyline. The ordering goes from highest to lowest values of *ConceptRank*, yielding the following ranking: T.TSARNAEV MIT S.COLLIER MIT students, since these entities have the highest values. The next four entities, (FATHER, MOTHER, AUNT, and D.TSARNAEV) have the same *ConceptRank*, in which case they can be inserted in any order. Given a different mix of implicit and explicit relationships, the ordering may change. In practical terms, *ConceptRank* favors the most well-connected entities, punishing the ones that are thinly-referenced in its spatial region. In the next section, we explain that only the top ranked entities (according to a threshold) are considered. All others are disregarded, preventing them from taking part in the story generation process.

5.4 Spatio-temporal Propagation

In this section, entities that were previously extracted from the datasources are organized such that they are not only spatially-correlated, but also time-ordered in a way that makes sense to the human mind. The key concept here is that entities evolve along space and time, and thus the notion of *spatio-temporal propagation* becomes an integral part of the storytelling process. Spatio-temporal propagation requires a strategy that prevents sequential contradiction, which is addressed with the use of *time windows* in Subsection 5.4.1. Time windows must be treated carefully as it raises several design questions, which are addressed in Subsection 5.4.2.

5.4.1 Devising Time Windows

One important aspect of intelligence analysis is the sequence in which real-life developments take place. In the *Boston Marathon Bombings* scenario, for instance, it is clear that the BOMBING event should precede the arrest of suspect D.TSARNAEV, and not the other way around. Temporal propagation over *Twitter* data is challenging for three reasons:

1. **Varying lengths**: in many instances, entities are spread throughout long periods of time (e.g., a war), while in others, the time span can be very short (e.g., a terrorism act). Therefore, varying-length time intervals must be accounted for;

2. **Bursty behavior**: often, entities display disparate frequencies in arrival rates. In an initial time period, for example, millions of tweets can be issued due to a high-visibility event (e.g., Barack Obama's election). But that same event may subside over time when it is no longer considered "news". Thus, distribution becomes important;

3. **Time synchronization**: many entities may be observed at the same time, in which case ordering them is not intuitive. Therefore, ties must be somehow dealt with.

One way to get around the above problems is to utilize a *time matrix*, which provides an intuitive way of aggregating spatial entities in flexible time intervals. In a *time matrix*, each column is a *time unit* and each row is a fraction of the *time unit*. Each cell of the matrix holds the entities observed at specific times. Fig. 5.8(a) shows an example where each column represents one day of the week (i.e., the *time unit*), and each row represents the time of the day. A *time matrix* permits entities to be observed as a sequence of interactions and can be made as short or as long as the situation dictates. One can then perform data analysis on the entire matrix or on a subset of rows and columns, which we denote as a *time window*. In the scope of this study, a *time window* is defined with a simple rule:

Definition 7. *Given a time matrix of interest (TM) composed of n time units (TU), a time window (TW) is composed of one or more* TU_i *where* $0 < i \leq n$. *In other words, a time window corresponds to a pre-defined time interval or a subset of it.*

For example, consider the *Boston Marathon Bombings*, where some of the developments took place over 7 days starting on April 15, 2013. We can establish its time matrix as **TM = *one week***, each *time unit* TU_i = ***one day***, and each column as the hours/minutes of the day, with n =**7**. Fig. 5.8 shows the corresponding time matrix, where $TU_1 = 15Apr$, $TU_2 = 16Apr$, etc... For more granular applications, the time matrix can be adjusted to one day and each *time unit* can be the minutes/seconds of the day. The point is that the user must determine the time units that make sense for the task at hand. Having established the time units, we must now define the length of each time window. A simple approach is to make each time window the same as a *time unit*. In Fig. 5.8(a), for instance, each time window *TW* corresponds to one *time unit* (e.g., $TW_1 = APR15$

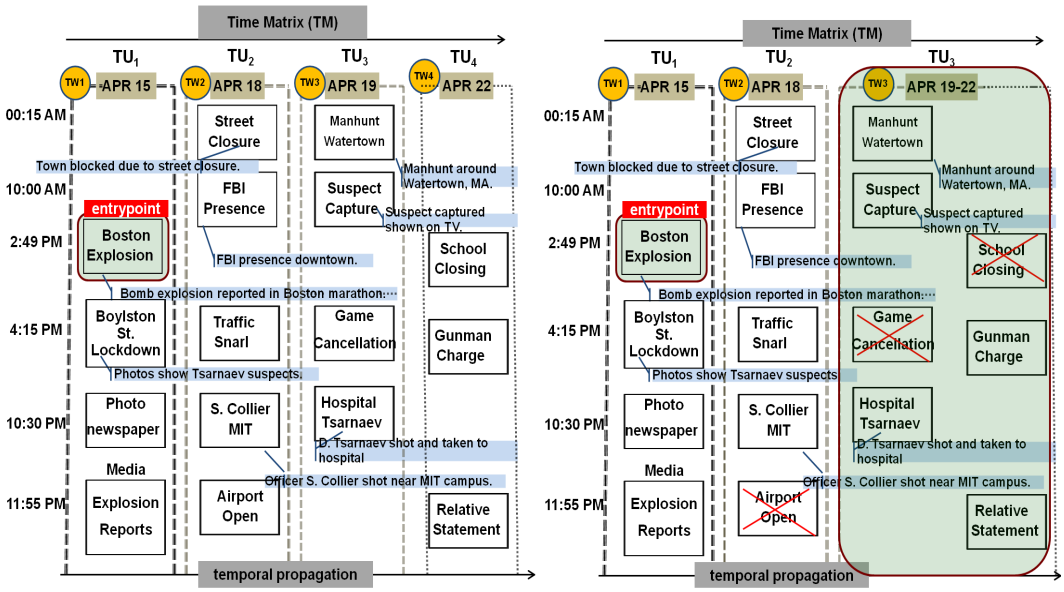


Figure 5.8: Visualization of a *Time Matrix*. (a) Temporal propagation of entities in 4 time windows TW1-TW4. Each entity is designated by a box and allocated to a *time unit* TU_i according to the entity's timestamp. (b) The crossed entities indicate that they have been pruned. Time units TU_3 and TU_4 are merged as a new single *time unit* TU_3.

or $TW_2 = APR18$). Alternatively, a time window can be a combination of several time units, as is shown in Fig. 5.8(b) where $TW_3 = APR19 - 22$.

The time window parameters above are decided on a per-application basis. Once established, each time window can be populated with the entities found according to the method in Subsection 5.3.1. This is easily accomplished by allocating each entity to the appropriate TW_i based on the entity's timestamp. On *Twitter* data, the timestamp is ideally extracted from text. Since that is not always available, the tweet's metadata timestamp can be used as a good-faith approximation. One additional caveat must be made: only entities that meet a minimum value of *ConceptRank* are inserted (*ConceptRank* is explained in Subsection 5.3.2). A visual example follows.

Fig. 5.8(a) depicts a partial time interval of four discrete days (Apr 15,18,19,22) related to the *Boston Marathon Bombings*. Some textual description is included for illustration purposes. It is assumed no data is available for the missing days (Apr 16,17,21). Here, we set $TM = 4$ days and set each each $TU_i = 1$ day on an hourly basis. Knowing that the Boston Explosion, which is set as the storyline's entrypoint, occurred on April 15 at 2:49 PM, we place that entity in TU_1. It is followed by the Boylston St. Lockdown at 4:15 PM, and so forth. The same is done for the rest of the days until all entities in the data space have been addressed for that time matrix. This organizational model is not only attractive for its simplicity, but it also serves as a look-up data structure where sequences of developments can be easily found. In Section 5.4.3, time windows are revisited and put to use after the computation of entity connectivity.

5.4.2 Time Windows Considerations

The model explained above provides an efficient view of time-ordered entities and events, which facilitates reasoning. However, it raises design questions for which decisions must be made and are explained below:

- **Time span**: entities may cover more than one time window. It is possible, for instance, that the Street Closure of April 18 last several days. In this case, allocation to a time window is done according to the entity's earliest observation time. That entity, therefore, is placed in TW_2 since its earliest occurrence is indeed April 18.

- **Concurrency**: entities may have the same timestamp, in which case there is no clear way to order them. In such scenarios, the following differentiation can be made: preference is given to the entities that contain either a *semantic constraint* (see *Definition* 3) or the most specific location. If the tie still cannot be broken, arbitrary ordering is taken as the last option. For example, if the user seeks semantic constraint "explosion", then entities with such a mention are placed in its time window before another entity that has the same timestamp, but with no such mention. Similarly, an entity located at *Boylston St.* precedes any simultaneous entity located in *Boston*, since the former location is more specific than the latter.

- **Frequency**: rare entities can be pruned since they provide little connectivity strength (connectivy, an important feature of this approach, is explained in Subsection 5.3.2). For example, assume that the Airport Open in TW_2, the Game Cancellation in TW_3, and the School Closing in TW_4 appear very few times. In this case, they are removed from the analysis, which is indicated by the red crosses in Fig. 5.8(b). Pruning removes non-interesting entities, thus saving processing cycles.

- **Merging**: two time windows TW_j and TW_k can be merged when they are deemed too sparse. For example, in Fig. 5.8(b), *Apr 19* and *Apr 22* had some entities pruned, leaving them relatively unpopulated as compared to the other TW_i. To save computing resources, they are combined into a single window, namely "*Apr19-22*", denoted by the shaded area. The time sequence of the remaining entities are still preserved.

- **Size**: in theory, a time window TW can hold any number of entities and can be composed of any number of time units, only limited by the length n of the time matrix. In addition, they do not have to have uniform lengths. However, long time windows, whether uniform or not, may generate excessively long storylines, which in turn tends to become less intelligible. The experiments of this study reveal that short time windows of one or two time units are not only more computationally efficient, but also allow more coherent storylines than longer time windows.

Parameter Tuning: Below, we briefly discuss the parameters that affect spatio-temporal storytelling in the context of this research:

1. **radius of study**: this parameter corresponds to the radius *r* in *Ripley's K* function. It determines the maximal area to be investigated, and, at least in theory, can be infinite. A long radius may find an unreasonably-high number of entities, raising concerns about computational complexity. A short radius may not find enough entities, all depending on the density of the dataset, and whether it is uniform or skewed in some way. In the context of *Twitter*, data tends to be very dense of entities and the radius is better kept short. It can then be increased in small increments, and experimented with. In our experiments, we start with a radius that encloses 1/4 of all entities for high-density datasets, and a longer radius that comprises 1/2 to 3/4 of all entities for increasingly sparse datasets. In general, this has been a good rule of thumb.

2. **relationship types**: the *ConceptRank* calculation must be able to differentiate between *implicit* and *explicit* relationships, which is determined by the user as a pre-processing step. *Explicit* are the important relationships that contribute strong knowledge to the storylines. Often, they are far and few in between. *Implicit* relationships are informational, but not as important, and tend to occur frequently. Ideally, a good *ConceptRank* should get most of its value from the strong relationships (i.e., explicit) and less from the weak ones (i.e., implicit). For that to happen in the *ConceptRank* formula of Eq. 5.3, the number of *explicit* relationships should be much lower than the number of *implicit* ones. When the two are close, *ConceptRank* devolves into simple *PageRank*.

3. **relationship between entities**: in the process of linking two entities, one challenging aspect is determining which relationship to link them with. There can be many options because the same entities can have different interactions at different times. For example, the dataset could have several connections between $\boxed{\text{D.TSARNAEV}}$ and $\boxed{\text{BOSTON}}$, such as $\overset{drove}{\rightarrow}$, $\overset{went}{\rightarrow}$, or $\overset{walked}{\rightarrow}$ which begs the question of which one to select for the storyline. The suggested approach is to pick the most frequent relationship between the two entities (stemmed). This often returns reasonable results and is a simple task. For better accuracy, one may want to use an external tool such as [143] to consolidate similar relationships into one, get the most frequent, and use it to link the entities.

4. **damping factor**: this is the Γ parameter in the *ConceptRank* calculation of Eq. 5.3. For all effects, Γ assigns an initial value to every entity in the concept graph, such that every entity has an initial amount of relevance to begin with. This value should be set in the range [0,1]. The suggested direction is to set Γ closer to 1 such that most of the *ConceptRank* comes from the relationships, and not from the the initial value itself, which, for all purposes, is arbitrary.

5. **time window**: a well-defined time unit should follow similar principles as the radius of study: it should not be so granular as to cover too few entities, but neither should it be so coarse as to encompass too many. Having too few entities per time unit may force many times units to be looked up, which is counter-productive. Having too many is unnecessary because a storyline is unlikely to include many entities (storylines are usually short). For

Twitter data, a good time unit represents one day for high-visibility events. For slow-moving processes (e.g., an election), time units of 15 days to 1 month can be adequate.

Spatio-temporal Challenges: Storytelling is not one single analytical tool. Rather, it can be better described as a framework founded on small principles: entities that have a minimum amount of spatial and temporal proximity, whose connectivity can be measured mathematically, but whose social interactions must be justified semantically. This is what determines coherence, which this study tries to achieve at every level, both mathematically and semantically. Coherence, then, is highly sensitive to the steps that comprise the storytelling process. For this reason, we briefly describe below some of the challenges, pitfalls, and points of contention that the reader may face while generating storylines:

1. Location and time extraction: Spatio-temporal storytelling requires entities to be explicity placed in space and time. Locations come in various flavors, such as *well-known places* (e.g., New York), *points of interest* (e.g., Statue of Liberty), common addresses (e.g., 123 Main Street), and geo-political boundaries (e.g., Cumberland County). They must be geocoded as latitude and longitude to allow for spatial operations. Ideally, granular locations (i.e., a specific address) should be used over coarser ones (i.e., a city) as they relay stronger accuracy. When many locations for the same entity can be identified, a simple, but often reasonable approach is to take the centroid of all locations as the approximate location for the entity in question. As for the time dimension, more than one timestamp can be present, in which case the latest one is often sufficient. Timestamps are less of a problem in near real-time data, such as recent tweets, because entities can be tagged with the tweet's issue time. For long-standing datasets, however, this assumption may not hold.

2. Entity identification: Because entities represent the building blocks of storylines, correctly identifying and extracting them from raw data is imperative. This is often accomplished with third-party *NLP* tools (e.g., [81], [128], [5]), some of which perform better than others under different situations. As a consequence, different tools should be evaluated for maximum performance. In addition, there is the ongoing problem of *entity disambiguation*, in which case the same entity is described differently in various datasets, preventing them from being identified as a single element. As much as possible, the user should strive to pre-process these entities in an attempt to minimize ambiguity. Scientific literature in this field is plentiful, but we do not endorse any specific works in this study.

3. Relationship binding: This is one of the most challenging aspects of storytelling. Entities must be connected to other entities through relationships, and deciding on what these relationships should look like directly impacts the computation of the *ConceptRank*. The reader should keep the following in mind. In any application, there should be a differentiation between more important and less important relationships. The more important ones should be limited in number (we denote them as *explicit*), so they can contribute more to the calculation of the *ConceptRank* of Eq. 5.3. The vast majority should be left as less important (e.g., implicit) since they contribute less information. One pitfall of this view is that there is no

Algorithm 5: Storyline Generation

inputs : Entity *entrypoint*, pruned and consolidated Time Matrix TM
output: List *storyline*

1: using *entrypoint* → get radius r from *spatial propagation* ;
2: *entities* ← identify entity set from radius r ;
3: compute *ConceptRank* for each e_k in *entities* ;
4: segregate each e_k into the appropriate tu_i of TM ;
5: **foreach** tu_i *and if* $|entities| < k$ **do**
6: | storyline ← add *top-k* entities in time order
7: **end**
8: storyline ← for each pair of entities, establish their relationship as their most frequent one ;
9: **if** *(storyline should proceed)* **then**
10: | **set** new *entrypoint* = e_{k+1};
11: | *iterate* → step 1
12: **end**
13: **output** *storyline*;

clear way to designate what should and should not be important other than relying on the reader's own understanding. To compound this problem, a pair of entities can have multiple relationships. When this happens, we have relied on the most frequent relationship to bind the entities in the storyline. There are other equally-valid approaches, however, such as using the most recent one or the one whose entities are mostly spatially-proximal.

4. Data pre-processing: Earlier in Subsection 5.3.1, we propose *Ripley's K function* to find dense regions where storylines can be investigated. To optimize the process, sparse datasets should be condensed by removing empty regions where few or no entities reside. Entities for which no location or time is available should be removed altogether. These steps can go a long way towards minimizing running time.

5.4.3 Spatio-Temporal Storyline Generation

This discussion puts together the ideas in subsections 5.3.1, 5.3.2, and 5.4.1 to generate storylines. Algorithm 5 takes as input the user's desired entrypoint, and an appropriately pre-defined Time Matrix. The essential steps are as follows: obtain the radius of study and identify the entities in that radius (Lines 1 and 2); compute the *ConceptRank* of the found entities and allocate the most important ones to an appropriate time window according to their timestamps (Lines 3 and 4); using each time window, build the storylines with temporal ordering (Line 6); for each pair of entities, select a relationship to insert in between them (the most frequent relatioship is often appropriate) (Line 8); if the storyline is too short or incomplete, a new entrypoint is established as the next highest ranking entity above the top-k ones (Line 10). The process iterates (Lines 9 to 11), otherwise, the storyline is output (Line 13).

The above process may generate long storylines, which may become less intelligible. However, the point at which this iterative process should stop depends on one's own understanding of fact completeness. Insertion of new entities into the graph requires a check to see if the entity already exists, which is done in constant time. Range searches may perform from O(log N) to O(N) depending on the number of location overlaps. Computation of the *ConceptRank* affects only

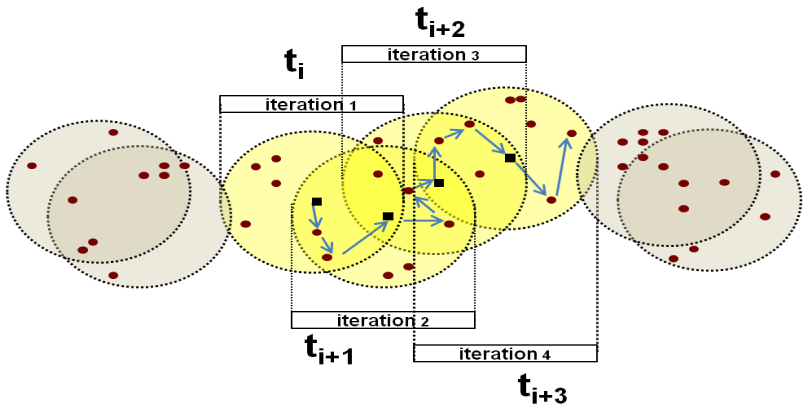


Figure 5.9: Hypothetical generation of a storyline through four iterations of the algorithm (t_i through t_{i+3}). Each circle corresponds to one iteration. Squares represent entrypoints and dots represent entities. Each iteration begins at an entrypoint and connects two other entities, before a new entrypoint is considered.

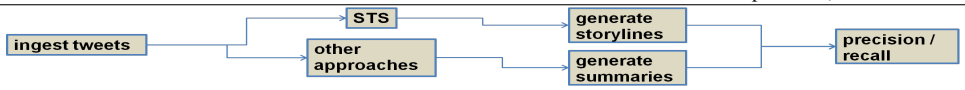
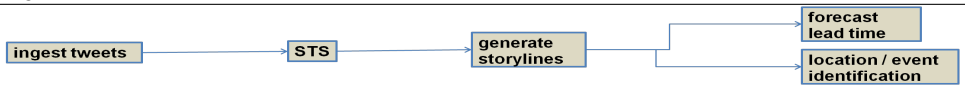
the inserted entities and the ones they link to either directly or indirectly. Fig. 5.9 shows the propagation of a storyline across four different regions in four iterations [t_i, t_{i+3}] of Algorithm 5. The entrypoints are represented by squares and the other entities by circles. At each iteration, the top 2 entities are linked followed by a new entrypoint, from where a new iteration begins. In this simple example, the four iterations generate one storyline composed of 12 entities (4 entrypoints + 8 other entities) and their relationships.

5.5 Empirical Evaluation and Technical Discussion

One of the initial claims of this research was that spatio-temporal storytelling can be gainfully applied to everyday analytical tasks. To follow through with that statement, the experiments are divided in three parts. Subsection 7.5.2 compares our approach to three existing methods of event summarization. Subsection 5.5.2 presents forecasting as the chosen task to verify how far in advance the generated storylines find an event before it is published in the news. Subsection 5.5.3 provides an in-depth analysis of the spatio-temporal characteristics of data and how they influence the generation of the storylines. To begin, the general experiment setup is given. **Experiment Setup** To provide a good variation of insights, the experiments are broken down in three parts, as listed in Table 5.2. The first task, event summarization, is done on the Ukraine political crisis. The second, event forecasting, uses mexican civil unrest as a case study. And the third, spatial analysis, discusses in technical detail data aspects and how they affect the algorithms. It uses tweets related to the Syria Civil War. For each part, the table shows a general flow of the steps taken to perform that task. And because each task involves different analyses, pertinent details and discussion are provided in the corresponding sections.

Data specification: The data sources utilized span the years of 2011 through 2014, queried directly from *Twitter* through its API webservice. The querying process used specific keywords such as “protest” and “fight” along with name places such as “Moscow” or “Kiev”. The number of

Table 5.2: Methodology and data specification of the experiments.

Task	Nature of Events	Years	No. Records	Measure	Validation
Event Summarization	Political Crisis in Ukraine	2014	100,000†	precision/recall	*Twitter Feed*
					
Event Forecasting	Mexico Civil Unrest	2013	119,578†	recall and forecast lead time	*GSR*
					
Spatial Analysis	Syria Civil War	2011-2012-2013	50,000†	analysis	–
					

†To demonstrate the high application potential of storytelling in different scenarios, the experiments use different data sets for different tasks.

records is approximately equal for each location. In this manner, appropriate event-related datapoints needed by the algorithms are captured. Retweets are removed to prevent artificial high frequency of terms. Each part of the experiments is performed with a different number of records to show variation. The nature of the data reflects items of interest to different communities such as intelligence, politics, law enforcement, and journalism. Some are civil protests and strikes, while others encompass harder violence such as attacks, shootings, and bombings. Events of a non-violent nature are also included to make sure that the algorithms in question are able to differentiate them as needed.

Comparative methods: For event summarization, the objective is to find out how well the proposed approach performs when compared to three other existing techniques. Currently, there is an extensive body of works related to text summarization [95] from where many options are available. The three selected methods encompass a mix of textual analysis, time reasoning, and synonymy. The first approach, *SUMMALLTEXT* (we denote it as *summ-text* going forward) uses a variation of TF $\times$ IDF to compare tweets. The second approach, *SUMMTIMEINT* (*summ-time*), uses a similar technique, but segments the tweets in different time windows and does processing based on each time window. They are described by Chakrabarti in [24]. Both output the top n tweets of maximum score to represent summaries. The third approach, described by Medvet [87] and what we denote as *EDCS-summ*, identifies highly-frequent words in a tweet, builds a set of synonyms from them, and outputs the tweets for sets that are also highly frequent. They also apply time segmentation. For event forecasting, a self-evaluation is performed on whether the proposed algorithms are able to point to an event before that event is actually published in the newswire. Because the goal is not to find the best forecasting approach, this part does not present comparisons to other existing methods. Rather, forecasting is portrayed as one more potential use of spatio-temporal storytelling. For spatial analysis, the discussion is standalone. It delves into the intrinsic characteristics of the dataset used in that part of the experiments, and elaborate on how those facets influence spatio-temporal storytelling.

Performance Measures: For the first task, the question to be answered is the following: which summarization approach has the highest precision and recall based on a given input event and

Table 5.3: Explanation of performance measures.

Measure	Meaning
precision^1 = $\frac{\lvert retrieved\ tweets \bigcap relevant\ tweets\rvert}{\lvert retrieved\ tweets\rvert}$	fraction of events correctly identified as relevant over all retrieved events (tweets).
recall^1 = $\frac{\lvert retrieved\ tweets \bigcap relevant\ tweets\rvert}{relevant\ tweets}$	fraction of events correctly identified as relevant over all relevant events (tweets).
recall^2 = $\frac{\lvert retrieved\ tweets \bigcap relevant\ tweets{-}GSR\rvert}{relevant\ tweets{-}GSR}$	fraction of events correctly identified as relevant in GSR over all relevant events in GSR.
Definitions	
Relevant event for precision^1 and recall^1: An event is deemed relevant if at least one record exists in the entire dataset that contains all keywords and locations from the input query.	
Relevant event for recall^2: An event is deemed relevant if at least one *GSR* record exists that contains all keywords and locations from the input query.	

location? If high precision and recall can be observed, then the retrieved tweets should be able to provide a reasonable summary of the topic in question. For example, if the input event is "bombing in Afghanistan", how many output tweets of each approach contain that event and that location? And how many are missed? For this purpose, traditional IR is used, defining precision^1=$\frac{|retrieved\ tweets \bigcap relevant\ tweets|}{|retrieved\ tweets|}$ and recall^1=$\frac{|retrieved\ tweets \bigcap relevant\ tweets|}{relevant\ tweets}$. The definition of a **relevant tweet** is one that contains all keywords and all locations of the input event. Note that an input can have one or more keywords along with one or more locations. For simplicity of discussion, we limit our experiments to one of each at a time. Table 7.2 provides a quick view of the performance measures and definitions.

For event forecasting, storylines are first generated using the proposed approach. Subsequently, the generated storylines are compared to the Gold Standard Report (*GSR*), which represents the ground truth data. *GSR* is a database of events and part of the *Intelligence Advanced Research Projects Activity* (IARPA) [54]. The measurement, forecast lead time, is simply a computation of how far in advance the storyline of an event gets generated in relation to the earliest occurrence of that event in the *GSR* database. Recall (denoted as recall^2) is similar to the first task, where a relevant tweet is one that contains all keywords and locations of the input event as part of a *GSR* record. Since the goal of this task is completeness (i.e., finding the most forecasts) and not preciseness (i.e., finding the most accurate forecasts), the calculation of precision is not published at this stage. For all experiments, no assumption is made on data distribution, but areas are selected where violent events are known to be of a high enough frequency such that summarization and forecasting are plausible for all comparative methods.

5.5.1 Comparison of Event Summarization Approaches on the Ukraine Political Crisis (2014)

In this subsection, the three event summarization approaches mentioned in the experiment setup (Subsection 5.5) are contrast to the proposed methods, spatio-temporal storytelling. One line of research complimentary to this work, but which often does not include spatial *storytelling*, is event detection, which is left for future work, and indicated to the reader for further consideration [78, 136]. The discussion is framed in terms of *precision* and *recall*, as specified in Table 7.2.

Table 5.4 lists a set of 10 event types, labeled $E1$ through $E10$, that are used as input to each of the four comparative methods: *STS*, which is our proposed work; *Summ-Text* [24], a cosine similarity variant of summaries; *Summ-Time* [24], a cosine variation with time-based segments;

Table 5.4: Comparison of precision and recall for four different approaches: *Spatio-temporal storytelling* (STS), *Baseline Summ-Text*, *Baseline Summ-Time*, and *EDCS-Summ*. Each row represent one type of event and a location of interest. The highest values are shown in bold.

	STS		**Summ-Text**		**Summ-Time**		**EDCS-Summ**	
Event	**precision**	**recall**	**precision**	**recall**	**precision**	**recall**	**precision**	**recall**
E1-assassination Kiev	**0.42**	0.54	0.40	**0.61**	0.34	0.58	0.41	0.48
E2-invasion Black Sea	0.52	0.66	**0.73**	0.65	0.37	**0.68**	0.67	0.62
E3-protest Russia	0.35	0.34	0.58	0.61	**0.71**	**0.70**	0.56	0.62
E4-attack Sevastopol	0.62	0.57	0.55	**0.65**	0.58	0.61	**0.68**	0.60
E5-confiscate Kharkiv	**0.70**	0.54	0.61	**0.70**	0.50	0.54	0.51	0.67
E6-fight Ukraine	**0.65**	**0.60**	0.64	0.51	0.39	0.46	0.40	0.59
E7-arrest Lviv	**0.65**	0.28	0.44	0.52	0.51	**0.62**	0.57	**0.64**
E8-explosion Donetsk	**0.63**	0.45	0.49	0.35	0.44	0.46	0.59	0.44
E9-occupation Simferopol	0.55	0.41	**0.64**	0.61	0.23	**0.70**	0.51	0.66
E10-blockade Crimea	**0.65**	0.33	0.59	0.38	0.22	0.71	0.50	0.62
Precision and recall above refer to **precision**^{1} and **recall**^{1} of Table 7.2								

and *EDCS-Summ* [87], which uses segmentation applied to synonym sets. Each event is composed of a single keyword and the name of a location. 100,000 tweets related to the Ukraine political crisis (2014) are used. For each event type, the table shows precision and recall values using the four comparative methods as explained earlier. The highest values are shown in bold type.

The way to interpret the table, exemplified for row 1, is as follows. First, E1 is taken (assassination Kiev) and storylines are generated using that event and location as the entrypoint to the proposed approach (*STS*). The set of generated storylines are then compared against the entire dataset to see how many tweets those storylines indeed summarize (i.e., contains an assassination keyword and mentions Kiev as a location or mentions any location that is enclosed by Kiev). Precision and recall are then computed. For the other approaches, the tweets that they output are retrieved, and again, checked for keywords and locations to compute precision and recall.

Discussion: At first glance, one can notice the fairly low levels of precision for *Summ-Time* for all event types, except for $E3$ (*protest Russia*). This approach clusters tweets based on time intervals, disregarding the clusters where events are not highly frequent. $E3$, on the other hand, is a very common occurrence of this event and place, which boosts its precision (we set the time interval to six months). For the other approaches, this event's precision is considerably lower for different reasons: *STS* fails to capture "explosion" as a highly-connected entity according to its *ConceptRank* measure. In addition, *Summ-Text* and *EDCS-Summ* suffer because the word "Russia" is not always accompanied by "protest". For *Summ-Time*, the situation is more favorable in terms of recall, as relevant items are often retrieved with greater success.

For *Summ-Text*, precision appears fairly stable across measurements, but with mixed signals. It is significantly high for $E2$ (invasion Black Sea) and $E9$ (occupation Simferopol), but decreases for $E1$ (assassination Kiev). The reason has to do with the fact that this approach relies on keyword matching, for which "invasion" and "occupation" are very common, but "assassination" is not. This method shows the highest recall on the table (0.70), which comes for event $E5$ (confiscate Kharkiv). Overall, this method presents the best recall of the four approaches, which may be useful in domains where completeness is more important than preciseness.

The fourth technique, *EDCS-Summ*, is interesting because it uses a dictionary approach to identify events. Thus an "attack" can be expanded with "assault" or "aggression", among other terms (we use *wordnet* [143] to expand terms with other synonyms). This feature explains the high

Table 5.5: Sample summaries (S1-S20) for the four comparative methods based on the query "violence Ukraine". The table shows that *STS* summaries are able to find locations and identify important events more effectively than the other methods. Tweets range from March 25 to March 28, 2014.

Comparative Methods	Summaries	Locations Found	Events Identified
STS†	S1 - UKRAINE continue ELECTIONS using VIOLENCE affect TATARS live RUSSIA. S2 - INTERVENTION threat KIEV contains VIOLENCE hit TV show CRIMEA. S3 - UKRAINE PROTEST united VLADIMIR PUTIN start VIOLENCE. S4 - MR AMAZAYEV works HIZB UT-TAHRIR plans DEMONSTRATION dogs VIOLENCE legalize #UKRAINE. S5 - #TYMOSHENKO speak U.S sanction MOSCOW promote VIOLENCE.	**6** (Russia, Kiev, Ukraine, Moscow, US, Crimea)	**4** (elections, protest, demonstration, intervention)
Summ-Text	S6 - @SteveKristan I've been to #Ukraine and found the people very warm and genuine. The violence needs to stop. S7 - @ShepNewsTeam: Elderly man remembers those lost in #Ukraine violence; a destroyed building looms in the background. S8 - @pinkyfajer: No matter what part of the world you're from #Violence Look's the same. my #Mexico, Ukraine. S9 - @agenfor people of Ukraine fought the regime of violence. We wanted new life.investigation regarding Maidan in process. S10 - @euHvR athlete leaves Ukraine in turmoil, violence: UPDATE 3/6: Keiara Avant left the Ukraine... http://t.co/mtb9PMtdaS.	**2** (Ukraine, Mexico)	**1** (turmoil)
Summ-Time	S11 - @TylerIbarra15 War and violence not the answer, love will solve negativity. Syria, I pray for u everyday. Ukraine 4u2. S12 - @LowMaintainLife the only thread of violence to Ukraine today is Russia. S13 - @Independent_ie Reiterates Rejection of Violence in Ukraine and the world #WakeUp http://t.co/EHoSehF0nc S14 - @andersostlund: @euHvR Sorry for sarcasm but you mean in #Russia. Violence in Ukraine was instigated in Russia. S15 - @Nonanon_anon: Too bad wasn't IMF banxters took brunt of violence in Ukraine. They will steal everything. #fauxgover.	**3** (Russia, Syria, Ukraine)	**1** (war)
EDCS-Summ	S16 - @muslimvoices The Russian might alone stops violence and preserve the life in the Ukraine #bbcqt. S17 - @carlbildt Hate hearing violence in the Ukraine. The Kiev bar reopened but its struggling. A mess... http://t.co/eZnmn7Lyg8. S18 - @zoomarang Yes SHE is ignorant, compares Issa to Ukraine violence for mic off but 4got THIS. https://t.co/R7KzYJG3XU. S19 - @NOLASpiceDesign @wolfblitzer Yes violence in Venezuela - orchestrated by the Terror-State Ukraine. S20 - @jesperjurcenoks Ukraine Crisis give leverage needed to fight cybercrime in area http://t.co/x8subj3Zax #infosec #security.	**3** (Venezuela, Kiev, Ukraine)	none
†our approach			

precision under $E4$, but also serves to explain why this method does not do well under $E1$ (assassination Kiev) or $E6$ (fight Ukraine). While "assassination" and "fight" have many synonyms in our database, terms in the other queries, such as "blockade" do not. Thus, these other terms do not yield many matches in the dataset. Another interesting fact is that this method shows the least amount of variation between precision and recall, which may be attractive for applications in which both of these measures are important.

Inspecting Table 5.4, it can be seen that *STS* provides six of the highest scores for precision. There reasons are twofold: the first is that the events on the table have high connectivity to many entities in the dataset. It implies that these events tend to receive a high *ConceptRank*, which helps them bubble up to the top of the important entities. Thus, they tend to show up on the storylines. The second factor, has to do with location. *STS* is a spatial technique in which places are regarded as geocodes (i.e., latitude and longitude coordinates), not plain keywords. Thus Ukraine covers any point of the country, while Donetsk represents any location within that city, and so forth. In essence, this has the effect of capturing a wider variation of events across many areas, regardless of how they are described in the dataset. These results are encouraging for three reasons: they reinforce the importance of the spatial aspect which the other methods do not target; they indicate that the other methods could use the output of our approach (storylines) as the input to theirs in order to incorporate the spatial contribution; they confirm our initial claim that storylines can be a valuable tool in many different activities. In this case study, storylines outperform the three other techniques of summarization in terms of precision.

The main goal of summarization is to capture essential ideas from the underlying text while disregarding unrelated points, what one would call noise. To this end, another claim can be made. Spatio-temporal storytelling is able to effectively capture two facets of the underlying data: the important locations and relevant events related to the query. Take, for example, Table 7.4 which lists a set of 20 summaries (S1-S20), five for each of the comparative methods. *STS* shows five storylines (S1-S5) related to the input event "violence Ukraine". The table shows that *STS* cap-

tures six locations related to the topic of discussion (the Ukraine political crisis). These locations are Russia, Kiev, Ukraine, Moscow, Crimea, and the US. The other approaches not only find less locations, but some of them are unrelated, such as Syria and Venezuela. The reason *STS* is able to find a more coherent set of locations has to do with *Ripley's K* function. It helps localize the investigated entities in short radii, which prevents far-away entities (such as the ones in Venezuela) from showing up. In addition, *STS* is also able to identify relevant events, such as elections, protests, demonstration, and intervention. This is because of *ConceptRank* which promotes these highly-connected entities into the storylines. The other approaches do not rely on connectivity, focusing more on word frequency, failing to capture many events that are not spelled out commonly enough. *Summ-Text* only finds "turmoil", while *Summ-Time* only captures "war", and *EDCS-Summ* finds no events.

Notice also that each summary is highly related to a real-world development, such as "elections in Ukraine", "violence on Tatars", "Hizb Ut-Tahrir plans demonstration", or "US sanctions Moscow". The other three approaches also speak of violence, as they also conform to the input query. However, their summarization content appears less coherent as they devolve the topic of violence into general statements such as "violence needs to stop", "hate hearing violence", or "violence looks the same...". From an application perspective, these are not real-world facts, but mere observations that would arguably lend little knowledge to an analyst. It should also be noted that, as seen in Table 7.4, spatio-temporal storylines represent true summaries: they yield a collection of the best connected people, events, organizations, and relationships without replicating the original tweets. The other approaches, on the other hand, are limited to finding the best original tweets that conform to the input query. In this sense, they operate more as search tools as opposed to a legitimate summarization technique.

5.5.2 Event Forecasting of Civil Unrest in Mexico (2013)

Storylines can be useful in many different applications. This section takes a lightweight view of *forecasting* and discusses how storylines are able to identify real-world developments before they are published in the news. Intuitively, *Twitter* data is available in near real time, while news reports lag behind for hours and sometimes days before their dissemination. The goal, then, is to generate storylines, identify which real-world stories they relate to, and determine how far in advance the storyline "forecasts" the real event.

The dataset is composed of 119,578 tweets related to *civil unrest*[1] in Mexico for the first three months of 2013. The targeted events are related to *education reform*, which has provoked social strife in Mexico, and documented as part of the Gold Standard Report (GSR) [54], which serves as our ground truth.

Table 6.3 shows a sample of 10 such events that are used for discussion. Each *GSR event* has an associated *reported by* source, an *event location*, and *published* date. For each *GSR event*, the table

[1] civil unrest denotes an event of social impact, such as a strike or a protest. Violence does not have to be included.

Table 5.6: 10 instances of civil unrest reported in the Gold Standard Report (GSR-IARPA). Each event is related to protests against educational reform in Mexico City in 2013 and other locations throughout the country. The *Forecast Lead Time* column shows that the storyline from tweets are generated even before the news article is published, often days in advance.

