

# Personalized Recommendation for Online Social Networks Information: Personal Preferences and Location Based Community Trends

Shaymaa Khater

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

Denis Gračanin, Chair  
Hicham G. Elmongui, Co-Chair  
Andrea L. Kavanaugh  
James D. Ivory  
Kristina Lerman

9 November 2015  
Blacksburg, VA

Keywords: Social Networks, Microblogs, Recommendation Systems

Copyright 2015, Shaymaa Khater

# Personalized Recommendation for Online Social Networks Information: Personal Preferences and Location Based Community Trends

Shaymaa Khater

(ABSTRACT)

Online social networks are experiencing an explosive growth in recent years in both the number of users and the amount of information shared. The users join these social networks to connect with each other, share, find content and disseminate information by sending short text messages in near realtime. As a result of the growth of social networks, the users are often experiencing information overload since they interact with many other users and read ever increasing content volume. Thus, finding the “matching” users and content is one of the key challenges for social networks sites. Recommendation systems have been proposed to help users cope with information overload by predicting the items that a user may be interested in.

The users’ preferences are shaped by personal interests. At the same time, users are affected by their surroundings, as determined by their geographically located communities. Accordingly, our approach takes into account both personal interests and local communities. We first propose a new dynamic recommendation system model that provides better customized content to the user. That is, the model provides the user with the most important tweets according to his individual interests. We then analyze how changes in the surrounding environment can affect the user’s experience. Specifically, we study how changes in the geographical community preferences can affect the individual user’s interests. These community preferences are generally reflected in the localized trending topics. Consequently, we present TrendFusion, an innovative model that analyzes the trends propagation, predicts the localized diffusion of trends in social networks and recommends the most interesting trends to the user. Our performance evaluation demonstrate the effectiveness of the proposed recommendation system and shows that it improves the precision and recall of identifying important tweets by up to 36% and 80%, respectively. Results also show that TrendFusion accurately predicts places in which a trend will appear, with 98% recall and 80% precision.

*To my great parents.*

*To my wonderful husband, Ahmad.*

*To my lovely kids, Ali and Abdullah.*

# Acknowledgments

First and foremost, I thank Allah for the numerous blessings He has bestowed upon me throughout my dissertation journey.

I would like to express my deepest gratitude to my advisor, Professor Denis Gracanin, for his excellent guidance and persistent support during my PhD study. Prof. Gracanin provided me the freedom through my Ph.D. to explore various research problems where he always gave stimulating and fruitful discussions. I have always enjoyed our weekly meeting in which he has helped me in all the aspects of this research. I have also benefited tremendously from his feedback, ideas, and criticism on my research, writing, and presentation skills.

I am deeply indebted to my Co-advisor, Professor Hicham Elmongui. From the first days of my PhD program in Egypt, his moral support, and continuous encouragement have been of extreme value for helping me getting through my PhD studies. He inspired me with his wisdom, hard work, and attention to details. He was always there to help and to give sincere advice whenever I need, not only on the academic level, but on the personal level as well. I am forever grateful for his support.

In addition, I would like to thank Professor Andrea Kavanaugh, Professor James Ivory, and Professor Kristina Lerman for serving in my Ph.D. committee and providing constructive feedbacks on my dissertation.

My deepest gratitude goes to my husband, Ahmad, and to our kids, Ali and Abdullah. Ahmad's unlimited support, encouragement and understanding was always the incentive to complete my

study. He was always supportive for me in the tough times, and was always keen to prioritize our family over himself. Without him, it would have been impossible for me to finish the five years of this long journey. I am grateful to my children, Ali and Abdullah. They have been told “Not right now, mommy’s working” often – too often. I am grateful for them for being understandable when I was so tired or busy to give them attention. Their smiles always give me the light and the power to achieve.

I am thankful to my parents, who spent endless efforts, all over the years, providing me with all the love, help, and care. They always gave without return, and their sacrifices are everlasting.

# Contents

<b>Contents</b>	<b>vi</b>
<b>List of Figures</b>	<b>xi</b>
<b>List of Tables</b>	<b>xiv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Recommendation Systems . . . . .	2
1.2 Recommendation Systems in Social Networks . . . . .	4
1.3 Challenges in Recommendation Systems . . . . .	5
1.4 The User as a Part Of Different Communities . . . . .	6
1.5 Twitter: A Case Study . . . . .	7
1.6 Trending Topics in Twitter . . . . .	9
1.7 Motivation . . . . .	10
1.8 Proposed Approach . . . . .	12
1.9 Dissertation Organization . . . . .	14

<b>2</b>	<b>Related Work</b>	<b>15</b>
2.1	Recommendation Systems in Online Social Networks . . . . .	16
2.1.1	Categorization by Approach . . . . .	17
2.1.2	Categorization by Objective . . . . .	21
2.1.3	Other Recommendation Systems . . . . .	25
2.2	Information and Influence Propagation in Social Networks . . . . .	27
2.3	Trending Topics in Social Networks . . . . .	28
2.4	Topic Modeling . . . . .	31
2.5	Event Detection . . . . .	32
<b>3</b>	<b>Problem Description</b>	<b>35</b>
<b>4</b>	<b>Tweets Analysis Subsystem</b>	<b>39</b>
4.1	Introduction . . . . .	39
4.2	Problem Description . . . . .	40
4.3	Topic Extraction . . . . .	42
4.4	Tweets Pooling . . . . .	42
4.5	Dynamic Level of Interest . . . . .	43
4.6	Personalized Tweet Recommendation . . . . .	45
4.6.1	Personalized Social Features . . . . .	46
4.6.2	Explicit Features . . . . .	46
4.7	Experimental Results . . . . .	47

4.7.1	Data Collection . . . . .	47
4.7.2	Dataset and Preprocessing . . . . .	50
4.7.3	Tweets Pooling . . . . .	51
4.7.4	Evaluating Topic Models . . . . .	52
4.7.5	Calculating the Dynamic Level of Interest . . . . .	54
4.7.6	Personalized Recommender Model . . . . .	54
4.7.7	Dynamic LoI and Other Features Effect . . . . .	55
4.8	Discussion . . . . .	55
4.8.1	Tweets Pooling Effect . . . . .	58
4.8.2	Number of Topics Variation Effect . . . . .	60
4.9	Summary . . . . .	61
<b>5</b>	<b>Trends Analysis Subsystem</b>	<b>63</b>
5.1	Introduction . . . . .	63
5.2	TrendFusion Framework . . . . .	64
5.3	TrendFusion Model . . . . .	65
5.4	Generating the Hazard Rate Graph . . . . .	66
5.5	TrendFusion Stages . . . . .	67
5.5.1	Stage 1: Collect and Store Trending Topics Stream from Locations . . . . .	68
5.5.2	Stage 2: Build Cascades . . . . .	68
5.5.3	Stage 3: Extract Parameters . . . . .	68
5.5.4	Stage 4: Model Learning/Using . . . . .	72