	GSR Event	Reported By	Event Location	Published	# Related Tweets	Forecasting Storyline	Generated Date	Forecast Lead Time
1	SNTE Protesters block Eje Central; demand pension pay.	milenio.com	Mexico City	Jan-03-2013	5,422	**EDUCATION** fighting **SNTE** paying **SALARY** lower **FUNDS**.	Jan-02-2013	1 day
2	Teachers protest in Michoacan; demand Christmas pay.	milenio.com	Michoacan	Jan-04-2013	1,410	**FIGHT** break **STUDENT** distribute **FUNDS** sending **MORELIA**.	Jan-01-2013	3 days
3	Stop at Oaxaca University affect more than 20 thousand students.	milenio.com	Oaxaca	Jan-12-2013	2,051	**EDUCATION** halt **UNIVERSITY** remove **STUDENT**.	Jan-08-2013	4 days
4	SNTE professors at Aguascalientes will march against education reform.	lajornada.com	Aguascalientes	Jan-14-2013	1,960	**TEACHER** protest **EDUCATION** lower **FUNDS**.	Jan-10-2013	4 days
5	SNTE teachers walk in Veracruz against education reform.	milenio.com	Veracruz	Jan-17-2013	2,737	**TEACHERS** lose **FUNDS** remove **BUDGET** impact **EDUCATION**.	Jan-10-2013	7 days
6	Teachers block Morelia-Toluca in Zitacuaro.	lajornada.com	Zitacuaro	Jan-19-2013	734	**ROAD** blocked **PROTEST** include **TEACHERS** ask **FUNDS**.	Jan-11-2013	8 days
7	Several incidentes reported during SNTE's march.	milenio.com	Pachuca	Feb-01-2013	1,155	**FIGHT** breaks **CITY** drain **FUNDS**.	Jan-28-2013	4 days
8	Teachers march against labor reform in Tlaxcala.	milenio.com	Tlaxcala	Mar-14-2013	3,938	**EDUCATION** march **TEACHER** lower **BUDGET**.	Mar-02-2013	12 days
9	In Acapulco, SNTE teachers from San Marcos will march.	lajornada.com	Acapulco	Mar-14-2013	1,021	**TEACHERS** march **CITY** protest **EDUCATION**.	Mar-13-2013	1 day
10	SNTE teachers march in Atlixco.	milenio.com	San Pedro Atlixco	Mar-28-2013	2,760	**SNTE** march **TEACHERS** participate **PROTEST**.	Mar-20-2013	8 days

shows a *forecasting storyline* of no more than five entities and generated by our algorithm from its respective # of *related tweets*. A tweet is related to the storyline if they share at least one entity in common at the same location. The *generated date* of the *forecasting storyline* is the timestamp of the most recent related tweet. In the *forecasting storyline*, entities are bolded in uppercase, relationships are not. The *forecast lead time* is the time difference to the *published* date. The starting location is *Mexico City* from where we consider a radius of 450 km that includes other major cities shown in Fig. 6.9.

Discussion: Item 1 has a *forecasting storyline* of four entities (*education*, *SNTE*, *salary*, *funds*), summarized from 5,422 tweets. These five entities are the ones of highest *ConceptRank*, and thus selected for the storyline. The relationships (*fighting*, *paying*, *lower*) are the most frequent ones between the adjoining entities. Note that storylines do not reflect stylized English language. Because they are linked based on spatial connectivity and time order, grammar rules cannot be easily enforced, though they often come out as properly-formed phrases. The *forecasting storyline* closely resembles the associated *GSR event*: even though they only share one entity, i.e. *SNTE*[2], both relay similar messages, that is, teachers protesting for higher wages. The most recent tweet in this set is dated Jan-02-2013, which when compared to the *GSR event*'s *published* date of Jan-03-2013 indicates that the *Twitter* storyline forecasts the real event by 1 day.

While the storyline of item 1 relates to Mexico City, the one of item 2 takes place farther away in the state of *Michoacan*. At first glance, the *forecasting storyline* is not related to the *GSR event* since they appear to have little in common. They are strongly connected, however, in two manners: the semantic closeness between "student fights" and "teacher protests", as well as by location. Most of the 1,410 related tweets have a location inside *Michoacan*, which is a state with many cities, one of them being *Morelia*, an entity that appears in the *forecasting storyline*. The *forecast lead*

[2]Sindicato Nacional de Trabajadores de la Educacion.

Figure 5.10: Spatial propagation of *education reform* protests. Starting from Mexico City, similar events are observed around the country. The map shows 10 of approximately 5,000 affected locations.

Table 5.7: Recall results based on 9,304 *GSR events* in four different categories. Recall R1(R2,R3) denotes that the storyline matches the GSR event with one(two,three) common entities in a designated radius, and within 30 days of the *GSR event*.

	Mexico			Other Countries*		
GSR event type	**R1**	**R2**	**R3**	**R1**	**R2**	**R3**
01-civil unrest (employment, housing, resources, other policies)	0.617	0.552	0.230	0.553	0.507	0.420
02-vote (local, national elections)	0.586	0.430	0.418	0.490	0.430	0.367
03-infectious human illness (rare and common diseases, pandemic)	0.502	0.428	0.400	0.537	0.472	0.311
04-economy (currency exchange, stock market)	0.772	0.512	0.497	0.402	0.324	0.405
*Argentina, Brazil, Chile, Colombia, Costa Rica, Mexico, Panama, Peru, and Venezuela.						

time is three days, showing that the algorithm was able to reconciliate tweets about fights due to educational factors, which were later reported by the media in the form of teacher protests. This example underscores the importance of location, which would otherwise make this linking difficult to justify.

The importance of the spatial aspect of this study must be emphasized, showing that all remaining items from 3 to 10 are highly-dependent on location. Note that none of those *forecasting storylines* have a location entity explicitly stated. However, their related tweets do contain at least one meta-data location that matches the location of the *GSR event*, and a timestamp that closely pre-dates the event's published date (within 30 days). This is particularly interesting in the case of item 9, whose *GSR event* is shown at *Acapulco*, but whose *forecasting storyline* does not reflect that location. But in fact, all of the 1,021 *related tweets* have a latitude/longitude that closely matches *Acapulco*. Also worth mentioning is the fact that there are a few entities very popular across the tweets, and as a consequence, appear commonly in the storylines. Four of them are *teachers*, *SNTE*, *protest*, and *funds*, which are commonly observed in *Aguascalientes*, *Veracruz*, *Zitacuaro*, *Pachuca*, *Acapulco*, and *San Pedro Atlixco*. The prominence of these entities as part of the storylines is very significant for a simple reason: it indicates that the spatio-temporal methodology is able to find storylines about civil unrest related to the education reform in Mexico using location as a decisive factor. Even though Mexico City is the most prominent area, the algorithm also identifies other important locations where events occurred with similar entities as the ones in Mexico City, as shown in the circle of Fig. 6.9. Even more encouraging is being able to identify them several days ahead of time, such as the "march against labor reform" in Tlaxcala (item 8), which is shown in the corresponding *forecasting storyline* 12 days earlier as "education march lower budget".

Apart from civil unrest, the GSR dataset catalogs events of other natures. Table 5.7 shows recall levels for the *STS* algorithm for 9,304 *GSR events* in four categories. Here, recall is defined according to recall[2] in Table 7.2. To deal with different parameters, a further refinement is made: in recall1 (R1), a storyline matches a *GSR event* under the following conditions: they must share at least one entity; the shared entity must be located within the investigated radius (e.g., 450 km in this case study); and the observation of the entity must pre-date the real event by no more than 30 days. In recall2 (R2) and recall3 (R3), at least 2 and 3 entities must be shared respectively. The radius and timestamp conditions remain the same as before. In short, we relate similar events that are close both spatially and temporally, as dictated by the proposed methodology.

Lessons Learned: Table 5.7 shows two sets of results: one for *Mexico*, since the previous forecast illustration was based on it; and one for other Latin American countries in order to add data variation. It shows a wide range in the recall values, with differing root causes:

- **Scope targeting**: storylines in R1 (i.e., the ones that share one entity with the *GSR event*) have good recall when only Mexico locations are considered, especially in the civil unrest and economy categories (0.617 and 0.772). The corresponding R1 values are significantly lower when other countries are included. The reason is the targeted events related to *education reform*, which is very prominent in Mexico, but not very common in the other countries. This gives us the first lesson: storytelling benefits when the scope is targeted. In other words, the examined topic should be specific enough to a region in order to maximize recall.

- **Match relaxation**: when the match is increased from one to a minimum of two or three entities (i.e., going from R1 to R2 to R3), there is a significant drop in recall. Making the parameters increasingly more strict prevents storylines from being identified as a match to the *GSR event*. This can be seen in the *vote* event type, for which R3 is very low for Mexico (0.418), and even lower for other countries (0.367). The second lesson is that the requirement of having more matches tends to find very coherent storylines, but will almost always find very few of them. This effect can be relaxed by increasing the number of investigated locations.

- **Region granularity**: these experiments consider a radius of 450 km from the country's capital. It should be apparent that such a large radius would have different effects in large countries, such as Argentina and Brazil, as opposed to smaller ones, such as Costa Rica and Panama. However, we find that large radii can be very appropriate in both situations, with one caveat: the application must be highly targeted for specific event types in order to avoid data explosion. For example, elections across the country would make sense when viewed in large areas. Therefore, the third lesson is that short radii may not find a significant number of entities that are global enough for forecasting. Local forecasting is certainly applicable, but may require more granular spatio-temporal reasoning.

- **Data variation**: data volumes and variation are important factors in recall levels. For example, in the case of event type 3 (*infectious human illness*), there are 1,916 *GSR events* for all

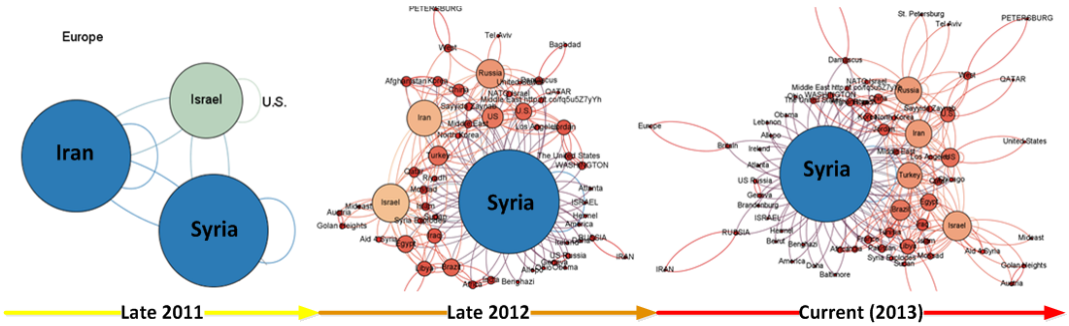


Figure 5.11: Spatio-temporal propagation of the *Syrian Civil War* from late 2011 to late 2013.

countries, out of which 128 refer to Mexico. However, our dataset does not have a significant number of tweets related to diseases or health topics. As a consequence, the algorithm is unable to generate very good storylines that can be matched to *GSR*, causing the recall levels to be low. At its worst, recall is only 0.400 for Mexico, and 0.311 for other countries. Even though this is a data problem, and not a weakness of the approach, it brings up the fourth lesson: low data volumes and poor variation creates storylines that lack meaning and appear disconnected from real events.

In general, the recall levels shown in Table 5.7 are very promising in the scope of spatio-temporal storytelling. Even when the values are low, they can be justified and remediated in different manners, such as by changing parameters (e.g., radius, countries), focusing on specific domains (e.g., vote, economy), relaxing or restricting match requirements, and verifying data volumes and variation. A well-tuned algorithm shows extremely high potential in a forecasting strategy.

5.5.3 Spatial Analysis on the Syrian Civil War (2011-2013)

This experiment discusses aspects of the spatio-temporal evolution of entities and events in the ongoing *Syrian Civil War*. At the time of this writing, it is an armed conflict between forces loyal to the ruling party and those seeking to oust it. The protesters demanded the resignation of President Bashal al-Assad. Although the conflict was originally confined within Syria's borders, it gained international attention with countries such as France, United States and Russia among others intervening to resolve the issue.

The dataset was composed of 50,000 tweets from where a concept graph of approximately 17,000 entities and 6,500 relationships was derived. Several runs were performed, starting with Damascus as the center, and allowing the radius to go long enough to include the Middle East, Europe and North America. The semantic constraints {*protests*, *chemical weapons*, *rebels*, *civilwar*, *Ba'athParty*} were used in storylines of length 5. Fig. 5.11 shows the spatio-temporal evolution of the *Syrian Civil War*. The dataset from late 2011 shows that only a few countries in close proximity to Syria, such as Iran and Israel, were paying attention to the Syrian protests. Over time the protests grew into a full-fledged civil war with increasing global impact extending into late 2012 and up until April 2013, the time of this writing. The latest stories regarding the use of chemical weapons have been corroborated by various leading news agencies around the world.

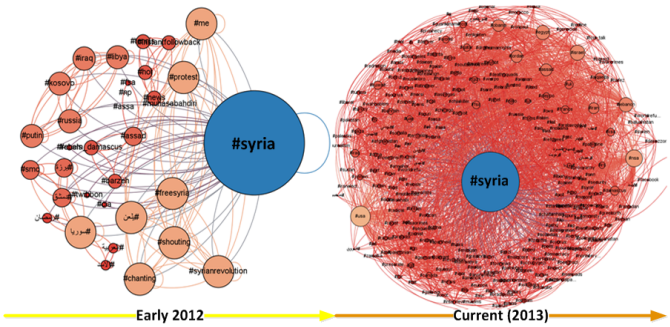


Figure 5.12: Temporal propagation of the **#Syria** hashtag showing an uptrend as the conflict gained international attention between 2012 and 2013.

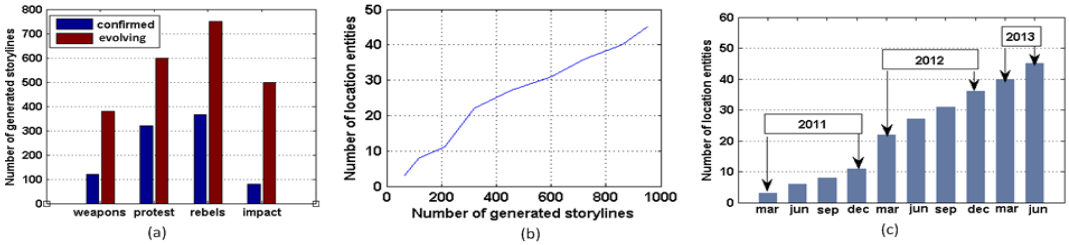


Figure 5.13: Effect on storytelling (a) by concepts (b) by locations (c) by location propagation over time

Temporal propagation of Hashtags: An advantage of the concept graph approach is that it allows one to perform detailed analysis by filtering the graph for specific types of entities and relationships. In addition, maintaining spatio-temporal information on all entities and relationships enables us to take a snapshot of the graph in different locations at any point in time. In this case, we are interested in the temporal propagation of the *Twitter* hashtags related to the *Syrian Civil War*. A hashtag is a form of metadata tag, or simply put, a word or phrase prefixed with the symbol #. Hashtags are neither registered nor controlled by any one group of users, and one of their most powerful effects is to group messages, since one can search for a hashtag and obtain the set of messages that contain it. When promoted by enough individual users, they can become "trend" and attract more users to the discussion. Over time, users discussing a specific hashtag typically start relating multiple hashtags, thus relating other discussions. In the case of the *Syrian Civil War*, it can be noticed that **#Syria**, **#Iran**, **#Assad**, and others start getting mentioned together as the civil war gains international attention. Two such snapshots are shown in Fig. 5.12, where the Syria hashtag suffers an explosion of co-mentions with many others from early 2012 to mid 2013. The significant difference in the number of hashtags between the two snapshots indicates the importance that the *Syrian Civil War* gained over time. This feature of the hashtags is used to evolve the concept graph which allows for an improvement in storytelling capabilities.

Effects of concepts on generated storylines: The high volume of tweets in the datasets generate a high number of storylines that cannot be easily confirmed in an automated manner. A Storyline that cannot be confirmed either manually or by automated means is what is denoted as an "evolving" story. This is illustrated in Fig. 5.13 (a), where the blue bars (confirmed stories) show

a much lower number of instances than the red bars (evolving stories). Selection of semantic constraints also influence this result. The abstract concept *impact* seems to have a very high proportion of evolving stories. However. these stories could further evolve into real news events in the future. If we focus the concepts more on the conflicts, we get a higher proportion of confirmed stories. This is illustrated in the bars for *protest* and *rebels*. The stories around *weapons* have recently seen an upsurge due to the recent allegations of use of chemical weapons.

Points of Consideration: In order to discover stories of interest, it is imperative to limit the search space to specific locations. For example, the concepts discussed in the previous section (*impact, protest, rebels, weapons*), at first glance, are very general, until they are combined with locations specific to the *Syrian Civil War*. If not bounded by the location (or radius), the semantic constraint *protest* could generate stories for the Syrian protests, Mexico protests, or any others around the world. Hence, the importance of the spatial component of our proposed work. In the context of the *Syrian Civil War*, the starting radius was fixed to the capital Damascus, and increased gradually to find connecting stories. Fig. 5.13 (b) shows that the number of stories increases with an increase in the number of locations.

The experimental results have helped characterize the importance of the spatio-temporal aspects on story coherence and recall. Proper understanding of the distribution of entities along with their relationships can help capture richer information content that could otherwise be missed. These extensive experiments demonstrate the potentially-high usability of the proposed methods, which fill a gap not currently addressed in the existing storytelling literature.

Experiment Summary The experiments in Subsection 7.5.2 demonstrated the potentially-high applicability of spatio-temporal storytelling, exemplified in an event summarization case study. *STS* yielded higher precision levels than existing methods (up to 28%) on highly-noisy small unstructured documents (i.e, *Twitter*). Rather than relying on textual content, *STS* introspects entities that are spatio-temporally tagged so to identify the ones with high levels of connectivity. Those entities are targeted for storyline generation regardless of how they are described in the underlying data source. In addition, *ConceptRank* helped differentiate the important relationships from the less relevant ones. This is an essential contribution to intelligence analysis, which often faces large data volumes, but have little ability to automatically segregate important connections among millions of possibilites.

Spatio-temporal storytelling's ability to capture the underlying links among entities is complemented by its flexible method of temporal propagation. Analysis with time windows promotes coherent storylines, which has the potential to uncover developments before they materialize. This was shown in the experiments of Subsection 5.5.2, where a set of ten events related to social unrest in Mexico in 2013 was identified. Even though forecasting was not the focus, *STS* proved highly successful in identifying events up to four days in advance of their publication in the news. In the most successful case, it was able to forecast a teacher's march for higher education budget with a twelve-day lead time.

The empirical analysis of Subsection 5.5.3 showed the propagation of concepts based on specific

items of interest, such as the dissemination of weapons or the activity of rebels. These concepts are pervasive in intelligence analysis and represent one area where spatio-temporal storytelling can be helpful. That section showed a case study that observed the evolution of the *Syrian Civil War* based on common hashtags. The volume of co-mentions with the **#syria** hashtag exploded significantly between 2012 and 2013. This is encouraging because it was also reflected in the generated storylines, which showed a high frequency of the #syria hashtag, and thus indicated that *STS* can effectively capture the importance of those entities.

5.6 Conclusion

In studying socio-political interactions from spatio-temporal propagation, this proposed work has been able to generate dynamic real-world storylines from *Twitter* sources that are of great significance to the intelligence community. Ranking is established based on different relationship types, and has proven effective on ill-formed datasets. Because spatial distribution is treated as an integral factor of the described algorithms, dense regions where storylines developed were identified. Further, this approach establishes time-coherent entity connections that otherwise may have been more challenging from purely textual approaches that do not consider the myriad locations such as the ones affected by the *Syrian Civil War*. Experiments on the *Mexico Civil Unrests* and the *Ukraine Political Crisis* demonstrated a high potential for applicability in tasks such as summarization and forecasting of current events. Future work will investigate more systematic methods of grounding the true "goodness" of generated storylines, and explore storyline coherence. Eventually, the objective is to establish storytelling as a robust tool for entity reasoning in a wide range of application domains.

Chapter 6

Spatial Similarity in Sequential Data Streams

Storytelling, the act of connecting entities through relationships, provides an intuitive platform to explore the dynamics of real-world developments. One of its shortfalls is the lack of a numerical similarity with which to compare and contrast stories, hindering its usage in geopolitical or social analysis. In this work, we take as input a set of spatio-temporal storylines related to violent events and show how they can be used in a wide range of analytical tasks. First, we devise a numerical similarity measure that can help identify related violent events, applying it to hierarchical clustering. Second, we demonstrate how storylines can help forecast violent events using common techniques such as *Bayesian* and *spatio-logical inference*. Third, we introduce a *spatial forecasting index* that observes how events propagate over space and time. Extensive experiments with social unrest in *Mexico* and wars in the *Middle East* compare and discuss the usefulness of each approach, highlighting differences, and demonstrating that they can be truly effective in exploratory analysis.

6.1 Introduction

Violent events are often the byproducts of complex factors of various natures, such as financial, political, and religious. For a violent event to take place, the right mix of signals must come together in order to elicit reaction. Take as an example Fig. 6.1, which depicts some of the locations of the *Poll Tax Riots* of Great Britain in 1990. Social unrest broke out after the government enacted a flat-rate tax on each adult. But before those acts of violence occurred, other developments led up to them: activists organized protests at *Trafalgar Square*, police closed a few of London's *Underground* stations, transit was rerouted in some streets, and shops closed in certain areas. The key idea here is that violent events tend to be associated to other spatially and temporally related nearby processes. These processes are composed of any number of constituent parts that, when identified properly, can help uncover the final event. While the above example is not surprising

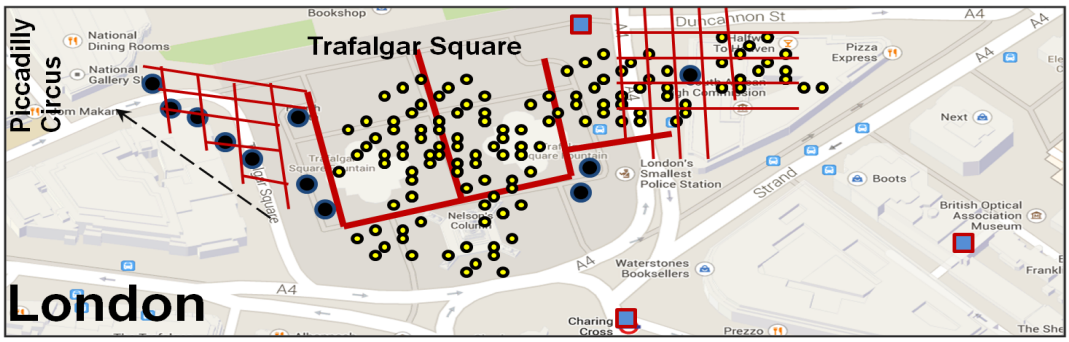


Figure 6.1: Approximate spread of the *Poll Tax Riots* of London in 1990. Red lines represent street closures around *Trafalgar Square*. Yellow dots denote concentration of protesters. Squares are closed subway stations, and black dots show locations of reported riots propagating north towards *Piccadilly Circus*.

(after all, protests can frequently lead to riots), acts of violence are not always transparent. The *Montreal Stanley Cup Riot* of 1993, for instance, developed quickly as the crowd celebrated a win, and had no apparent reason to engage in violence, when in fact it did. Violent events can take on many characteristics, four of which are observed in the above example:

1. **event cascading:** single developments provide little insight into the overall event. On their own, the *street closures* of the above example are not alarming. But when combined with other developments, such as *gathering of protesters* and *closed shops*, a much bleaker picture begins to delineate;

2. **event propagation:** developments evolve in spatial regions through nearby areas, fading into distance. Shops, for instance, are closed near the event, but not far away from it;

3. **event sequencing:** the temporal sequence in which developments occur is essential to explain facts. *Disruption in transportation*, for example, commonly takes place after protesters have gathered, but less frequently before;

4. **event interaction:** developments represent interactions among entities: police try to contain protesters, rioters throw stones, looters attack shops, etc. Some interactions provoke strong reactions, while others do not.

Given the spatio-temporal sequence of developments as described above, one interesting question is whether an event, violent or not, can be predicted based on previous knowledge of seemingly related developments. In other words, would it be possible to foresee looting at London's *Piccadilly Circus* knowing that major protests took place earlier at *Trafalgar Square*? Making such determinations has proven elusive even with the most advanced reasoning systems available today.

While forecasting has been an art as much as a science, we can measure the feasibility that an event will occur by expanding the four characteristics mentioned above to the following hypothesis: an event can be identified by the **constituent parts** that lead to it, observing their **spatial propagation** and **time coherence**, and taking into account their **semantic interactions**. The goal of this study is to reason over spatio-temporal sequences of developments that can lead to other events, and

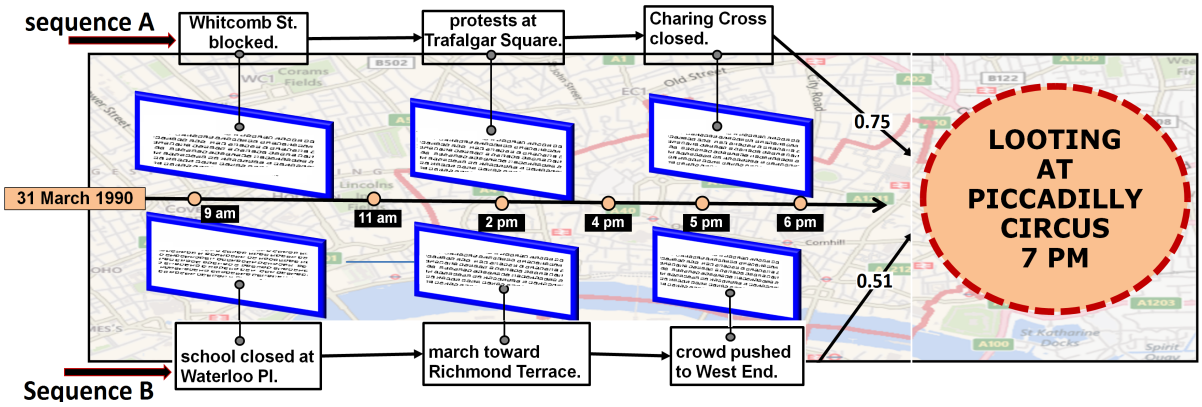


Figure 6.2: Example of forecasting from spatio-temporal storytelling on two event sequences A and B. The 0.75 and 0.51 values indicate the beliefs with which sequences A and B respectively forecast the looting. For its higher value, sequence A is deemed a better predictor than sequence B.

provide a probabilistic view of their occurrence. For a focused discussion, we base our case study on violent events. To be more specific, Fig. 6.2 gives an example of what we work towards. The figure shows two event sequences **A** and **B** across a short timeline (31 March, 1990). **Sequence A** is composed of three developments taking place along the day: *Whitcomb St.* is blocked, protests occur at *Trafalgar Square*, and *Charing Cross* station is closed. They culminate in looting in the vicinity of *Piccadilly Circus* at 7 pm. **Sequence B** has three different events in different locations, but also lead to the same looting at *Piccadilly Circus*. Given the two sequences (and possibly others), our goal is to give each sequence a numerical quantification of its ability to forecast the looting. We would like to say that **Sequence A** forecasts looting at *Piccadilly Circus* with a certain value, while **Sequence B** forecasts the same looting with a lower value than **Sequence A** . Thus, **Sequence A** is a better predictor than **B**. These values can be either a probability or an index, two approaches that we explore later. The above sequences represent streams of information that *tells a story*, and thus, we begin by framing this problem as one of *storytelling*, which we explain below.

Broadly speaking, *storytelling* is the process of connecting entities through their characteristics, actions, and events [134] in order to create meaningful streams of information. In the *Poll Tax Riots* example above, a possible storyline would be the sequence $\boxed{\textbf{\textit{activists}}} \xrightarrow{organize} \boxed{\textbf{\textit{protest}}} \xrightarrow{containedby} \boxed{\textbf{\textit{police}}} \xrightarrow{closed} \boxed{\textbf{\textit{streets}}}$, where entities {activists, protest, police, streets} are connected through semantic relationships {organize, containedby, closed}, and tagged with a location and timestamp. *Information retrieval* and web research have studied this problem, i.e., modeling storylines from documents and search results, and linking documents into stories [67][49][51] (the terms *stories* and *storylines* are used interchangeably). A violent event can be viewed as a vector of three important dimensions: the **spatial regions** where entities interact; **temporal coherence** which dictates the proper ordering of developments; and the **interactions** that lead to social outcomes. In this study, we enforce all of these three dimensions and focus on spatio-temporal *storytelling* related to violent events, presenting the following contributions:

1. **Devising a similarity measure between storylines**: Massive numbers of storylines demand an automated method to compare and relate them. Using spatial distances and *Dynamic Time*

Warping (DTW), we show how such similarity can be devised, serving to collapse thousands of storylines into fewer and more manageable numbers, at times in the single digits. In this manner, violent events can be told more coherently.

2. **Designing spatio-temporal methods to analyze events**: Because the dynamics of violent events are too complex for simple modeling, treating them in short spans of space and time is more conducive to human understanding and permits better real-world applicability. We show how such applicability can be achieved using spatio-temporal analysis techniques that combine our proposed storyline similarity measure with *Hierarchical Clustering*, *Bayesian Inference*, and our own *Spatial Forecasting Index*. They are demonstrated in four different forecasting strategies which has not been explored in spatio-temporal storytelling.

3. **Reasoning with spatio-logical inference**: Key to understanding violent events is to differentiate their relevant circumstances while filtering out the unimportant ones. We apply *spatio-logical inference* to determine the likelihood that parts of an event will occur, and by extension, if the final event is probable or not. In this manner, the analyst can focus on hundreds of important happenings rather than thousands (or millions) of uninformative developments.

4. **Performing extensive experiments over disparate datasets**: Because violent events are reported in various formats, we perform several experiments using both structured and unstructured data sources. Analysis of violent events is done on *Twitter* data and on *Global Database of Events, Language, and Tone* (*GDELT*) [71], the latter being a well-established dataset of conflicts and social unrest, from which we target events in the Middle East and other parts of Asia.

In this study, we briefly show how storylines are generated. For full details, however, we refer the reader to the spatio-temporal framework described in [116] and its originating work in [67], which we use as the basis for this research. The methods presented, however, are applicable to other storyline-generating approaches. Our focus is on spatio-temporal techniques of storyline usage to demonstrate how they can be helpful in real-world applications, using violent events as our domain. This article is organized as follows. In Section 6.2, we describe related works and point their differences to our approach. Section 6.3 describes our storyline similarity measure and show its use in hierarchical clustering. We begin discussion on forecasting in Section 6.4, detailing four different approaches, and present extensive experiments in Section 6.5. A conclusion is finally given in Section 6.6.

6.2 Related Works

Storytelling is not a single analytical tool with predefined tasks. It can be better described as a platform of knowledge exploration for fact finding, association discovery, and inferencing. Moreover,

its goals can range widely according to the domain of application: law enforcement may want to connect criminal behavior; health officials may be interested in drug interactions; and marketers may benefit from repercussion of their products in social media. As such, storytelling depends on a combination of social analysis and the technical quantitative fields. The work proposed in this study, therefore, spans many areas of expertise, from graph analysis to geographic networks. Our research best lines up with the approaches described below.

Storytelling and **Connecting the Dots**: The phrase 'storytelling' was introduced by Kumar *et al*. [67] as a generalization of *redescription mining*. At a high level, *redescription mining* takes as input a set of objects and a collection of subsets defined over those objects with the goal of identifying objects described in two or more different ways. Such objects may signal shared behavior, which can be a powerful tool in the context of *storytelling*. In [51], *Hossain et. al.* develop this idea to connect two unrelated *PubMed* documents where connectivity is defined based on a graph structure, using the notions of hammocks (similarity) and cliques (neighborhoods). This work was generalized to entity networks in [50] and specifically targeted for use in intelligence analysis. The authors' motivation is that current technology lacks better support for entity linkage, explanation of relationships, exploration of user-specified entities, and automated reasoning in general. The tools used in this work include concept lattices as a network where candidate entities are identified with three nearest neighbor approaches (Cover Tree, *k*-Clique, and NN Approximation). The *Soergel Distance* measures the strength between entities, while *coreferencing* serves to identify entities mentioned in various parts of the text using differing terms. These works link entities according to a desired neighborhood size and distance threshold. In many of these works, edge weight is based on a variation of term frequency $\times$ inverse-document-frequency (*TF-IDF*). This class of works represent *traditional storytelling* approaches that do not address the geospatial perspective.

In the realm of frequent pattern mining, research related to our work comes from *Cascading Spatio-Temporal Pattern Discovery (CSTP)*, proposed by Mohan *et al* [98]. *CSTP* identifies partially-ordered subsets of event types that are colocated and sequential. The goal of this approach is not to perform storytelling per se, but its focus on event association is a significant step in that direction. *CSTP* accepts boolean event types and computes a measure of *interestingness* for a pattern, namely a *Cascade Participation Ratio*, as the probability of observing a CSTP within an entire dataset of event types, such as events that lead to crime occurrences. In our approach, a point of differentiation is that event types are not necessarily boolean. We accept the notion of soft logic in which events have variable "truths" based on one's personal belief of what may have happened (or not). This approach has been optimized in several aspects, such as by minimizing the number of candidate patterns. With modifications, CSTP can be a valuable tool complementary to our work with respect to our proposed *spatial forecasting index* and to the rule generation of the *spatio-logical inference*. This is a potential item of exploration as part of our future work.

Connecting the dots-type approaches focus on document linkage rather than entity connectivity. They apply textual reasoning as a strong facet of the targeted methods, which departs from a spatio-temporal view of events. Link strength utilizes the notion of *coherence* across documents, which is proposed by [120]. In this work, stories are modeled as chains of articles, where the ap-

pearance of shared words across documents help establish their relatedness. Extending that work, they also propose related methods to generate document summaries, i.e. *Metro Maps*, in [122] and [121], which target scientific literature. Some of the goals are to measure the importance of a article in relation to the corpus, find the probability that two articles originate from the same source, and identify research lines. Overall, *connecting the dots* methods rely on the abundance of robust content. The types of datasets used in our study (*Twitter* data and *GDELT*), however, break the assumption of robust content, limiting the amount of textual reasoning that can be performed. Thus, *connecting the dots* is less than ideal for environments that rely on such data feeds.

Inferencing and **Forecasting**: While the goal of this study is not to compare the best forecasting strategies, we briefly discuss some interesting forecasting approaches. Some authors prefer the terms 'event prediction' while others speak of 'causality' in relation to forecasting. One such work proposed by Radinsky *et al*. reasons over the causes of events described in news articles [105]. They present an algorithm that takes as input a causality pair to find a causality predictor. Objects are defined to be similar if they relate to a third object in the same way. This departs from our approach, which does not compare entities, but rather investigates if *behavior* is similar when entities are in close spatial proximity. Further, their work utilizes external knowledge databases, such as *LinkedData* to obtain information about well-known objects and entities. Our approach is mostly unsupervised in that all knowledge is self-contained in the targeted datasets.

Another work worth mentioning is prediction from textual data described in [103, 102]. The authors propose to capture the effects of an event by propagating it through a hierarchical model, namely an *abstraction tree*, that contains events and rules. It then finds matching nodes that can produce possible effects. In our work, we also propose a rule-based method, but do not rely on a trained model that stores rules for subsequent use. Our idea is to compare events, which may be viewed as nodes, where each event has a weight based on spatial distance. We favor this methodology as it does not depend on the availability of entity attributes or physical characteristics.

In our discussion, we note the importance of *Bayesian Inference* in forecasting. Among classical methods, it is one the strongest foundations for *cause-effect* relationships that one can use to justify forecasting. Determining that A happens because B and C also happen is a powerful statement in many areas of knowledge, although it must be taken carefully. *Bayesian Inference* in its traditional form, however, is challenging for a few reasons: (1) it needs many instances of the same events to occur in like sequences to establish certainty; (2) without modification, it does not consider subjective criteria, such as behavioral knowledge or entity characteristics. Things "are" or "are not"; (3) it does not take into account spatial reasoning. Every element, no matter where they reside, are regarded equally. In terms of violent events, these three aspects represent challenges that must be dealt with. For this reason, we do utilize *Bayesian Inference* as a forecasting method, but do not solely rely on it. Our discussion will also include three other approaches (*distance-based Bayes*, *spatial forecasting index*, and *spatio-logical inference*) that mitigate *Bayes*'s effects.

Trust Management: A common issue in spatio-temporal storytelling has to do with the reliability and trustworthiness of entities. Especially in social networks, many facts are reported erroneously

while others are deliberately malicious. This raises the prospect that storytelling should incorporate trust as a means to enforce truthful stories. Identifying illegitimate nodes and events not only strengthens coherence, but also serves to reduce the data space by eliminating unwanted entities. For the purposes of this research, trust can be viewed in terms of connectivity, propagation, and ranking, as explained below:

1. **Connectivity**: Trust can flactuate with new experiences and decay with time [127]. In temporal terms, new experiences are more important than old ones, since the relevance of facts become obsolete gradually. Various techniques have been used to model connectivity as a function of time. The works of Wishart et al. [142] and Kamvar et al. [59] operate on the aging of interactions. Giving more weight to recent interactions is studied by Song et al. [126], and Zhang and Fang [151]. In some models, such as PeerTrust [144, 145], users are allowed to choose the temporal window for dynamic aging of old interactions in a customizable manner. An alternative approach is PowerTrust [152], where a trust computation is performed periodically to ensure that the computed trust values are up-to-date.

2. **Propagation**: Trust spreads across space much in the same way that entities travel in a certain direction. For example, assume that person A has a trusting relationship with person B in one area, who in turn is also trusted by person C in a nearby place. It can be stated that the storyline that these three entities represent is legitimate not only in their locations, but also in other areas to where they propagate. This notion of trust propagation is explored by Josang et al. [57], where a recommendation system allows propagation according to explicit conditions. Various trust models [[117], [92], [113], [149]] have used this property. For the purposes of storytelling, sliding windows, as explained in Chapter 5, can have significant benefits in terms of more precise entity reasoning.

3. **Ranking**: Trust is typically variable. An entity may trust another entity more than the former is trusted back. When both are trustworthy, they will converge to high mutual trust. In this sense, trust serves to establish ranking, and possibly enhance the *ConceptRank* model of Chapter 5. This type of trust has been identified in various hierarchies within organizations [147], and can potentially serve as a differentiator among the highest and lowest connected entities in a network.