5.6	Snowball Cascade Model . . . . .	72
5.6.1	SC Model Definition . . . . .	74
5.6.2	GT Model Definition . . . . .	75
5.7	Evaluation . . . . .	75
5.7.1	Trending Topics Dataset . . . . .	75
5.7.2	Using TrendFusion Framework . . . . .	78
5.7.3	Experiments . . . . .	79
5.7.4	Results and Discussion . . . . .	81
5.8	Summary . . . . .	87
<b>6</b>	<b>Personalized Recommendation</b>	<b>88</b>
6.1	Introduction . . . . .	88
6.2	Multilevel Trends Filtering . . . . .	89
6.3	Tweets and Trends Recommendation . . . . .	90
6.4	Effect of Topic Modeling on Recommendation . . . . .	92
6.5	TrendFusion System . . . . .	92
6.6	Summary . . . . .	95
<b>7</b>	<b>TrendFusion User Study</b>	<b>97</b>
7.1	Purpose . . . . .	97
7.2	Method . . . . .	97
7.2.1	Participants . . . . .	97

7.2.2	Procedure . . . . .	98
7.3	Users Tasks . . . . .	98
7.4	Results and Discussion . . . . .	99
<b>8</b>	<b>Conclusion</b>	<b>102</b>
	<b>Bibliography</b>	<b>106</b>
	<b>Appendix A User Study</b>	<b>122</b>
A.1	TrendFusion System Setup . . . . .	122
A.2	User Questionnaire . . . . .	126
A.2.1	Part I: General usability questions . . . . .	126
A.2.2	Part II: TrendFusion application usability . . . . .	129
	<b>Appendix B Questionnaire Results</b>	<b>131</b>

# List of Figures

1.1	Interaction between recommendation systems and social media . . . . .	4
1.2	Interaction between social network user and different communities . . . . .	7
1.3	Twitter homepage when a user login with the message stream . . . . .	8
1.4	Sources of information on Twitter . . . . .	8
1.5	Trendfusion interaction with sources of information on Twitter . . . . .	10
1.6	The General Framework . . . . .	12
2.1	Content-based recommendation . . . . .	18
2.2	Collaborative based recommendation . . . . .	19
2.3	Example of trending topics in Twitter . . . . .	28
4.1	Tweets analysis structure . . . . .	41
4.2	Timeline window for the user . . . . .	51
4.3	Perplexity for LDA and our model . . . . .	52
4.4	User dynamic level of interest in topics . . . . .	54
4.5	Average precision and recall for the three classifiers . . . . .	56

4.6	Average gain in precision and recall with including Dynamic LOI feature . . . . .	57
4.7	Classification Analysis . . . . .	59
4.8	Number of topics variation effect effect . . . . .	61
5.1	An information cascade represented by a Directed Acyclic Graph (DAG). . . . .	65
5.2	The stages of TrendFusion model. . . . .	67
5.3	Time tracking of trends' appearances in locations $i, j$ . . . . .	71
5.4	Steps of the Snowball and General Threshold model . . . . .	74
5.5	10 Major US cities according to population . . . . .	76
5.6	Histogram of the distances between the 48 cities . . . . .	77
5.7	Average precision and recall for TrendFusion and GT models considering cascade steps and all cascades respectively . . . . .	80
5.8	Rank of each parameter used in the classification process . . . . .	84
5.9	Predicted trends analysis . . . . .	86
6.1	Trendfusion sources of information . . . . .	89
6.2	Multilevel trends filtering . . . . .	90
6.3	The general framework . . . . .	91
6.4	Average precision and recall for TrendFusion with and without adding topics, with considering distance features only and the General Threshold model . . . . .	93
6.5	TrendFusion system . . . . .	96
7.1	Range bars representing precision and recall values as reported by TrendFusion users	100

7.2	Range bars representing the ease of interface use and location variation as reported by users (lower value is better). . . . .	101
A.1	Register a new TrendFusion user . . . . .	123
A.2	Authorize TrendFusion user . . . . .	124
A.3	Check user timeline . . . . .	124
A.4	User login . . . . .	125

# List of Tables

1.1	Recommendation systems approaches . . . . .	3
4.1	Twitter APIs description and rate limit . . . . .	49
4.2	Characteristics of different pooling scheme . . . . .	52
4.3	Example for top ten words for five topics . . . . .	53
5.1	WOEIDs of 10 major US cities . . . . .	77
5.2	Geo Bounding Boxes of 10 major US cities . . . . .	78

# Chapter 1

## Introduction

Online social networks are experiencing an explosive growth in recent years in both the number of users and the amount of information shared. Through these message streams, the users can connect with each other, share, find content and disseminate information. Some of these sites provide social links (e.g. Facebook, LinkedIn, MySpace). Others are used to share content (e.g. Youtube, Flickr). Understanding users' behavior in these sites is one of the important research challenges.

Unfortunately, the explosion of information does not necessarily improve the quality of our life. With limited human attention, finding relevant information and knowledge from the huge amount of available information can be frustrating and extremely time-consuming. Because of this information explosion, the users became overwhelmed with the huge amount of information they have to follow, and hence they are spending a lot of time and effort to get just the information they are interested in. Sometimes, the inability to make clear and accurate decisions could even increase people's stress level.

It was also stated by the European Network and Information Security Agency [57] and Pro-Con.org [82] that the risks of using online social networks are, but not limited to:

1. Social media enables the spread of unreliable and false information. It was found that false

- rumors related to crisis events spread so quickly on Facebook and Twitter.
2. Online social networks are associated with developing addictive behavior; if not managed properly it will result in a decline in the user's productivity.
  3. Online social networks may lead to social networking spam; that is the propagation of unsolicited messages.
  4. Social networking sites encourage people to waste time. According to a survey described in [37], 36% of people surveyed listed social networking as the biggest waste of time. Users consume 20 to 25 minutes on average to return to their original task. When alerted to a new post or message on Twitter or Facebook it could take around two hours to fully return attention to the original task.

Thus, it becomes crucial to have an intelligent system that is able to learn user preferences, and based on these preferences, to automatically filter irrelevant information or suggest useful information to this user in a timely manner. Under these circumstances, recommendation systems have been proposed as a key tool to overcome information overload.

## 1.1 Recommendation Systems

The technology of recommendation systems lies at the convergence of multiple areas such as cognitive science [103], approximation theory, information retrieval [107], and relates to management science and marketing. These recommendation systems had then emerged as an independent research area in the mid-1990s when researchers started focusing on recommendation problems that depends on the ratings structure. This recommendation problems were then reduced to the problem of estimating ratings for the items that have not been seen by the user.

Different online applications have used recommendation systems to help users find resources and save them from the information overload problem. Usually, a recommendation system predicts



Table 1.1: Recommendation systems approaches

<b>Approach</b>	<b>Basic idea</b>	<b>Limitations</b>
Content-Based	recommend items similar in contents to the previous choices of the user	content analysis over-specialization
Collaborative Filtering	recommend items to the users based on other users' recommendations who have similar preferences	new item problem new user problem

the users' preferences by mining their profiles, previous behaviors and social connections. For example, when viewing a product on Amazon.com, other products are recommended if the other users buy them with the product being viewed.

During the past decade, different models of recommendation systems had evolved. These recommendation systems can be classified into two main categories. The first category classifies the recommendation systems by the objective of the recommendation, which can include locations, users, activities, or social media. The second category classifies the recommendation systems by the methodologies employed, including content-based, collaborative filtering-based or hybrid methodologies. In content-based filtering, the recommendations are based on the item itself rather than the preferences of the users. However, these techniques suffer from the overspecialization and cold start problems.

In collaborative filtering, recommendations are made by measuring similarities between users preferences. The limitation for this method is its dependency on the amount of ratings present. This raises again the problem of cold start. It is difficult to produce an accurate recommendation for a new user who has very few or no ratings at all.

The third category is the hybrid approach. These methods combine both the content and collaborative filtering approaches to make better recommendations. This is done by trying to avoid the drawbacks in content-based and collaborative filtering approach exclusively. Table 1.1 represents a summary for the different approaches for the recommendation systems.

## 1.2 Recommendation Systems in Social Networks

In our daily life, we used to rely on recommendations from friends and relatives to choose the best item to buy. Nowadays, we often use Internet to make buying decisions. However, when using Internet we usually see many available products with nearly the same characteristics thus making it difficult to make a decision.

For these reasons, social networks became an important source for generating recommendations. Using social networks to understand the relations between users and their friends as well as the information obtained about them can improve the knowledge about users' behaviors and ratings. Also, integrating recommendation systems into social networks can provide new observations and thus decisions that cannot be achieved through using traditional recommendation systems [55]. Figure 1.1 shows the interaction between the recommendation systems and the social media.

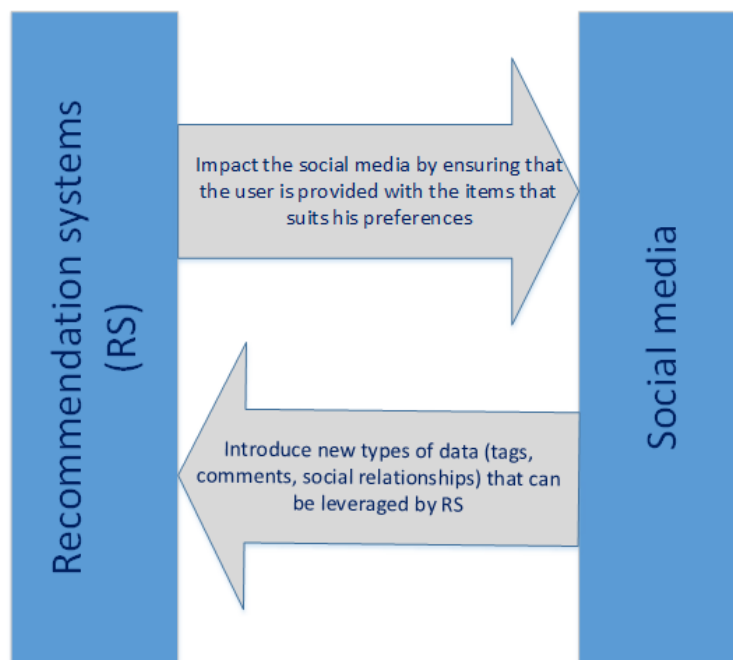


Figure 1.1: Interaction between recommendation systems and social media

Research studies have also found that different properties of social networks encourage the integration of recommendation systems with social networks. These studies are varied and address

areas such as network structure, trust, information credibility, event detection, social tagging, etc. The recommendation systems in these studies aim at returning items that are similar to the users' preferences. Most of these studies were conducted mainly on studying the social network structure and recommending friends to the user based on the similarities of interests [53, 74]. However, these studies didn't consider the personnel user's preferences. A part of our work studies how to capture the change of user's interests over time, and how to make recommendations based on these interests.

### 1.3 Challenges in Recommendation Systems

Recommendation systems try to help users to deal with information overload and to provide personalized recommendations, contents, and services. However, It is rather challenging when the numbers of the users and items are large. Many researchers had addressed the key challenges on recommendation systems including cold-start problems, user's preferences modeling, personalized recommendation and so on. Other challenges include:

1. **Content analysis and data sparsity:** The algorithms for the recommendation systems suffer from the ability to measure item similarity. Content-based methods depends on explicit item descriptions. However, such descriptions may be difficult to obtain for abstract items like ideas or opinions.
2. **Trust issues in recommendation systems:** Although users always prefer recommendation made from trusted friends rather than recommendations made by strangers [112], most of the recommendation techniques make recommendations to the user mainly based on other users' preferences. These users have similar rating data with the target user, regardless of the trust between these users.
3. **Challenges to adapting the recommendation system to the dynamical aspects of users and items:** The dynamic nature exists in both users and items. From the user's perspective,

the user's preferences or interests change over time. The sensitivity of the item with respect to time also changes. Some items are time-sensitive and expire quickly. Other items are of continuous interest, such as classic story books. Recommendation systems should be able to adapt to these dynamic factors and make effective dynamic recommendations.

Some approaches had been proposed to tackle these problems. Some of these approaches used the user's preferences or ratings, along with the correlation with their friends to design a better recommendation system [55]. Other approaches had used the hybrid systems that combine collaborative and content-based methods, to avoid limitations of content-based and collaborative systems [33, 110]. For example, in Twitter social networks, Hong *et al.* [59] had used the content features, temporal features, publishers features and tweets features to predict the popularity of messages measured by the number of future retweets, and hence recommend Tweets of interest to the user. Yet recommendation systems in online social networks still face many challenges.

## 1.4 The User as a Part Of Different Communities

The term 'Community' first appeared in the book "Gemeinschaft und Gesellschaft" published in 1887 [119]. There is no unique definition of community which is widely accepted in social networks. A variety of definitions of community have been proposed according to different sides. In general, a community can be regarded as a social unit of any size that defines a group of individuals that share common characteristics, beliefs, values, needs, etc. Classically, communities were determined by geographical boundaries. Currently, online social media networks has also created virtual environments for establishing online virtual communities.

Geography determines our local communities and plays an important role in various aspects of our lives. As Tobler's first law of geography states: "Everything is related to everything else, but near things are more related than distant things" [118]. An individual is usually considered a part of his local geographical community. Moreover, in addition to the local community to which a person belongs, a user of online social media networks is also considered as a part of an online



Figure 1.2: Interaction between social network user and different communities

community. Generally, an individual human being is subject to and is influenced by a different set of ideologies promoted in his/her different communities, as shown in Figure 1.2.