Differences: Each of the above research fields provides solutions to the various tasks involved in storytelling. Challenges and requirements come in different flavors as a result of application demands or data characteristics. Our work, for instance, requires geolocation of entities as it relies on a spatio-temporal model where both geographical proximity and time ordering are favored. In this sense, our focus is on methods for which spatial influence and time sequencing can be intuitively justified by semantic analysis. Given the many differences in what each technique can contribute, we do not show competing approaches. Rather, we present complementary techniques that demonstrate how storylines can be a valuable analysis tool, covering a spatio-temporal niche which remains largely untapped.

6.3 On Relating Violent Events

This section introduces the definitions and nomenclature used throughout the remainder of this work and provides a visual representation of our tasks. We further propose a method that establishes numerical similarity between storylines and apply it to *hierarchical clustering* for subsequent use. At a high level, the work proposed in this study follows the steps shown in Fig. 6.3, explained below.

6.3.1 Analysis Framework

To generate storylines, we reuse our previous work detailed in [116]. We do not delve into details, but summarize the approach here. It takes as input a dataset with entities for which locations and timestamps are available or can be obtained. Locations are geocoded into latitudes and longitudes, and entities are extracted, stored, and indexed spatially. Relationships between entities are also extracted. A relationship is an interaction between two entities, such as when "person-1 talks to person-2", in which case the relationship is "talks". An entity graph is then built by linking the extracted entities to the extracted relationships. For each entity in the graph, a *ConceptRank* (i.e., a variation of *PageRank*) is calculated. The storylines are formed in 3 steps: (1) the user selects an entity to be the entrypoint, i.e, the point from where the story begins; (2) from the entrypoint, the algorithm applies *Ripley's K function* to find an optimal radius within which the concentration of entities is high; (3) within that radius, the entrypoint is linked to the top-k entities of highest *ConceptRank*, sorted in time order. This set of linked entities is the final storyline, which has the general format $\boxed{\text{entity-1}} \overset{relationship-1}{\rightarrow} \boxed{\text{entity-2}} \overset{relationship-2}{\rightarrow} \boxed{\text{entity-3}}$. The length of the storyline may vary without bound.

Once the storylines are available, the first task is to calculate a numerical similarity between storylines (Subsection 6.3.2). Because the number of generated storylines can be massive (especially from *Twitter* data), an intermediate step is taken to perform hierarchical clustering (Subsection 6.3.3). The clustering process serves two purposes: segregate the events in the storylines into related groups and allow processing to be done on a per-cluster basis, which is more manageable. In the final step, three methods are explored to reason over violent events, providing intuitive justifications for their use. Those methods, *distance-based Bayesian inference* (along with traditional *Bayesian inference*), *spatial forecasting index*, and *spatio-logical inference* provide the foundation for our forecasting strategies of subsections 6.4.1, 6.4.2, and 6.4.3.

Ontological Resolution: In order to capture the likeness of concepts described differently, this research uses entity resolution based on ontological categories. An ontology is a hierarchical structure that groups similar concepts in the same branch, organizing them from most general to most specific items. As a data structure, it has been used extensively in a wide range of domains for many different purposes, such as classification, data annotation and mediation, and reasoning over inconsistency [35].

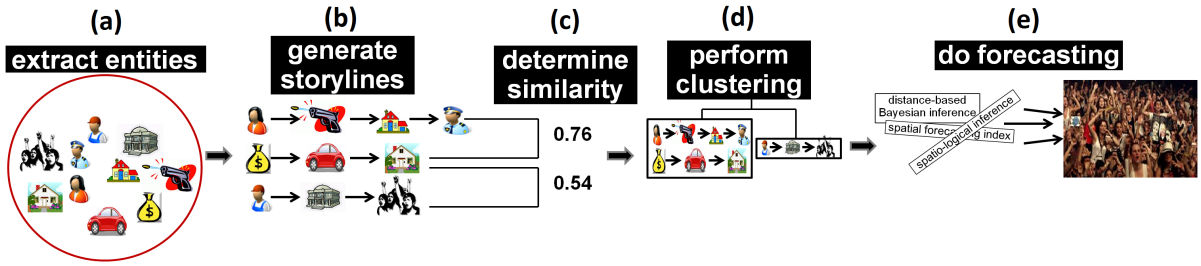


Figure 6.3: Forecasting process using spatio-temporal storylines: (a) entities are extracted; (b) storylines are generated; (c) a numerical similarity determines distance among storylines; (d) storylines are hierarchically clustered; (e) forecasting is performed using three methods.

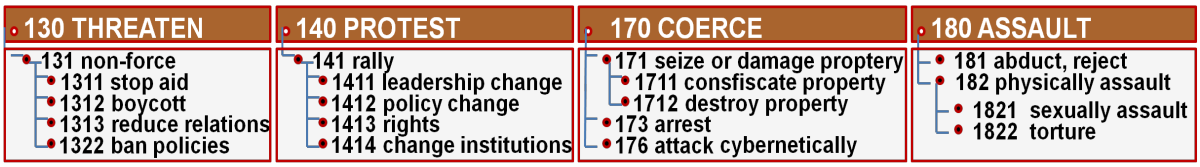


Figure 6.4: Ontological hierarchy of four *GDELT* events. Each category is composed of one or more groups in different levels of resolution. Concepts become more specific with increasing depth. Partial list.

In this work, ontologies are used as a means to combine similar concepts in order to treat them as one. We select the hierarchical structure given by *GDELT* (which is also one of our experimental datasets), and provides an extensive categorization of spatio-temporal events. Fig. 6.4 illustrates four of *GDELT*'s categories. In the top-level category (*130-THREATEN*), there is one subcategory (*131 non-force*) along with four other items as leaf nodes. The other top-level categories (*140-PROTEST*, *170-COERCE*, and*180-ASSAULT*) show somewhat similar structures.

From an application perspective, items in the same branch are deemed similar, and can be treated as the same type. Thus every "boycott" (1312) can be considered a "stop-aid" (1311) with the assumption that loss of information is tolerable. Similarly, every "abduction" (181) or "torture" (1822) are simply instances of "ASSAULT". In many datasets, events are described in varying terms, and thus combining them, can be useful if no specificity is required. When high resolution is desired, combining only certain elements, but not all, can also serve a purpose. Throughout this study, we take the following direction. Whenever at least 1,000 data points are available for the same violent events, we do not combine others from different categories. This is the case with "protests", which are abundant, and need not be supplemented. For "reduce relations", however, not enough information is available, and is thus consolidated with other items, such as "boycott". In our experiments, ontological resolution has proven more useful with *Twitter* data, which tends to be noisy and highly ambiguous, but not so much with *GDELT*, which is well structured and segregated.

Storylines can be powerful in understanding violent events due to their flexible nature: entities can be added or removed, regions of introspection can be expanded or shrunk, time windows can be varied, all while observing how social interactions play out to help explain facts. The true value of storylines, however, only surfaces when they are put to use. In order to either compare or relate violent events portrayed in different storylines, it becomes essential to establish a numerical similarity between pairs of storylines. The next section describes a method for such purpose, and

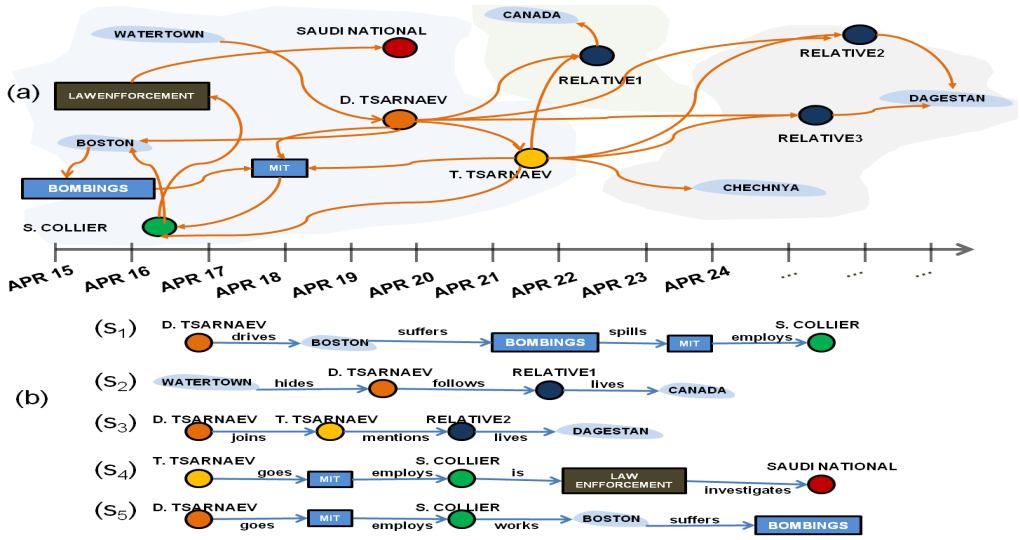


Figure 6.5: *Boston Marathon Bombings* spatio-temporal sequence. In (a), each shape represents an entity observed in the data source. The edges denote relationships between the entities. In (b), S_1 through S_5 represent five storylines connecting different entities. The English verbs define their relationships and correspond to the edges of the concept graph in (a).

subsequently explain its use in hierarchical clustering that groups similar violent events. **Definitions**: In the scope of our study, a storyline describes an event (or development) as the interaction among entities linked by relationships. A violent event is one that causes hardship at the individual, organizational, or governmental levels. Physical harm does not need to be involved. Unless otherwise stated, the following definitions will apply going forward:

Definition 8. *An entity e represents a person, location, organization, event, or object described in a document. Only entities for which a location and a timestamp can be obtained are considered in this study.*

Definition 9. *A relationship, connection, or link defines a unit of interaction between two entities and is denoted by* $e_i \xrightarrow{interaction} e_j$. *All relationships* $e_i \xrightarrow{interaction} e_j$ *are intended to be directional.*

Definition 10. *A trigger event or final event represents a real-world development extracted from text, such as an "explosion" or a "protest". They can be user-defined or application-specific based on an external ontology.*

Definition 11. *A storyline is a time-ordered sequence of n entities* $\{e_1, ..., e_n\}$ *where consecutive pairs* (e_i, e_j) *are linked by one relationship. The number of entities n is the length of the storyline.*

6.3.2 Storyline Similarity

During the storyline generation process, many storylines may arise, which demands a numerical method to compute their similarity. Determining similarity is an essential task with a wide range of uses: performing classification, identifying outliers, removing duplicates, among others. In this

section, we demonstrate a method of computing similarity between storylines and later apply the computed values for clustering purposes.

Traditional similarity measures compare entities in different ways: (1) by differences in numerical attributes, such as a person's age or other specific traits [28]; (2) by frequency of attribute types, as in the works of *Goodall* and *Leacock* [44, 70]; (3) by nominal values based on the colocation of entity types in space [115]. These approaches, however, pose obstacles which prevents us from using them in this study. First, we do not expect the availability of attribute data. In social networks, for instance, entities can be very poorly-described, making attribute comparisons not adequate. Second, events can be highly-infrequent, making frequency approaches less than ideal. Third, these methods perform either over documents or entities, but not storylines.

In light of these issues, we refrain from using the above approaches, and propose a measure of similarity between storylines based on *Dynamic Time Warping* (*DTW*), which has been utilized successfully to compare time-series in various applications [61]. *DTW* is applicable to this research because it is able to gauge the difference between two sequences of different lengths, with non-matching entities, and separated by a time shift. In general, these properties are also observed in storylines, and thus our selection of *DTW*.

At a high level, *DTW* takes as inputs two data sequences, and outputs a numerical distance between them. Consider, for instance, Fig. 6.5, which depicts a temporal sequence of developments related to the *Boston Marathon Bombings* of April 15, 2013. Fig. 6.5(a) shows an entity graph where the nodes represent various entities involved in that event. Several outcomes came out of it: two suspects were eventually identified (D.TSARNAEV and T.TSARNAEV), LAW ENFORCEMENT investigated a SAUDI NATIONAL, Police Officer S.COLLIER was shot near the MIT Campus, and some of the suspects' RELATIVES were identified in CANADA and DAGESTAN. Fig. 6.5(b) shows five storylines (S1 through S5) generated from the entity graph of Fig. 6.5(b). Looking closely at each storyline, some of them appear related, as they share certain entities, while others appear more disparate. Straight comparisons between them is difficult because these storylines are neither all the same length, nor do their entities align uniformly. *DTW* takes care of both of these problems by finding an optimal alignment between the entities based on a minimum distance function, even when they are out of sequence or some elements are missing.

Given two storylines S_a and S_b composed respectively of entities $\{a_1,...,a_n\}$ and $\{b_1,...,b_m\}$, *DTW* in its simplest form specifies the distance between S_a and S_b as follows:

$$DTW(n,m) = \begin{cases} 0, & \text{if } (n,m) = (1,1). \\ \infty, & \text{if } (n,m) = (n,0) \text{ or } (0,m). \\ dist(n,m) + min\{DTW(n-1,m), DTW(n,m-1), DTW(n-1,m-1)\}, & \text{otherwise.} \end{cases} \tag{6.1}$$

where $dist(n,m)$ is a distance function between the n^{th} entity of S_a and the m^{th} entity of S_b.

As an illustration, we apply *DTW* to compute the distance between storylines S_1 and S_5 from Fig. 6.5(b). For simplicity, but not required, both S_1 and S_5 have 5 entities (n=5,m=5). The computation is better visualized as a 5-by-5 matrix, shown in Fig 6.6(a). Each cell (i,j) of the

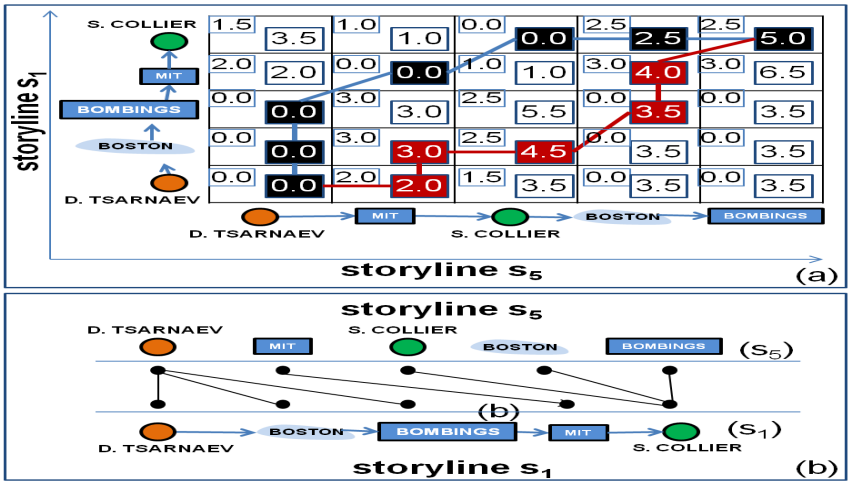


Figure 6.6: (a) Entity matrix of storylines S_1 and S_5. Values on the upper left corner of each cell represent *Euclidean* distances between the corresponding entities. Values on the main part of the cell are the DTW values. (b) Time-warped mapping between the two storylines.

matrix contains the *Euclidean distance* between entities a_i and b_j in its upper left corner. This *Euclidean distance* corresponds to the dist(n,m) of Eq. 6.1 (other distance types may be used). Our goal is to apply Eq. 6.1 to fill in each cell of the matrix with $DTW(i, j)$. For example, DTW(1,1) = 0 and $DTW(2,1) = dist(2,1) + min\{DTW(1,1), DTW(2,0), DTW(1,0)\} = 2$. When all cells have been filled in, the last element in the array, in this case [5,5] = 5, represents the distance between the two storylines. The higher the distance the less similar the storylines.

Fig. 6.6(a) also shows two paths from the first element of the array (1,1) to the last(5,5). Starting from [1,1] and connecting each cell to its lowest-DTW neighbor, we find the optimal mapping between the two storylines, which in this case is represented by the black squares. Note that other paths are possible, such as the red one, but those traverse through cells of higher DTW values, which is undesirable. Fig 6.6(b) shows how the entities are mapped to one another according to the matrix. D.TSARNAEV, for example, is linked to itself, to BOSTON and to BOMBINGS. Note that this mapping is time-warped, i.e. not aligned, for two reasons: first, the entities are ordered according to time of observation, and thus do not align linearly; second, the comparison metric is the *Euclidean* distance, which favors nearby entities. For *storytelling*, it indicates how entities are spatially-connected as opposed to how similar they truly are in terms of their physical characteristics or any other factors.

6.3.3 Clustering Violent Events

The $DTW(n, m)$ values obtained in Subsection 6.3.2 allows us to compare storylines based on spatial distance. Using these distances, our goal here is to segregate the storylines such that the nearby ones fall in the same group, which would arguably indicate that those storylines discuss related ideas. One way to accomplish that is to perform *hierarchical clustering*, which is desirable for its ability to organize data in levels [97], and can be helpful in separating entity sequences. Other types of clustering are applicable as well. Performing *hierarchical clustering* now will allow

us to do forecasting later on using only storylines and events from the same clusters, which may help increase coherence.

A simple, yet powerful variation of *hierarchical clustering* is the *agglomerative* approach, which begins with one storyline per cluster and works to merge them successively. The algorithm follows four simple steps: **(1)** allocate each storyline to its own cluster; **(2)** using the *DTW* distances, identify the nearest clusters and merge them into one; **(3)** recompute the distances from the new cluster to all the other remaining clusters; **(4)** repeat (2) and (3) until the desired number of clusters has been reached or there is only one cluster left. The computation of step (3) follows a *single-link* strategy, in which the distance between two clusters is equal to the shortest distance from any storyline in the first cluster to any storyline in the second cluster.

The above clustering process is well established and, to a certain extent, should yield the right groupings of related storylines. However, one modification is necessary to make sure that only proximally-close events fall in the same clusters. For example, one may want to put together all storylines with an instance of "bombing", "explosion", or "attack", what we denote as **trigger events**, that took place within a delimited spatial region. For such task, we cannot simply retrieve storylines with those specific keywords as we risk correlating totally disparate violent events, such as a "market bombing in Iraq" with a "factory explosion in Texas". Thus, not only are the concepts (*i.e.*, keywords) important, but so are the geospatial locations. To fix this problem, we can take step (2) of the clustering process and make the following modification. Cluster A is merged with cluster B: (a) if they share at least one trigger event; (b) if A is within distance d of cluster B. We first consider both criteria at the same time. If no merging occurs, each criterion is then considered individually. In this manner, we enforce separation when a specific scenario is desired.

The above example illustrates how *trigger events* along with spatial reasoning drive similar elements together. It also has the advantage of relating similar events earlier on in the merging process than conventional clustering would in many instances. There is no limit on the potential applicability of how the clustered storylines can be used: find similar documents, identify common patterns, relate social interactions, perform summarization, among many other tasks. In the next section, we discuss methods that utilize *spatio-logical inference*, *Bayesian Inference*, and a simple, yet powerful *spatial forecasting index* that more systematically reason over violent events useful in real-world applications.

6.4 On Forecasting Violent Events

Now that we have similar storylines clustered by distance from Section 6.3, we would like to investigate its applicability to different forecasting strategies. A key consideration here is determining if a sequence of events has any causal relationship to a subsequent one, in which case the former would serve to forecast the latter. Consider, for example, the *Boston Marathon Bombings* example, which was the result of two persons acquiring explosive devices, delivering them to specific locations, and setting off the attacks. In practice, we seek the extent to which

PERSON $\overset{acquire}{\rightarrow}$ DEVICE $\overset{deliver}{\rightarrow}$ PLACE necessarily implies a $\overset{set-off}{\rightarrow}$ ATTACK, which would indicate some sort of cause and effect. Without heavy data analysis and strong supporting evidence, this type of **causality** is nearly impossible to demonstrate [66]. Instead, just as important is to demonstrate **association**, which is a more loose concept and can be intuitively justified with the following:

1. **Event support:** An event cannot happen at random. It requires prior support, whether financial, logistical, or others, and therefore, associations between the event and its support is automatically built into the process. Mathematically, it can be stated that when n entities are observed in a spatial region, then there exists an entity $n+1$ which is bound to be observed as well. This denotes *Bayesian Inference* and by extension *distance-based Bayesian Inference*;

2. **Event influence:** Events may affect other events propagating through different regions. This means that an event in one area can influence a different event in a different area, allowing us to compute a *spatial forecasting index*, which we describe later;

3. **Event interpretation:** Abusing the notion of inference, we state that violent events unfold as a consequence of prior developments, which implies that an inherent association exists among them. Each event can then be viewed as an independent interpretation embedded with a certain amount of uncertainty. This leads us to formulate forecasting in terms of *spatio-logical inference* in which a large number of possibilities that explain a violent event can be reduced to the most probable causes;

The above items ground associations between entities and events, leaving us to explain our forecasting strategies in the next subsections. For this purpose, there are two end goals: (1) find the likelihood that a storyline (and its associated events) will happen again, in which case we say that the storyline *forecasts itself*; (2) find the likelihood that a storyline in one location influences another storyline in a nearby location, i.e., one storyline *forecasts another*. To achieve these two goals, four methods are presented: traditional *Bayesian Inference*, *distance-based Bayesian Inference*, *spatial forecasting index*, and *probabilistic logic*, which we discuss below. Note that, previously, we explained *DTW* as one of our similarity measures. Going forward, we investigate others in order to demonstrate a greater variation of similarity measures for which storytelling is applicable.

6.4.1 Forecasting Violent Events with Distance-based Bayesian Inference

In traditional *Bayesian Inference*, forecasting is done by viewing each storyline as a *Bayesian Network*, in which each entity represents a node specified by a *Conditional Probability Distribution*(*CPD*). Mathematically, if a storyline is described by three entities Ⓐ→Ⓑ→Ⓒ, we may want to find out its likelihood of occurring again, which is given by the joint probability of that entire storyline:

$$P(A,B,C) = P(A) \times P(B|A) \times P(C|B) \tag{6.2}$$

or, alternatively, one may want to simply find the probability of observing Ⓒ knowing that Ⓑ was observed in the past:

$$P(C|B) = \frac{P(B|C) \times P(C)}{P(B)} \tag{6.3}$$

Given the above, forecasting can be done either for single entities or events or for the entire storyline. In either case, we need to know the a-priori frequencies of all entities associated with a storyline. Referring back to the five storylines of Fig. 6.5(b), and assuming those five storylines represent the entire available dataset, we have the prior knowledge that police officer S.COLLIER was killed. Now, we would like to know the likelihood that another police officer will be murdered in the near future. The best answer lies with S_4, which is the only storyline that contains a LAW ENFORCEMENT presence and also someone related to a previous similar crime (T.TSARNAEV). Numerically, the forecast corresponds to the joint probability of that storyline in relation to all the other four storylines: P(T.TSARNAEV) × P(MIT ‖ T.TSARNAEV) × P(S.COLLIER ‖ MIT) × P(LAW ENFORCEMENT ‖ S.COLLIER) = $\frac{2}{5} \times \frac{1}{5} \times \frac{3}{5} \times \frac{1}{5} = 0.0096$. We can finally state the following: given recent data, there is less than a 1% chance of another police officer being murdered in the vicinity of the Boston area. In its traditional form, *Bayesian Inference* works well for highly-frequent storylines, but poses two problems for storytelling: entities must match perfectly and it does not consider the aspect of location. Next, we explore an approach that relieves these issues.

An intuitive approach to forecasting is to simply search the data space for similar storylines that reoccur in constant time intervals. In this case, similar storylines are defined as the ones composed of the same entities or a subset thereof. For instance, if FLOOD $\xrightarrow{causes}$ CHOLERA $\xrightarrow{promotes}$ VIOLENCE $\xrightarrow{affecting}$ MOZAMBIQUE is observed every 5 years, then we can assume this pattern will be observed again in the next five-year interval. In many applications, however, perfect sequences are seldom found, which forces us to relax the definition of storyline similarity with two modifications: (1) two storylines are similar if the location of at least one entity in one storyline is within a d distance of the location of an entity in the other storyline; (2) and apart from location, the two storylines must share at least one entity. And unlike what traditional *Bayesian Inference* would require, entities must not match perfectly. As long as the entities belong to the same 'concept' or 'category', they are deemed to be the same, as explained in Subsection 6.3.1. Similarity, in this case, is determined by an ontological structure appropriate for a specific application domain.

In the above discussion, as long as two storylines are in close spatial proximity and share at least some characteristics, then we assume that they are similar enough. For example, assume that for any given day, either storyline S_1 = Ⓐ→Ⓑ→Ⓒ→Ⓓ or storyline S_2 = Ⓐ→Ⓩ→Ⓒ→Ⓓ has appeared for the past year (for simplicity, we use letters for entity names and do not show relationship tags above the arrows). Assume also that Ⓐ is the location on both S_1 and S_2. Since they have the same location, and share two other entities (Ⓒ and Ⓓ), then we can say that S_1 and S_2 forecast each other and that either one of them will be observed again the next day. In practice, we search for storylines that are not only nearby each other, but also share a minimum number of entities, which define a common theme of discussion. Since these storylines are now consider the "same"

by virtue of similarity, traditional *Bayesian Inference* can be applied on them to find the probability of occurrence. This is what defines *Distance-based Bayesian Inference*, which is attractive for its simplicity and very promising in the analysis of violent events. As part of our experiments, we show this method as a forecasting strategy.

6.4.2 Forecasting Violent Events with Spatial Forecasting Index

The previous approach finds storylines and verifies if they could reoccur. A more powerful aspect of forecasting violent events, however, is to measure influence, i.e., whether the observation of a storyline in one place influences the occurrence of another storyline in another place. The *Boston Marathon Bombings*, for instance, provoked a myriad of reactions ranging from street closures around the blast site to a shootout in Watertown, a nearby area. In other words, an event in area A triggered other events in areas B, C, D, etc. At a high level, this is spatial correlation [124] framed in terms of entities and their interactions, rather than through traditional comparison of specific attributes, as in the work of [139].

As mentioned earlier, our datasets are not rich in attributes, so we work at the entity and relationship levels instead. We must consider the following: if the influence of area A on area B is high(low), then there is high(low) likelihood that whenever A experiences a storyline, there exists other storyline(s) that B will experience. Our goal then is to find out the storyline(s) that B will experience and identify the violent events behind them. Note that the storylines observed by A could, but need not be the same as the storylines observed by B. This is what we denote by a storyline that forecasts another, rather than forecasting itself. In order to gauge the level of influence between locations given their respective storylines, we propose a *spatial forecasting index* as described below.

The first consideration is that influence is stronger when entities are located within a reasonably-short distance of one another, and thus location is an important aspect. The second is that, for associations to happen, there must exist a minimum amount of commonality that bridges the two locations. In other words, events must not only be spatially close, but must also share entities. These ideas are combined to design our index. In this approach, we must first establish the following definitions:

Definition 12. *The distance between storyline S_x, composed of entities $E = \{e_1,...,e_n\}$ and location l_y, denoted dist(S_x,l_y), is the shortest distance between any $e_i \in E$ and any point in l_y.*

Definition 13. *The distance between two storylines S_x and S_y, composed respectively of entities $E_x = \{e_1,...,e_n\}$ and $E_y = \{e_1,...,e_n\}$, and denoted dist(S_x,S_y), is the shortest distance between any $e_i \in E_x$ and $e_k \in E_y$.*

Def(s). 12 and 13 establish distance as a function of the closest entity to a specific location or to another entity in space. We treat distance in spatial terms, and prefer *metric* measures such as *Euclidean*, since they conform to symmetry, which simplifies distance computations. In practical

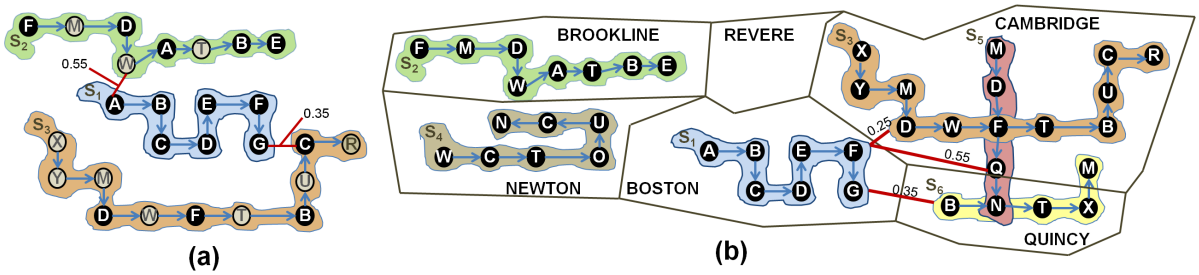


Figure 6.7: Hypothetical set of storylines located in different regions. (a) Three storylines of various lengths and different numbers of entities. (b) Six storylines spread across various cities. The circles denote entities and the edges represent relationships. Red lines denote the shortest normalized distances between corresponding storylines.

use, however, other metrics can be just as applicable. The *spatial forecasting index* (SFI) between two storylines S_x and S_y is then defined as:

$$SFI(S_x, S_y) = \log\left\{\frac{1}{dist(S_x, S_y)} \times n\right\} \tag{6.4}$$

where n is the normalized number of shared entities between S_x and S_y. Eq. 6.4 indicates that shorter distances and high numbers of shared entities contribute to a larger value, which indicates a stronger level of forecasting, and is indeed the desired effect. As an example, Fig. 6.7(a) shows three storylines, S_1, S_2, and S_3, all of different lengths. S_1 and S_2 share 5 entities (Ⓐ,Ⓑ,Ⓓ, Ⓔ, and Ⓕ). The two closest entities between S_1 and S_2 are Ⓐ and Ⓦ, which at a distance of 0.55 (normalized on a [0,1] scale), determine the distance between these two storylines. Calculating their *SFI*, therefore, yields $SFI(S_1, S_2) = \log\left\{\frac{1}{0.55} \times 5\right\} = 0.96$. Repeating the calculation for S_1 and S_3 results in $SFI(S_1, S_3) = \log\left\{\frac{1}{0.35} \times 4\right\} = 1.05$. Comparing the two results, we can then claim that storyline S_1 forecasts storyline S_3 better than it forecasts S_2. In everyday language, these results would be akin to stating that whenever events of the first storyline happen in one location, they are more likely to be followed by events of the third storyline. Note that the *SFI* values are not restricted to the range [0,1], and thus, are not probabilities. Rather, they are a spatial measure of influence that can be used to compare storylines. True probabilities can be computed using *Bayesian Inference* as described previously, but while it considers frequencies, it does not take into account the spatial factor of the storylines.

Eq. 6.4 requires that two storylines be supplied ahead of time. In exploratory analysis, however, one may want to investigate not simply two storylines, but rather the influence of a source location on a target location based on their storylines. A classical example are protests, which many times originate peacefully in a small area and spread as looting, fights, and other acts of violence in various directions. In such scenario, influence is better understood as a location-to-location process based on a random source storyline S. For location to location, what we initially have is one storyline and we want to find out the influence of its location on other nearby locations. To achieve this redefined notion of influence, we reuse our SFI index above with the following algorithm:

1. starting from a user-specified source storyline S of interest in a desired area of study, we first identify the closest location to S that meets the following condition: the identified location

must reference at least one storyline that shares at least one entity with S. Call the identified location L_{target} and the location of the closest entity in S as L_{source}.

2. retrieve the set of all storylines that refer to location L_{target}. Call that set $ALL{-}STORYLINES$.

3. using $ALL-STORYLINES$, compute the *spatial forecasting index* of L_{source} on L_{target} w.r.t. S:

$$SFI(L_{source}, L_{target}, S) = \sum_{i=1}^{|ALL-STORYLINES|} SFI(S, ALL-STORYLINES_i) \tag{6.5}$$

The above algorithm operates in the following manner: given a source storyline, it finds the closest nearby region that also has storylines with similar entities (at least one). It then investigates all of the discovered storylines for that nearby region, calculating their SFI values, and summing them up into one aggregated value. This aggregated value represents a numerical measure of storyline influence between the originating location (source) and the investigated location (target). Again, the higher the *spatial forecasting index* the stronger the level of forecasting. A visual example follows.

Fig. 6.7(b)shows five regions (*Brookline*, *Newton*, *Revere*, *Cambridge*, and *Quincy*) around the *Boston* area. Except for *Revere*, all areas contain at least one storyline, and some of their entities take part in more than one storyline. This is the case of 🅕, which is observed at different times in *Brookline*, *Boston*, and *Cambridge*. Imagine that an analyst would like to understand how the events in *Boston* imply events in those other areas. Following the algorithm above, the analyst would first identify the closest area to *Boston* that has a storyline which share one or more entities with a *Boston* storyline. It turns out that storylines of all areas share entities with the *Boston* storylines. In this case, the chosen location is *Cambridge* since it is the closest to *Boston* considering driving distance (when several locations are equally distant, the one with the highest number of common entities is selected before a random choice is made). Thus, according to *step 1*, $L_{source} = Boston$ and $L_{destination} = Cambridge$. As per *step 2*, we now retrieve all storylines associated with *Cambridge*, which according to Fig. 6.7(b) are S_3, S_5, and S_6. In the last step, we compute all SFI values between S_1 and each of S_3, S_5, and S_6 (Eq. 6.4), and sum them up (Eq. 6.5). Considering the storyline distances shown by red lines in Fig. 6.7(b), the computations would be: $SFI(Boston, Cambridge) = SFI(S_1, S_3) + SFI(S_1, S_5) + SFI(S_1, S_6) = \log\left\{\frac{1}{0.25} \times 4\right\} + \log\left\{\frac{1}{0.55} \times 2\right\} + \log\left\{\frac{1}{0.35} \times 1\right\} = 2.21$. One could certainly perform the same calculations for any other areas, e.g., *SFI(Boston,Brookline)*, and compare their *spatial forecasting index*.

The algorithm outputs one *SFI* value for each pair of storylines and all storylines are considered within those locations. Optimizations can be done, such as pruning locations known to be uninteresting, or removing storylines known to be uninformative. Intuitively, this approach allows the analyst to see how events propagate in time and space, providing a numerical value of confidence that developments in the first location will be followed by developments in the second one. For this reason, we state that the *SFI* has a strong forecasting potential, and thus its name. For example,

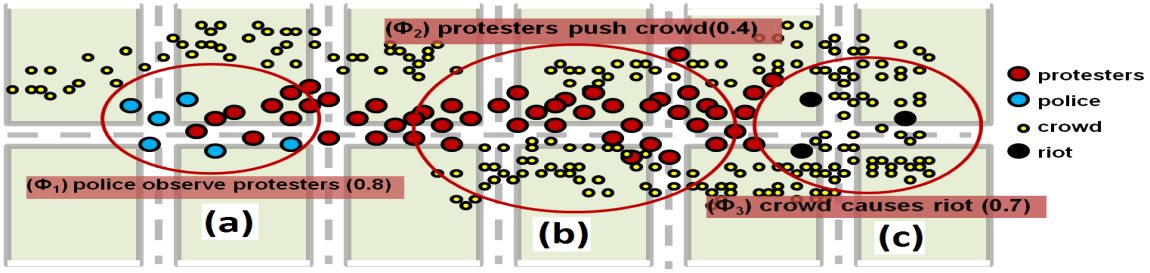


Figure 6.8: A spatial diagram of entity interactions enclosed in ovals: (a) and (b) represent *trigger events*; (c) is the *final event*. Each has a text description, is denoted by ϕ_1, ϕ_2, and ϕ_3, and has a *soft truth* value. The sequence conveys a storyline in which as police observe protesters, and protesters push against the crowd, a riot ensues.

we could specifically state that a "bombing in *Boston*" forecasts "law enforcement in *Cambridge*" with confidence x. To avoid specific scenarios unlikely to repeat (such as the *Boston Marathon Bombings*), we can better generalize that assertion such that $action_1$ in location A forecasts $action_2$ in location B, when dist(A,B)$\leq$ distance d and their $SFI\leq$ threshold t.

Generating these heuristics allows us to extend our approach as models that can be applied to other datasets for forecasting purposes. In the experiments section, we demonstrate how storylines observed in certain locations forecasts other disparate events. These experiments use real datasets related to social unrest in Mexico.

6.4.3 Forecasting Violent Events with Spatio-logical Inference

As explained earlier, violent events can be viewed as the end result of larger processes composed of one or more *trigger events*. In the *Poll Tax Riots*, for example, we identified some of those *trigger events*, two of which were that activists organized protests and police closed some streets. Intuitively, each of these *trigger events* contribute a certain amount of momentum to the riots, with some weighing in more heavily than others. Our goal then is to make use of these weights, which we will call "*soft truths*", such that, when put together, the final violent event can be deemed probable or not. A *soft truth* is simply a *numerical belief* in the range [0,1] that two entities will interact in a particular way. Thus, one person may have seen police observing protesters with a *soft truth* of 0.75, while another person is not sure the police was involved, lowering the *soft truth* to 0.25. The combination of event sequences and *soft truths* allows us to generate rules and determine how well they lead to the violent event (i.e., their *distance to satisfaction*), which we explain below.

Rule Inference: Informally, our problem can be expressed as follows: given a storyline composed of several interacting entities, we seek a method to combine the individual *soft truths* of each interaction and make a decision of whether the consolidated interactions are compatible with the violent event or not, i.e., if they can generate the violent event. Consider Fig. 6.8 which depicts different sets of entities (*police*, *protesters*, *crowd*) interacting among themselves in the streets. There are three interactions, denoted ϕ_1, ϕ_2, and ϕ_3, each described in text with an associated *soft truth* value. The *soft truths* can be obtained from various sources: historical frequencies, input of domain experts, and random sampling, among others. We wish to find an algorithmic way to an-

swer the following question: is the combination of "*police observe protesters*" (ϕ_1) and "*protesters push against crowd*" (ϕ_2) enough for the crowd to "*cause a riot*" (ϕ_3)? Formally, this problem can be modeled in *First Order Logic* with the following statement:

$$\textbf{observe(police,protesters)} \wedge \textbf{push(protesters,crowd)} \implies \textbf{cause(crowd,riot)} \qquad \text{(Rule } r_1\text{)}$$

The above statement establishes a logical rule (r_1) that relates two *trigger events* via an "and" relationship ($\wedge$) to the *final event*, which is the riot. All of these events are in the format ***predicate*(entity$_x$,entity$_z$)**. It should read that *entity*$_x$ performs the *predicate* on *entity*$_z$, meaning that when police observe protesters and protesters push against the crowd, it implies that a riot will break out. This type of statement represents hard logic, *i.e.*, it determines whether developments will or will not happen, such as in a binary fashion. In terms of violent events, hard logic in many instances is not applicable because one can seldom state with certainty that a riot will or will not occur. For this reason, instead of hard logic, a more appropriate direction is to relax the binary restriction, and permit interactions to have a *soft truth* in a continuous fashion. Relaxing these restriction allows us to rewrite Rule r_1 as in the two examples below:

$$0.25\text{: } \textbf{observe(police,protesters)}(0.8) \wedge \textbf{push(protesters,crowd)}(0.4) \implies \textbf{cause(crowd,riot)} \\ (0.7) \quad \text{(Rule } r_2\text{)}$$

$$0.44\text{: } \textbf{observe(police,protesters)}(0.9) \wedge \textbf{push(protesters,crowd)}(0.3) \implies \textbf{cause(crowd,riot)} \\ (0.1) \quad \text{(Rule } r_3\text{)}$$

Generalizing them, we have:

$$\text{RW: } \phi_1(\mathbf{e}_a,\mathbf{e}_b)(w_1) \wedge ... \wedge \phi_n(\mathbf{e}_u,\mathbf{e}_v)\ (w_n) \implies \phi_{n+1}(\mathbf{e}_w,\mathbf{e}_z)\ (w_{n+1})$$

where RW is the rule weight, ϕ_i is either a *trigger event* or the *final event*, e_i represents an entity (or set of) and w_i is a *soft truth* value. Note that *trigger events* always appear in the antecedent of the rule (*i.e.*, before the $\implies$ sign), and the *final event* always appear in the consequent of the rule (*i.e.*, after the $\implies$ sign). Subsection 6.4.3 shows a method on how to select *trigger events* and *final events* in order to generate rules. In Rules r_2 and r_3 respectively, the *trigger events* have *soft truths* (0.8, 0.4, 0.9, 0.3) and the *final events* have *soft truths* (0.7, 0.1). The rules themselves have weights 0.25 and 0.44. In practice, the rules put in formal notation statements about what "people think" or "may have seen" or "has happened" given uncertainty. There could be different rules that also lead to the same riot, such as:

$$0.65\text{: } \textbf{seen\_with(weapons,protesters)}(0.8) \wedge \textbf{push(protesters,crowd)}(0.4) \implies \\ \textbf{cause(crowd,riot)}\ (0.7) \qquad \text{(Rule } r_4\text{)}$$

Given its higher rule weight, Rule r_4 is preferable to r_2 and r_3 (possibly because it involves weapons!). In a real application, thousands of such rules can be generated, which requires a numerical method to determine how good each rule actually is. In practice, we must find out whether the *trigger events* satisfy the riot, and if not, their *distance from satisfaction*. What we have described so far is derived from *Probabilistic Soft Logic* (PSL) [63]. PSL allows us to find if a rule's *trigger events* satisfy the *final event*, in which case we can then state that the rule forecasts the *final event*.

Given a set of *trigger events* ϕ=$\{\phi_1, ..., \phi_n\}$, the assignment of $\phi_i \rightarrow [0,1]^n$ represents the allocation of a *soft truth* value to an interaction between two entities. This allocation is called an *interpretation* $I(\phi_i)$. PSL uses the *Lukasiewicz t-norm* and *co-norm* to relax the traditional logical conjunction ($\wedge$) and disjunction ($\vee$) into continuous values as follows:

$$I = \begin{cases} \phi_1 \ \widetilde{\wedge} \ \phi_2 = max\{0, I(\phi_1) + I(\phi_2) - 1\} \\ \phi_1 \ \widetilde{\vee} \ \phi_2 = min\{I(\phi_1) + I(\phi_2), 1\} \\ \widetilde{\neg} = 1 - I(l_1) \end{cases} \tag{6.6}$$

The ˜ symbol is applied to denote the relaxed version of the normal logical operators, which allows us to assert the following:

Definition 14. *Given a rule r, composed of a set of trigger events* Φ=$\{\phi_1,...,\phi_n\}$ *and a final event* ϕ_{final} *where each* ϕ_i *and* ϕ_{final} *have an interpretation in [0,1], r is satisfied if and only if* $I(\phi_1,...,\phi_n) \leq I(\phi_{final})$.

Definition 14 states that the interaction established by the entities in the *final event* (ϕ_{final}) must have at least the same *soft truths* as the interactions of its constituent *trigger events* (ϕ_1,...,ϕ_n). The rule's distance to satisfaction for interpretation I is given by:

$$d_r(I) = max\{0, I(\phi_1, ..., \phi_n) - I(\phi_{final})\} \tag{6.7}$$

As an example, take Rule r_2, for which we wish to compute its distance to satisfaction $d_r(I)$. I(ϕ_1,ϕ_2)=max{0, 0.8+0.4-1}=0.2. Since 0.2≤0.7, we say that the rule is satisfied and $d_r(I)$=0. This contrasts with Rule r_3. where I(ϕ_1,ϕ_2)=max{0, 0.9+0.3-1}=0.2, and $d_r(I)$ = max{0,0.2-0.1} = 0.1. Rule 3 is more distant to satisfaction than Rule 2.

Interpretations can be challenging to deal with because different people have different opinions and different perceptions of facts. From an algorithmic perspective, however, all interpretations are equally valid until some are shown to be more feasible than others. For that purpose, we can calculate a distribution over all interpretations and identify the most probable ones. Given a set of Rules R = $\{r_1,...,r_n\}$, each composed of one or more *trigger events* and one *final event* in ϕ=$\{\phi_1,..,\phi_n\}$, the probability density function over all interpretations of the rules in R is given by:

$$f(I) = \frac{1}{Z} e^{[-\sum_{r \in R} \ WR \ (d_r(I))^p]} \tag{6.8}$$

where *WR* is the rule's weight, Z is a normalization constant so that interpretations sum up to 1, and p is a *loss function* that affects the rule's distance from satisfaction, in {1,2}. When p=1,

interpretations that completely satisfy one rule are preferred over others that contribute a positive distance from satisfaction. When p=2, distance from satisfaction is squared, which favors all rules to some degree. The value of Z, which is derived from *Markov Random Fields*, can be obtained from:

$$Z = \sum_{I=1}^{n} e^{[-\sum_{r \in R} \ WR \ (d_r(I))^p]} \tag{6.9}$$

Deriving the most probable interpretation is mathematically equivalent to maximizing $f(I)$. Knowing this distribution allows one to pick the maximal interpretations, which can be subsequently used for many purposes, such as ranking, filtering, or even classification of violent events. In the experiments section, we apply them in the context of stortylines to forecast probable violent events.

Generation of Candidate Rules: The discussion in Subsection 6.4.3 explains how to find the "goodness" of a rule for comparison purposes. Obviously, it requires a set of rules as input, and therefore, rules must be available as a pre-condition. Rule generation is an open research field, ranging from pattern mining [133, 23] to distributed processing [77, 52]. We do not endorse any specific methods, but would rather show an effective spatial approach for that purpose. This approach injects a spatial distance component into the rules, and thus the name *spatio-logical inference*. Fast forward to Table 6.6 and the discussion in Subsection 6.5.3 for a brief visual example.

Algorithm 6: Candidate Rule Generation

inputs: set of $STORYLINES = \{s_1,...,s_n\}$ where each s_i is composed of events $\phi_1, ..., \phi_m$ tagged by locations and timestamps in an area of study, number of desired rules n, size of rule s, distance d, event-pair *Probability-Matrix*
output: set of weight-based rules $RULES$

Initialize

1: $|Rules| = 0$; $\phi_{final} \leftarrow \phi_k$; // select one event in the dataset to be the final event

Pre-processing

2: **while** *STORYLINES exist* **do**
3: **foreach** *pair* $(\phi_i,\phi_j) \in \{s_i\}$ *where* $\phi_i \neq \phi_j$ **do**
4: *Distance-Matrix* ← store(normalizedDistance(ϕ_i,ϕ_j)) ; // calculate the distance between each pair of entities.
5: **end**
6: **end**

Main Stage

7: **while** $|Rules| \leq n$ **do**
8: List{Trigger-Events} = query(Distance-Matrix,ϕ_{final},s,d) ; // perform a query for the s closest events within distance d of the final event.
9: rule ← concatenate(List{Trigger-Events},"∧",ϕ_{final}) ; // combine all trigger events to the final event with an "and" relationship.
10: **foreach** $(\phi_i,\phi_j) \in$ *rule,* $\phi_i \neq \phi_j$, **do**
11: **set** soft-truth(ϕ_i,ϕ_j) = Probability-Matrix[(ϕ_i,ϕ_j)] ; // set the *soft truth* for each interaction in the rule by looking up the probability of its composing events in the probability matrix.
12: **end**
13: **set** $rule_{RW} = \frac{1}{avgDistEvents(rule,Distance-Matrix)}$; // set the rule's weight as the inverse of the average normalized distance among all its composing events
14: $RULES$ ← rule ; // store the formed rule.
15: increment d ; // increase the search distance and perform another query.
16: **end**
17: **output** $RULES$;

More formally, this process obeys the steps of Alg. 6, explained below.

The algorithm takes as input a set of storylines composed of many events, where each event is associated to a location, such as latitude and longitude. The user must also input the following items: the number of desired rules to be generated (n), a matrix of probabilities where each cell contains the likelihood of observing the corresponding events (event pair *Probability Matrix*), and the desired size s of each rule. Rule size is defined as the number of *trigger events* that composes the rule, *i.e.*, the number of events concatenated by the $\wedge$ relationship. In the previous example, the size of Rule 4, for instance, is 2. The algorithm first initializes two items: *RULES*, a data structure to hold the final rules, as empty; and the user-selected *final event* ϕ_k (line 1).

In the pre-processing stage, we wish to compute the distance between all events in the area of study, shown in line 3, to be used later. For better efficiency, we suggest that the events be clustered as explained in Section 6.3.3, and that only a limited number of clusters be used. The results are stored in a *Distance-Matrix* (line 4). Rule generation is accomplished in the main stage. First, using the *Distance-Matrix*, a query finds a number s of events (*i.e.*, a number that matches the rule size) within a user-specified spatial distance d of the *final event*. The results are stored in List{Trigger-Events} (line 8). The rule is then formed by concatenating the found *trigger events* in the list to the *final event* ϕ_{final} via the "and " ($\wedge$) operator (line 9). What remains to be done is to set the *soft truths* for each event in the rule. This is represented in lines 10 and 11 by doing a lookup in the probability matrix already provided. The overall rule weight is obtained by averaging the distances of all events for that rule, which can be obtained from the Distance-Matrix (line 13). The formed rule is then stored in the output data structure *RULES* (line 14) and the distance is incremented for a new search for more *trigger events* (line 15). The process continues until the desired number of rules has been reached, at which point the *RULES* are output in line 17.

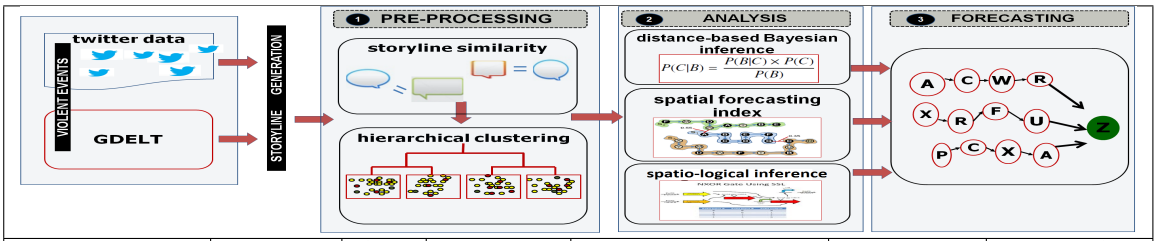
6.5 Empirical Evaluation and Technical Discussion

One of our initial claims was that spatio-temporal storytelling can be gainfully applied to everyday analytical tasks. To follow through with that statement, we select forecasting to showcase the applicability of our proposed methods. As mentioned previously, the goal is not to find most accurate forecasting strategy, but rather to demonstrate how spatio-temporal storytelling can be used in this domain. Towards that goal, we investigate how the four methods described previously (*Bayesian inference*, *distance-based Bayesian inference*, *spatial forecasting index*, and *spatio-logical inference*) can be employed in the forecasting of real-world developments. We begin by listing the general experiment setup and follow with three subsections detailing variations in measurements and different analysis points for each.

6.5.1 Experiment Setup

The experiments follow the specifications of Table 6.1 and the steps in the associated image. Initially, *Twitter* and *GDELT* data related to violent events are ingested and used in the generation

Table 6.1: Methodology and data specification of the experiments. The image shows the three-step forecasting process using spatio-temporal storytelling from *Twitter* and *GDELT* data: (1) a numerical similarity determines distance among storylines, which are hierarchically clustered; (2) clustered storylines are used as input to three methods: *distance-based Bayesian Inference*, *spatial forecasting index*, and *spatio-logical inference*; (3) the three methods output a numerical forecast of the sequences most likely to generate a final violent event.



Comparative Methods	Number of Records	Source	Years	Nature of Violent Events	Measure	Validation Set
SFI	100,000	*Twitter*	2011, 2012, 2013	education reform in Mexico	recall-1	*GSR*
SFI	119,758	*GDELT*	2011	war (Afghanistan and Middle East)	recall-2	*GDELT*
Li	50,000	*GDELT*	2011	war (Afghanistan and Middle East)	precision-1/recall-3	*GDELT*
Bayes, *dbB*, *SFI*, *Li*	50,000	*GDELT*	2011	war (Afghanistan and Middle East)	precision-1/recall-3	*GDELT*
Bayes=traditional *Bayesian Inference*, *dbB=distance-based Bayesian Inference*, *SFI=spatial forecasting index*, *Li=spatio-logical inference*.						

of storylines. Again, the storyline generation process follows the approach in [116], and is briefly explained in Subsection 6.3.1. As part of pre-processing, there are two tasks: establish numerical similarity among storylines and perform hierarchical clustering so that storylines can be more effectively handled. In the analysis stage, the clusters are fed as input to each of the comparative methods. The comparative methods, in turn, output a numerical forecast of events that are probable according to our datasets.

Data specification: Two data sources are utilized: tweets spanning the years of 2011, 2012, and 2013; and *GDELT* data from 2011. We perform several experiments with a varying number of records used in each, as shown in Table 6.1. The data contain a high variation of content: violent events reported in tweets and *GDELT* interactions, events of a non-violent nature, and a large number of other records not directly associated to any particular event, but with entities linked to other records that have violent events. We target events of two types: education-related protests in *Mexico* and wars in *Asia*.

Comparative methods: Evaluation is done on the four methods proposed previously (*Bayesian Inference* (*Bayes*), *distance-based Bayesian inference*(*dbB*), *spatial forecasting index* (*SFI*), *spatio-logical inference* (*Li*), which we call the 'comparative methods') for which two directions are taken: first, we investigate and discuss some of the comparative methods separately, and second, compare all of them together. The following approach is taken: we apply a subset of the data to our comparative methods to see what they are able to forecast, and then validate those findings in a different subset of the data not used previously. This is somewhat akin to a train-and-test approach. The number of records for each subset is specified in the experiment sections where they are discussed. The first examined method, *SFI*, uses both *Twitter* and *GDELT* data employing different record sets as shown on the table. We also solely investigate *Li* using 50,000 *GDELT* records. For the part of the experiments that compares all methods, we utilize 50,000 *GDELT* records that are different from the previous ones used on *Li*. We do not assume any specific data distribution, but we do select areas of study where violent events are known to be of a high enough frequency such that forecasting is actually plausible.

Table 6.2: Explanation of performance measures.

Measure	Meaning
recall-1 = $\frac{identified-as-relevant}{(identified-as-relevant)+(relevant-but-not-identified)}$	fraction of events correctly identified as relevant over the set of all relevant events.
recall-2 = $\frac{\lvert SFI(unobserved\ events)\geq\lvert x\rvert}{\lvert unobserved\ events\rvert}$	fraction of the *unobserved events* that have an *SFI* value of x or more.
recall-3 = $\frac{similar\ identified}{similar\ identified+similar\ missed}$	fraction of similar events that were identified over the total number of similar events.
precision-1 = $\frac{similar\ identified}{all\ retrieved}$	fraction of similar events that were identified over all retrieved records.
Definitions	
Successful forecast: An initial storyline successfully forecasts a target storyline (i.e., forecast is deemed relevant) if the initial storyline is linked to the target storyline through at least one common entity and the target storyline has an event identified in *GSR* or in *GDELT*.	
Relevant event: An event is deemed relevant if the storyline it belongs to makes a successful forecast of a target storyline.	
Similar events: Two events are deemed similar if they both belong in the same ontological branch (see Subsection 6.3.1) and are located within *distance to satisfaction* $\leq t$ of one another, where t is a user-defined threshold.	
Unobserved event: An event is considered unobserved if it is not included in previous analysis, such as the calculation of *SFI* values.	

Performance Measures: The evaluation's goal is to provide a variation of discussions, and thus different directions are taken. In some of the experiments, we are only interested in visualizing successful forecasts from an intuitive perspective. For others, we select **recall** as our performance measurements, leaving out **precision** for simplicity. Those are for the evaluation of individual methods. Yet, when all comparative methods are considered, we use both **precision** and **recall**. In terms of storytelling, it must be noted that both precision and recall should be interpreted carefully, as different viewpoints may arise based on what one would consider a "true" forecast as opposed to a "missed" forecast. Likewise, storylines do not possess standard definitions for what should be considered "relevant" or "similar". For this reason, we define in Table 6.2 our usage of precision, recall, and what we consider a "successful forecast", a "relevant event", "similar events", and "unobserved events". As much as possible, our goal is to reflect the definitions of precision and recall according to traditional *Information Retrieval*, and provide a clear picture of what is being measured.

6.5.2 Forecasting using *SFI*

Unlike traditional probability, in which a score of 1.0 indicates full certainty, the *spatial forecasting index* (*SFI*) is not constrained by an upper bound. As a result, one *SFI* value on its own has little meaning. To be useful, it must be compared to or contrasted with other *SFI* values. In a perfect world, higher *SFI* indices always indicate that a violent event will occur with higher likelihood than lower *SFI* indices. We look for the highest *SFI* values that we can calculate, and investigate if those high values translate to correct forecasts.

For this part of the experiments, the dataset is composed of approximately 100,000 tweets related to *civil unrest*[1] in *Mexico* for parts of 2011, 2012, and 2013. Using these tweets, storylines are generated based on the approach described in [116]. We target events related to *education reform*, which has provoked social strife in *Mexico*, and are documented as part of the *Gold Standard*

[1] civil unrest denotes an event of social impact, such as a strike or a protest.

Table 6.3: Demonstration of the *spatial forecasting index* in point-to-point mode between an initial storyline and 10 target storylines. The initial storyline is displayed across the top row. The GSR event represents a development reported in the media that reflects the target storyline.

Initial Storyline S_i: **STRIKE** affect **TEACHERS** demand **SALARY** higher **FUNDS**.			**Location:** ***Mexico City***		
	Target Storyline	**Location**	**Distance to Mexico City (km)**	**SFI**	**GSR Event**
S_1	**EDUCATION** fighting **SNTE** paying **SALARY** lower **FUNDS**.	Mexico City	0	3.60	SNTE Protesters block Eje Central; demand pension pay.
S_2	**SNTE** march **TEACHERS** participate **PROTEST**.	San Pedro Atlixco	28	1.52	SNTE teachers march in Atlixco.
S_3	**EDUCATION** march **TEACHER** lower **BUDGET**.	Tlaxcala	113	1.07	Teachers march against labor reform in Tlaxcala.
S_4	**ROAD** blocked **PROTEST** include **TEACHERS** ask **FUNDS**	Zitacuaro	129	1.02	Teachers block Morelia-Toluca in Zitacuaro.
S_5	**FIGHT** breaks **CITY** drain **FUNDS**.	Pachuca	87	1.01	Several incidents reported during SNTE's march.
S_6	**TEACHERS** lose **FUNDS** remove **BUDGET** impact **EDUCATION**.	Veracruz	313	0.76	SNTE teachers walk in Veracruz against education reform.
S_7	**EDUCATION** halt **UNIVERSITY** remove **STUDENT**.	Oaxaca	365	0.56	Stop at Oaxaca University affect more than 20 thousand students.
S_8	**TEACHER** protest **EDUCATION** lower **FUNDS**.	Aguascalientes	425	0.50	SNTE professors at Aguascalientes will march against education reform.
S_9	**FIGHT** break **STUDENT** distribute **FUNDS** sending **MORELIA**.	Michoacan	439	0.49	Teachers protest in Michoacan; demand Christmas pay.
S_{10}	**TEACHERS** march **CITY** protest **EDUCATION**.	Acapulco	297	0.48	In Acapulco, SNTE teachers from San Marcos will march.

Report (GSR) from the *Intelligence Advanced Research Projects Activity* (IARPA) [54], which serves as our ground truth.

We perform two sets of experiments: **(1) point-to-point mode**: investigate pairs of initial and target storylines, and for each pair, calculate their *SFI* values. For the top-k pairs of highest *SFI* values, find the corresponding events in GSR. Then determine if a forecast was successful or not: if the target storyline is linked to the GSR event through at least one entity and have the same distance (or less) to the initial storyline, then we claim that the forecast was successful based on that *SFI* value. We can then verify what *SFI* values corresponds to a successful forecast and the cutoff where the value is too low for a successful forecast; and **(2) point-to-region mode**: start from a set of initial storylines and calculate the *SFI* values to storylines of nearby regions (in our case, countries). Then compare the *SFI* values between the different regions, find matching events for the different regions in GSR, and justify which ones were forecast or not as before. For each task, four steps are involved: **(a)** select an initial storyline. **(b)** calculate the SFI values between the initial storyline and the other storylines. **(c)** select a number of top-k *SFI* values and verify if those locations were the place of a violent event that is documented in the GSR list. In other words, we seek the regions whose *SFI* values translate to good **recall-1**$=\frac{identified-as-relevant}{identified-as-relevant+relevant-but-not-identified}$, which we define as the fraction of events correctly identified over the set of events that should have been identified as relevant, but were not.

Table 6.3 illustrates on the top row an initial storyline, which we call S_i. This storyline, which was observed in *Mexico City* in January 2013, reports a teachers' strike for better financial conditions. Each row of the table shows a target storyline (S_1 through S_{10}) generated from tweets, the target storyline's location, its distance to the location of the initial storyline (*Mexico City*), the *SFI* value between the initial storyline and the target storyline, and a GSR event that confirms the veracity of the target storyline.

SFI Forecasting in Point-to-Point Mode: In this subsection, we are interested in investigating forecasting based on the locations of two specific storylines at a time, thus the "point-to-point" designation. Table 6.3 is sorted in decreasing *SFI* values. Immediately, it can be seen that the

lowest *SFI* value that we were able to do a successful forecast with is 0.48 (the last row in the table). Since S_1 has the highest *SFI* value, the first conclusion is that S_i forecasts the target storyline S_1 better than it forecasts any of the other nine target storylines. In other words, a **STRIKE** by the **TEACHERS** for better **SALARY** and **FUNDS** is a strong indicator of **EDUCATION**-related fighting by the **SNTE** (workers' union) for better **SALARY** and **FUNDS**, which is documented in the corresponding GSR event. Note that both S_i and S_1 have the same location (*Mexico City*), with zero distance of each other, which boosts their *SFI* value according to Eq. 6.4. They also share most entities, shown in uppercase letters.

Discussion: At a distance of 28 km, S_2 is only somewhat farther from *Mexico City*, but has a much lower *SFI* score (1.52) than S_1. This is because the longer distance between *San Pedro Atlixco* and *Mexico City*, added to the fact that S_i and S_2 only share two entities (TEACHERS-TEACHERS and PROTEST-STRIKE), drive their *SFI* lower. One notable item is S_{10}, whose storyline has the lowest *SFI* value of all (0.48), even though its distance to S_i (297 km) is much shorter than S_6, S_7, S_8, and S_9. It indicates that location is not the only determining factor in our forecasting strategy, though an important one. Looking at the table, it is generally true that longer distances determine lower *SFI* values, which should be expected. However, this assumption breaks in S_4 and S_5, which seem to hold a contradiction. The former is located farther away from S_i than the latter, but has a higher *SFI* value. The difference, again, is due to the number of shared entities with S_i, which is higher for the former than for the latter.

Based solely on this dataset, the premise is that, as a violent event, the **STRIKE** in *Mexico City* described in S_i is more likely to be followed by fighting by the **SNTE** also in *Mexico City* (S_1) than by a march by the **SNTE** in *San Pedro Atlixco* (S_2). We could go further and state that a **PROTEST** by **TEACHERS** in *San Pedro Atlixco* (S_2) is more probable than a **TEACHER**'s march against lower **BUDGET** in *Tlaxcala* (S_3). Such observations can be generalized into a forecasting model of how organizations mobilize people in social settings, which can be further applied in tasks such as classification or rule association mining.

Note that the above statements do not come solely from the comparison of a few storylines. Rather, it compares storylines that represent thousands of entities involved in the same violent events, which were reported in tweets and in the media, and compressed into short storylines. We claim success because all of the target storylines are highly reflective of a real GSR-documented event, which is shown in the last column of the table.

SFI Forecasting in Point-to-Region Mode: In the previous discussion, point-to-point mode investigates specific pairs of storylines, allowing their comparisons two at a time. We now switch to point-to-region mode, in which the objective is to investigate the *SFI* values from an initial storyline to all other storylines contained in a specific region. Thus, given the same initial storyline as in the previous example, we would like to know if the **STRIKE** from the **TEACHERS** for better **SALARY** and **FUNDS** in *Mexico City* propagates to other regions as similar events, or even cause different events to happen. The higher the *SFI* value for a region, the higher our belief that storylines in that region will reoccur, thus our forecast.

Table 6.4: Demonstration of the *spatial forecasting index* in point-to-region mode between an initial storyline based out of Mexico City and 10 target locations. The initial storyline (S_i) and its location (L_i) are displayed across the top row. For each target location, the table shows a related GSR Event.

Initial Storyline S_i:**STRIKE** affect **TEACHERS** demand **SALARY** higher **FUNDS**.					**Location L_i: Mexico City**
	Target Location	**Number of storylines in target location**	**Avg. Number of Shared Entities**	**Avg. SFI**	**GSR Event**
L_1	Mexico City	545	1.4	2.71	With protest, SNTE initiates informative campaign about education reform.
L_2	Pachuca	275	2.4	2.31	SME and CNTE protest in front of the Government.
L_3	San Pedro Atlixco	601	2.5	1.79	Protest takes place in front of Sagarpa's building.
L_4	Tlaxcala	325	2.0	1.26	CNTE teachers protest for eight hours, Metrobus service altered.
L_5	Zitacuaro	291	1.3	0.98	Teachers meet in front of the nation's Supreme Court.
L_6	Acapulco	255	1.2	0.87	Strike to continue at Autonoma University.
L_7	Oaxaca	184	1.2	0.75	DF Teachers will stop city center on Monday.
L_8	Michoacan	98	1.2	0.69	Teachers maintain pay dispute despite police confrontation.
L_9	Veracruz	402	1.5	0.54	Strike breaks out at Conalep plant in DF.
L_{10}	Aguascalientes	127	1.8	0.44	CNTE marches from Zacatecas to San Lazaro, maintain ground.

Figure 6.9: Spatial propagation of *education reform* protests. Starting from Mexico City, similar events are observed around the country. The map shows 10 of approximately 1,000 affected locations.

Discussion: Some of the results are shown in Table 6.4. The first thing to notice is that L_1, in the vicinity of *Mexico City*, has 545 storylines that drive the highest average *SFI* value in the set (2.71). Noting that L_i and L_1 have the same location (*Mexico City*) and thus no distance between them, their high *SFI* value is not surprising. In practice, it would be similar to stating that violent events often spread to nearby areas, such as rioting along connected streets. A more interesting case is L_2, which contains a significantly smaller number of storylines (275), but not a much lower *SFI* value than L_1 (2.31). Two reasons explain this difference: first, *Pachuca* is not very far from *Mexico City* (87 km); second, Pachuca's storylines have a high average number of shared entities with S_1 (2.4). They help boost the *SFI* value calculated with Eq. 6.5. *Veracruz*, in L_9, has a high number of storylines related to *education reform*, but its long distance to *Mexico City* (313 km) and a low number of shared entities with S_1 (1.5) gives it a low *SFI* score (0.54), making it challenging to associate events in *Mexico City* to any of *Veracruz*'s events.

We must emphasize the importance of the spatial aspect of this study, showing that all items from L_1 to L_{10} are highly-dependent on location. In this dataset, many of the storylines have no location explicitly stated. However, their related tweets do contain at least one metadata location that matches the location of the *GSR event*, and a timestamp that closely pre-dates the report of the event. This is particularly interesting in the case of L_{10}, whose *GSR event* is shown in *Zacatecas*, but whose *target location* is shown in *Aguascalientes*, which are only 43 km apart. The prominence of these storylines in close proximity of one another is significant for a simple reason: it indicates

Table 6.5: Recall results based on 119,758 *GDELT events* in four different categories. Recall is defined as the percentage of the unobserved events that have an *SFI* score equal to or greater than the average *SFI* score.

GDELT event type	**Source Country**†	**Target Country (Storyline)**	**Observed Events / Unobserved Events**	**Avg. SFI**	**Recall**
THREATEN (political dissent, repression, military force, occupation, attack, mass violence)	AFG	Iran (s_1)	8,245 / 14,465	3.15	0.57
		Pakistan (s_2)	7,129 / 15,842	1.75	0.45
PROTEST (political dissent, rally, hunger strike, passage obstruction)	IRN	Afghanistan (s_3)	5,745 / 9,575	2.03	0.60
		Iraq (s_4)	6,054 / 9,924	2.43	0.61
		Pakistan (s_5)	5,347 / 7,638	1.90	0.70
		Turkey (s_6)	2,118 / 5,573	2.71	0.38
COERCE (seize property, impose sanctions, ban political parties, enact martial law, arrest)	IRQ	Iran (s_7)	10,218 / 12,615	3.21	0.81
		Kuwait (s_8)	3,151 / 5,626	0.60	0.56
		Syria (s_9)	7,211 / 11,093	2.74	0.65
		Turkey (s_{10})	1,616 / 3,298	1.12	0.49
ASSAULT (hijacks, torture, killings, suicide bombings)	PAK	Afghanistan (s_{11})	2,744 / 3,563	2.98	0.77
		India (s_{12})	5,091 / 20,364	3.43	0.25
		Iran (s_{13})	144 / 182	2.45	0.79

†Afghanistan, Iran, Iraq, Pakistan

s_1: TALIBAN capture MAZHAR-I-SHARIF occupy IRANIAN CONSULATE kill DIPLOMATS.	s_2: STUDENTS protest FORCES kill QASIM KHAN secure BORDER.
s_3: TEHRAN hosts REFUGEES clash POLICE threaten ECONOMY.	s_4: AIRCRAFT fire MISSILE hit STARK kill PERSONNEL.
s_5: AGENTS kills PAKISTANIS chasing GUARDS reported FISHING.	s_6: IRAN starts OIL supply TURKEY monitor BLAST.
s_7: U.S. warns IRAN fight ISRAEL destroy WEAPONS.	s_8: BA fly KUWAIT seize CITY hold PASSENGERS.
s_9: IRAQ accuse SYRIA plan BOMBING rock MINISTRY.	s_{10}: KADEK wins ELECTION combat PKK declares CEASE-FIRE.
s_{11}: NATO attack SALALA engage CHECKPOST wound SOLDIERS.	s_{12}: MUMBAI conspire PAKISTAN deprive EXTREMIST enter HOTEL.
s_{13}: OFFICIAL shot MAN ran BALUCHISTAN taken NARCOTICS.	

that our *SFI* model based on spatial distance and shared entities can uncover related violent events that could reoccur in nearby areas in the future. If an analyst is interested in at most three regions of interest, Table 6.4 allows us to speculate that from our initial storyline S_1, violent events with an *education reform* theme are more likely to take place in *Mexico City*, *Pachuca*, and *San Pedro Atlixco*, with decreasing order of confidence. The analyst may want to prioritize those regions.

While the *GSR* dataset catalogs civil unrest developments, we also experiment with *GDELT* [71], which is a more comprehensive database of events. It covers most regions of the world in more granular categories, many of which have a violent nature. One example of a *GDELT* event is an occurrence of *ethnic cleansing* on January 24, 2005, by Iraqi forces on individuals of Iranian origin. The event took place in latitude 31.0914 and longitude 46.0872, in the *Dhi Qar* province of *Iraq*. In this study, we use facts of this nature to generate storylines, calculate their *SFI* values to nearby regions, and then verify if *GDELT* matches other similar events for the regions of highest *SFI* values. If it does, we say that the event was forecast correctly given that *SFI* value. We use a subset of *GDELT* events, which we call *Observed Events*, to calculate *SFI* for a region, and then use a different set of *GDELT* events, which we call *Unobserved Events*, to calculate recall as explained further below.

Table 6.5 lists four *GDELT Event Types* documented for several countries. The first row, for instance, indicates 8,245 THREATEN-type events perpetrated by an actor [2] in *Afghanistan* (AFG) on an actor in *Iran*. Starting from an initial storyline (shown on the bottom of the table) that took place in the *Source Country*, we calculate the *SFI* values to each of the *Observed Events* in the *Target Country*. Thus for row 1, we use S_i to calculate the *SFI* values to all the 8,245 observed events in *Iran*, which yields an average *SFI* of 3.15. Recall is then the percentage of the *Unobserved Events* that have an *SFI* value of x=3.15 or more, which we generalize as **recall-2** = $\frac{|SFI(unobserved\ events) \geq x|}{|unobserved\ events|}$. The *SFI* values can also be compared, allowing us to state that violent events between AFG and *Iran* in row 1 is more probable than violent events between AFG and Pakistan

[2] An actor can be a political organization, the military, militias, terrorist organizations, and individuals, among others.

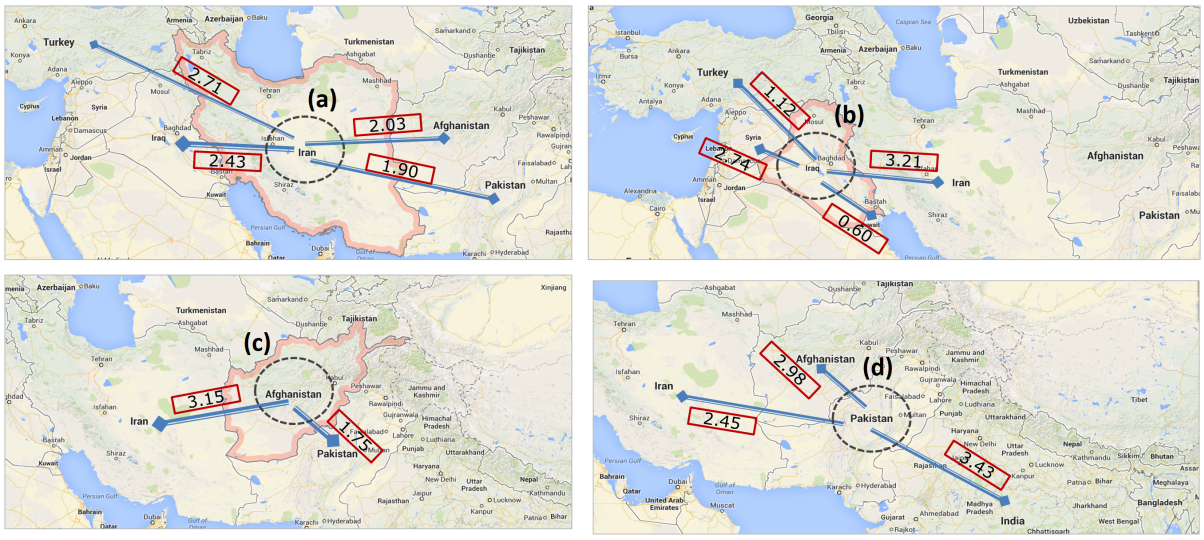


Figure 6.10: Spatial propagation of violent events in four Asian countries: (a) protests originating in *Iran*; (b) coercion in *IRAQ*; (c) threats in *Afghanistan*; (d) assaults in *Pakistan*. The boxed numbers represent the *SFI* scores between the countries. Higher *SFI* values indicate higher potential for a successful forecast.

in row 2, where the *Avg. SFI* is lower (1.75), for that category of events.

At first glance, Table 6.5 shows that the three rows of highest recall (S_7, S_{11}, and S_{13}) also have relatively high *SFI* values. This type of consistency is highly desirable as it may signal that violent events in areas of high *SFI* values have a high potential to be identified and thus forecast. This consistency, however, must be interpreted carefully as high *SFI* values do not necessarily imply high recall. This is the case with S_6, which are PROTEST-related events between *Iran* and *Turkey*. Its recall value is poor (0.38) because most of the 5,573 unobserved events are not in the PROTEST category. We would not be able to assert a successful forecast for those. A similar scenario can be seen for S_{12}, where the events between *Pakistan* and *India* are of various natures. One lesson to be learned here is that distribution of event types is an important factor. It is important to filter out storylines that are completely different from the domain in question.

In terms of forecasting, different observations can be made. Our first storyline (S_1) tells about a *TALIBAN* attack on a *CONSULATE* affecting *DIPLOMATS*. Since we are able to recall 57% of unobserved events that have similar entities, we assert a 57% chance that an event with those entities will reoccur in a nearby location, thus our forecast. Looking down Table 6.5, we can make other forecasts, such as a 65% chance of an *Iraq*-led attack on a *Syria* target, as examplified in S_9. Indeed, we can identify several of such events, as a *Taliban* attack on the *U.S. Consulate* in 2010, and a militia-led suicide bombing by an Iraqi national in Syria in 2011. Table 6.5 also indicates that the regions between *Pakistan* and *India* provides the best chances for a succesfull forecast in the category of *ASSAULTS*, since these two regions have the highest *SFI* values for that category (3.43). The same is true for *Iran* and *Turkey* for the category of PROTEST (2.71) and *Iraq* and *Iran* for COERCE (3.21). Fig. 6.10 depicts spatial propagation of events based on the four *Source Countries* of Table 6.5 using their corresponding storylines. Visually, higher *SFI* values indicate better chances of a successful forecast. Values can be compared always starting

from the same source country propagating to others, or constrasted across different sources and different destinations.

6.5.3 Forecasting with Spatio-logical Inference

In the previous section, we investigated forecasting from a spatial propagation perspective. In this section, we apply *spatio-logical inference* to transform storylines into weight-based rules, which we then use to do forecasting.