As people spend more time online, data regarding the two dimensions — geography and social relationships — are becoming increasingly precise allowing us to build reliable models to describe their interactions. On the other hand, as the ‘virtual’ distance between users has dramatically decreased, research shows that geographical locality still matters in our choice of friends [120], as well as topical interests [58].

## 1.5 Twitter: A Case Study

In our experiments, we used Twitter social networks as our case study. Twitter is one of the most popular online microblogging social network launched in July 2006, with over 316 million monthly users and more than 500 million postings per day as of 2015 [1,43]. Twitter poses a question to its users, “what’s happening?”, and the answer to this question is restricted to 140 characters called

tweets. In terms of social connectivity, Twitter allows a user to follow any number of other Twitter users, called friends. When the user first login to his homepage, he sees a list of tweets from the logged-in user's friends. The messages are displayed as a "stream" on the user's Twitter page. Figure 1.3 shows the twitter homepage for a logged in user, with the message stream from his friends.



Figure 1.3: Twitter homepage when a user login with the message stream

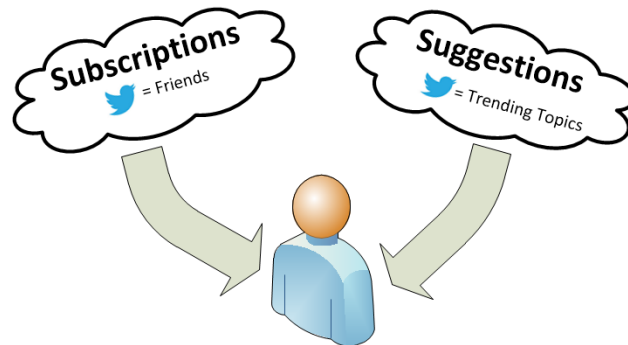


Figure 1.4: Sources of information on Twitter

As in Figure 1.4, Twitter users receive information feeds, either by:

1. **Subscriptions:** A subscription is related to the updates that the user request from his online community (i.e by following friends, groups). Twitter allows users to post and consume Twitter messages or “tweets” generated by other users. A user can then reference another users in their tweets by appending the @ symbol (called mention symbol) to the other user’s username. This creates a link from their message to the referenced user’s account. A retweet is a message from one user that is ‘forwarded’ by a second user to the second user’s followers, commonly using the ‘RT @username’ text as prefix to credit the original poster. In addition to posting tweets, users can also interact with the stream of tweets they are receiving in their timeline by replying (commenting on a tweet posted earlier by self or others), retweeting (resending an earlier tweet posted by other to followers, giving credit to the original publisher) or favoriting (liking an earlier tweet) the tweets.
2. **Suggestions:** Used by the social media to send the users information that they might be interested in. One of the important suggestions by the social media is the trending topics that are suggested to the user based on geographic location.

## 1.6 Trending Topics in Twitter

As the nature of the user’s posts in Twitter is quick and transient, Twitter became an information system that provides a ‘real time’ reflection of the interests and thoughts of its users, as well as their attention. As a consequence, Twitter serves as a rich source for exploring the mass attention of millions of its users, reflected in ‘trends’ that can be extracted from the site.

A trending topic on Twitter is a word, phrase or topic that is posted multiple times. The trends appearing on Twitter are the terms that occur with the highest frequency in the tweets. These trends become popular either through a concerted effort by users, or because of an event that prompts people to talk about one specific topic. They can be Twitter memes, local or global events, or tweets related to the celebrity. A Twitter meme is a phrase or a sequence of words representing an emergent topics which spreads quickly through Twitter as a hash-tag, and then disappear after

few days. These memes are not necessarily related to any real-world event or news. When tweets from celebrities are reposted by a large number of users who follows them, this reposting causes the terms in a celebrity tweet to become a trending topic.

A part of our research about the relations between geographical locations based on the user activity focuses on the trending topics appearing in each geographical location, and how they can be used to detect rules that relates different geographical locations.

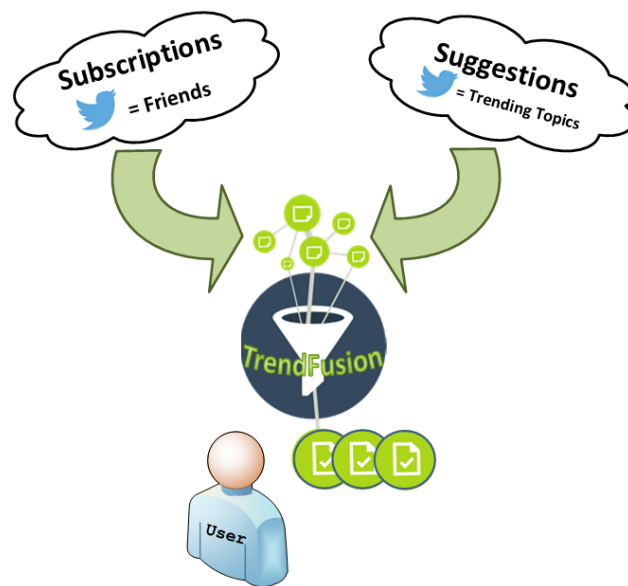


Figure 1.5: Trendfusion interaction with sources of information on Twitter

## 1.7 Motivation

The online social networks face different challenges in providing the streams of information to the user. While tweets may contain valuable information, many of them are not of interest to the users. A large number of tweets can overwhelm users since they interact with many other users and they have to read ever increasing content volume on their timeline [115]. Thus, the difficulty in recommending content that are of interest to users became a key challenge for social networks sites.



One of the important challenges in recommending the suitable personalized contents to the user in online social networks is the ability to adapt to the dynamical aspects of the user and items. The value of item content may change over time. For instance, some items are time-sensitive and expire quickly with a life-time as short as several hours, such as breaking news. Some items are of continuous interest for a long time for many people, such as classic technical papers that continue to be referenced decades after their initial publication. Furthermore, people's intentions are usually different. Some users tend to get updated information, so they will mostly read breaking news. Others look for technical information, and they tend to refer to long-term documents. People's interests also evolve over time. One person who was interested in a certain topic one year ago may not care about the same topic today. Time also plays an important role in collaborative filtering. People who at one point had similar tastes that at another more recent point diverged should receive recommendations based on a new set of preferences, such as classic technical papers.

Moreover, the user's preferences are also affected by their surrounding environments, which are determined by their geographically located communities. Another challenge was that traditional social network analysis mainly studies network structure and properties without the consideration of geographical distance between nodes. Although the idea of 'Death of Distance' proposed in 2011 [25] claims that geographical distance plays a less important role due to the communication revolution and the rapid development of the Internet, which could make of our world a 'global village', studies on spatial structure of networks demonstrated that there is a strong correlation between geographical attributes and network properties, indicating the significance of considering the spatial properties of networks for future applications [44]. Researchers have further studied the distinctions between online and offline social networks [34], and discovered that geographical property does play important roles when constructing the social connection between two users especially in explaining their preferences. Another challenge was to study how the change in the geographical community preferences can affect the individual user interests. These community preferences are generally reflected in the localized trending topics.

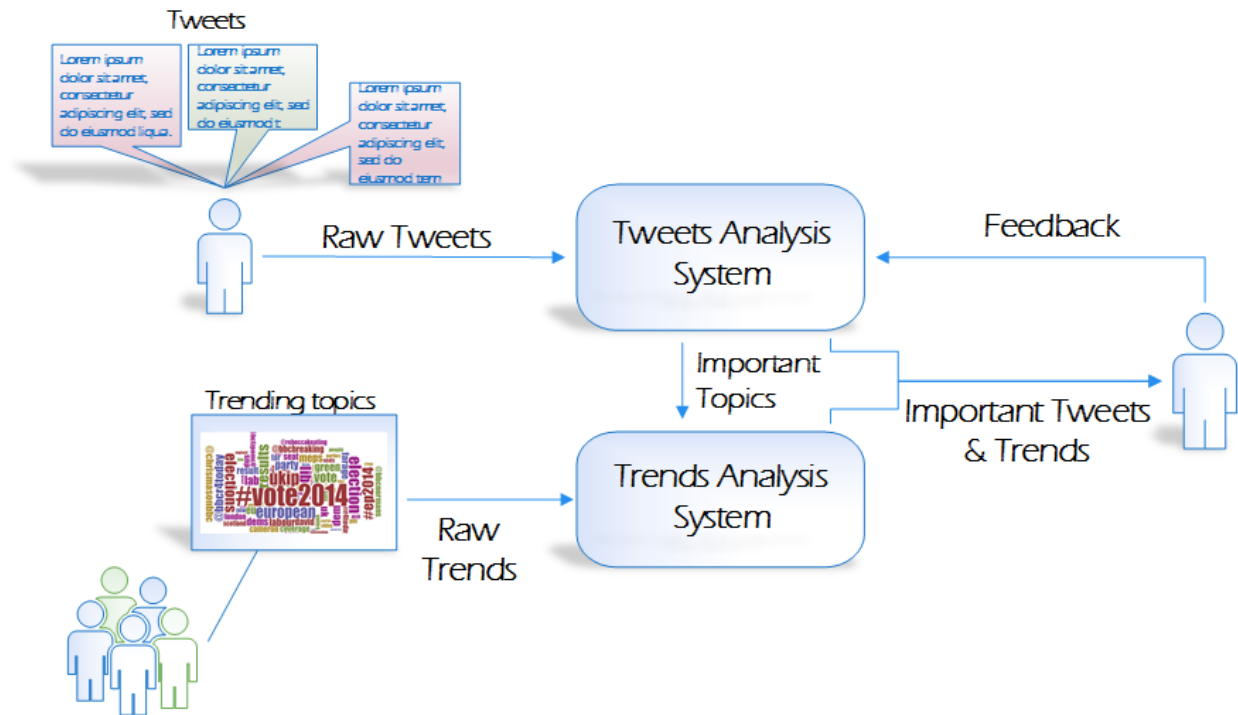


Figure 1.6: The General Framework

## 1.8 Proposed Approach

Our research objective is to develop a conceptual model and the corresponding framework to help the social media users to quickly find the information they are interested in. This is done by taking into consideration the different sources of information feeds to the user. Figure 1.5 illustrates the usage of different information feeds in our model. Our goal is to provide the user with personalized recommendation for online social network information. This is based on both the individual and geolocated community levels. The general structure is shown in Figure 1.6.

On the individual level, the proposed approach provides better subscriptions view for the user (tweets analysis system in Figure 1.6). Accordingly, we propose a new model of dynamic personalized recommendation system that provides the user with the most important tweets. The proposed model captures the user's interests, which change over the time, and shows the messages that correspond to such dynamic interests.

In addition to this, and in order to fully customize the user experience, we expanded our research to analyze how the change in environment can affect user's experience. Specifically, we study how the change in the geographical community preferences can affect the individual user preferences. Our research focuses on improving the suggestions provided by the social media to the users (trends analysis system in Figure 1.6). Our assumption is that enhancing the trending topics suggested by the social media to user based on their location will reflect positively on their online experience. We approached this point by investigating the interplay between local community interests and public trends. Consequently, we developed a model for predicting localized trends diffusion from one localized community of users to other geographically separated communities of users. We show that observing the local trends in some locations (e.g., cities), can be used to predict where these trends will appear next. Finally, the interesting topics for the user discovered by the tweets analysis system are then used by the trends analysis system to personalize the trends suggested to the user.

The most important aspect of our model is prediction of trends that will appear in a location, before even users in that location start mentioning that topic. This is extremely useful in many cases, such as building a proactive localized recommendation system for topics or for early prediction of social events (e.g., protests).

Our contributions are as follows:

1. The notion of dynamic Level of Interest (LoI) for the recommendation made to the Twitter user, in which we build a user specific time variant (dynamic) level of interest graphs for each topic constituting the tweets. This is based on utilizing the weights of topics in the user's tweets to determine its level of importance to the user.
2. A model that incorporates the dynamic change in users' interests in tweets topics, along with other social features and tweets related features to recommend interesting tweets for the user.
3. A new information diffusion model (*Snowball Cascade Model*) in online social networks that is suitable to model the diffusion between geographically separated communities, rather

than relying on the users' social network structure.

4. TrendFusion, a predictive model that can predict whether the trending topics will appear in a certain city in the future, along with the activeness time, i.e., the time it will appear.
5. A web application that recommends to a user the most interesting tweets according to past interests, along with predicting the trending topics and their related tweets that will appear in a location selected by the user.
6. An evaluation of the TrendFusion performance and its quality in terms of recommendation and prediction using a user study.

## 1.9 Dissertation Organization

In Chapter 2, we introduce the background of recommendation systems, and discuss the different approaches for the recommendation systems in online social networks, the different topic modeling techniques, event detection techniques, trend detection and information credibility. Chapter 3 presents the motivation for the proposed research. We then presents our model for personalized dynamic recommendation model in Chapter 4. Moreover, A complete description of the TrendFusion model and its components will be described in Chapter 5. Chapter 6 describes our approach towards the personalized recommendation through integrating tweets and trends recommendation systems. A description of TrendFusion web system is also included. Description for the usability study conducted to measure the performance of the TrendFusion system, along with the observations and findings are described in Chapter 7. Finally, conclusions and future work are provided in Chapter 8.

# Chapter 2

## Related Work

In this chapter, we provide an overview of the related research. The main objective of our research is to provide the user with personalized recommendation for online social network information. Accordingly, we first give an overview of different recommendation systems techniques in online social networks. Then, since our work focuses on the interplay between personal interests and public trends, we study the information propagation process in social networks and trending topics in geolocated communities.

Another two directions that are orthogonal to our work are:

- Topics modeling and extractions.
- Global or localized event detection.

Though the scope our research does not cover the real time event detection, different approaches and methods learned in that field can benefit our work. We also relied heavily on topics modeling from social media. Thus the last two subsections in this chapter are covering event detection and topic modeling.