We begin with a set of 50,000 *GDELT* events of category type ASSAULT (broken down into three subcategories) that took place in *Afghanistan*. Out of those records, we use 30,000 to extract rules, find events of high probability of occurrence using *spatio-logical inference*, and use those to find the number of similar events that exist in the remaining 20,000. Our measures are: **recall-3**=$\frac{similar\ identified}{similar\ identified+similar\ missed}$ as the number of similar events that were identified over the total number of similar events among the 20,000; **precision-1**=$\frac{similar\ identified}{all\ retrieved}$ as the number of similar events that were identified over all retrieved records. By similar events, we denote events of the same ontological resolution (see Subsection 6.3.1) located within *distance to satisfaction* $\leq t$ of one another, where t is a threshold. In the experiments, we evaluate different distance thresholds and present the results.

To extract rules from our dataset, we use Alg. 6, for which we give a brief example. Consider the three *GDELT* event types shown in Table 6.6 and geolocated in the corresponding image, which is *Afghanistan*, our region of study. The frequency for each event type is shown in parenthesis. Because the two closest events are Ⓐ and Ⓑ, at a distance of 115 km, these two events make up the body of the rule. The remaining one, event Ⓒ, becomes the implication:

$$\textbf{carryout-vehicular-bombing}(\text{AFGMOS,AFGREB}) \wedge \textbf{use-as-human-shield}(\text{AFGREB,AFGCVL}) \implies \textbf{attempt-to-assassinate}(\text{AFGCVL,AFGMIL})$$

To add the *soft truths*, we look at Table 6.6 and see that the probability of event Ⓐ = $\frac{15}{45}$ = 0.33, Ⓑ = $\frac{5}{45}$ = 0.11, and Ⓒ = $\frac{25}{45}$ = 0.55. The overall weight of the rule is the average distance between the three events, normalized in the range [0,1], which can be calculated as 0.76, assuming a min distance of 0 km, and a max distance of 278 km. Thus the final rule looks like:

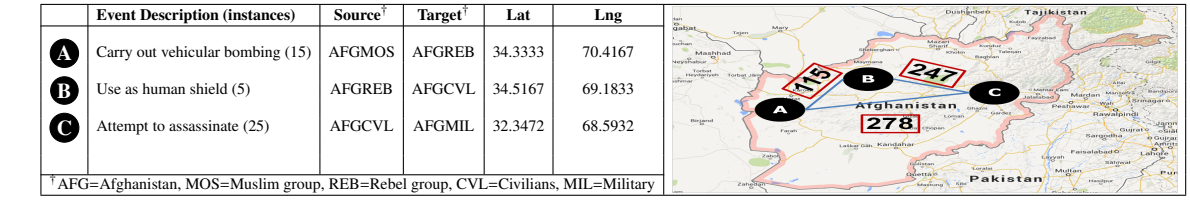
$$0.76: \overbrace{carryout\ vehicular\ bombing(AFGMOS, AFGREB)}^{0.33} \wedge \overbrace{use\ as\ human\ shield(AFGREB, AFGCVL)}^{0.11} \implies \overbrace{attempt\ to\ assassinate(AFGCVL, AFGMIL)}^{0.55}$$

We then use the above rule to find its *distance to satisfaction* as described in subsection 6.4.3. In the experiments we set the overall weight of every rule as 1.0 (every rule is equally important) and focus on the soft truths instead. Our forecasts are the rules with the least distance to satisfaction.

Table 6.6: Example of three *GDELT* events located in different areas of *Afghanistan* in 2011. The number in parenthesis is the total number of events of that kind reported in *Afghanistan* for that year. The image shows the distance in km between the different events.

	Event Description (instances)	Source†	Target†	Lat	Lng
A	Carry out vehicular bombing (15)	AFGMOS	AFGREB	34.3333	70.4167
B	Use as human shield (5)	AFGREB	AFGCVL	34.5167	69.1833
C	Attempt to assassinate (25)	AFGCVL	AFGMIL	32.3472	68.5932

†AFG=Afghanistan, MOS=Muslim group, REB=Rebel group, CVL=Civilians, MIL=Military



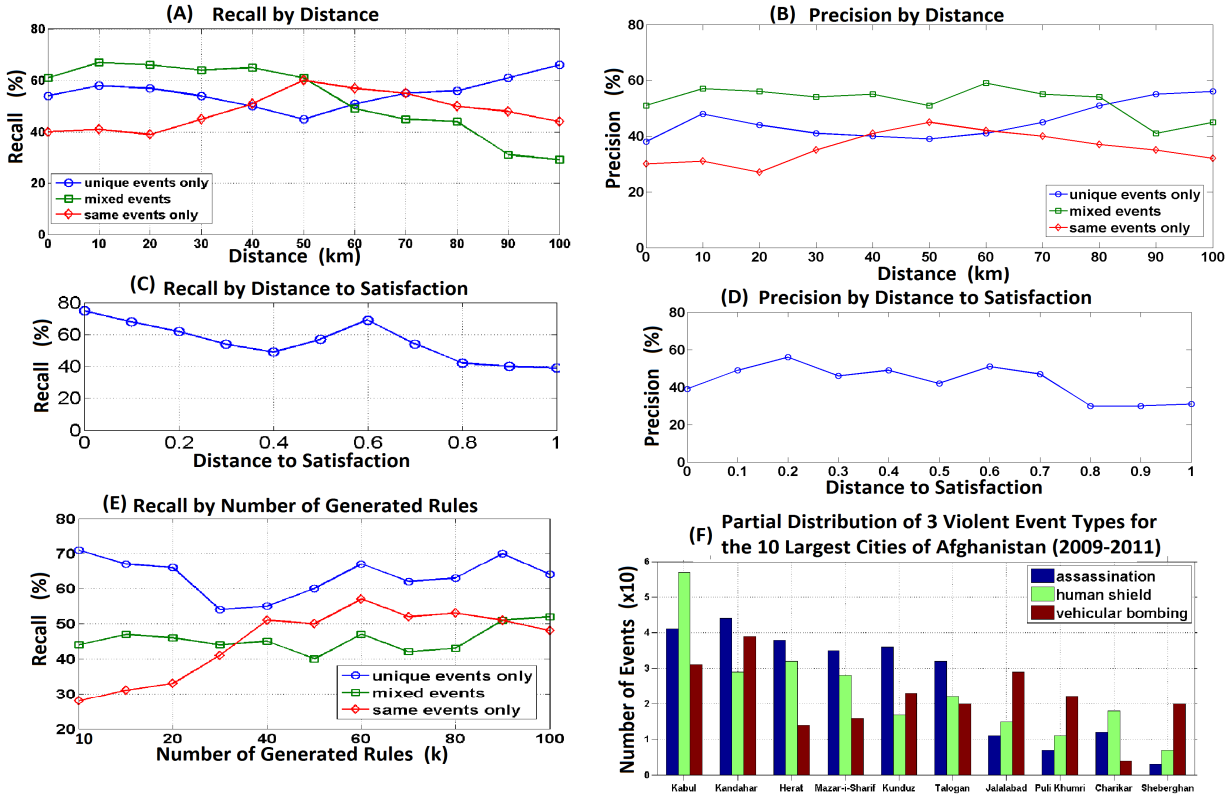


Figure 6.11: Results from *spatio-logical inference*. (A) Effect of distance between events on recall. (B) Effect of distance between events on precision. (C) Effect of the *distance to satisfaction* on recall. (D) Effect of the *distance to satisfaction* on precision. (E) Effect of the number of generated rules on recall. (F) Distribution of violent events in cities of Afghanistan.

Based on that, we use *precision* and *recall* as evaluation measures. It should be clear that the above example is a simple scenario with only three events. Given vast numbers of events, the number of rules can easily explode. Optimizations should be done, such as shortening distances or filtering out specific event types in order to alleviate computation costs. Our weights are based on frequencies and spatial distances, but it is possible that different approaches may be better suited for different domains of knowledge.

Discussion: In the context of violent events, a key consideration is whether relevant events can be forecast, knowing that relevance is a highly subjective matter. For measurement purposes, we define relevance in a comparative scale based on either *Euclidean* distance or *distance to satisfaction*: lower values are always more relevant than higher ones. Association among events can be investi-

gated in three configurations: (1) all events are the same, such as when instances of fights result in other fights; (2) all events are different, such as when a fight and police crackdown result in a riot; (3) otherwise, events are mixed. Assume that there exists a set of *trigger events* (ϕ_1 to ϕ_n) that lead to a *final violent event* (ϕ_{final}) with a *d Euclidean* distance or *distance to satisfaction*. Then one can assert a successful forecast for other unseen *final violent events* provided that the *trigger events* lead to the same *final violent events* with the same or lower distance d. In other words, comparing the association between two sets of events, if the events match (or partially match) on at least one *trigger event* and distance is just as low, then a forecast is made. If no events match or distance d is off, then the forecast is a miss.

Using the above ideas, we retrieve all events from our dataset that fit those conditions, and count how many we were able to forecast, and how many were missed. For simplicity, we limit the views to a range where the max distance between any two events is 100 km. Fig. 6.11 shows six plots with different measurements for discussion. High recall values indicate that previously-unseen events are being found without going over a distance limit. This is shown in Fig. 6.11(A), in which recall values range from 40% to 66%. In the range where distance between events lies between 0 to 50 km, recall remains fairly constant at around 62% for mixed events. It indicates that, for many of the generated rules, their constituent events lead to the same *final violent event* with a distance of 50 km or less. For events of the same type, recall trends upward up to 50 km, but only gets worse thereafter. More intriguing are events of unique type, in which recall is good with short distances (0 - 20 km) or long distances (80 - 100 km), but often worse between (21 - 79 km). The lesson learned from this example is the following: the *soft truth* values established in the rules seem to be appropriate for the initial part of the graph (shorter distances) and the late stages (longer distances), but may not be ideal for mid distances. Those values are candidates for adjustment. Proceeding to Fig. 6.11(C), the *distance to satisfaction* trends down most of the way with the exception of a spike at 0.6. The downward portion relates to the notion that fewer of the *final violent events* are being found, or when found, the *distance to satisfaction* is too high (*i.e.*, above the limit established by the rule that found it). The analyst may want to investigate the events associated with low recall to see if adjusting the *soft-truth* values affords better results. It is possible that the values are indeed correct, and that the low recall comes as a result of violent events in the unseen data not matching the ones in the observed data.

Similar trends as the above is also seen in Fig. 6.11(B), which shows precision by *Euclidean* distance. In general, one would expect high recall for short distances and vice-versa. Intuitively, government in Kabul experiences many bombings over time, but ones which are not necessarily related to other bombings in far-away cities, such as Charikar. However, our data indicate that, in many instances, longer distances between events display higher precision than shorter ones. This is the case in Fig. 6.11(B) where the highest precision for mixed events is approximately 60% with a distance of 60 km. For unique events, this fact is even more pronounced, since the highest precision (57%) lines up with the highest distance (100 km). This is indicative of a particular type of event that takes place in many locations (*e.g.*, protests against corruption taking place across multiple cities): the violent events match with similar conditions (*i.e.*, similar *trigger events*) even when the cities are far apart. Fig. 6.11(D) shows the effects of *distance to satisfaction* on precision.

Table 6.7: Examples of events that were forecast correctly or missed based on the generated rule shown across the top row. The final violent event is **destruction of property** as shown in the implication of the generated rule $G1$.

	Generated Rule	**Distance to Satisfaction: 0.25**
$G1$	engage(AFGGOV,RADMOS) ∧ demand-release(AFG,COP) ⟹ destroy-property(AFGREB,RADMOS)	

	Events Correctly Forecast by Rule $G1$
$F1$	halt-negotiation(AFGCOP,UAF) ∧ demand-release(AFG,COP) ⟹ confiscate-property(AFGGOV,RADMOS) (0.13)
$F2$	engage(AFGGOV,RADMOS) ∧ impose-embargo(AFGSPY,AFGCRM) ⟹ seize(AFGGOV,AFGINSTALUAF) (0.17)
$F3$	cooperate-militarily(AFGCOP,AFG) ∧ impose-curfew(AFGSPY,AFGCVL) ⟹ destroy-property(AFGGOV,RADMOS) (0.12)
$F4$	ban parties(AFGCOP,UAF) ∧ demand-material-coop(AFGGOVBUS,AFGCVL) ⟹ destroy-property(AFGREL,RADMOS) (0.24)

	Missed Forecasts	**Reason for Miss**
$M1$	engage(AFGGOV,RADMOS) ∧ reject(AFG,AFG) ⟹ mobilize-armed-forces(AFG,RADMOS) (0.20)	wrong *final violent event*
$M2$	halt-negotiation(AFGCOP,UAF) ∧ use-tactics-violent(AFG,COP) ⟹ destroy-property(AFGGOV,RADMOS) (0.20)	no match on *trigger events*
$M3$	expel(AFGMIL,AFGELI) ∧ rally-opposition(AFGGOV,AFGREF) ⟹ demand-release(AFGGOV,RADMOS) (0.38))	high *distance to satisfaction*
$M4$	engage(AFGEDU,AFGMIL) ∧ reduce-econ-aid(UAF,AFGREF) ⟹ destroy-property(AFGGOV,RADMOS) (0.31)	high *distance to satisfaction*
	AFG=*Afghanistan*, BUS=*business*, COP=*police force*, CRM=*criminal*, CVL=*civilian*, ELI=*elites*, GOV=*government*, MIL=*military*, MOS=*muslim*, RAD=*radical*, REB=*rebels*, REF=*refugee* , SPY=*spy*, UAF=*unidentified armed force*	

This trend does not deviate significantly from the *Euclidean* distance approach, even though high precision at times does come from lower distances. The fact that the two approaches have similar results is encouraging because it indicates that our reasoning is valid.

The above discussion points to the importance of relating event types, locations, distances, and frequencies in the discussion of violent events. We use these components to generate event-based rules. Fig. 6.11(E) summarizes recall in terms of the number of generated rules according to event type. This time, distance is disregarded, which has a different effect on the results. When distance is not considered, recall is consistently high when events have different types, but suffer considerably for mixed ones, with a higher variation for same event types. In practice, it seems to relay the message that "a forecast is safe when **a and b lead to c**, but not when **a and a lead to c** or **b and b lead to c**". The closest that the three lines come together is at approximately 33k generated rules, where recall ranges from 41% to 54%. This is a significant difference from the distance approach, which underscores the importance of spatial analysis. For illustrative purposes, Fig. 6.11(F) depicts the distribution of three events for the 10 largest cities in *Afghanistan*, which were used in this dataset. It shows, for instance, that (for this partial dataset) *vehicular bombings* are mostly frequent in *Kandahar*, *Kabul*, and *Jalalabad* (in this order), while *Kabul* itself sees most of the *human shield* events. While this graph is not the complete dataset used in the experiments, it gives the reader a sense of the spatial locations being investigated and the event types we were looking for.

Finally, we display some of the events our approach is able to forecast in Table 6.7. Starting from a sample generated rule ($G1$), whose *final violent event* relates to *destruction of property*, having a *distance to satisfaction* = 0.25, the table first shows a set of four rules that were correctly forecast ($F1$, $F2$, $F3$, $F4$). $F1$, for example, tells about some sort of "negotiation" that involves an action of "release", which eventually ended up as "confiscation of property". Without the benefit of external knowledge, we do not know the details of this case. However, we can affirm with confidence that this event is very close in concept to the original rule $G1$, which also has a "release" component, involves "destruction of property", and has lower *distance to satisfaction* than the original rule $G1$ (0.13 as opposed to 0.25). These events took place in 2010 in Afghanistan at a distance of 34 km from each other. The same is true for $F3$, which also deals with "destruction of property", though coming from totally separate *trigger events* related to "military cooperation" and a "curfew". $F2$ and $F4$ have slightly higher *distance to satisfaction*, albeit still below the limit of 0.25 established

by $G1$.

Further down the table, we show four other rules that were not considered valid forecasts based on our pre-established conditions. The first one, $M1$, does not have a similar *final violent event* to $G1$, and thus we do not have a basis to compare distances given that the rules share little in common (only one *trigger event*). $M2$ shares no *trigger events* at all with $G1$, and thus is not valid because our approach needs at least one element in common. $M3$ and $M4$ both are too distant in terms of *distance to satisfaction* from 0.25, and thus are rejected as well.

In *storytelling*, the high number of entities and events is always of concern. It is important, thus, to understand the number of rules that are generated and how they affect recall, which is shown in Fig. 6.11(E). The plot separates whether the events considered are of the same nature (*e.g.*, bombing followed by another bombing), unique natures (*e.g.*, bombing followed by an assassination attempt), or a mix of them. When event types are mixed, recall remains fairly constant despite the increase in the number of generated rules. It hints at the distribution of the data: events are well spread out throughout space. An analyst studying many event types concurrently may find this fact interesting. The situation is vastly different when the events are all the same or are all different. In this case, recall displays greater variation (28-57% and 54-71%, respectively). The graph also shows that fewer rules is not necessarily better than more rules (as one might expect). In fact, some of the best recall values can be seen exactly at the end of the graph when the number of generated rules hits 100k. The not-all-clear message here is that violent events are better explainable with different types of events, and not with the reoccurrence of the same event types.

6.5.4 Comparison of the Different Forecasting Strategies

In this subsection, we put in perspective the four forecasting strategies explained earlier. Our goal is not to find the best forecasting strategy, but rather to contrast them. One line of research complimentary to our work, but which often does not include spatial *storytelling*, is event detection, to which we point the reader for further reading [78, 136]. We frame our discussion in terms of *precision* and *recall*, as done before.

Table 6.8 lists a set of 10 event types, labeled $E1$ through $E10$, from our *GDELT* dataset that we target as *final violent events*. We use 50,000 records: 30,000 as input and 20,000 for validation. For each event type, the table shows precision and recall values, calculated as explained earlier, using the four technical approaches discussed in Section 6.4. The highest values are shown in bold type. For easier visualization, the data on the table is also shown as bar plots in Fig. 6.12. *Bayes* denotes traditional *Bayesian Inference*, where we consider combinations of events whose probability of occurrence is 10% or less (higher than 10% was less significant in our dataset). The other three methods (*dbB*, SFI, and *Li*) are considered according to the nature of their distances. For *Distance-based Bayes* (*dbB*), precision and recall values are provided in two groups: the first group encompasses all events whose normalized distances (*nd*) are less than 0.5 (on a range of [0,1]), and the second group is for *nd* greater than 0.5. Similarly, *spatial forecasting index* is considered for values initially less than 1.5, and later greater than 1.5, while *spatio-logical inference* breaks at a

Table 6.8: Comparison of precision and recall for four different approaches: traditional *Bayesian Inference* (Bayes), *distance-based Bayesian Inference*, *spatial forecasting index*, and *spatio-logical inference*. Except for Bayes, precision and recall is shown for each method before and after a midpoint distance of the dataset (normalized distance *nd*, *spatial forecasting index* sfi, and *distance to satisfaction* df). For each row, the highest values are shown in bold letters.

	Bayes		**distance-based Bayes (dbB)**				**spatial forecasting index (sfi)**				**spatio-logical inference (Li)**			
	*		$nd \leq$**0.5**		$nd >$**0.5**		$sfi \leq$**1.5**		$sfi >$**1.5**		$df \leq$**1.0**		$df >$**1.0**	
Event	**precision**	**recall**	**precision**	**recall**	**precision**	**recall**	**precision**	**recall**	**precision**	**recall**	**precision**	**recall**	**precision**	**recall**
E1-attempt to assassinate	0.37	0.64	0.35	**0.81**	0.44	0.68	0.51	0.58	0.44	0.76	**0.55**	0.77	0.36	0.65
E2-carry out vehicular bombing	0.42	0.76	0.61	0.75	0.63	0.58	0.57	0.72	0.24	0.51	**0.66**	**0.80**	0.33	0.64
E3-engage in violence	0.25	0.44	0.48	0.71	0.61	0.75	0.54	0.72	0.38	0.58	0.66	0.79	**0.70**	**0.85**
E4-conduct strike or boycott for rights	0.52	0.67	0.45	0.65	0.55	0.71	**0.78**	0.61	0.44	0.59	0.67	**0.79**	0.56	0.65
E5-destroy property	0.29	0.64	0.51	0.70	0.40	0.64	**0.61**	0.77	0.43	0.80	0.45	0.52	0.54	**0.81**
E6-impose blockade, restrict movement	0.54	0.70	0.55	0.61	0.49	0.66	0.60	0.69	0.56	**0.77**	**0.61**	0.70	0.56	0.45
E7-impose state of emergency or martial law	**0.61**	0.68	0.34	0.60	0.55	0.72	0.57	0.68	0.59	**0.77**	0.54	0.69	0.45	0.68
E8-impose restrictions on political freedoms	0.49	0.53	0.59	0.65	0.54	0.61	0.58	0.41	0.59	0.75	0.60	0.72	**0.73**	**0.81**
E9-use as human shield	0.40	0.59	0.54	0.66	0.53	**0.74**	**0.61**	0.71	0.57	0.70	0.50	0.63	0.52	0.59
E10-threaten to reduce or break relations	0.29	0.46	**0.69**	0.39	0.58	0.73	0.52	0.66	0.52	0.71	0.55	**0.78**	0.46	0.66
Precision and recall above refer to **precision-1** and **recall-3** of Table 6.2														

distance to satisfaction of 1.0. These values are selected because they are approximate midpoint distances between the events in our dataset, and thus a reasonable breakoff point for investigative purposes.

The way to interpret the table, exemplified for row 1, is as follows. Upon running *Bayesian Inference* for event $E1$ (*attempt to assassinate*) in the initial set of 30,000 events, the results indicated 5,101 combinations (not shown in table) of *trigger events* that led to $E1$ with a probability $\geq 10\%$. However, when validating against the remaining 20,000 records, those combinations only contained 985 out of 2662 events with a probability$\geq 10\%$ (and that shared at least one event with the generating combination), yielding a precision of 0.37. For recall, 985 combinations were found, but 1539 should have been identified, resulting in a recall of 0.64. For the other approaches, instead of a simple probability, the criteria are the normalized distances (*nd*), SFI values, and *distance to satisfaction* (df).

Discussion: The first noticeable point is the fairly low levels of precision for traditional *Bayes* for all event types, except for $E7$ (*imposing state of emergency or martial law*). Especially for $E3$ (*engage in violence*), very few combinations of events lead to that type of event, making it hard to identify. One exception of high precision is $E7$, which is highly frequent with similar event types. Overall, the reason for the low precision values is that traditional *Bayes* requires the same events in the same sequence for the probabilities to be high. For violent events, however, sequence can seldom be guaranteed, rendering Bayes less than ideal. The situation is more favorable in terms of recall, as relevant items are often retrieved with greater success.

For *distance-based Bayes*, precision is often higher than for traditional *Bayes*, but with mixed signals. It significantly improves for $E2$ and $E10$, but decreases for $E7$. The reason has to do with the distance, which impacts $E7$ negatively when $nd \leq 0.5$ (0.34), and not as much when $nd \geq 0.5$ (0.55). Verifying the dataset, it can be seen that events within low distance of $E7$ are not commonly observed, thus impacting precision. The best precision level is 0.69 for events that lead to $E10$ (*threaten to reduce or break relations*). Indeed, this item is very common in the dataset, possibly due to its subjectiveness, which can lead to a high number of interpretations. In general, recall is consistently high, not seeing much impact from either side of the *nd* threshold.

The *spatial forecasting index* demonstrates the highest precision of any of the approaches for events

$E4$, $E5$, and $E9$. Its positive aspect is consistency even when it is not the highest. It is never lower than 0.51 for $sfi \leq 1.5$. However, it seems to suffer for larger distances ($sfi \geq 1.5$) where it drops to only 0.24 on $E2$ (*carry out vehicular bombings*). SFI is highly sensitive to how many events are far apart versus nearby, and seems to favor the latter. The data distribution is certainly a factor here, and has to do with spatial colocation. For example, we see many instances of the same pairs of events that lead to $E7$ (*impose state of emergency or martial law*) with high values of SFI. This explains the 0.59 precision of $E7$ and $E8$. It is true that some of the SFI values are low ($E2$ and $E3$, for $sfi \geq 1.5$), but those can be explained by the low numbers of similar events in the validating dataset. Its recall values are good across all events with the exception of $E8$ (0.41), which have on average a very low SFI score of 0.19 (not shown on table).

Two observations can be made about *spatio-logical inference* (*Li*): first, precision shows good consistency for low *distances to satisfaction*, which is desirable in terms of forecasting. However, one should also expect low precision for high *distances to satisfaction*, which in general does not occur when $df > 1.0$. While high precision is normally a good thing, we would prefer df to oppose precision hand-in-hand (low to high, and high to low). This does not happen with $E3$ and $E8$ (for $df > 1.0$), where precision values are unexpectedly high (0.70 and 0.73, respectively). Indeed, these values come from many rules that are established by far-apart events of the same ontological category with high *soft truth* values, and thus their high precision. *Li* is very stable in terms of recall, and interestingly, especially when distances are long. While Table 6.8 only shows a limited number of results, our overall experience point to *Li* as having the best recall results.

Key Observations and Potential Issues: We evolved our discussion from storylines composed of events with a strong spatial component to them (*i.e.*, the distances). As such, our first inclination is to favor the three distance-based approaches (*dbB*, SFI, and *Li*), and leave traditional *Bayes* on a second plan. *Bayes*, however, is elegant for its simplicity, though sensitive to event order, which can complicate reasoning of violent events and *storytelling* in general. It would be attractive to single one of them out as the most robust forecasting strategy, one which captures all associations with high certainty. While such an answer is not feasible, several considerations can be made based on knowledge of the dataset and the adjustment of parameters:

1. The experiments of Subsection 6.5.4 demonstrate that, for datasets across large spatial regions (across countries, for example), *dbB* has shown to provide higher precision than other approaches. For low event distances (for example, crime hotspots in Washington D.C.), on the other hand, either SFI (Subsection 6.5.2) or *Li* (Subsection 6.5.3) shows better performance.

2. When high recall is more important, we achieved better results with *dbB* or SFI. However, we have little basis to advocate for one versus the other.

3. Ontological resolution (Subsection 6.3.1) contributes significantly to precision and recall. Combining several events into like categories increases the chances of finding similar occurrences. An important implication, however, is that such combinations result in loss of information, and must be taken carefully.

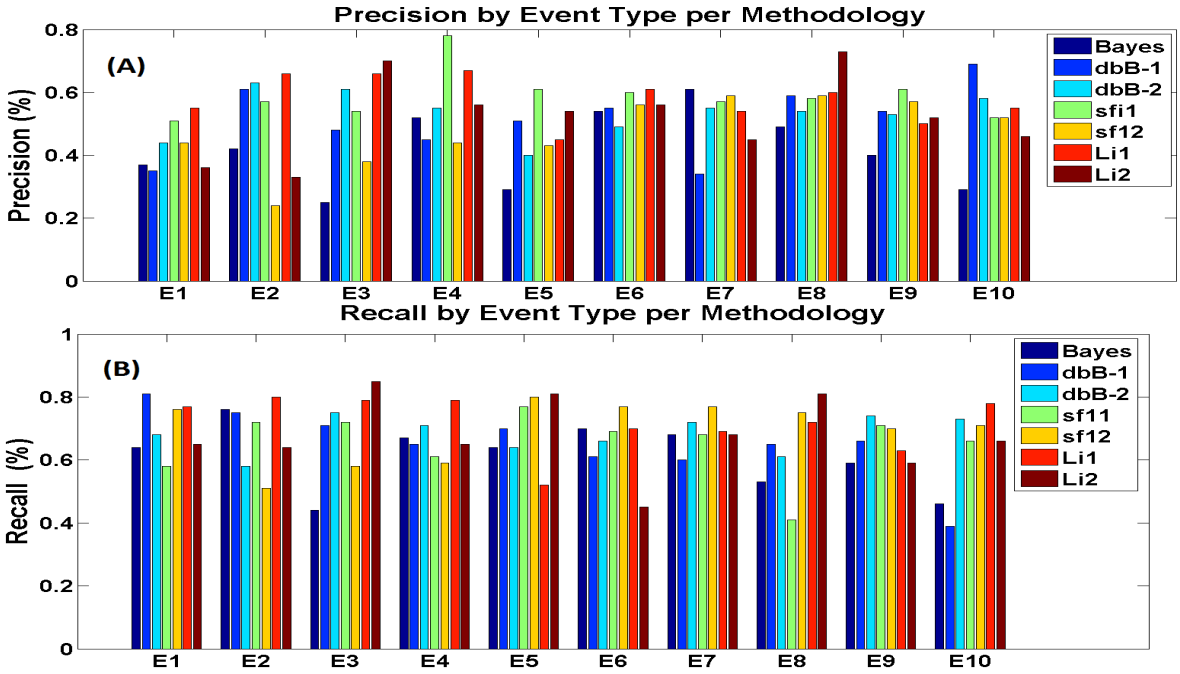


Figure 6.12: Bar plots corresponding to the 10 event types of Table 6.8. (A) Precision for each event type based on the 7 variations of methodologies employed. (B) Recall values. dbB-1 and dbB-2 correspond respectively to $nd \leq 0.5$ and $nd > 0.5$. $sfi1$ and $sfi2$ correspond to $sfi \leq 1.5$ and $sfi > 1.5$. $Li1$ and $Li2$ represent $df \leq 1.0$ and $df > 1.0$.

4. Data distribution must be analyzed thoroughly. In our experiments, we selected 30,000 data points to train, and 20,000 to test. Obviously, the training data contain an inherent amount of bias that affects how storylines are generated and rules extracted. Often, this problem is not as pronounced when distribution is highly uniform. Since uniformity cannot always be guaranteed, one way to avoid this problem is to improve sampling during the storyline generation process. A simple approach is based on the following steps: (1) sample k points from the dataset randomly; (2) for each point, generate m number of storylines into one group; (3) repeat steps 1 and 2 n times to generate n groups. The result will be a total of m×n storylines that, when performed in enough iterations, should capture enough of the true distribution of the data. Therefore, the storylines will be less prone to bias, which we surmise will lead to better application results, whether related to forecasting or other analytical tasks.

5. In the experiments of Subsection 6.5.4, we compare the different methods based on midpoint thresholds, such sfi values less than or greater than 1.5. These are values that worked well for our experiments, but can certainly be manipulated or even parameterized as user-defined inputs. These distances are highly dependent on the application domain and, whenever possible, should be experimented with extensively until optimal values are identified.

6.6 Conclusion

Storytelling has proven to be a valuable tool in the study of violent events. We have been able to demonstrate successfully how storylines capture the dynamics of violent events reported in social media and traditional databases. Our major contributions include a similarity measure and spatio-temporal modeling to reason over storylines, along with extensive experiments over disparate datasets. Our approach introspects interactions among entities and measures their influence to determine if they are compatible with the occurrence of violent events. The proposed numerical similarity for storylines is based on *Dynamic Time Warping* and *Euclidean distance*, and can be applied to common data analysis methods, such as *hierarchical clustering*. From an application perspective, we demonstrate how storylines can be used in four different forecasting strategies: traditional *Bayesian inference*, *distance-based Bayesian inference*, *spatial forecasting index* and *spatio-logical inference*. Experiments on civil unrest in *Mexico* and wars in the *Middle East* demonstrated our high potential for exploratory analysis. For future work, we plan to investigate distributed storytelling on massive datasets and more systematic storyline similarity metrics. Eventually, our objective is to establish storytelling as a robust tool for entity reasoning in a wide range of application domains.

Chapter 7

Spatial Similarity in Coherent Paths

Storytelling is the process of connecting entities through relationships to discover meaningful streams of information, a pervasive goal of intelligence analysis. One of its most challenging aspects is to identify coherent stories, that is, those that are bound to only one (or perhaps a limited few) themes of discussion. This problem mainly arises because the number of potential relationships among entities can be massive, which may lead to large numbers of stories whose themes stray in many directions. This study takes as input a set of entities, proposes a method to generate spatio-temporal storylines, and devises a technique to quantify their coherence. In addition, it introduces the concept of *boundaries* as a means to uncover hidden spatial relationships between disconnected entities. Experiments with *Twitter* data demonstrate that spatio-temporal storytelling is highly applicable to many different analytical tasks. A case study of event summarization on the 2014 political crisis of Ukraine finds well-described storylines as compared to other approaches, illustrating the effectiveness of storyline coherence for exploratory analysis.

7.1 Introduction

Broadly speaking, storytelling combines entities, events, and objects, linking them with relationships in order to generate meaningful streams of information [134]. Law enforcement, for instance, investigates people and organizations to identify illicit drug trading; and medical research observes symptoms to track the causes of diseases. Many of these activities can find reasonable answers to their specific problems with the help of storytelling. See Fig. 7.1 for the types of automated stories envisioned in this study.

The process of telling a convincing story is challenging for a few reasons, three of which authors have long struggled with. First, a good story revolves around just a limited number of themes, avoiding unnecessary clutter. Second, the themes themselves are key. They should propagate through space and time in a continuous manner using accessory facts as support. And third, the length of the story must be treated carefully. Long stories are acceptable as long as the theme(s) in

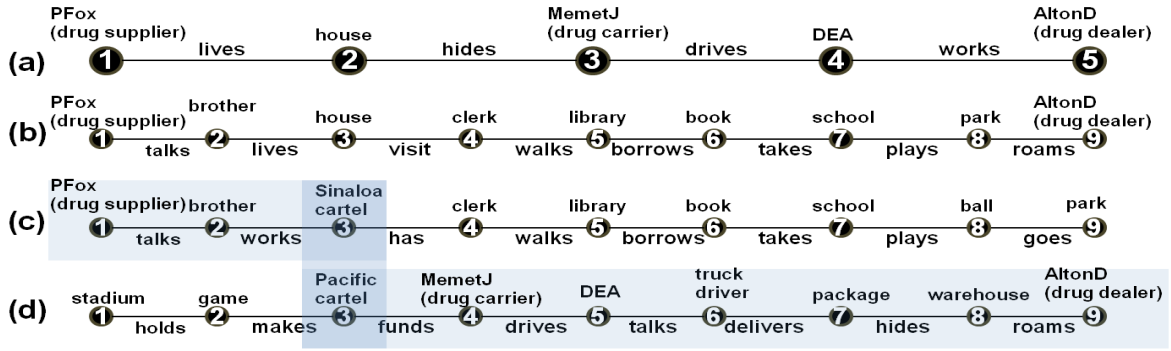


Figure 7.1: Four hypothetical storylines connecting (PFox) to (AltonD). Storyline (a) has 5 entities connected by four relationships. Storyline (b) connects the same entities, but uses more links in between. Storyline (a) appears more coherent than (b) because (a) maintains the theme of discussion solely around drug trafficking. Combining storylines (c) and (d) makes the drug operation become better delineated as the storyline evolves.

question is kept on focus. These three aspects, which could serve as guidelines in traditional writing, are explored in this paper in an algorithmic fashion more suitable for automated storytelling.

The goal of this study is not to write stories in the literary sense. Rather, it is to perform automated reasoning between a starting and finishing entities such that a highly-coherent line of facts can be revealed between them. Highly coherent means that these stories include no more than k themes of discussion (the experiments will target two or less), and that as much as possible, these themes keep repeating along the way. A theme of discussion is simply a given subject of interest, such as 'elections' or 'sports talk' and associated concepts. In the literary sense, [60] defines coherence as a function in which "statements connect with the ones preceding it ... to establish a master plan". This definition, while adequate in a writing scenario, is too subjective for real use, and must be further specified quantitatively. This study reformulates it as follows: coherence is the ratio of words that belong in a theme of interest over the total number of words being considered (see Definition 21). This definition will be adapted for storylines in Subsection 7.4.2.

The framework proposed here does the following: takes as input disconnected data records containing spatio-temporal entities (i.e. tagged with location and time), extracts them, and builds a graph using the entities as nodes and their relationships as edges. These edges are words which are transformed into numerical scores, which assists in later calculations of our proposed semantic coherences. The graph is then traversed to find streams of information that would otherwise not be apparent if the entities were viewed separately in their individual documents. In this sense, this framework is not creating stories, as they already exist in a latent state. Rather, it applies automated numerical methods to stitch together the relevant actors and actions such that a valid story becomes clear, much in the same way a professional investigator would do manually.

To begin the discussion, a modeling strategy on how to enforce storyline coherence is needed (the terms 'stories' and 'storylines' are used interchangeably). Take for instance Fig. 7.1, which depicts four hypothetical storylines related to drug trafficking activities. These storylines are comprised of entities (i.e., people, places, organizations, objects) such as ❶ and ❷, where adjacent entities are connected by a conceptual relationship, such as 'lives' and 'hides'. Storylines may have different lengths, which is defined as the total number of entities. Thus, the length of storyline (a) = 5,

whereas the length of (b) = 9.

Storyline (a) is the type that is highly desirable to find due to its coherence: almost all of its entities are related to drug trafficking (*PFox drug supplier*, *MemetJ drug carrier*, U.S. Drug Enforcement Administration (*DEA*), *AltonD drug dealer*) and associated activities. In other words, the drug trafficking theme propagates well throughout the storyline, and it thus appears plausible. Storyline (b), on the other hand, appears incoherent. While it involves some of the drug dealing entities, it includes a higher proportion of other entities, such as 'library', 'book' and 'ball', that arguably bear little relevance to the drug scenario. As for storylines (c) and (d), they also appear incoherent when considered separately. But when the two are combined into one, as depicted by the shaded area, they also carry high coherence. Note that the shaded combination of (c) and (d) are connected via the entities denoted as *Sinaloa cartel* and *Pacific cartel*, what we call a *boundary* (discussed later). Even though this combination is much longer than (a), they are just about as convincing.