## 2.1 Recommendation Systems in Online Social Networks

Recommendation systems had first emerged as an independent research area in the mid-1990s when researchers started focusing on recommendation problems that depends on the ratings structure. This recommendation problems were then reduced to the problem of estimating ratings for the items that have not been seen by a user [8]. Intuitively, this estimation is usually based on the ratings given by this user to other items. Once we can estimate ratings for the yet unrated items, we can recommend to the user the item(s) with the highest estimated rating(s).

Moreover, the recent popularity of online social network sites and the overwhelming amount of information available today made it difficult for users to find useful information. As a solution to this problem, many recommendation systems were introduced to help users find interesting information.

These recommendation systems can either be general (non personalized) or personalized [66]. The former methods do not consider the characteristics and preferences of the customers, whereas the latter tightly depends on the user profile. An example of non-personalized recommendation method is to return top ten songs of the current month. In order to create this kind of recommendation the statistical methods are commonly used where the recommendation is based on statistical factors like average or summary ratings [109].

The personalized recommendation is based either on the demographic information about users or on the analysis of the past behavior of the user and his social relationships in order to predict their future behavior (collaborative and content based filtering) [109].

We categorize the recommendation systems into two categories. The first category classifies the recommendation systems by the objective of the recommendation, which can include recommending locations, users, activities, or social media. The second category classifies the recommendation systems by the methodologies employed, including content-based, collaborative filtering-based or hybrid methodologies. The following sections will describe each category in detail, along with the research efforts done in this field.

## 2.1.1 Categorization by Approach

The major methodologies used by the recommendation systems can be categorized into the following three groups: 1) content-based filtering, 2) collaborative filtering, and 3) the hybrid approach. Collaborative filtering and content-based approaches are often used in personalized recommendation.

### 2.1.1.1 Content-Based Filtering

In content-based filtering, the items that are recommended are similar to what a user liked in the past. The recommendation in these systems are based on the item itself rather than the preferences of other users. To select such items, content-based filtering measures item-to-item similarity by analyzing the content of textual information of the items. This includes the keywords representing the users characteristics (age, gender, location, etc) and items characteristics (product, price, appearance, etc). For example, to recommend a movie  $m$  to user  $u$ , the content-based recommendation system will get the previously rated movies by user  $u$  and then the movies with highest similarity to the user preferences are recommended. Figure 2.1 represents the content-based recommendation approach.

Different techniques were used to measure the user's preferences and the candidate item's characteristics, including cosine similarity measure, Bayesian classifier, decision trees and so on. They are often used to recommend items containing textual information, such as books, web sites and news. However the limitations of these techniques are:

1. **Analysis of Content problem:** It is difficult to extract and analyze the features of multimedia content such as audio, image and video in order to measure the user's and item's contents. Another problem with the content analysis is that, if two different items are represented by the same set of features, they are indistinguishable. Since each item is represented by the most important keyword, content-based systems cannot distinguish between a good item and a bad one, if they use the same terms [111].

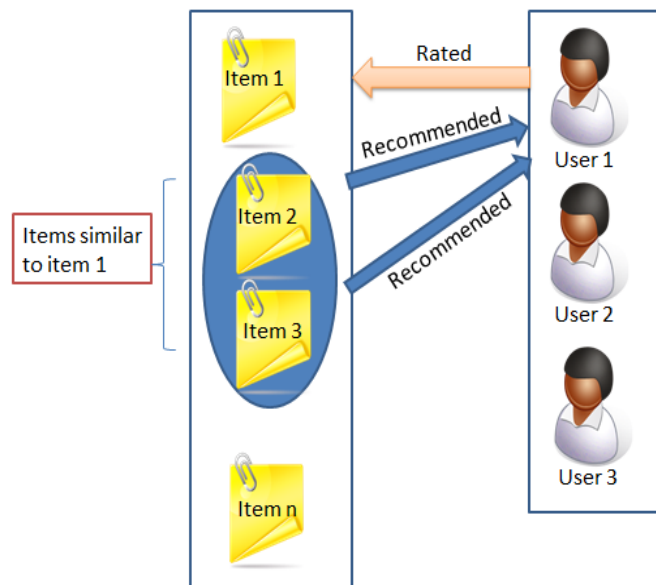


Figure 2.1: Content-based recommendation

2. **Overspecialization problem:** The recommendation for the user is limited to the items that are similar to those already rated. It is not possible to recommend items that are different from those that were rated by the users before. A person with no idea about a certain item will not be recommended this item as new. Sometimes the overspecialization problem includes the items when they are too similar to something the user has already seen. This problem had been addressed in many researches. Billsus *et al.* filtered both the items that are too different from the user's interest and those that are too similar to what was recommended before to the user [21]. Zhang *et al.* also proposed five measures for determining the redundancy of a new document with respect to a previously seen stream of documents [134].
3. **Cold Start problem:** The user have to make a sufficient number of ratings before the recommendation system can detect his preferences. It is not possible to make recommendation for new users who do not have rating for any items yet.



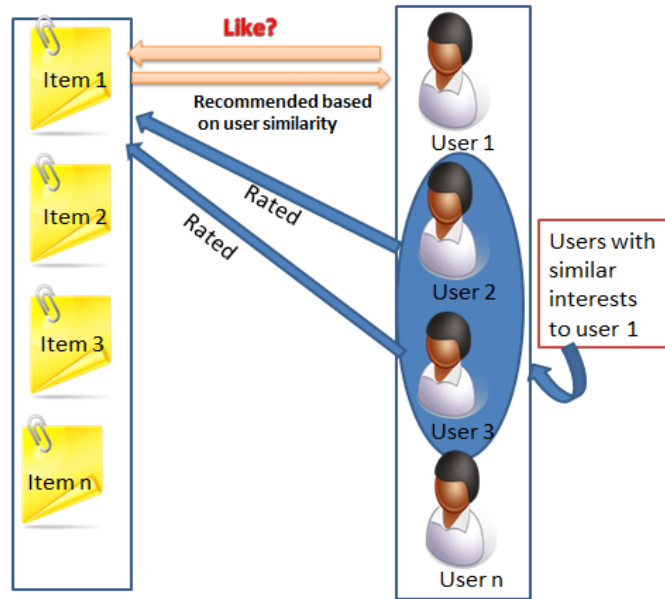


Figure 2.2: Collaborative based recommendation

### 2.1.1.2 Collaborative Based Filtering

Collaborative filtering makes recommendations by measuring similarities of users preferences [39]. As it analyzes patterns of favorable items without analyzing any content properties of items, it has been possible to discover these items without analyzing any content properties of them. Figure 2.2 represent the collaborative approach. Different collaborative systems have been developed in academic field. The systems introduced in [73, 111] were considered the first systems to use collaborative filtering algorithms to automate prediction and recommendation.

According to [130], algorithms of these techniques have been generally grouped into memory-based and model-based. The memory-based method measures the similarity in previous ratings for the same item for different users. The aggregated ratings from these similar users can then be used as a prediction for the target user's rating. In model-based method, a model (e.g., Bayesian model, probabilistic model) is learned from the previous ratings in order to predict the target user's ratings. Collaborative filtering techniques are mostly used in cases with some domains (e.g. music, cultural events, etc.) where content information is unavailable, or there is no content properties for it. It is clear that collaborative systems do not have some of limitations of that of the content-based

systems have. However, one of the limitations for this method is its dependency on the amount of ratings present [78]. This raises again the problem of cold start. It is difficult to produce accurate recommendation for a new user who has very few or no ratings at all. Other limitations include:

1. **New User problem:** In collaborative recommendations, the system must analyze the user's preferences from the ratings that the user gives. So, to address this problem, many of the present techniques use the hybrid approach that combine both the content and the collaborative one.
2. **New Item Problem:** Collaborative systems depend mainly on user's preferences to make recommendations. Therefore for a new item that is not rated by a number of users, the recommendation system would not be able to recommend it. Hybrid approaches are addressed for this problem as well.
3. **Sparsity:** The sparsity problems of the recommendation systems are due to lack of gathered information and the inability to provide inferences for either users or items. One way to overcome the problem of sparsity is to use user's profile information besides their rating skills to calculate user similarity [96]. That is, two users can be considered similar not only if they rated the same item similarly, but also if they belong to the same geographic area.

Another approach for exploring similarities among users was made by Huang *et al.* [61]. They dealt with the sparsity problem by applying an associative retrieval framework that explore transitive associations among consumers through their past transactions and feedback.

### 2.1.1.3 Hybrid Approach

This approach combines both content and collaborative filtering approaches to make better recommendations by trying to avoid the drawbacks in content-based and collaborative filtering approach exclusively.

Different algorithms that fall into the hybrid recommendation category had been introduced [69,96,

110]. These algorithms employed different methods to combine content-based and collaborative filtering approaches. Some of these methods can be classified as follows:

1. **Combining Recommendation systems approach:** One way to build a hybrid system approach is to first implement content and collaborative approach separately, and then combining their output (the ratings) into one recommendation [33,96].
2. **Adding content based properties to collaborative methods:** These methods use the content based profiles of the user to calculate the similarity between the users. These similarities are then combined with the unrated items to make the final recommendation.
3. **Developing a single recommendation model that combines both content and collaborative approaches:** This is considered the most widely used method. Smyth *et al.* proposed the PTV system that uses this approach to assemble a recommended program of television viewing [114]. It uses content-based techniques based on textual descriptions of TV shows and collaborative information about the preferences of other users. The final recommendation is then a combination of the two methods. Basu *et al.* also proposed using content-based and collaborative characteristics (e.g., the age or gender of users in a single rule-based classifier in order to make recommendation [17].

## 2.1.2 Categorization by Objective

### 2.1.2.1 User Recommendations

User recommendations have been extensively studied in the context of online social networks. Users are interested in finding not only their close friends but also new contacts not yet known to them. A user may follow other users whom he or she does not know but who share interesting topics. Based on the user's needs, different recommendation algorithms are used. Some of these studies were interested in recommending popular users. Other focused on recommending friends

(followees) for the user or discovering communities and groups. They used the social network of the user to find common friends and recommend known ones.

Chen *et al.* introduced friend recommendation systems that provide the user with promising potential friends based on their user profiles [30]. Garcia *et al.* identified some features that might be useful for recommending followee [43]. The intuition of the paper was that if a target user has many popular and active followees, other popular and active followees should be recommended to the user. If the target user has only popular followees, only popular followees should be recommended. A similar approach can be applied for target users with active followees. They found that the popularity (the followers and followees count ratio), and the activity of the user (the number of tweets he posted since the creation of his account), are the most relevant features used for recommendation.

Hannon *et al.* presented Twittomender system that recommends followees using both content-based and collaborative-based approaches [54]. In the content-based approach, users are represented by their own tweets, their followers' tweets, their followees' tweets, or combination of all of them. Recommendation is then made based on the similarity between user and targeted user's tweets. In the collaborative-based approach, the users are represented by followees IDs, followers IDs or combination of them. Each user is then represented by a set of his follower/followee IDs. Then, TF-IDF weighting scheme is used to find users with similar follower/followee IDs.

Kim *et al.* proposed two recommendation system models for Twitter, TWITOBİ and TWILITE, using probabilistic modeling based on LDA and matrix factorization. The models recommend the top-K users to follow and the top-K tweets to read for a user. In TWITOBİ, the model estimates the probability that a user  $u$  generates a word  $w$  in his/her tweets, whereas TWILITE is an algorithm that estimates the topic preference distributions of users to generate tweet messages as well as the latent factor vectors of users to establish friendship relations [70, 71]. Golder *et al.* introduced a structural approach to contact recommendations in Twitter by presenting reciprocity, shared interests, and filtered people as methods for recommending followees [46]. The reciprocity method assumes that a user will follow back his or her followers. Shared interests methods states that

people are considered similar or shares the same interest if they are following the same people. Similarly, users who share the same audience or followers are considered similar. A user is then recommended to follow his similar users. Filtered people of a user are also described as the users whose tweets are retweeted by the followees of this user. A user may be interested to follow those filtered people who are the followees of the user's followees because they may also share the same interest.

Krutkam *et al.* recommends followees based on the number of followers that the user has, the number of lists or groups that the user is listed in [74]. In their work, they didn't consider the personnel user's preferences. Instead, they used the above methods to suggest the most popular users. They proved that recommendation based on the number of followers significantly outperforms recommendation based on the number of lists the user is in.

### 2.1.2.2 Activity Recommendation

Activity recommendation systems refer to recommending activities to a user that he may be interested in, taking into consideration the user's interests and location. These recommendation systems are mostly related to the Location based social networks (LBSNs), in which it acts as an information retrieval operation of one or more activities that are appropriate for a query location (e.g., sightseeing, and shopping).

Pozdnoukhov *et al.* explored the the spatial-temporal distribution of the topical content of the geo-tagged tweets [97]. They proved that the topics, and thus activities, are often geospatially correlated.

Zheng *et al.* proposed a collaborative-based approach to extract the features for the locations, and to provide activity recommendation in LSBNs [136]. They relied on three matrices: location activity matrices, location-feature matrix and activity-activity matrix. The location activity matrix is used to correlate the user's activity to a spatial location. Location-feature matrix is used to connect locations and categories (e.g. restaurants, cafes and movie theater). The basic idea in this

matrix is that locations of the same category are likely to have the same activity possibilities. The activity-activity matrix shows the correlations between different activities. The probability that certain activity will be performed at a certain location given that a user has performed some other activity can then be predicted.