The storylines of Fig. 7.1 can be generated from an entity graph using conceptual relationships as the edges between the entities. Extracting storylines from a graph, then, becomes a task of finding a path between any two entities. While finding a path is not necessarily a problem, finding a highly-coherent path is an elusive task. There are several challenges that affect coherence, chief among them are the following:

(1) **Lack of a relevance model**: In textual sources, relationships between two entities are word concepts, not numbers. An example would be a "drug supplier carries drugs to drug dealer". Because these relationships are conceptual, and thus too subjective for automation, measuring their relevancy becomes challenging. These relationships must be quantified from their word descriptions so that they can be ranked;

(2) **Excessive number of themes**: relationships can be described at different conceptual resolutions involving different themes of interest. In Fig. 7.1, for example, one may be interested in all storylines concerning drug trafficking operations in a transportation setting. But other themes are possible, such as drugs and politics or drugs and education. This potentially-high number of themes demands a method with which storylines can be drawn according to the desired topics so to enforce maximal coherence in different scenarios;

(3) **Spatio-temporal discrepancies**: certain relationships are only legitimate when they take place in a certain location and time. A drug sale, for instance, requires both seller and buyer to come together at around the same time. If they are deemed far apart, this relationship may become doubtful, making a real-life investigation inconclusive. Thus, space and time generate discrepancies which must be taken into account;

The premise of this study is that storyline coherence can be maximized by minimizing the negative effects of the three issues described above. Simply put, a framework is needed to compute coherence from three factors: relationship strength, spatio-temporal continuity, and theme propagation. These ideas summarize our contributions, which are formally stated as follows:

(1) **Devising a spatio-temporal model for relationships**: this work proposes a method that generates a semantic signature for each entity based on its spatial location, temporal difference to other entities, and number of typed relationships. Using these signatures, a measure of relationship strength between any pair of entities, what we denote as *binding strength*, can be calculated using common methods such as L^2-*norm*. This allows the comparison of millions of entities so that their storylines can be ranked by relevance;

(2) **Designing a numerical method of storyline coherence**: key to enforcing cohesiveness is to take into account each relationship strength along the storyline's path. Here, a method is proposed that utilizes storyline distance, storyline length, and theme propagation to calculate the semantic coherence of a storyline. In this manner, different storylines can be compared and reasoned on how truly convincing their are based on their specific themes;

(3) **Combining disparate storylines to increase coherence**: On its own, a storyline may be impertinent to specific scenarios. When combined with others, however, important information may surface. This research shows that when disconnected storylines (or parts of) are combined via proximally-close entities, what is denoted as a *boundary*, the resulting storyline can be highly coherent. This can potentially uncover hidden tracks of knowledge that might remain unidentified otherwise;

This paper is organized as follows. In Section 7.2, related works are discussed. Section 7.3 describes in technical detail how each of the above contributions can be achieved. Experiments and an in-depth analysis of the results are presented in Section 7.5. A conclusion is finally given in Section 7.6.

7.2 Related Works

Storytelling involves performing several tasks. As such, it can be described as a platform of knowledge exploration for fact finding, association discovery, and inferencing. Moreover, its goals can range widely, making it dependent on a combination of semantic analysis and the technical quantitative fields. The work proposed in this paper spans many areas of expertise, but best aligns with the approaches described below.

7.2.1 Storytelling and Connecting the Dots

The phrase 'storytelling' was introduced by [67] as a generalization of *redescription mining*. At a high level, *redescription mining* takes as input a set of objects and a collection of subsets defined over those objects with the goal of identifying objects described in two or more different ways. Such objects may signal shared behavior, which can be a powerful tool in the context of *storytelling*.

Hossain et al. [51] develop the above idea to connect two unrelated *PubMed* documents where connectivity is defined based on a graph structure, using the notions of hammocks (similarity) and cliques (neighborhoods). This work was generalized to entity networks in [50] and specifically targeted for use in intelligence analysis. The authors' motivation is that current technology lacks better support for entity linkage, explanation of relationships, exploration of user-specified entities, and automated reasoning in general. The tools used in this work include concept lattices as a network where candidate entities are identified with three nearest neighbor approaches (Cover Tree, k-Clique, and NN Approximation). The *Soergel Distance* measures the strength between entities, while *coreferencing* serves to identify entities mentioned in various parts of the text using differing terms. These works link entities according to a desired neighborhood size and distance threshold. In many of these works, edge weight is based on a variation of term frequency $\times$ inverse-document-frequency (*TF-IDF*). This class of works represent *traditional storytelling* approaches that do not address the geospatial perspective.

In the realm of frequent pattern mining, research related to our work comes from *Cascading Spatio-Temporal Pattern Discovery (CSTP)*, proposed by [98]. *CSTP* identifies partially-ordered subsets of event types that are colocated and sequential. The goal of this approach is not to perform story-telling per se, but its focus on event association is a significant step in that direction. *CSTP* accepts boolean event types and computes a measure of *interestingness* for a pattern, namely a *Cascade Participation Ratio*, as the probability of observing a *CSTP* within an entire dataset of event types, such as events that lead to crime occurrences. In our approach, a point of differentiation is that event types are not necessarily boolean. We accept the notion that entities have latent relationships based on one's personal belief of which links between entities are possible, even when the underlying dataset does not support them. This approach has been optimized in several aspects, such as by minimizing the number of candidate patterns. With modifications, *CSTP* can be a valuable tool complementary to our work with respect to entity generalization which can be helpful in relationship inferencing.

Connecting the dots-type approaches focus on document linkage rather than entity connectivity. They apply textual reasoning as a strong facet of the targeted methods, which departs from a spatio-temporal view of events. Link strength utilizes the notion of *coherence* across documents, which is proposed by [120]. In this work, stories are modeled as chains of articles, where the appearance of shared words across documents help establish their relatedness. Extending that work, they also propose related methods to generate document summaries, i.e. *Metro Maps*, in [122] and [121], which target scientific literature. Some of the goals are to measure the importance of a paper in relation to the corpus, find the probability that two papers originate from the same source, and identify research lines. Overall, *connecting the dots* methods rely on the abundance of robust content. The types of datasets used in our study, such as *Twitter* data, however, break the assumption of robust content, limiting the amount of textual reasoning that can be performed. Thus, *connecting the dots* is less than ideal for environments that rely on such data feeds.

7.2.2 Link Analysis

Often relying on graphs as a modeling abstraction, this class of work studies the connections among entities ([91], [27]) and the identification of patterns ([42], [26]). While spatio-temporal storytelling reasons over which entities to link and how to link them, link analysis help in understanding the connections that have been made. This leads to the notion of ranking.

Ranking in terms of link analysis has been popularly applied to web pages since the seminal works of [18] and [64]. The former computes the value of the importance of a web page based on its links and an initial damping factor. The latter also consider the page's links, but is dependent on an initial query that generates a *root set*, and is augmented by other pages that point to the *root set*.

Within the same family of the above approaches, there have been other proposed methods. The *Indegree Algorithm* is a simple heuristic that considers the *popularity* factor as a ranking measure [84]. For social media, *popularity* is a gray area: works well for high-visibility events, but may fail miserably for events that are important, but that do not get much exposure. For *storytelling*, this type of applicability is possible, but too subjective in terms of ranking. The *HITS Algorithm* by [64] introduced the notion of *hub and authority*, where authorities are the pages that hold "legitimate" information, and hubs are the pages directing the user to the authorities. In terms of *storytelling*, this type of ranking would be challenging since there is no clear-cut way to determine which entities would be authorities and which would be hubs. It represents an open line of research, but outside the scope of this document.

7.2.3 Event Detection and Summarization

The goal of storytelling is to find meaningful streams of information that are neither explicitly stated in text nor apparent to the naked eye. It can be used in many different ways, such as event detection and summarization, and demonstrated in the experiments section.

In terms of event detection, event expansion and topic trending are two commonly-studied aspects. Event expansion starts with limited bits of information about an event and seeks to expand it using social media data. Topic trending, on the other hand, monitors large volumes of social streams to find the most popular themes of discussion. The work of [114] targets the detection of earthquakes in Japan using common classification techniques. Events are defined by the user by selecting keywords. TEDAS by [76] describes a system for detecting new events related to crime and natural disasters, and identify their importance. It first crawls tweets, classify them as event-related or not, and stores spatio-temporal information. Users then issue queries that contain location, time, and keywords, which the system uses to retrieve and display related events. The importance of event reporting over *Twitter* is questioned by the work of [101]. The authors claim that the benefit of tweets comes from increased coverage, not timeliness. They devise a system that clusters both tweets and news articles, and measure their overlap to discover the coverage of one versus the other. Comparisons can then be done on their spread over time. Twevent by

[75], a different approach, proposes segment-based event identification. Initially, it detects bursty events and clusters them using frequency and content similarity. The similarity between segments is computed using their associated tweets, while Wikipedia is searched to verify which events are realistic or not. In [138], the authors monitor specific locations of high tweeting activity. They further analyze clusters of those tweets, using machine learning to detect if the identified posts during high activity represent real events or not.

Textual summarization has been well studied in IR, using a wide variety of techniques, such as latent semantic analysis and machine learning as in the works of [43, 34]. Event summarization, as an extension, has gained strength in recent years due to social networks. *TwitInfo* describes a system that allows users to navigate a repository of tweets, where the system discovers high peaks of twitter activity ([85]). In addition, the system allows geolocation and sentiment visualization. A more comprehensive approach to event summarization is detailed in [24]. The authors propose a segment-based approach where summarization takes places within each segment. This technique can take on different variations. The first uses cosine similarity as a straightforward method. The second applies a similar approach, but considers tweets that fall within a specific time window. A third approach uses a Hidden Markov Model (HMM) where each state can be a sub-class of events (e.g., "touchdown" in a football event). An alternative technique also based on time segmentation is given by [87], but with the added assistance of synonym expansion for keywords. For each of these approaches, the output is the set of tweets that best summarizes the events.

Differences: Each of the above research fields provides solutions to the various tasks involved in storytelling. Challenges and requirements come in different flavors as a result of application demands or data characteristics. The work in this paper, for instance, requires geolocation of entities as it relies on a spatio-temporal model where both geographical proximity and time ordering are favored. In this sense, the focus is on methods for which spatial influence and time sequencing can be intuitively justified by semantic analysis. The vast majority of the other methods are textual by nature. Given the many differences in what each technique can contribute, this paper does not describe a competing approach. Rather, it presents complementary techniques that demonstrate how storylines can be a valuable analysis tool for intelligence analysis, show how they can be coherent, and cover a spatio-temporal niche which remains largely untapped.

7.3 Spatio-temporal Modeling

The discussion about spatio-temporal modeling begins with Subsection 7.3.1, where the definitions used in the remainder of this paper are given. Subsection 7.3.2 describes the general workflow of the proposed approach. Further, Subsection 7.3.3 provides a detailed explanation on how to devise semantic signatures for entities and determine their binding strengths.

7.3.1 Definitions

In the scope of our study, an entity network is a graph $G(E,R)$ where entities $E=\{e_1, ..., e_n\}$ can be linked to one another through relationships $R=\{r_1, ..., r_n\}$ defined by conceptual interactions, and thus called an *entity graph*. Given a set of documents $D=\{d_1, d_2,..., d_n\}$, the following definitions apply:

Definition 15. ***Entity*** *- An entity e represents a person, location, organization, event, or object described in at least one document $d_i \in D$. Only entities for which a location and a timestamp can be obtained are considered in this study.*

Definition 16. ***Relationship*** *- A relationship, connection, or link defines a unit of interaction between two entities and is denoted by $e_i \overset{interaction}{\longrightarrow} e_j$.*

Definition 17. ***Entrypoint*** *- An entrypoint is any entity e in the dataset from where the story evolves. Similarly, an endpoint is any entity e where the story ends. They are both application-dependent. For instance, in the storyline of Fig. 7.1(a), the entrypoint is* PFox drug supplier *and the endpoint is* AltonD drug dealer*.*

Definition 18. ***Storyline*** *- A storyline is a time-ordered sequence of n entities $\{e_1, ..., e_n\}$ where consecutive pairs (e_i,e_j) are linked by at least one relationship.*

Definition 19. ***Storyline Length*** *- The length of a storyline, denoted sl, is the number of entities that compose that storyline.*

Definition 20. ***Storyline Distance*** *- The distance of a storyline, denoted sd, is the sum of the values of all edges linking the composing entities of that storyline.*

Definition 21. ***Coherence*** *- Coherence of theme t w.r.t. document d is the ratio of words in document d that belongs to theme t over the total number of words in document d. For example, if t = {illicit drugs} and d = "The new economy has legalized the use of marijuana, but heroin remains the biggest threat", then coherence(d,t) = $\frac{2}{15}$ = 0.13. The number '2' corresponds to the two underlined words of d, while the number '15' is the total number of words in d. This definition of coherence will be refined for storylines in Subsection 7.4.2.*

7.3.2 Spatio-temporal Storytelling Workflow

In the process of telling a story, there are at a minimum two basic considerations to be made: first, the **actors** involved in the plot must be identified; second, the **actions** involving the actors must be established such that the plot can take shape. In this study, actors are the entities extracted from the dataset, such as people, organizations, and objects. All entities have a location and timestamp, which denote their place and time of observation. Actions are the relationships between those entities, such as when two people *talk* or an organization *interacts* with a person.

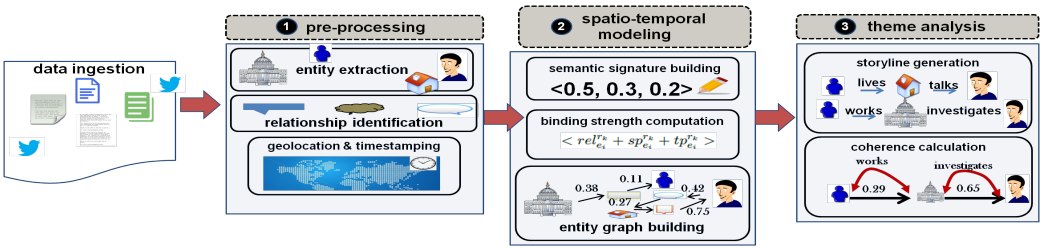


Figure 7.2: Conceptual approach for theme analysis: (1) entities and relationships are extracted and geo-time coded from the ingested data; (2) semantic signatures are devised for each entity, from which binding strengths are calculated, and a graph is built; (3) storyline coherence is calculated.

At a high level, this study goes through four stages of activities: entities are extracted from the dataset, an entity graph is built, storylines are generated from the graph, and two coherence measures are computed for each generated storyline. The next subsections will explain each of these steps, as depicted in Fig. 7.2. It is assumed the availability of a dataset which can be structured, such as a database, or unstructured, such as *Twitter* data or text documents.

In the **pre-processing stage**, entities are identified along with their corresponding relationships. Additionally, the locations and timestamps of each entity are extracted either from text or from document metadata. Locations are geocoded as latitude and longitude, and later used in the computation of their spatial distances. **Stage 2** utilizes the relationships, geolocations, and timestamps to build semantic signatures for each entity (Subsection 7.3.3), which are subsequently used to compute their binding strength. An entity graph is then built using the binding strengths as edge values. With a graph in hand, **Stage 3** generates storylines for which two numerical coherence scores are calculated and evaluated against others as part of our theme analysis. (Subsection 7.4.2). It is further shown how seemingly disconnected storylines can be merged to include more information and still remain coherent. In the next subsections, each step of the spatio-temporal model (Stage 2) and theme analysis (Stage 3) are discussed in detail.

7.3.3 Semantic signatures

The first task we propose is to quantify the connectivity between entities, which is denoted as their *binding strength*. In later stages, storyline coherence will be calculated using a sequence of entities and their binding strengths.

In order to determine binding strength, three features can be utilized: the relationships that an entity establishes with others; each entity's spatial location; and each entity's time of observation. All of these three factors are important because, in many intelligence analysis scenarios, for a story to be coherent its entities must be close in space and time while their interactions take place. Consider, for instance, Fig. 7.3, which depicts hypothetical interactions between the Sinaloa drug cartel, several people (CDenn, AltonD, etc), and organizations (DEA, Pacific Cartel). The entity graph shows four links originating at Y.Parks that connect this person to other entities such as LNord and DEA, among others, labeled respectively as *r1* (talks), *r2* (interacts), and *r3*

(works). The entity graph, however, does not show how strong these bonds are. To transform these relationships into numerical scores, the three features listed above create a *semantic signature* for every relationship of every entity in the graph.

The semantic signature is a vector that summarizes both the number and types of relationships that an entity makes with other entities, along with their spatial distances and temporal differences (all normalized in the range [0,1]). In later stages, these signatures will be used to compute a *binding strength* and two coherence scores. These two scores are based on one or more themes of discussion, which are given to the application as input parameters. Fig. 7.3 provides the following information: entity Y.Parks has four relationships (*r1*, *r1*, *r2*, *r3*). Assuming that we are only interested in *r1* for the time being, the table shows two *r1* instances out of a total of four relationships. Each one of those two *r1* instances is associated to a spatial distance, which is the distance between Y.Parks and RMarin as well as the distance between Y.Parks and LNord. The average of those two distances is shown in the table as 0.81 (again, normalized). The temporal distance indicates the difference in timestamps between the observations of the two entities. Thus, if Y.Parks was seen at 12:35 PM and LNord at 12:40 PM, their time difference is 5 minutes, which the table shows as a normalized value of 0.35. The last column of the table combines those three values into the signature vector for entity Y.Parks with respect to *r1*, which is <0.5,0.81,0.35>. The same is done for entity LNord, and in practice, this process would need to be replicated across all entities.

Formally, the semantic signature of entity e_i w.r.t. relationship r_k is defined as:

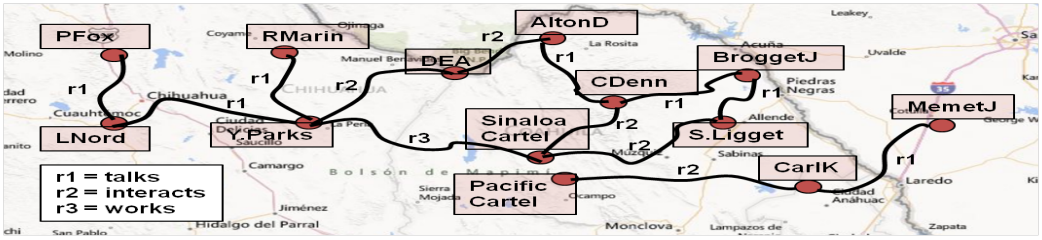
$$ssig(e_i, r_k) =< rel_{e_i}^{r_k} + sp_{e_i}^{r_k} + tp_{e_i}^{r_k} > \tag{7.1}$$

$$rel_{e_i}^{r_k} = w \times \frac{count(r_k)}{|R|} \tag{7.2}$$

$$sp_{e_i}^{r_k} = f(spatial\ distance^{r_k}(e_i, e_j)) \tag{7.3}$$

$$tp_{e_i}^{r_k} = f(temporal\ distance^{r_k}(e_i, e_j)) \tag{7.4}$$

where R is the set of all relationships of e_i, $r_k \in R$, count(r_k) represents the number of r_k relationships established by entity e_i, w is a user-defined weight to differentiate relationships, and f is a



entity	r1 instances / all instances	average r1 spatial distance	average r1 temporal distance	r1 semantic signature
Y.Parks	2/4	0.81	0.35	< 0.5, 0.81, 0.35>
LNord	2/2	0.47	0.62	<1.0, 0.47, 0.62 >

Figure 7.3: Entity graph depicting interactions between the Sinaloa drug cartel, a law enforcement organization (DEA), and several hypothetical individuals. Using relationship r1 as an example, the table summarizes its instances for entities Y.Parks and LNord. Also shown are the average spatial and temporal distances between the entities as well as their semantic signatures.

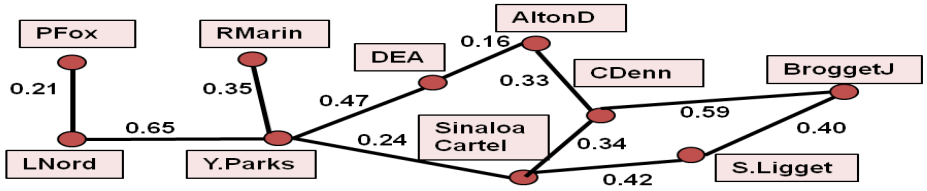


Figure 7.4: Entity graph corresponding to Fig. 7.3. Instead of conceptual relationships, the edges show the binding strength computed from the semantic signatures of each entity.

function of the spatial or temporal distances among all entities e_j that are linked to e_i through r_k. This function can be an aggregate such as standard deviation or average of the spatial or temporal distances.

Intuitively, each signature vector provides a quick snapshot of the actions that an entity participates in. Thus if entity A "meets" entity B, the signature gives a picture of how many times they met, how far apart they were geospatially, and the time difference of when they were observed. In the context of a real application, it would be akin to identifying the "whats, wheres, and whens" of the activities taking place. This process allows each entity to be embedded with its own semantic signatures, which can then be used to calculate the binding strength $bind(e_i,e_j,r_k)$ between a pair of entities w.r.t a specific relationship:

$$bind(e_i, e_j, r_k) = f(ssig(e_i, r_k), ssig(e_j, r_k)) \tag{7.5}$$

Eq. 7.5 represents a function f that takes as input the semantic signatures of the entities in question and outputs a single numerical value of their binding strength. Any function that compares vectors can be used. A well-established approach is the L^2*-norm* (a.k.a. *Euclidean distance*), which is attractive for its compliance with truly metric properties required of many geospatial applications:

$$bind(e_i, e_j, r_k) = \frac{1}{n} \times \left[\sum_{m=1}^{n} \left| e_{im} - e_{jm} \right|^2 \right]^{\frac{1}{2}} \tag{7.6}$$

As an example, the binding strength of the two entities in the table of Fig. 7.3 can be computed as follows:

bind(Y.Parks , LNord ,*talks*) = $\frac{1}{3} \times [(0.5\text{-}1.0)^2+(0.81\text{-}0.47)^2+(0.35\text{-}0.62)^2]^{\frac{1}{2}} = 0.22$

The above score of 0.22 is a measure of high tightly-bound those two entities are based on their 'talks' interaction. This value is highly sensitive to the number of other 'talks' that Y.Parks and LNord have with other people: the more they resemble each other (i.e, they talk to equal or close numbers of people), the lower the score will be. And since this value represents *distance*, a low value represents high *similarity*. The same can be said for spatial and temporal distances: the farther away these entities are in space and time, the higher the score, and thus the lower their similarity.

Now that a binding strength can be calculated for each pair of entities, Fig. 7.3 can be transformed into Fig. 7.4, whose edges show the binding strengths in lieu of relationship names. For completeness, we added not only the 0.22 value computed above, but also filled in values for all of the edges. This graph provides

a platform with which storytelling can be performed by analyzing the possible paths available among the entities. In later subsections, we explore these paths and provide a measure to compute their levels of coherence given a set of themes of interest.

7.4 Spatio-temporal Theme Analysis

In this section, two objectives are envisioned. First, Subsection 7.4.1 proposes an optimization step which shows that a large data space does not imply that every entity must be investigated. The data space can be segmented with limited views of the entities by applying spatial clustering. Second, Subsection 7.4.2 uses the *semantic signatures* calculated in Subsection 7.3.3 and apply them in the design of storyline coherence, as explained below.

7.4.1 Entity optimization with spatial clustering

The approach described previously proposes the investigation of every spatio-temporal entity in the dataset. In reality, however, inspecting every entity may be a cost-prohibitive task given large data volumes. To alleviate this problem, a method that limits the number of entities that should be included in the analysis is needed. One could, for example, admit entities of a certain type, such as 'people', and reject other types, such as 'organizations'. Another idea would be to remove all entities with relationships irrelevant to the task at hand. A large number of such rules can be devised, but it can be challenging to determine the appropriate ones. To avoid these ad-hoc heuristics, but still limit the number of entities to be investigated, we propose a spatio-temporal clustering-factor approach based on *Ripley's K* function, as explained below.

In general, the *K* function [111] takes as input a set of data points along with an initial radius of interest from a starting point, and outputs a clustering factor for that radius. This clustering factor is a numerical score that reflects the density of data points for a spatial region based on the selected radius. Different radii can be evaluated to find the area where the most clustering takes place. This area where the most clustering takes place is then selected for storytelling, i.e., only entities in that area will be considered in the analysis. Take as an example Fig. 7.5(a), which depicts several data points scattered throughout a 3×3 km area. Circle A has a radius of 1 km and shows a high density of data points, whereas circle B and C, which have radii of 2 and 3 km respectively, are sparser. Since circle A already contains most of the data points, we can limit our analysis to those points, and, at this stage, disregard the others. In terms of storytelling, the data points are entities, and mathematically, *Ripley's K* function can be stated as:

$$K(r) = \sqrt{\frac{A \sum_{i=1}^{n} \sum_{j=1}^{n} w(i,j)}{\pi n(n-1)}}, i \neq j \tag{7.7}$$

where r is a desired radius originating at a chosen entrypoint, n is the total number of entities in the data space, A is the entire area of study, and $w(i, j)$ represents a weight. $w(i, j) = 1$ if spatial distance(e_i, e_j) $< r$, and 0 otherwise. Fig(s) 7.5(b) and (c) show two simple calculations of the *K-coefficient* for 3 persons $\{P_1, P_2$, and $P_3\}$ located in a (3 km x 3 km) area A. In 7.5(b), the chosen radius is 1 km. The calculation follows: using each entity P_i as the center of a 1 km circle, count the number of other entities P_j within that radius, adding 1 if their distance is less than the radius, zero otherwise. In that range, P_1 "can see" 2 others

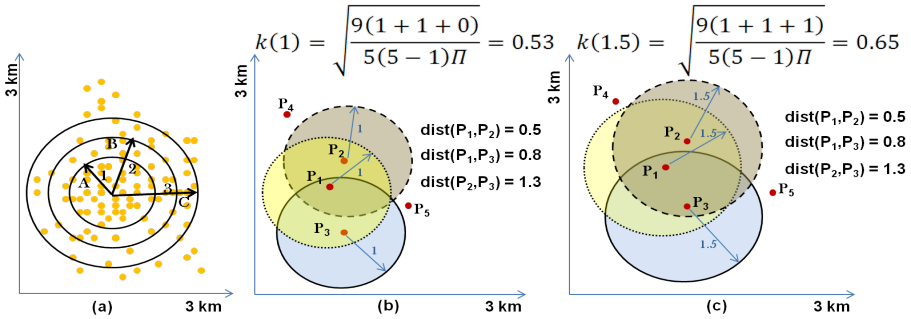


Figure 7.5: Spatial scaling for different radii. (a) Circle A depicts high entity density, becoming more sparse in circles B and C. (b) and (c) shows the calculation of *Ripley's K* function for a 1 and 1.5-km radius respectively.

(P_2 and P_3), since their respective distances ($dist(P_1, P_2)$ and $dist(P_1, P_3)$) are both less than $r = 1$. Using P_2 as the center of a 1km-radius, P_2 "sees" only P_1. The same is true for P_3, which yields $K(1) = 0.53$. In Fig. 7.5(b), the radius is increased to 1.5 km, and the calculations are repeated, yielding a $K(1.5) = 0.65$. Therefore we would use the 1.5 km radius for its higher clustering factor.

From an application perspective, if an analyst were to investigate the link between entities PFox and AltonD in Fig. 7.4, he/she would set PFox 's location as the center point of the circle, and try different radii to find nearby entities. If AltonD were not found initially, the process would be repeated from the farthest entity in the radius towards AltonD. Besides limiting the region of study, *Ripley's K* function provides another important feature: it enforces locality, i.e., storylines are formed with entities that are spatially close to one another since it is bounded by a radius. Locality is a hard requirement in many types of geospatial applications, but can be omitted otherwise. In our experiments, we show how storyline coherence can be impacted by variations in the radius of study.

7.4.2 Coherence Model

With a graph such as the one in Fig. 7.4, generating a storyline between two entities becomes the task of finding a path connecting those two entities. By definition, any path linking two entities represents a potential storyline between them.

The graph of Fig. 7.4 shows that there can be many competing paths connecting two entities. For example, a storyline from PFox to AltonD has two possible versions:

(s_1) PFox $\xrightarrow{0.21}$ LNord $\xrightarrow{0.22}$ Y.Parks $\xrightarrow{0.47}$ DEA $\xrightarrow{0.16}$ AltonD

(s_2) PFox $\xrightarrow{0.21}$ LNord $\xrightarrow{0.22}$ Y.Parks $\xrightarrow{0.24}$ Sinaloa Cartel $\xrightarrow{0.34}$ CDenn $\xrightarrow{0.33}$ AltonD

According to Def. 19, s_2 is longer than s_1 since s_2 has more entities (6 as opposed to 5). Commonly, readers tend to favor shorter stories over longer ones due to their brevity. In intelligence analysis, however, no assumption can be made on whether 'shorter' is better than 'longer' or vice-versa. In the context of the drug trafficking scenario, for instance, s_2 connects several people in the path of the Sinaloa cartel, while s_1 does not. In this scenario, an investigator may find s_2 more interesting than s_1 since s_2 relays more information, even though it is the longer of the two storylines. This example motivates us to consider the length

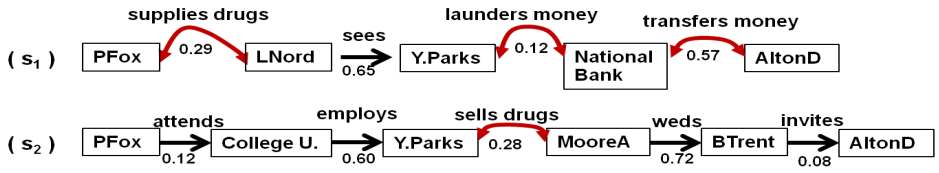


Figure 7.6: Two example storylines related to a drug trafficking scenario connecting PFox to AltonD. Storyline s_1 has three relationships of interest, denoted by the red edges, while s_2 has only one. The numbers indicate the binding strength between entities.

of a storyline (*sl*) as an important factor of our analysis.

A second factor that must be contended with is storyline distance (*sd*), which according to Def. 20, equals the sum of the edges between the entities of a storyline. Thus for s_1, its *sd* = 0.21+0.22+0.47+0.16 = 1.06, while for s_2 the *sd* = 0.21+0.22+0.24+0.34+0.33 = 1.34. Again, s_2 is more distant than s_1 for the same entrypoint and endpoint, but gives us no clue to which one is better. While both *sl* and *sd* provide valuable information, they do not on their own lead to coherent stories, which is our ultimate goal. For this reason, the theme-coherence approach requires further refinement, as explained below.

The initial claim of this paper was that a coherent story maintains focus on only one or possibly a few themes of discussion. Incorporating too many plots makes the story less intelligible to the human mind. An application that focuses on pollution and medicine, for example, should most likely not involve sports or real estate (at least not to a great extent). This simple observation points us in the following direction: whenever more than one version of a story can be drawn, the most coherent is the one that propagates the desired theme(s) with the highest frequency. And not only that, longer stories as well as more distant entry and endpoints should propagate the theme more frequently than shorter stories. A visual example follows.

Take Fig. 7.6 as an example which depicts two storylines (s_1 and s_2) between entrypoint PFox and endpoint AltonD. Assume that these two entities are being investigated for **drug trafficking** and **financial dealings**, which correspond to two themes of interest. The first storyline, s_1, encompasses three relationships related to the two themes: *supplies drugs*, *launders money*, and *transfers money*. Those are the relationships represented by curved edges. The second storyline, s_2, has only one, which is *sells drugs*. At a quick glance, one could say that s_1 is more coherent than s_2 because s_1 propagates the two themes more frequently than s_2 does. However, this statement is flawed because s_1 has shorter length (5 entities) than s_2 (6 entities), and thus the two should not be compared directly. To be fair, the comparison should be in terms of 'participation ratios' instead of absolute numbers. Fig. 7.6 shows that all of s_1's entities participate in one of the two themes of interest, and thus its participation ratio = $\frac{5\ participating\ entities}{5\ total\ entities}$ whereas s_2's ratio = $\frac{2\ participating\ entities}{6\ total\ entities}$. Clearly, s_1 has a higher ratio than s_2, which allows us to state that s_1 is indeed more coherent than s_2 based on those two themes of interest. The participation ratio discussed above is what we propose as *Lengthwise Semantic Coherence* (LSC) for a storyline and defined as follows:

$$LSC(s_i) = \frac{\sum_{n=1}^{|E|} 1 \ \ where\ r_k \in T}{|E|} \tag{7.8}$$

Given a set of themes of interest T={t_1,...,t_n}, a set of entities E={e_1,...,e_n} and a set of relationships R={r_1,...,r_n} that compose storyline s_i, LSC(s_i) is the number of entities in storyline s_i that have a relationship r_n that belongs in the theme set divided by the total number of entities in s_i. *LSC* resides in the range [0,1].

Formally, the *Lengthwise Semantic Coherence* for storylines s_1 and s_2 would be computed as $LSC(s_1)$= $\frac{5}{5}$ = 1.0 and $LSC(s_2)$= $\frac{2}{6}$ = 0.33. Note that storyline s_2 is numerically less coherent than s_1. This is because s_2 has

only two entities (Y.Parks , MooreA) connected by a theme-related relationship (*sells drugs*). Everything else seems irrelevant, revolving around people attending college or involved in a wedding. While these extra facts may be true, they only lend scant knowledge to the investigation, and are not as interesting in a drug trafficking scenario.

Storyline length is not the only factor that impacts storyline coherence. A second factor has to do with storyline distance (*sd*), which according to Def. 20 is the sum of the edges from entrypoint to endpoint. For s_1, *sd* = 0.29+0.22+0.12+0.57 = 1.20 and for s_2, *sd* = 0.12+0.60+0.28+0.72+0.08 = 1.80. Based only on *sd*, storyline s_2 is the most coherent of the two stories. Again, this comparison is unfair since s_2 has more entities than s_1, which would tend to accumulate longer distances. To get around this issue, we follow a similar strategy as before, and calculate a participation ratio based on storyline distance instead of storyline length. For this purpose, we define the *Distancewise Semantic Coherence* (DSC) for storyline s_i as follows:

$$DSC(s_i) = \frac{\sum_{m=1,n=1}^{|E|} \sum_{k=1}^{|T|} bind(e_m, e_n, t_k)}{\sum_{m=1,n=1}^{|E|} \sum_{k=1}^{|R|} bind(e_m, e_n, r_k)}, \quad m \neq n \tag{7.9}$$

which corresponds to the sum of the binding strengths (edges) between entities in storyline s_i that have a relationship r_k as part of the set of desired themes T divided by the sum of all binding strengths in s_i. This value also falls in the range [0,1].

The two proposed approaches of coherence measurement (*LSC* and *DSC*) give different views of how a theme propagates throughout the storyline. It must be noted that more robust results can be achieved if the themes are expanded either with an ontology or a dictionary approach, such as [143]. In this manner, concepts such as "finance" can be expanded with related terms, such as "money" or "payment". Because *LSC* is length-based, it only looks at how entities connect via conceptual relationships. *DSC*, on the other hand, embeds spatio-temporal knowledge, favoring not simply conceptual relationships, but those which are in close proximity of space and time. *DSC* requires distance and time computations, which translates to higher computational complexity. *LSC* is less resource-intensive as it only requires relationship frequencies, but may not be adequate for applications that demand geospatial reasoning. In the experiments section, both of these methods are contrasted, and a technical discussion about them is provided.

7.4.3 Storyline merging with boundaries

In the storytelling process, a common occurrence are storylines that have no connections to other storylines. Put in other words, none of a story's entities have relationships with the entities of another story. In real life, however, lack of data support does not necessarily imply lack of entity connectivity, or by extension, lack of storyline relatedness. Some of these relationships are latent and important information can be gained if they are uncovered. For example, assume an ongoing investigation of a possible drug trafficking operation between PFox and MemetJ, as illustrated in Fig. 7.7. The figure shows that a single path linking those two entities does not exist because the part of the graph that involves PFox is fully disconnected from the part of the graph that includes MemetJ. As a consequence, there are no apparent storylines that can be drawn between them, which would complicate the analyst's job. To be able to link the two entities in question, the goal here is to operate on uncertainty, and surmise that a latent relationship exists between at least one entity from one graph to an entity in the other graph. If such a relationship can be devised, then the link between the desired entities can be established and investigated by the analyst.

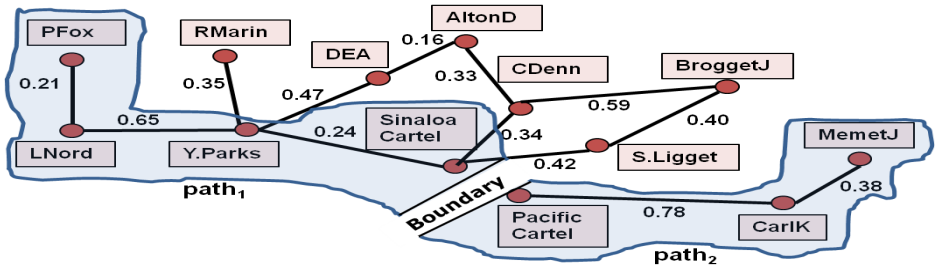


Figure 7.7: Storyline connecting entities PFox to MemetJ. Even though no true paths exist to link these two entities, a boundary bridges the Sinaloa Cartel to the Pacific Cartel, allowing $path_1$ to be merged with $path_2$, and creating a storyline from PFox to MemetJ.

One way to suggest latent relationships is to reuse the same parameters discussed previously, namely the entities' spatial distances and their times of observation. Using those parameters, the following approach is taken: a latent relationship can be discovered whenever two entities are located within spatial distance d and observed within a temporal distance t of each other. In Fig. 7.7, for instance, assume that the Sinaloa cartel operates in nearby areas to the Pacific cartel, say within 3 miles of each other, and that they run several of their drug deals around the same timeframes, for example, in March 2014. Due to their spatio-temporal proximity, a latent relationship is assumed to exist between them. In reality, such relationship may turn out to be not true, but this does not prevent investigating it (intelligence analysts often rely on seemingly unknown relationships). This would permit a join of $path_1$ to $path_2$ of Fig. 7.7, and generate the analyst's desired storyline from PFox to MemetJ. Because this latent relationship sits at the division between two disconnected entities, it is called a *boundary* to differentiate from the other truly observed relationships, and defined formally as:

Definition 22. *A boundary establishes a relationship between two disconnected entities e_i and e_j under two conditions: (1) $spatialDist(e_i,e_j) \leq d$; (2) $temporalDist(e_i,e_j) \leq t$. The function $spatialDist(e_i,e_j)$ outputs a spatial distance whereas $temporalDist(e_i,e_j)$ is a time difference function, and d and t are user-defined distance and time range thresholds, respectively.*

Since a boundary is a type of relationship, it must also have a binding strength. This value is calculated in the same manner as explained in Subsection 7.3.3. Recalling it, a semantic signature is derived for each of the boundary entities, and their *Euclidean distance* is computed. The only difference is that, since in this case no prior relationships exists between the entities in question, the value $rel_{e_i}^{r_k} = 0$ for both entities. In essence, boundaries represent a type of speculation that attempts to address uncertainty. As mentioned previously, there is no guarantee that establishing boundaries will find better storylines or uncover useful information. However, it does provide a valuable tool for exploratory analysis that goes beyond what the underlying data provides.

The above definition provides a way to establish boundaries, which may prove valuable, but raises another question. Because many entities may be located proximally to one another in space and time, the number of possible boundaries can be high and must be taken carefully (other optimizations can alleviate this problem, but are not discussed in the scope of this paper). Fig. 7.7, for example, shows one boundary between the Sinaloa cartel and the Pacific cartel. However, there could be others, such as the Sinaloa cartel and CarlK or Y.Parks and CarlK, as long as they meet the parameters of Def. 22. Since there is no clear way to determine which boundaries truly represent a real-life relationship, an initial assumption is made that they are all legitimate, investigate the possible storylines that they yield, and compare their semantic

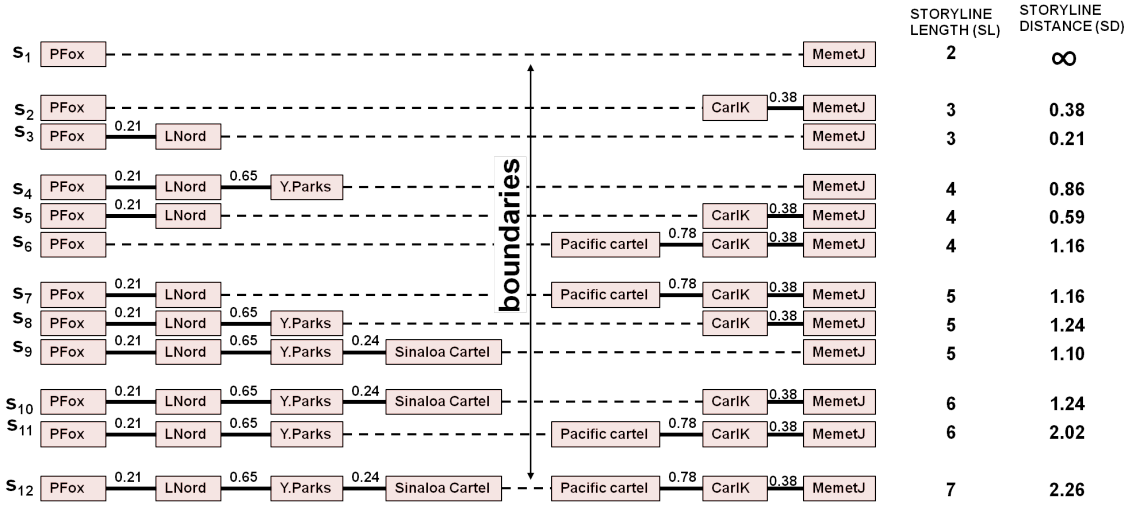


Figure 7.8: Example of 12 storylines connecting PFox to MemetJ based on Fig.7.7. The connections are only possible with the use of boundaries, represented by the dividing line. On the right, storyline length (SL) and storyline distance (SD) are specified for each sequence.

coherence as explained in Subsection 7.4.2. The top-k storylines of highest semantic coherence would then be accepted and the remaining ones rejected. A visual example follows.

Consider a task where the objective is to connect PFox with MemetJ according to Fig. 7.7. Assume that every entity in $path_1$ and $path_2$ are proximally close to one another according to Def. 22. Since no path exists between PFox and MemetJ, we need to identify their possible boundaries, which in this case involves all distinct pairs of entities in the corresponding shaded areas.

Fig. 7.8 shows a set of 12 storylines, denoted s_1 through s_{12}, generated from the entities in the shaded paths of Fig. 7.7. A solid line represents a true relationship with its binding strength, whereas a dashed line represents a boundary between the corresponding entities. Storyline s_{12}, for example, shows a possible path from PFox to MemetJ if we take into account a boundary between the Sinaloa cartel to the Pacific cartel. Alternatively, if we consider a boundary between Y.Parks and MemetJ, then storyline s_4 is the one that provides the desired connectivity. The 12 storylines represent every possible combination of boundary scenarios for this example. The figure also shows that the storylines vary in length from 2 to 7 entities and that distances range from 0.38 to 1.83 (note that for s_1, the sd=∞, since there is no binding strength between its entities). Using these values, the semantic coherence for each storyline is calculated, the storylines are ranked from highest to lowest, and the k ones of highest value are selected. Finally, a judgment can be made as to which storylines should be kept and which ones should be discarded based on their semantic coherence.

The above approach is formalized with Alg. 7, which takes as input the entities for which storylines are desired (entrypoint and endpoint), two disconnected sets of entities P and Q from the underlying graph, and two user-defined parameters: spatial distance and temporal difference that define a boundary. The algorithm outputs a set of storylines and their semantic coherence. In lines 1-5, several data structures are put into place to hold the final storylines, coherence values, identified boundaries, and temporary lengthwise and distancewise semantic coherences. The real work begins at $Step$ 1 where each pair of entities from the input sets P and Q (lines 6-7) are tested for their spatial and temporal distances (line 8). If the pair in question conforms to the user-defined thresholds d and t, a boundary has been identified, and it is added

Algorithm 7: Semantic Coherence Calculation Using Boundaries

inputs : entity entrypoint, entity endpoint, set of **ENTITIES P** = $\{p_1,...,p_n\}$ and set of **ENTITIES Q** = $\{q_1,...,q_n\}$ where $P \cap Q$=Ø, spatial distance d, temporal difference t
output: set of **S**TORYLINES and associated semantic coherences

Initialize

1: List($STORYLINES$, $semantic\_coherence_i$) = Ø ; // holds the output storylines and their calculated coherences
2: List $BOUNDARIES$ = Ø ; // data structure to hold the identified boundaries
3: Number lsc_i = 0 ; // temporary storage for lengthwise semantic coherence
4: Number dsc_i = 0 ; // temporary storage for distancewise semantic coherence
5: Number $semantic\_coherence_i$ = 0 ; // value of the combination of lsc_i and dsc_i

Step 1 - Identify the boundaries

6: **forall the** $p_i \in P$ **do**
7: **while** q_k *in* Q **do**
8: **if** $spatial\_distance(p_i,q_k) \leq d$ *and* $temporal\_distance(p_i,q_k) \leq t$ **then**
9: $BOUNDARIES \longleftarrow$ add(p_i,q_k) ;
10: **end**
11: **end**
12: **end**

Step 2 - Generate storylines with boundaries and calculate coherences

13: $STORYLINES \longleftarrow$ generate_storylines(entrypoint,endpoint, $BOUNDARIES$);
14: **foreach** s_i *in* $STORYLINES$ **do**
15: lsc_i = calculate_LSC(s_i) ;
16: dsc_i = calculate_DSC(s_i);
17: $semantic\_coherence_i = \frac{lsc_i+dsc_i}{2}$;
18: List($STORYLINES$, $semantic\_coherence_i$) $\longleftarrow$ s_i,$semantic\_coherence_i$;
19: **end**
20: **output** $STORYLINES$ in descending order of $semantic\_coherence_i$;

to the $BOUNDARIES$ set (line 9). With the boundaries in hand, the algorithm generates storylines, which is the task of connecting the entrypoint to the endpoint via a path that includes each boundary (line 13). Subsequently, each of the generated storylines is given a lengthwise and distancewise semantic coherence score (lines 15-16), the two of which are combined into a single score by average (line 17). Apart from average, other aggregate types may be substituted in, or the individual values can be used separately (in our experiments, we analyze them separately). Line 20 finally outputs each storyline along with its semantic coherence in sorted order.

7.5 Empirical Evaluation and Technical Discussion

Now that this study claims to have coherent storylines, it must be demonstrated that this coherence is applicable to everyday analytical tasks. The analytical task selected is event summarization. The intuition is the following: if the storylines generated from a set of documents are highly coherent for a specific set of themes, then the storylines should serve as a summary of those documents. In Subsection 7.5.1, the experiment details are presented. Subsection 7.5.2 shows that the generated storylines can outperform common event summarization techniques. Subsection 7.5.3 provides an in-depth analysis of boundaries and how they perform in terms of true positive and false positive storylines.

7.5.1 Experiment Setup

Current literature does not provide standards for what a good storyline should look like. Nor does it specify a baseline with which results can be compared. For this reason, coherence scores cannot be simply stated as good or bad. What can be done, however, is to generate storylines and verify if coherence translates to how well they perform in a given analytical task. Many tasks could be utilized, such as the clustering of

Table 7.1: Experiment details.

Experiment Specification	
Task 1	Storyline coherence on event summarization
Comparative Methods	*sts-lsc*, *sts-dsc*, *summ-text*, *summ-time*, *edcs-summ*
Dataset	Ukraine Political Crisis 2014
Number of Records	100,000 tweets
Measures	Precision / Recall
Task 2	Storyline coherence using boundaries
Dataset	Ukraine Political Crisis 2014
Number of Records	100,000 tweets
Measures	True Positives/False Positives and dsc/lsc

similar events or a search for related documents. For this study, event summarization was selected since summaries are intuitively a good way of capturing the main ideas of the underlying text, one of the goals of spatio-temporal storytelling. Boundaries were also explored to verify if they would result in true positive storylines or not. Table 7.1 provides a bird's-eye view of the experiment setup.

Data specification: The data source is related to the Ukraine political crisis of 2014, and spans the months of February, March, and April 2014. It was queried directly from *Twitter* through its API webservice. The querying process used specific keywords such as "protest" and "fight" along with place names such as "Moscow" or "Kiev". The nature of the data reflects items of interest to different communities such as intelligence, politics, law enforcement, and journalism. Some are civil protests and strikes, while others encompass more violent themes, such as attacks and shootings. Events of a non-violent nature were also included to make sure that the algorithms in question would be able to differentiate them as needed.

Comparative methods: For event summarization, the question to be answered was how well two variations of this proposed work (explained later) performed when compared to three other existing techniques. Currently, there is an extensive body of works related to text summarization ([95]) from where many options are available. The three selected methods encompass the following mix. The first approach, *SUMMALLTEXT* (denoted as *summ-text*) uses a variation of TF $\times$ IDF to compare tweets. It takes as input the tweet corpus (100,000 records), the set of words in the tweet corpus, and the desired number of output tweets. The second approach, *SUMMTIMEINT* (*summ-time*), uses a similar technique, but segments the tweets in different time windows and does processing based on each time window. Its inputs are the tweet corpus (100,000 records), the set of words in the tweet corpus, a minimum activity threshold (meaning that only segments with a minimum number of tweets are considered. We set this number to 1000 tweets), and the desired time segment (1 hour). They are described by [24]. Both output the top n tweets of maximum score according to their heuristics to represent summaries. The third approach, described by [87] and denoted as *edcs-summ*, identifies highly-frequent words in a tweet, builds a set of synonyms from them, and outputs the tweets for sets that are also highly frequent. They also accept the tweet corpus as input, and apply time segmentation for which 1 hour is set. Another reason we selected these methods is because they are truly event summarization methods, as opposed to textual summarization methods.

Performance Measures: The evaluation was done to find out which summarization approach would show the highest precision and recall based on a given input event and location. If high precision and recall could be observed, then the retrieved documents (in this case, tweets) should be able to provide a reasonable summary of the entities and locations present in the input query. For example, if the input event is "bombing in Kiev", how many output tweets of each approach contain that event and that location? And how many tweets that contain that event and location in the dataset are missed? For this purpose, we use traditional IR

Table 7.2: Explanation of performance measures.

Measure	Meaning
precision = $\frac{\lvert retrieved\ tweets \bigcap relevant\ tweets\rvert}{\lvert retrieved\ tweets\rvert}$	fraction of events correctly identified as relevant over all retrieved events (tweets).
recall = $\frac{\lvert retrieved\ tweets \bigcap relevant\ tweets\rvert}{relevant\ tweets}$	fraction of events correctly identified as relevant over all relevant events (tweets).
Relevant event: An event is deemed relevant if at least one record exists in the validation dataset that contains all keywords (or synonyms based on *Wordnet*) and locations or sublocations from the input query.	

Table 7.3: Comparison of precision and recall for five different approaches: *Spatio-temporal storytelling* (*sts-dsc* and *sts-lsc*), *summ-text*, *summ-time*, and *edcs-summ*. Each row is based on the results of one type of event and one location of interest. The highest values are shown in bold.

	sts-dsc		sts-lsc		summ-text		summ-time		edcs-summ	
Event	**p**	**r**	**p**	**r**	**p**	**r**	**p**	**r**	**p**	**r**
E1-assassination Kiev	**0.42**	0.54	0.31	0.57	0.40	**0.61**	0.34	0.58	0.41	0.48
E2-invasion Black Sea	0.52	0.66	0.54	0.70	**0.73**	0.65	0.37	**0.68**	0.67	0.62
E3-protest Russia	0.35	0.34	0.29	0.60	0.58	0.61	**0.71**	0.70	0.56	**0.62**
E4-attack Sevastopol	0.62	0.57	0.61	0.34	0.55	**0.65**	0.58	0.61	**0.68**	0.60
E5-confiscate Kharkiv	**0.70**	0.54	0.41	0.69	0.61	**0.70**	0.50	0.54	0.51	0.67
E6-fight Ukraine	**0.65**	**0.60**	0.34	0.59	0.64	0.51	0.39	0.46	0.40	0.59
E7-arrest Lviv	**0.65**	0.28	0.40	0.60	0.44	0.52	0.51	**0.62**	0.57	**0.64**
E8-explosion Donetsk	**0.63**	0.45	0.38	**0.71**	0.49	0.35	0.44	0.46	0.59	0.44
E9-occupation Simferopol	0.55	0.41	0.51	0.69	**0.64**	0.61	0.23	**0.70**	0.51	0.66
E10-blockade Crimea	**0.65**	0.33	0.30	0.62	0.59	0.38	0.22	**0.71**	0.50	0.62
Parameters	STS: 100,000 tweets, entrypoint = lat/lng of the query location, radius = 50 km, output storylines = 1000 Summ-Text: 100,000 tweets, word set of tweets, output tweets = 1000 Summ-Time: 100,000 tweets, word set of tweets, activity threshold = 1000 tweets, time segment = 1 hr, output tweets = 1000 EDCS-Summ: set of words in the 100,000 tweets, output tweets = 1000									
	p = precision, r = recall									

to define precision=$\frac{\lvert retrieved\ tweets \bigcap relevant\ tweets\rvert}{\lvert retrieved\ tweets\rvert}$ and recall=$\frac{\lvert retrieved\ tweets \bigcap relevant\ tweets\rvert}{relevant\ tweets}$. The definition of a **relevant tweet** is one that contains all keywords (or synonyms based on [143]) and all locations or sublocations of the input event. Note that an input can have one or more keywords along with one or more locations. For simplicity of discussion, the experiments displayed are limited to one of each at a time. Table 7.2 provides a quick view of the performance measures and definitions. For all experiments, no assumptions were made about data distribution, but areas of study were selected such that violent events were known to be of a high enough frequency for summarization to be plausible for all comparative methods.

7.5.2 Storyline Coherence On Event Summarization

In this subsection, the three event summarization approaches mentioned in the experiment setup were contrasted against two variations of this work, spatio-temporal storyline generation. The discussion is framed in terms of *precision* and *recall*, as specified in Table 7.2.

Table 7.3 lists a set of 10 event types, labeled $E1$ through $E10$, that were used as input to each of the five comparative methods: *sts-dsc* and *sts-lsc*, which are from this proposed work; *summ-text*, a cosine similarity variant of summaries; *summ-time*, a cosine variation with time-based segments; and *edcs-summ*, which uses segmentation applied to synonym sets. For the two variations of our approach, *sts-dsc* denotes that only storylines of highest *distancewise semantic coherence* were considered. For illustration purposes, only the top four are shown, but in the experiments the top-k = 5,000 results were investigated. The same is true for *sts-lsc* (*lengthwise coherence*) as well as for the other comparative methods. Each event is composed of a single keyword and the name of a location. The data used corresponds to 100,000 tweets related to the Ukraine political crisis of 2014. For each event type, the table shows precision and recall values using the five comparative methods as explained earlier. The highest values are shown in bold type.

The way to interpret the table, exemplified for row 1, is as follows. E1 was first taken (assassination Kiev) and storylines were generated over the 100,000 records of the dataset. That event and location became the entrypoint to our approaches (*sts-dsc* and *sts-lsc*). The set of generated storylines were then compared against the dataset to see how many tweets those storylines indeed summarized (i.e., contained an "assassination" keyword or a wordnet synonym of "assassination", and mentioned Kiev or an enclosed area of Kiev). Precision and recall were then computed. For the other approaches, the process was similar: first their output tweets were taken and then checked to see how many they truly summarized, using that information to compute precision and recall.

Discussion: At first glance, one can notice the fairly low levels of precision for *summ-time* for all event types, except for $E3$ (*protest Russia*). This approach clusters tweets based on time intervals, disregarding the clusters where events were not highly frequent. $E3$, on the other hand, was a very common occurrence of this event and place, which boosted its precision (time interval was set to one month). For the other approaches, this event's precision was considerably lower for different reasons: *sts* failed to capture "assassination" as a highly-connected entity according to its *semantic signature* features. It also failed to capture other synonyms, such as "killing". In addition, *sts-lsc*, *summ-text* and *edcs-summ* suffered because the word "Russia" was not always accompanied by "protest". For *summ-time*, the situation was more favorable in terms of recall, as relevant items were often retrieved with greater success.

For *summ-text*, precision appeared fairly stable across measurements, but with mixed signals. It was significantly high for $E2$ (invasion Black Sea) and $E9$ (occupation Simferopol), but decreased for $E1$ (assassination Kiev). The reason was due to this approach's reliance on keyword matching, for which "invasion" and "occupation" were very common, but "assassination" was not. This method showed one of the highest recalls on the table (0.70), which came for event $E5$ (confiscate Kharkiv). Overall, this method presented the best recall of the five approaches, which may be useful in domains where completeness is more important than preciseness.

The fourth technique, *edcs-summ*, is interesting because it uses a dictionary approach to identify events. Thus an "attack" can be expanded with "assault" or "aggression", among other terms. This feature explains the high precision under $E4$, but also serves to explain why this method did not do well under $E1$ (assassination Kiev) or $E6$ (fight Ukraine). While "assassination" and "fight" were expanded to other terms, these expanded terms failed to yield many matches in the dataset. Another interesting fact was that this method showed the least amount of variation between precision and recall, which may be attractive for applications in which both of these measures are important.

Inspecting Table 7.3, it can be seen that *sts-lsc* provided a stable trend of high recall, in general comparable to *summ-text* and *summ-time*. They are textual approaches, and thus this result is not surprising. *Sts-lsc*, however, showed poor precision for the most part. *Sts-dsc*, on the other hand, provided the 6 highest scores for precision. The reasons are twofold: the first one is that the events E1-E10 specified on the table showed high connectivity to many entities in the dataset. It implies that these events tended to receive high values in their *semantic signature*, which helped them bubble up to the top of the important entities. Thus, they tended to show up on the storylines. The second factor has to do with location. *sts* is a spatial technique in which places are regarded as geocodes (i.e., latitude and longitude coordinates), not plain keywords. Thus Ukraine covers any point of the country, while Donetsk represents any location within that city, and so forth. In essence, this had the effect of capturing a wider variation of events across many areas, regardless of how they were described in the dataset. These results are encouraging for three reasons: they reinforce

Table 7.4: Sample summaries (S1-S20) for the five comparative methods based on the query "violence Ukraine". The table shows that both *sts-dsc* and *sts-lsc* are able to identify a larger number of events and locations than the other approaches. Tweets range from March 25 to March 28, 2014.

Comparative **Methods**	**Summaries**	Events Identified	Locations Found
sts-dsc†	S1 - CORRUPTION spread KIEV found PRIORITY spread UKRAINE talks EU SUMMIT. S2 - POLICE confiscate CAMERA show RUSSIA step UKRAINE fund INSURGENCY. S3 - GROUP fight UKRAINE GOVERNMENT plan DEMONSTRATION organize DONETSK lend BANK. S4 - PARATROOPER attacked SEPARATIST greet PROTEST affect CRIMEA CURRENCY.	**6** (corruption, confiscate, insurgency, fight, demonstration, protest)	**5** (Ukraine, Kiev, Russia, Donetsk, Crimea)
sts-lsc†	S5 - KLITSCHKO pulls ELECTION takes UKRAINE registers PRESIDENT radicalize PARTY. S6 - ELECTION haste lead NEW MAIDAN erupt VIOLENCE faces UKRAINE S7 - UKRAINE commit CRIME react RUSSIA threaten INVASION disarm NATIONALISTS. S8 - RIGHT SECTOR give GUNS lodge UKRAINE plan DISARMAMENT follow AGREEMENT.	**4** (election, invasion, disarmament, agreement)	**2** (Ukraine, Russia)
summ-text	S9 - @PatDollard: problem solved: Obama Sends Biden To Fix Ukraine http://t.co/J9cCd0KZ5P #tcot #pjnet S10 - @JohnKerry The situation with Ukraine ultimately benefit only China, lose the rest of the world. S11 - @StoneMartyn: West self-congratulates for talking tough whilst #Russia prepares long term strategic operation in Ukraine. S12 - @JustinTrudeau @pmharper when not accompanied demonstrated poor judgement on #Ukraine and no relevant experience.	**1** (operation)	**3** (Ukraine, Russia, China)
summ-time	S13 - @PruStrategist: Stocks face near-term risks with Ukraine crisis, #China growth and credit concerns. S14 - @LowMaintainLife the only thread of violence to Ukraine today is Russia. S15 - @fast_ua support of international financial aid for Ukraine - Reuters http://t.co/no4K7d1Src. S16 - @benberdankweed All who are concerned about The Fed, China, Ukraine, etc. I genuinely believe cannabis sector is a hedge against these	**2** (crisis, financial aid)	**3** (Ukraine, China, Russia)
edcs-summ	S17 - @tatrg It seems, if Ukraine will decide to blockade Crimea, Russia will not be able to supply even vital things either. S18 - @alexnicest that whole Ukraine should be under an international aviation watch. S19 - @Tymchatyn: Russia Greets The Ukraine Crisis With a Shrug http://t.co/O1pnPaHpsA. pathetic? S20 - @shustry: Maybe you care what's going on in east Ukraine. But the folks in east Ukraine are sort of meh about it.	**3** (blockade, aviation watch, crisis)	**3** (Ukraine, Crimea, Russia)
†our approaches			

the importance of the spatial aspect which the other methods do not target; they indicate that the other methods could use the output of our approach (storylines) as the input to theirs in order to incorporate the spatial contribution; they confirm our initial claim that storylines can be a valueable tool in many different activities. In this case study, storylines outperformed the three other techniques of summarization in terms of precision.

Table 7.3 showed precision and recall results for single queries. Fig 7.9 provides a more comprehensive view over many random queries. Fig 7.9(a) shows how precision and recall varied as the number of generated summaries grew starting at 500 and ending at 5,000. The spatial radius was fixed at 100 km. The trend points to some common patterns. First, for all approaches, precision was consistently higher with lower numbers of summaries, which is not exactly surprising. More interestingly, however, is the fact that precision was low and approximately equal across all approaches when the number of generated storylines hit 5,000, which corresponds to the end of the graph. A drop in precision could be seen for all approaches when recall surpassed the 0.70 level. However, this drop was less pronounced for *sts-dsc*, which had its lowest point at approximately 0.39, whereas the other approaches went lower (0.20). The graph also shows that no significant difference in recall levels was apparent for all approaches.

Fig. 7.9(b) illustrates a similar plot, but fixed the number of generated summaries at 500 while varying the radius of study from 10 to 100 km. Again, the plot shows that *sts-dsc* had low variation in terms of precision, ranging from 0.59 down to 0.47. *Summ-text* was also stable on precision, but at a lower range (from 0.49 to 0.37). And while *summ-text* showed good recall values for specific queries, such as E5 in Table 7.3, its recall performance translated to lower values when many queries were utilized. In general, the results favored *sts-dsc* and *summ-time* with higher numbers of summaries (approximately 3,500) and longer radii (approximately 80 km). Beyond those points, *sts-lsc* and *summ-text* showed comparable performance. One point that can be learned from these results is that *sts-dsc* becomes more robust when the data provides a higher variation of spatial regions, which the other approaches appeared to be less able to capture.

The main goal of summarization is to capture essential ideas from the underlying text. To this end, Table 7.4 shows that storylines are able to coherently capture main topics of discussion as well as capture important

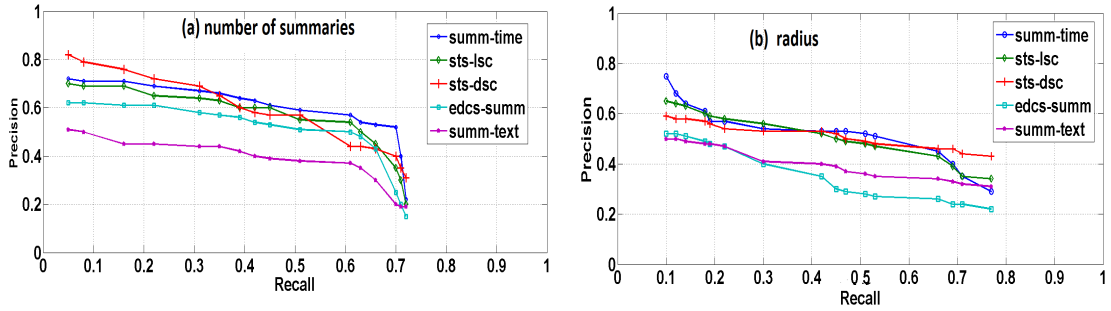


Figure 7.9: Precision and recall (p/r) shown for the four comparative approaches. In (a), p/r values are shown as the number of generated summaries are varied from 500 to 5000. In (b), the p/r values correspond to summaries generated when the radius of study grows from 10 to 100 km.

locations and events related to the input query. The table lists a set of 20 summaries (S1-S20), four for each of the comparative methods. Our approaches, *sts-dsc* and *sts-lsc*, show a total of 8 storylines (S1-S8) related to the input event "violence Ukraine". Notice that each summary is highly related to a real-world development, such as "elections in Ukraine", "Russia threatens invasion", "protest affect crimea", or "Ukraine funds insurgency". The other three approaches also speak of violence, as they also conform to the input query. However, their summarization content appears less coherent as they devolve the topic of violence into general statements such as "demonstrated poor judgement", "stocks face risk", or "I believe cannabis...". From an application perspective, these are not real-world facts, but mere observations that would arguably lend little knowledge to an analyst.

Since the focus of this study is spatio-temporal, it can also be seen that the two *sts* summaries were able to capture not only the locations of the input query, but also other nearby locations where similar events occurred, even when those events were not specified in the input query. The last two columns of Table 7.4 illustrate that both *sts-dsc* and *sts-lsc* were effective in identifying those events and locations. For example, in relation to the input query "violence Ukraine", while *sts-dsc* captured six related events (corruption, confiscate,insurgency, fight, demonstration, and protest), *sts-lsc* captured four (election, invasion, disarmament, and agreement). The next best approach, *edcs-summ* captured only three events (blockade, aviation watch, and crisis), while the remainder two, *summ-text* and *summ-time*, had only one and two respectively (operation, crisis and financial aid). The *sts-dsc* approach was also able to detect a wider range of locations (five in total), while the other approaches only showed a limited ability to do so. For example, *sts-dsc* found not only Ukraine (the input location), but also enclosed areas (e.g., Kiev, Crimea, and Donetsk) as well as neighborhing Russia. The other approaches only yielded Ukraine, Russia, Crimea, and China. The reason our approach can do a more effective identification of locations is the geocoding aspect. Unlike the other approaches which are textual by nature, *sts-dsc* does not rely on keyword frequencies or matches. Rather, it treats place names as entities tagged by latitude and longitude. In this manner, locations can be equated with its enclosed parts, such as the fact that Donetsk should be considered similar to Ukraine since the former is part of the latter, and should be included in the analysis.

7.5.3 Storyline Coherence Using Boundaries

Subsection 7.4.3 presented an algorithm to uncover relationships between disconnected entities, which was denoted as boundaries. These boundaries help in the suggestion of links that are not explicitly stated in the dataset, but which may exist in real life, and may be of importance. A classical example would be two

people believed to be involved in drug trafficking, even though no proof is available to substantiate their participation. In intelligence analysis, the lack of proof should not keep an investigator from pursuing the case further.

The problem with establishing boundaries is that entities must be investigated two at a time, but no clear-cut method exists to decide on which entities to choose. Connecting every pair combination could be cost prohibitive. To avoid connecting all entities and alleviate this problem, the spatial clustering approach of Subsection 7.4.1 was used in this part of the experiments. That method (Ripley's k function) finds the set of disconnected entities within a given radius, which was set to 50 km here. A maximal time difference of 24 hour was used, meaning that only entities observed within a 24-hour period of one other were considered. Boundaries were created and then storylines were generated. Subsequently, their *distancewise* and *length-wise semantic coherence* were calculated. For these generated storylines using boundaries, the goal was to verify whether they would turn out to represent a real-world development or not. A real-world development is one for which a news article that describes it can be found in the mainstream media. For example, if a storyline indicates that two politicians are involved in corruption via a boundary relationship (politician-1 $\overset{pays-off}{\longrightarrow}$ politician-2), and an an online newspaper reports this fact, then we can claim that the boundary turned out to be a legitimate link and the storyline is true. This verification process was done by searching *Google* using their search API. Each verification was a test in which the boundary counted as a true positive (TP) if the storyline could be verified, or a false positive (FP) otherwise. The Ukraine dataset used previously was again used here. Two measurements were sought: (1) the percentage of storylines that were TP; (2) the score ranges of *dsc* and *lsc* that translated to TP (i.e., the boundary turned out to be indeed a legitimate link).

Discussion: Table 7.5 shows five sample storylines (S1 - S5) that were generated, their corresponding *dsc*/*lsc* scores, and whether they were a TP or FP. The table also shows a news source and date which published an article that confirmed that the storyline was indeed true. In the storylines, boundary entities are shown in uppercase and boundary relationships, which were used as the themes of interest, are underlined. The weight for each theme was set to 1, meaning that all themes were considered equally important. This is the "w" parameter in Eq. 7.2.

For storyline S1 in the first row of the table, two entities ("CRIMEA" and "CATHOLICS") were connected via an "arrest" relationship. Note that part of this storyline is difficult to interpret, since we are not sure of the meaning of "scythian travel time". Nevertheless, our success came from the main theme of the storyline, which corresponds to the arrest of catholics, where the word "arrest" is underlined in the table. Indeed, this fact was confirmed by an article published in the catholicnews.com on Mar 25, 2014. Thus, this case was considered to be a true positive. The values of *dsc* and *lsc* (0.64 and 0.58, respectively) were not particularly high for this storyline. This was mainly because the entities "Crimea" and "catholics" were tagged with far-apart locations (albeit close in time), making their two coherence scores low.

The second storyline, S2, exemplifies a true negative. In this case, we attempted to link MVS (Ministry of Internal Affairs, Ukaine) to the #RUSSIAORTHODOX entity, which seems to refer to a religious group. Even after trying several themes, such as group, arrest, protect, control, attack (and others), we could not find any news article that could point us to any recent interaction between these entities. Since no confirmation of their interaction were made, the entities are linked by an unlabeled arrow in the table, and it was deemed to be a false positive. The values of *dsc* and *lsc* are not shown since a relationship could not be determined.

Table 7.5: Five example storylines (S1-S5) generated by the methods of Section 7.3 using boundaries. For each storyline, its *dsc* and *lsc* scores are shown along with a validation source and whether it is true positive or false positive (TP/FP).

	Storyline	dsc	lsc	validation	date	TP/FP
S1	ukrainians flee **CRIMEA** arrest **CATHOLICS** live scythian travel TIME.	0.64	0.58	catholicnews.com	Mar 25, 2014	TP
S2	#crimea are people engage **MVS** ⟶ **#RUSSIAORTHODOX** lead religion.	*	*	*	*	FP
S3	**FSB** kill **DEMONSTRATOR** prevail government give #aidukraine.	0.70	0.59	povesham.com	Mar 26, 2014	TP
S4	**KLITSCHKO** seized **RIVNE** join ternopil demand military summon capital return russia.	0.55	0.51	theguardian.com	Jan 24, 2014	TP
S5	#crimea arrest activists support journalists reveal **KREMLIN** invade **PLAN**.	0.80	0.71	dailymail.co.uk	Feb 25, 2014	TP

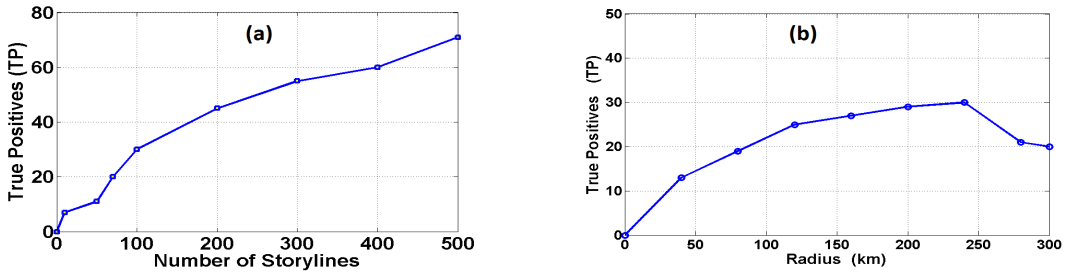


Figure 7.10: Progression of true positive (TP) results for two different parameters. (a) As the number of generated storylines grow, the number of TP remains fairly constant in the range of 14-16%. (b) The number of TP for the Ukraine dataset increases up to a spatial radius of approximately 240 km, after which a significant drop is observed.

The last three storylines, S3 through S5, appear more succint from a human-mind perspective. They tell of the security agency "FSB killing demonstrators", the seizing of "Rivne" by "Klitschko", a politician, and a possible "invasion of Crimea by the Kremlin". These statements could be verified by their corresponding validation news source specified in Table 7.5, and were all considered TP. The two coherence scores varied significantly for each storyline. It had the highest marks for S5 because this storyline was composed of entities that were very close to one another both spatially and in terms of timestamps. If one is interested in highly-coherent stories that are also spatially proximal, S5 is the best option and S3 the weakest of all.

In general, one interesting trend that cannot be seen by only looking at the table is the following: whenever the coherence scores drop below 0.4, it becomes challenging to confirm the story, even if manually. From one side, this is good news because it allows us to establish a correlation between coherence and true positive rates. We could, for example, use this 0.4 cutoff to prune out storylines whose coherence are lower, since they are difficult to verify as legitimate. From another side, this correlation is mostly based on observation (i.e., empirical) and not necessarily due to a theoretical justification. Thus, it must be taken carefully. Coherence is a factor of space, time, and relationship counts and types, and thus these factors must be studied in more depth, if that theoretical foundation is to be reached. It should also be strongly emphasized that our algorithms do not advocate for the truthfulness of the above storylines. In other words, we cannot state that an "invasion" will or will not occur. Rather, our algorithms uncover meaningful relationships among entities that are reported on *Twitter*, whether true or not. In intelligence analysis, they are important because they would give the investigator several directions which may be worth pursuing. We further raise two points about our experiments:

1. **Variations in true positives**: Fig. 7.10(a) shows how the number of true positives changed as an increasing number of storylines were generated. Noticeably, we find that approximately 16% of storylines that were forced through a boundary actually became a true positive. This growth appeared mostly linear (at least for this dataset), and indicates that extra storyline generation would be need-

eded if one desired extra TP cases to appear. While this number may seem low, the following must be considered: a boundary starts simply from the mere belief that two entities are interacting in an uncertain way. And in this sense, getting a 16% success rate from uncertainty is in fact very encouraging. Devising better methods of boundary identification can prove very valuable, and is one of our future directions.

2. **Changes in the radius**: Fig. 7.10(b) illustrates how the TP levels were affected when we considered different spatial regions of study. In this part, we considered the entrypoint to be an entity in downtown Kiev. Then we generated storylines with entities located in different radius increments of 50,100,...,300 km, using boundaries. The graph shows that the more the radius was increased the more TP cases were found. One may be tempted to think that a longer radius would tend to gather more entities, which might lead to more stories that could be confirmed. In fact, this did not happen. This is shown in graph (b) where the radius grows to approximately 240 km, after which the number of TP instances begins to drop. Increasing the radius of study often translates to higher processing costs, but not always to better results. This is one of the reasons this study targets storylines that are tightly bound with short spans of time and space, as they tend to yield higher coherence.

7.5.4 Experiment Summary

Overall, the experiments have successfully demonstrated that coherence provides significant benefits to spatio-temporal storytelling, and that it should be enforced as much as possible. Subsection 7.5.2 showed that *distancewise semantic coherence* can provide as much as an 18% increase in precision levels over existing summarization methods, while *lengthwise semantic coherence* is comparable in most situations. The use of spatial localization allows the algorithm to concentrate on entities that are closely-linked, and introspect events in which those entities participate. In this manner, the algorithm is better able to find locations where events take place. In general, our approach was able to find approximately 22% more locations and 9% more events than methods that rely on textual descriptions alone. In addition, our methods indicate that high coherence, whether distance or length-based, can be easily extended to other analytical tasks, such as document retrieval or pattern mining.

Subsection 7.5.3 described *boundaries* as another important use of spatio-temporal storytelling. By analysing seemingly disconnected entities, but which were spatially and temporally close, *STS* was able to uncover storylines not present in the dataset. Empirically, it has been observed that storylines whose coherence drops below 40% are unlikely to be a legitimate stream of information. While additional experimentation is needed on this number, it can have a significant impact on intelligence analysis, which operates on the following of leads that are highly uncertain. Establishing coherence thresholds can thus assist in pruning the dataspace to only the storylines that fall above the 40% coherence. The experiment results are promising on social media settings, but it should be noted that this work is generalizable to other environments where spatial modeling is center stage. Devising more resilient coherence scores provides a research direction that can make this framework even more robust.

7.6 Conclusion

This research proposed storyline coherence as a numerical measure of how convincing a stream of information is. By utilizing spatio-temporal storytelling, it devised a model with which entities are introspected for their relationships, spatial distances, and temporal differences to measure their binding strength. In addition, it introduced the concept of boundaries that can help uncover hidden relationships. These methods were successfully demonstrated on a case study related to the Ukraine political crisis of 2014. Storyline coherence is a significant success factor in intelligence analysis, giving us future motivation for alternative approaches to coherence models in specific application domains.

Chapter 8

Conclusion and Future Work

This chapter summarizes our contributions in determining similarity measures among spatial entities. In addition, several research directions have been identified and are discussed below as items of future work.

8.1 Contributions

This dissertation focuses on effective mechanisms to determine the similarity of entities scattered in space and constrained by time. Formal definitions were presented for various types of spatial entities, and a suite of algorithms were proposed to related them accurately based on contextual circumstances. The contributions of this research can be summarized in five parts: (1) spatial similarity in ontological spaces; (2) spatial similarity in categorical domains; (3) spatial similarity in entity networks; (4) spatial similarity in sequential data streams; (5) spatial similarity in semantic paths. A detailed summary is given below.

8.1.1 Spatial Similarity in Multidimensional Spaces

Descriptive representation of entities often lacks attributes, but includes hierarchies that associate elements by levels. These hierarchies provide descriptions of the entities, helping with their analysis when attributes are not available. This is the case with ontologies and similar tree structures. These hierarchies, however, make no assumptions of the types of relationships that elements can make when they reside in different parts of the hierarchical structure. They utilize no concept of distance or feature differences. For instance, two storms described in different weather reports cannot be easily compared to one another when their respective sources utilize disparate hierarchical representations. GIS systems have often tried to bridge their gap by analyzing spatial characteristics. Alternatively, textual systems introspect keyword features to find matches. This work proposed the concept of spatio-dimensional feature expansion that combines not only the spatial and textual dimensions of the data, but also their ontological similarity. Unlike other existing methods, this technique combines space and typed features with a numerical score of similarity for hierarchical categorization.

This thesis proposed similarity for entities that are proximally close, share the most characteristics, and belong to the same hierarchical level. This would be akin to finding not only things that are similar, but of the closest possible category. This approach was evaluated on a *CityGML* urban dataset of 50,000 buildings. The results showed that the algorithm was able to correctly group these buildings in different bins ("hotels", "churches", "houses", "apartments") without looking at labels, very close to a classification strategy, but simpler in nature. This contribution is significant because the proposed similarity computation is applicable in many domains, such as in clustering and pattern identification.

8.1.2 Spatial Similarity in Categorical Domains

In many application domains, spatial entities are described by categorical terms that are either non-deterministic or too subjective for automated analysis. The medical fields provide a classical example where drug names abound, but their similarity cannot be easily established without complex chemical analysis. A number of methods have been proposed to identify similarity for categorical data points. However, these methods suffer from several potential deficiencies: low frequency of data points, uneven distribution, and lack of attributes. The majority of these methods fall in two categories:

- **frequency-based approaches**: these methods relate entities based on the number of times each entity is observed in the dataset. [80, 44, 38]. Interpretation can be done with different variations. In some, infrequent values are deemed more relevant due to their rare nature. This is applicable in an outlier detection scenario. Alternatively, the similarity value is higher on frequent matches, while mismatches make it lower. A drawback of this interpretation is that many datasets contain more mismatches than matches, which has a swamping effect on the results. Nevertheless, this method has been successfully applied to document similarity.
- **ontology-based approach**: this class of methods observe where entities reside in the scope of a hierarchical structure, such as an ontology [17, 70]. Two entities that match each other in one or more dimensions have maximal similarity. When they have no dimensions in common, the similarity decreases based on the number of categories to which they could belong in the ontology. Such approaches tend to have better usage is small category sets where similarity is not punished by mere breadth and depth of the hierarchical structure.

In this study, we designed a method that combines the strength of both of the above approaches that addresses categorical similarity. It considers not only entity frequency, but also by takes into account the co-occurrence, and depth of the entity in the ontological tree. In the evaluation, we experimented in two different manners: (1) when entities co-occur in the same document; (2) when entities co-occur within the same spatial region based on different radii. The results demonstrated that even under low frequencies or uneven distribution, similarity translated to higher correlation in a dataset of 380,000 entities scattered in five U.S. cities and evaluated on a drug ontology of 170 levels.

8.1.3 Spatial Similarity in Graph Networks

Social networks have brought to surface the well-established effectiveness of graph structures. And while building a graph is not a problem per se, deciding on who to connect and how to connect them has been an

elusive task. The main problem is that entities can be linked in a theoretically unlimited number of ways (e.g., because they share the same attributes, belong in the same document, etc...), and deciding on these rules is neither intuitive nor practical. Existing methods of link analysis consider entity connectivity from different viewpoints:

- Node relevance: each entity is assigned an importance score based on the number of links that it receives from other nodes. One of the most popular approaches has been *PageRank* [18], which was designed to operate on web pages. Alternative approaches also consider links, but with different heuristics, such as restricting linkage according to a user query.
- Hub and authority: this technique comprises a family of approaches where authorities are pages that hold important information, and hubs are pages that direct users to the authorities. In this manner, a hierarchical structure establishes who is relevant and who is not. The *hits* algorithm [64] disseminated this type of strategy.

While the above items equate strength to high numbers of connections, they do not differentiate on the types of those connections. In spatial analysis, this often represents a pitfall because certain connections are more important than others. In this study, we implemented a novel technique, namely *ConceptRank*, which embeds relevance to each link. In this manner, an entity benefits from higher importance whenever its connections to other entities are deemed important, not simply on straight counts.

This proposed approach also provided a method to constraint the number of entities in order to alleviate high computational complexity. It applies *Ripley's K* function to find a spatial region where most entities reside. Only the entities in that region are then considered in the analysis. These methods were evaluated on two *Twitter* datasets of approximately 100,000 records each. It sought to determine if the connected entities represented a good summary of the underlying data. The results showed precision levels approximately 22% higher than other existing summarization methods.

8.1.4 Spatial Similarity in Sequential Data Streams

In the scope of this study, a data stream is also referred to as a 'storyline', which is a sequence of spatial entities connected by relationships in time order. The goal of these storylines is to reveal the actions or activities in which two entities participate. In semantic analysis, the number of storylines that can be written for a set of entities can be overwhelming. This problem motivated us to explore methods of combining storylines for more effective processing, which in turn required a numerical similarity measure.

The first contribution of this work extended the concept of *Dynamic Time Warping* with spatial distance so to constrain entities in small regions more appropriate for localized GIS use. In practice, this method takes as input two data sequences, and outputs a numerical distance between them. This distance is then used in hierarchical clustering to group the most similar sequences in groups that can be analyzed separately.

The second contribution relates entities and events to measure their level of influence on one another. A practical application would be to perform inferencing of facts, such as in determining whether 'the concentration of people' and 'the presence of police' implies a 'riot'. For this purpose, we devised three different methods of inferencing:

- **Distance-based Bayesian inference**: two data sequences are similar if the location of one is within a distance *d* of the other, and they share at least a minimum number *n* of entities. This definition relaxes traditional Bayesian inferencing by allowing probabilities to be computed on sequences of entities that are not the same. Relaxation is important because in many applications an exact sequence may never happen again (e.g., another Boston Marathon Bombing), but a similar event may (e.g., a murder during a Boston triathlon).

- **Spatial Forecasting Index**: this measure quantifies similarity in terms of influence: if area A influences area B, then whenever area A experiences a storyline, there exists another storyline that B will experience. For example, a riot in street A causes shops to close down in nearby streets B and C. In this approach, influence is a function of spatial distance, where storylines that are spatially close and share the most entities provide the most influence on each other.

- **Spatio-logical inference**: given a sequence of events and entities, this method combines the interactions between the entities to determine if an event is likely to happen or not. These interactions are based on logical rules, where each rule is embedded with a numerical belief in the range [0,1]. Similarity is then established by taking these numerical beliefs and computing a *distance to satisfaction*, that is, if the beliefs of the composing entities is lower than the belief of the target event, than the rule is satisfied, and the event is plausible.

The three methods above inject spatio-temporal components in the analysis so to account for GIS scenarios in which localization must be enforced. Most other approaches miss the identification of important relationships since they do not observe the evolution of entity similarity across different regions and along different timeframes. Experiments on *Twitter* data about social unrest in Mexico and *GDELT* data on Afghanistan wars found well-described storylines with high precision levels (up to 79%).

8.1.5 Spatial Similarity in Coherent Paths

When two spatial entities are connected through various other entities and relationships, the trajectory between them is called a semantic path. By extension, a semantic path becomes a coherent path if the relationships among their composing entities refer to the same concepts. For example, when two connected individuals (directly or indirectly) commonly have interactions that refer to diseases, then their path is coherent in a medical scenario. If their interactions stray in many directions (e.g., sports, finance), then their path is deemed less coherent for that same medical scenario. This notion is exploited in this study as a concept of *semantic coherence*. Very little research has been attempted to provide a numerical similarity for data sequences based on how coherent they are.

This study introduced two novel measures to determine how convincing a storyline is. The first one, *lengthwise semantic coherence*, finds a ratio of the number of entities that participate in a desired set of themes over all entities. This method is purely textual, very efficient in terms of computation cost, but limited in its ability to identify disparate locations. The second method, *distancewise semantic coherence*, calculates the spatial distance of the entities which participate in a desired set of themes over the sum of all distances between all entities. This approach is computationally more costly, but affords significant performance accuracy. Experiments evaluated coherence in terms of event summarization using a dataset of 100,000 tweets related to the Ukraine political crisis of 2014. Results showed that our proposed solution can outperform

existing approaches of event summarization by 19% (in terms of precision) and approximately 5% (in terms of recall).

8.2 Future Work

The current research work can be extended in several directions, including semantic analysis for relationship identification, statistical methods of data stream coherence, and unsupervised stream generation.

For semantic analysis of relationship identification, the goal is to find the most effective ways to connect a group of entities. For instance, connecting them by document co-occurrence, or by shared attributes, or by temporal proximity, etc... Making an excessive number of connections in intractable. Not connecting enough may miss important relationships. Therefore, methods that can propose an optimal compromise can be valuable. One potential solution could be *Hidden Markov Models*, which can detect latent states in entity connections, and may be able to present such ideal relationships. An alternative approach could be *Latent Dirichlet Allocation* (LDA), a generative topic model which may be used to generate relationships in an unsupervised manner. In its traditional form, however, LDA yields topics that are too general for specific domains, and must be modified to attend the needs of similarity analysis.

Data streams that are intelligible to the human mind are imperative for the success of similarity analysis. These data streams must display semantic coherence, one of the topics explored in this research. Beyond the methods that were presented, other techniques can be helpful in coherence analysis. One of them is *belief propagation*, a statistical method that uses prior distributions to determine the possible states of an entity in a data stream. In brief, the state of an entity is determined by the "opinions" or "beliefs" of its surrounding entities. In theory, this could be used to determine semantic similarity, which is the goal of this work.

Finally, this research can be enhanced with methods of storyline generation that are unsupervised. In the previously discussed approaches, the generation of storyline was dependent upon user inputs, such as an entrypoint and a radius of study. While input gives the user external control of the application, there are situations in which the system should be able to provide them in an optimal basis. Work in this research demonstrated that hierarchical clustering can be effective for this purpose. However, other clustering methods should be investigated. A good candidate could be density-based methods, since similarity analysis is often performed in spatial regions of high entity density. The benefits could come not only from performing analysis by cluster (i.e. higher efficiency), but also by separating the important entities from the irrelevant ones (i.e, better filtering).

Bibliography

[1] Dawn: Drug abuse warning network. US Dept. of Health and Human Services. http://www.samhsa.gov/data/DAWN.aspx - Last accessed July 01, 2012.

[2] Farmers market and local food marketing. US Dept. of Agriculture. http://www.ams.usda.gov/AMSv1.0/farmersmarkets - Last accessed June 25, 2012.

[3] M. R. Ackermann, J. Blömer, and C. Sohler. Clustering for metric and non-metric distance measures. In *Proc. of the 19th Annual ACM-SIAM Symposium on Discrete algorithms*, pages 799–808, 2008.

[4] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. On finding lowest common ancestors in trees. In *Proc. of the 5th Annual ACM Symposium on Theory of Computing (STOC)*, pages 253–265, 1973.

[5] Alchemy api, 2013. http://www.alchemyapi.com/ - Last accessed July 29, 2013.

[6] Apache. Xerces2 java parser. http://xerces.apache.org/xerces2-j/ - Last accessed July 20, 2012.

[7] D. Appelt, J. Hobbs, J. Bear, D. Israel, M. Kameyama, D. Martin, K. Myers, and M. Tyson. Sri intl. fastus system: Muc-6 test results and analysis. In *6th Conf. on Message Understanding*, MUC6 '95, pages 237–248, 1995.

[8] W. Bae, P. Vojtechovsky, S. Alkobaisi, S. Leutenegger, and S. Kim. An interactive framework for raster data spatial joins. In *Proc. of the 15th Symposium on Advances in Geographic Information Systems (ACM GIS)*, pages 4:1–4:8, Seattle, USA, 2007.

[9] D. Beech, A. Malhotra, and M. Rys. A formal data model and algebra for xml. http://infolab.stanford.edu/infoseminar.Archive/FallY99/malhotra-slides/malhotra.pdf - Last accessed July 20, 2012.

[10] C. Beeri, Y. Kanza, E. Safra, and Y. Sagiv. Object fusion in geographic information systems. In *Proc. of the 13th Intl. Conf. on Very Large Databases (VLDB)*, pages 816–827, 2004.

[11] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[12] P. Bernstein, S. Melnik, P. Michalis, and C. Quix. Industrial-strength schema matching. *SIGMOD Record*, 33(4):38–43, 2004.

[13] J. Bleiholder, S. Szott, M. Herschel, F. Kaufer, and F. Naumann. Subsumption and complementation as data fusion operators. In *Proc. of the Conf. on Extending Database Technology (EDBT)*, pages 513–524, 2010.

[14] A. Boedihardjo and C. T. Lu. Aoid: Adaptive on-line incident detection system. In *Proc. of the Intl. Conf. on Intelligent Transportation Systems*, pages 858–863, 2006.

[15] P. Bohannon, J. Freire, J. Haritsa, P. Roy, and J. Simeon. Legodb: Customizing relational storage for xml documents. In *Proc. of the 28th Intl. Conf. on Very Large Data Base*, pages 1091–1094, 2002.

[16] A. Bonifati and S. Ceri. Comparative analysis of five xml query languages. *SIGMOD Rec.*, 29(1):68–79, Mar. 2000.

[17] P. Bouquet, G. Kuper, M. Scoz, and S. Zanobini. Asking and answering semantic queries. In *Proc. of Meaning Coordination and Negotiation Workshop (MCNW'04) in conjunction with ISWC '04*, 2004.

[18] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30:107–117, 1998.

[19] B. Brodaric, S. Cox, J. Laxton, E. Boisvert, T. Duffy, B. Johnson, S. Richard, and B. Simons. Standardizing geologic data interchange: The cgi datamodel collaboration. In *GIS and Spatial Analysis IAMG Conf.*, 2005.

[20] J. Callan and T. Mitamura. Knowledge-based extraction of named entities. In *Intl. Conf. on Knowledge Management*, 2002.

[21] T. H. Cao. *Conceptual Graphs and Fuzzy Logic: A Fusion for Representing and Reasoning with Linguistic Information*. Springer, Boston, Massachusetts, 2010.

[22] J. Carvalho and A. Silva. Finding similar indentities among objects from multiple web sources. In *Proc. of the Intl. Workshop on Web Information and Data Management (WIDM)*, pages 90–94, 2003.

[23] A. Ceglar and J. F. Roddick. Association mining. *ACM Computing Surveys*, 38(2), July 2006.

[24] D. Chakrabarti and K. Punera. Event summarization using tweets. In *Proc. 6th AAAI Int. Conf. on Weblogs and Social Media*, 2011.

[25] D. Chamberlin, P. Fankhauser, M. Marchiori, and J. Robie. Xml query (xquery) requirements. http://www.w3.org/TR/xquery-requirements - Last accessed July 20, 2012.

[26] J. Chan, J. Bailey, and C. Leckie. Discovering correlated spatio-temporal changes in evolving graphs. *Knowledge and Information Systems*, 16:53–96, 2008.

[27] J. Chan, J. Bailey, and C. Leckie. Using graph partitioning to discover regions of correlated spatio-temporal change in evolving graphs. *Intelligent Data Analysis*, 13:755–793, 2009.

[28] Y. Chen, T. Suel, and A. Markowetz. Efficient query processing in geographic web search engines. In *Proc. of the ACM Intl. Conf. on Management of Data (ICMD)*, pages 277–288, Chicago, USA, 2006.

[29] CityGML. Virtual 3d city models. http://www.citygml.org/. Last accessed July 26, 2012.

[30] G. Cong, C. Jensen, and D. Wu. Efficient retrieval of the top-k most relevant spatial web objects. *Proc. of the VLDB Endowment*, 2(1):337–348, 2009.

[31] J. E. Córcoles and P. González. A specification of a spatial query language over gml. In *Proc. of the 9th ACM intl. symp. on Advances in geographic information systems*, ACM GIS, pages 112–117, 2001.

[32] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sept. 1995.

[33] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[34] D. Das and A. Martins. A survey on automatic text summarization, 2007.

[35] J. Davies, R. Studer, and P. Warren. *Semantic Web Technologies. Trends and Research in ontology-based systems*. Wiley, 2006.

[36] A. Doan, P. Domingos, and A. Halevy. Reconciling schemas of disparate data sources: A machine-learning approach. In *Proc. of the ACM SIGMOD Intl. Conf. on Management of Data*, pages 509–520, Santa Barbara CA, USA, 2001.

[37] M. Egenhofer and M. A. Rodriguez. Determining semantic similarity among entity classes from different ontologies. *IEEE Transactions on Knowledge and Data Engineering*, 15(2):442–456, March 2003.

[38] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo. A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. In *Applications of Data Mining in Computer Security*. Kluwer, 2002.

[39] W. Fan, M. Garofalakis, M. Xiong, and X. Jia. Composable xml integration grammars. In *Proc. of the 13th Intl. Conf. on Information and Knowledge Management (CIKM)*, pages 2–11, Washington D.C, USA, 2004.

[40] B. Fazzinga, S. Flesca, and A. Pugliese. Retrieving xml data from heterogeneous sources through vague querying. *ACM Transactions on Internet Technology*, 9(2):7:1–7:35, 2009.

[41] M. Fernández, D. Florescu, J. Kang, A. Levy, and D. Suciu. Catching the boat with strudel: experiences with a web-site management system. In *Proc. of the 1998 ACM SIGMOD intl. conf. on Management of data*, ACM SIGMOD, pages 414–425, 1998.

[42] B. George, J. Kang, and S. Shekhar. Spatio-temporal sensor graphs (stsg): A data model for the discovery of spatio-temporal patterns. *IDA*, 13:457–475, 2009.

[43] Y. Gong and X. Liu. Generic text summarization using relevance measure and latent semantic analysis. In *24th Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 19–25, 2001.

[44] D. Goodall. A new similarity index based on probability. *Biometrics*, 22(4):882–907, 1966.

[45] G. Groh, F. Straub, and B. Koster. Spatio-temporal small worlds for decentralized information retrieval in social networking. In *ACM GIS'12*, pages 418–421, 2012.

[46] Z. GuoDong, S. Jian, Z. Jie, and Z. Min. Exploring various knowledge in relation extraction. In *43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 427–434, 2005.

[47] P. Haase, R. Siebes, and F. van Harmelen. Peer selection in peer-to-peer networks with semantic topologies. In *Proc. of the Intl. Conf. on Semantics in a Networked World (ICNSW)*, volume 3226 of *LNCS*, pages 108–125, Paris, June 2004. Springer Verlag.

[48] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2001.

[49] M. S. Hossain, C. Andrews, N. Ramakrishnan, and C. North. Helping intelligence analysts make connections. In *Workshop on Scalable Integration of Analytics and Visualization*, AAAI '11, pages 22–31.

[50] M. S. Hossain, P. Butler, N. Ramakrishnan, and A. Boedihardjo. Storytelling in entity networks to support intelligence analysts. In *Proc. of the 2012 ACM Intl. Conf. on Knowledge Discovery and Data Mining*, KDD, 2012.

[51] M. S. Hossain, J. Gresock, Y. Edmonds, R. Helm, M. Potts, and N. Ramakrishnan. Connecting the dots between pubmed abstracts. *PLoS ONE*, 7(1), 2012.

[52] Y. Huang, Z. Lu, and H. Hu. A new permutation approach for distributed association rule mining. In *ACM Intl. Conf. on Information and Knowledge Management*, CIKM '05, pages 351–352, 2005.

[53] S. Hwang. Using formal ontology for integrated spatial data mining. *Computational Sciences and Its Applications (LNCS)*, 3044:1026–1035, 2004.

[54] Open source indicators program (osi).

[55] A. Islam, D. Inkpen, and I. Kiringa. Applications of corpus-based semantic similarity and word segmentation to database schema matching. *The Intl. Journal on Very Large Data Bases (VLDB)*, 17(5):1293–1320, 2008.

[56] H. Jiang, H. Lu, W. Wang, and J. X. Yu. Path materialization revisited: an efficient storage model for xml data. *Aust. Comput. Sci. Commun.*, 24(2):85–94, Jan. 2002.

[57] A. Josang, E. Gray, and M. Kinateder. Analysing topologies of transitive trust. In *Workshop on Formal Aspects in Security and Trust*, FAST, pages 9–22, 2003.

[58] N. Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *ACL 2004 on Interactive Poster and Demonstration Sessions*, ACLdemo '04, 2004.

[59] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Intl. Conf. on World Wide Web*, WWW '03, pages 640–651, 2003.

[60] T. Kane. *The New Oxford Guide to Writing*. Oxford Press, 1988.

[61] E. Keogh and M. Pazzani. Scaling up dynamic time warping for datamining applications. In *Knowledge Discovery in Data Mining (KDD '00)*, pages 285–289, 2000.

[62] L. Khan and Y. Rao. Web information management: a performance evaluation of storing xml data in relational database management systems. In *Proc. of the 3rd Intl. Workshop on Web Information and Data Management (WIDM)*, pages 31–38, ATlanta GA, USA, 2001.

[63] A. Kimmig, S. H. Bach, M. Broecheler, B. Huang, and L. Getoor. A short introduction to probabilistic soft logic. In *NIPS Workshop on Probabilistic Programming: Foundations and Applications*, 2012.

[64] J. Kleinberg. Authoritative sources in a hyperlinked environment. In *SIAM '98*, pages 668–677, 1998.

[65] Y. Kou, C.-T. Lu, and R. D. Santos. Spatial outlier detection: A graph-based approach. In *Proc. of the 19th Intl. Conf. on Tools with Artificial Intelligence (ICTAI)*, pages 281–288, Patras, Greece, 2007.

[66] V. Kreinovich and O. Kosheleva. Computational complexity of determining which statements about causality hold in different space-time models. *Theoretical Computer Science*, 405(1-2):50–63, Oct. 2008.

[67] D. Kumar, N. Ramakrishnan, R. F. Helm, and M. Potts. Algorithms for storytelling. *IEEE TKDE*, 20(6):736–751, 2008.

[68] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *18th Intl. Conf. on Machine Learning*, ICML '01, pages 282–289, 2001.

[69] R. Lake. Introduction to gml markup language. http://www.w3.org/Mobile/posdep/GMLIntroduction.html - Last accessed July 17, 2012.

[70] C. Leacock and M. Chodorow. Combining local context and wordnet similarity for word sense identification. In C. Fellfaum, editor, *MIT Press*, pages 265–283, Cambridge, Massachusetts, 1998.

[71] K. Leetaru and P. Schrodt. Gdelt: Global database of events, language, and tone, 1979-2012. In *Proc. Intl. Studies Assoc. Annual Conf. (ISA)*, 2013.

[72] L. Leitao, P. Calado, and M. Weis. Structure-based inference of xml similarity for fuzzy duplicate detection. In *Proc. of the 16th ACM Conf. on Information and Knowledge Management (CIKM)*, pages 293–302, 2007.

[73] V. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966.

[74] F. Li, J. Yang, and J. Wang. A transductive framework of distance metric learning by spectral dimensionality reduction. In *Proc. of the 24th Intl. Conf. on Machine Learning (ICML '07)*, pages 513–520, Corvallis, OR, December 2007.

[75] P. Li. Multiple relationship based deduplication. In *Proc. of the 4th SIGMOD Workshop on Innovative Database Research (IDAR)*, pages 25–30, Indianapolis, USA, 2010.

[76] R. Li, K. H. Lei, R. Khadiwala, and K. Chang. Tedas: A twitter-based event detection and analysis system. In *Proc. 28th IEEE Conf. on Data Engineering (ICDE)*, pages 1273–1276, 2012.

[77] S. Li, T. Wu, and W. M. Pottenger. Distributed higher order association rule mining using information extracted from textual data. *ACM SIGKDD Explorations Newsletter*, 7(1):26–35, June 2005.

[78] Z. Li, B. Wang, M. Li, and W.-Y. Ma. A probabilistic model for retrospective news event detection. In *ACM SIGIR Conf. on Research and Development in Information Retrieval*, SIGIR '05, pages 106–113, 2005.

[79] M. Lieberman and J. Sperling. Augmenting spatio-textual search with an infectious disease ontology. In *Workshop of the Intl Conf. on Data Engineering (ICDE)*, pages 266–269, 2008.

[80] D. Lin. An information-theoretic definition of similarity. In *Proc. of the 15th Intl. Conf. on Machine Learning (ICML '08)*, pages 296–304, San Francisco, CA, 1998.

[81] Lingpipe, 2013. http://alias-i.com/lingpipe/index.html - Last accessed February 15, 2013.

[82] C.-T. Lu, R. F. D. Santos, L. Sripada, and Y. Kou. Advances in gml for geospatial applications. *Geoinformatica*, 11:131–157, 2007.

[83] A. Luniewski, P. Schwarz, K. Shoens, J. Stamos, and J. Thomas. Information organization using rufus. In *Proc. of the 1993 ACM SIGMOD intl. conf. on Management of data*, ACM SIGMOD '93, pages 560–561, 1993.

[84] M. Marchiori. The quest for correct information on web: Hyper search engines. In *In WWW '97*, pages 1225–1235, 1997.

[85] A. Marcus, M. Bernstein, O. Badar, D. Karger, S. Madden, and R. Miller. Twitinfo: Aggregating and visualizing microblogs for event exploration. In *ACM Conf. on Human Factors in Computing Systems (CHI)*, 2011.

[86] J. McHugh, S. Abiteboul, R. Goldman, D. Quass, and J. Widom. Lore: A database management system for semistructured data. *SIGMOD Record*, 26:54–66, 1997.

[87] E. Medvet and A. Bartoli. Brand-related events detection, classification and summarization on twitter. In *Proc. of the The 2012 IEEE/WIC/ACM Intl. Joint Conf. on Web Intelligence and Intelligent Agent Technology*, WI-IAT '12, pages 297–302, 2012.

[88] G. Mendel. Experiments in plant hybridization. http://www.mendelweb.org/Mendel.html. Last accessed June 06, 2012.

[89] Microsoft. Msxml 4.0 sdk. http://msdn.microsoft.com/en-us/library/windows/desktop/ms763742(v=vs.85).aspx - Last accessed July 20, 2012.

[90] S. Miller, M. Crystal, H. Fox, L. Ramshaw, R. Schwartz, R. Stone, and R. Weischedel. Algorithms that learn to extract information: Bbn: Tipster phase iii. In *Proc. of a Workshop on Held at Baltimore, Maryland: October 13-15, 1998*, TIPSTER '98, pages 75–89, 1998.

[91] G. D. Mondo, M. Rodriguez, C. Claramunt, L. Bravo, and R. Thibaud. Modeling consistency of spatio-temporal graphs. *Data and Knowledge Eng.*, 84:59–80, 2013.

[92] L. Mui, M. Mohtashemi, and A. Halberstadt. A computational model of trust and reputation for e-businesses. In *35th Annual Hawaii Intl. Conf. on System Sciences (HICSS'02)-Volume 7 - Volume 7*, HICSS '02, pages 188–, 2002.

[93] D. Murray and J. Chow. An xml-driven data translation engine for gml 2. In *Proc. of the Urban and Regional Information Systems Association*, 2005. http://www.gisdevelopment.net/proceedings/gita/2003/innovt/innovt43pf.htm - Last Accessed July 20, 2012.

[94] NASA. Semantic web for earth and environmental ontology (sweet). http://sweet.jpl.nasa.gov/ontology/. Last accessed July 22, 2012.

[95] A. Nenkova and K. McKeown. Automatic summarization. *Foundations and Trends in Information Retrieval*, 5(2):103–233, 2011.

[96] OGC. Open geospatial consortium web feature service specification. http://www.opengeospatial.org/standards/wfs#downloads. Last accessed July 22, 2012.

[97] Y.-J. Oyang, C.-Y. Chen, and T.-W. Yang. A study on the hierarchical data clustering algorithm based on gravity theory. *Lecture Notes in Computer Science (LNCS)*, 2168:350–361, 2001.

[98] J. S. P. Mohan, S. Shekhar and J. Rogers. Cascading spatio-temporal pattern discovery. *Transactions on Knowledge and Data Engineering (TKDE)*, 24(11):1977–1992, 2012.

[99] J. Partyka, N. Alipanah, L. Khan, B. Thuraisingham, and S. Shekhar. Content-based ontology matching for gis datasets. In *Proc. of the 16th ACM SIGSPATIAL Intl. Conf. on Advances in geographic information systems*, GIS '08, pages 51:1–51:4, 2008.

[100] Z.-R. Peng and M.-H. Tsou. *Internet GIS: Distributed Geographic Information Services for the Internet and Wireless Network*. John Wiley and Son, New York, 2003.

[101] S. Petrovic, M. Osborne, R. McCreadie, C. Macdonald, I. Ounis, and L. Shrimpton. Can twitter replace newswire for breaking news? In *7th Intl. AAAI Conf. On Weblogs And Social Media (ICWSM)*, 2013.

[102] K. Radinsky, S. Davidovich, and S. Markovitch. Learning causality for news events prediction. In *World Wide Web Conf. (WWW)*, pages 909–918, 2012.

[103] K. Radinsky, S. Davidovich, and S. Markovitch. Learning to predict from textual data. *Journal of Artificial Intelligence Research (JAIR)*, 45:641–684, 2012.

[104] K. Radinsky and E. Horvitz. Mining the web to predict future events. In *WSDM '13*, pages 255–264, 2013.

[105] K. Radinsky and E. Horvitz. Mining the web to predict future events. In *Conf. on Web Search and Data Mining*, WSDM '13, pages 255–264, 2013.

[106] E. Rahm and P. Bernstein. A survey of approaches to automatic schema matching. *The Intl. Journal on Very Large Data Bases (VLDB)*, 10(4):334–350, 2001.

[107] E. Rahm, H. Do, and S. Massmann. Matching large xml schemas. *ACM SIGMOD Record*, 33(4):26–31, 2004.

[108] N. Ramakrishnan, D. Kumar, B. Mishra, M. Potts, and R. F. Helm. Turning cartwheels: an alternating algorithm for mining redescriptions. In *KDD '04*, pages 266–275, 2004.

[109] T. Reed and K. Gubbins. *Applied Statistical Mechanics: Thermodynamic and Transport Properties of Fluids*. Butterworth-Heinemann, Boston, Massachusetts, 1973.

[110] P. Rigaux, M. Scholl, and A. Voisard. *Spatial Databases with Application to GIS*. Morgan Kaufmann, San Mateo, CA, 2002.

[111] B. D. Ripley. *Statistical Inference for Spatial Processes*. Cambridge University, 1989.

[112] D. Roth and W.-T. Yih. Probabilistic reasoning for entity and relation recognition. In *Intl. Conf. on Computational Linguistics*, COLING '02, 2002.

[113] J. Sabater and C. Sierra. Reputation and social network analysis in multi-agent systems. In *Intl. Joint Conf. on Autonomous Agents and Multiagent Systems*, AAMAS '02, pages 475–482, 2002.

[114] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: Real-time event detection by social sensors. In *Proc. of the 19th Intl. Conf. on World Wide Web*, WWW '10, pages 851–860, 2010.

[115] R. D. Santos, A. Boedihardjo, and C.-T. Lu. Towards ontological similarity for spatial hierarchies. In *ACM GIS Workshop on Querying and Mining Uncertain Spatio-Temporal Data*, QUeST, pages 26–33, 2012.

[116] R. D. Santos, S. Shah, F. Chen, A. Boedihardjo, P. Butler, C.-T. Lu, and N. Ramakrhishnan. Spatio-temporal storytelling on Twitter. Technical report, Virginia Tech, 10 2013. http://vtechworks.lib.vt.edu/handle/10919/24701.

[117] M. Schillo, P. Funk, I. Stadtwald, and M. Rovatsos. Using trust for detecting deceitful agents in artificial societies. *Applied Artificial Intelligence*, 14:825–848, 2000.

[118] A. Schmidt, M. Kersten, M. Windhouwer, and F. Waas. Efficient relational storage and retrieval of xml documents. In *In ACM SIGMOD Workshop on the Web and Databases (WebDB)*, pages 47–52, 2000.

[119] V. Seghal, L. Getoor, and P. Viechnicki. Entity resolution in geospatial data integration. In *Proc. of the 14th Intl. Symp. on Adv. in Geographic Information Systems (ACM GIS)*, pages 83–90, 2006.

[120] D. Shahaf and C. Guestrin. Connecting the dots between news articles. In *KDD'10*, pages 623–632, 2010.

[121] D. Shahaf, C. Guestrin, and E. Horvitz. Metro maps of science. In *KDD'12*, pages 1122–1130, 2012.

[122] D. Shahaf, C. Guestrin, and E. Horvitz. Trains of thought: Generating information maps. In *WWW'12*, pages 899–908, 2012.

[123] S. Shekhar and S. Chawla. *Spatial Databases: A Tour*. Prentice Hall, 2003.

[124] S. Shekhar, R. R. Vatsavai, N. Sahay, T. E. Burk, and S. Lime. Wms and gml based interoperable web mapping system. In *Proc. of the 9th ACM intl. symp. on Advances in geographic information systems*, ACM GIS, pages 106–111, 2001.

[125] M. Skounakis, M. Craven, and S. Ray. Hierarchical hidden markov models for information extraction. In *Proc. of the 18th Intl. Joint Conf. on Artificial Intelligence*, IJCAI'03, pages 427–433, 2003.

[126] S. Song, K. Hwang, R. Zhou, and Y.-K. Kwok. Trusted p2p transactions with fuzzy reputation aggregation. *IEEE Internet Computing*, 9(6):24–34, Nov. 2005.

[127] S. Staab, B. Bhargava, L. Lilien, M. Winslett, M. Sloman, T. Dillon, E. Chang, F. Hussein, W. Nejdl, D. Olmedilla, and V. Kashyap. The pudding of trust: Managing the dynamic nature of trust. *IEEE Intelligent Systems*, 19(5):74–88, 2004.

[128] Stanford ner, 2013. http://nlp.stanford.edu/software/CRF-NER.shtml - Last accessed February 20, 2013.

[129] A. Tagarelli and S. Greco. Toward semantic xml clustering. In *Proc. of the 6th SIAM Intl. Conf. on Data Mining (SDM), year = 2006, pages = 188–199*.

[130] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison Wesley, 2005.

[131] S. Thakkar, C. Knoblock, and J. Ambite. Quality-driven geospatial data integration. In *Proc. of the 15th Intl. Symp. on Adv. in Geographic Information Systems (ACM GIS)*, pages 16:1–16:8, Seattle WA, USA, 2007.

[132] H. Thompson and R. Tobin. Current status of xsv. http://www.ltg.ed.ac.uk/ ht/xsv-status.html - Last accessed July 20, 2012.

[133] M.-C. Tseng and W.-Y. Lin. Efficient mining of generalized association rules with non-uniform minimum support. *Data and Knowledge Engineering*, 62(1):41–64, July 2007.

[134] S. Turner. *The Creative Process: A Computer Model of Storytelling and Creativity*. Psychology Press, 1994.

[135] R. Vatsavai. Gml-ql: a spatial query language specification for gml. http://www.cobblestoneconcepts.com/ucgis2summer2002/vatsavai/vatsavai.htm - Last accessed July 20, 2012.

[136] K. N. Vavliakis, A. L. Symeonidis, and P. A. Mitkas. Event identification in web social media through named entity recognition and topic modeling. *Data and Knowledge Engineering*, 88:1–24, Nov. 2013.

[137] W3C. Document object model. http://www.w3.org/DOM/faq.html - Last accessed July 20, 2012.

[138] M. Walther and M. Kaisser. Geo-spatial event detection in the twitter stream. In *Advances in Information Retrieval*, volume 7814 of *Lecture Notes in Computer Science*, pages 356–367. Springer Berlin Heidelberg, 2013.

[139] B. Wang and X. Wang. Spatial entropy-based clustering for mining data with spatial correlation. In *Proc. of the 15th Pacific-Asia Conf. on Adv. in knowledge discovery and data mining*, PAKDD'11, pages 196–208, 2011.

[140] K. Weinberger and L. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, December 2009.

[141] WHO. International classification of diseases. World Health Organization. http://www.who.int/classifications/icd/en/ , Last accessed May 25, 2012.

[142] R. Wishart, R. Robinson, J. Indulska, and A. Jøsang. Superstringrep: Reputation-enhanced service discovery. In *Australasian Conf. on Computer Science - Volume 38*, ACSC '05, pages 49–57, 2005.

[143] Wordnet: A lexical database of english, 2012. http://wordnet.princeton.edu/. Last accessed July 22, 2012.

[144] L. Xiong and L. Liu. A reputation-based trust model for peer-to-peer ecommerce communities [extended abstract]. In *Proc. of the 4th ACM Conf. on Electronic Commerce*, EC '03, pages 228–229, 2003.

[145] L. Xiong and L. Liu. Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Trans. on Knowl. and Data Eng.*, 16(7):843–857, July 2004.

[146] Y. Xu and Y. Papakonstantinou. Efficient keyword search for smallest lcas in xml databases. In *Proc. of the 2005 ACM SIGMOD Intl. Conf. on Management of data*, SIGMOD, pages 527–538, 2005.

[147] I. Yaniv and E. Kleinberger. Advice taking in decision making: Egocentric discounting and reputation formation. *Organizational Behavior and Human Decision Processes*, 83(2):260–281, 2000.

[148] M. Yoshikawa and T. Amagasa. Xrel: a path-based approach to storage and retrieval of xml documents using relational databases. *ACM Trans. Internet Technol.*, 1(1):110–141, Aug. 2001.

[149] P. Yu, M. Singh, and K. Sycara. Developing trust in large-scale peer-to-peer systems. In *Symposium on Multi-agent Security and Survivability*, pages 1–10, 2004.

[150] D. Zelenko, C. Aone, and A. Richardella. Kernel methods for relation extraction. In *Proc. of the ACL-02 Conf. on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 71–78. Association for Computational Linguistics, 2002.

[151] Y. Zhang and Y. Fang. A fine-grained reputation system for reliable service selection in peer-to-peer networks. *IEEE Transactions on Parallel and Distributed Systems*, 18(8):1134–1145, 2007.

[152] R. Zhou and K. Hwang. Powertrust: A robust and scalable reputation system for trusted peer-to-peer computing. *IEEE Trans. Parallel Distrib. Syst.*, 18(4):460–473, Apr. 2007.