### 2.1.2.3 Location Recommendation

These systems are used to suggest stand-alone locations which provide a user with individual locations (e.g. point of interest such as restaurants or cities), or sequential locations (such as recommending travel routes and sequences) that match their interests and constraints.

In stand alone recommendation systems, some systems use the similarities in the user's profile and the location metadata, such as description and semantic text and tags to recommend a new place for the user. Park *et al.* used a Bayesian network model to match the user's profile data (age, gender, preferences) with the different categories of restaurants and recommending the best that suits the user's preferences [93]. Kodama *et al.* proposed an approach to recommend location based on semantic data and make a final recommendation using a skyline operator that takes into account both the price and the distance of the candidate location [23, 72].

Backstrom *et al.* measured the relationship between geography and friendship by using the user's spatial data and the network of associations between members of the social network [15]. Using these measurements, they can predict the location of an individual.

Leung *et al.* proposed the Collaborative Location Recommendation (CLR) framework for location recommendation [81]. The framework considers activities and different user classes to get refined recommendations. The authors presented a dynamic clustering algorithm, namely Community based Agglomerative-Divisive Clustering (CADC), to cluster the trajectory data into groups of similar users, similar activities and similar locations.

Ye *et al.* incorporated the user preferences, social influence and geographical influence to recommend points of interests [131]. They proposed a power-law probabilistic model to capture the ge-

ographical influence among Points of Interest. Finally, the authors evaluate their proposed method over the Foursquare and Whrrl datasets, and discover among others that geographical influence is more important than social influence and that item similarity is not as accurate as user similarity due to a lack of user check-ins.

Yuan *et al.* developed a collaborative recommendation model to recommend POIs for a given user at a specified time in a day [132]. They defined a new problem, the time-aware POI recommendation, that considers the temporal influence in user activities. In addition to the temporal influence, they also enhanced the recommendation model by considering geographical information and the social influence (i.e. users tend to visit nearby POIs). The authors found that if two users have similar temporal behavior, they are likely to visit similar POIs at the same time.

### 2.1.3 Other Recommendation Systems

Due to the rapid growth of the social networks content such as blogs, there emerged a need to design personalized recommendation systems to recommend only useful content to users. Morales *et al.* uses tweets to build user profiles and recommend interesting Yahoo news articles to users based on the supervised learning method [36].

Other than the previous recommending systems, some systems recommend news articles to the user based on the posts generated by that user. Chen *et al.* proposed a URL recommendation system “Zerozero88” that recommends interesting URLs to the user [31]. First candidate set of URLs are chosen according to their popularity in the social neighborhood of the user. The candidates URLs are then ranked according to the topic relevance to the user tweets. Duan *et al.* proposed a strategy to rank tweets by applying learning to rank algorithm [38]. Naveed *et al.* presented a LiveTweet application that can predict the probability of the messages being retweeted by a user [91]. Their application were based on training a Naive Bayes model to learn the basic content features of the users tweets. However, all these methods uses content based methods for recommendations. They don’t consider personalization of tweets, nor do they consider the effect of spatial and temporal

change in getting the user's interest.

Hashtag recommendation is another type of recommendation systems offered in Twitter. Hashtags are used in twitter to categorize tweets according to the user's interest. Zangerle *et al.* used the contents of the user's tweet to recommend the interesting hashatag [133]. The system uses TF-IDF scheme to measures the similarity between the tweets. Hashtags are then extracted from these tweets and ranked using a similarity ranking score.

Some previous work focused on studying the retweet behavior is also relevant to our work. Hong *et al.* used the content features, temporal features, publishers features and tweets features to predict the popularity of messages measured by the number of future retweets [59]. Suh *et al.* built a predictive retweet model based on the content and contextual features. Their work is based on finding useful features that enable them to predict whether a tweet will be retweeted regardless of who will retweet the tweet [117]. However, this work was still missing the personalization factor. Yang *et al.* modeled twitter as a directed graph of users and tweets as nodes, and the retweets as edges. Using this graph, they find the tweets that might be of interest to a wide range of users by analyzing the retweet relations in the graph [129]. Lee *et al.* introduced a variant of contents-based analysis for news recommendation. Instead of utilizing the information about the news from the original news contents read by the user, their model uses the information obtained from the user's tweets, retweets and hashtags to extract the important keywords and build a personal profile from them [77]. However, this model did not employ the user's social network in the recommendation system.

Limited work has been done in dynamically personalized tweet recommendation. The study done by Abel *et al.* is the most relevant to our problem [7]. They analyzed how users profiles changes over time, and how to recommend news articles for topic based profiles. Our model is different in that it tries to capture the change of each user's interest in different topics over time, and recommend interesting tweets based on this interest. Uysal *et al.* explored user-publisher and user-tweet features to rank the Twitter feed for each user based on their probability of being retweeted [122]. Compared to our model, it just uses the explicit features of the tweets without considering the



personalized features for each user.

## 2.2 Information and Influence Propagation in Social Networks

In recent years, information propagation on social networks has been attracting much attention in academic and industrial circles [80]. Understanding the mechanisms of information propagation is vital to finding the factors affecting the information propagation process. These factors, in turn, provide a better explanation for predicting information popularity [16].

Two factors that affect the information propagation process: the importance of the information, and the level of interactions between users. The studies of the first factor mainly consider the analysis of the messages propagation and the decay with respect to the time since the posting of the message [51]. Most of these approaches are descriptive. However, our approach is predictive.

For the second factor, the level of interactions between users, the current research efforts focus on the interactions between the users, along with the geographic, demographic, topical and contextual features that affect the propagation between the users [11]. For example, Galuba *et al.* proposed a propagation model that predicts which users will tweet about which URL based on the history of past user activity [42]. Agarwal *et al.* studied the problem of identifying influential bloggers in the blogosphere [9].

As our model analyzes and predicts the localized diffusion of trends in social networks between locations, our work is different in that it doesn't take into account the social structure of the social networks. It is prohibitively complex to include the social structure connections relating the locations. Another point is that the location information posted by the user is not always available or accurate. Our work is also different from the research that studies relationship between geography and information diffusion [26], as our model considers other non-geographical parameters.

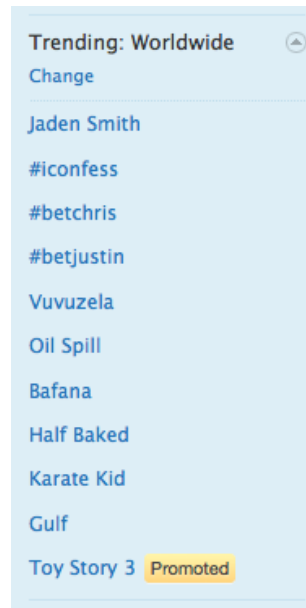


Figure 2.3: Example of trending topics in Twitter

## 2.3 Trending Topics in Social Networks

Since the nature of the user's posts in Twitter is quick and transient, it can be considered as an information system that provides a real time reflection of the interests and thoughts of its users, as well as their attention. As a consequence, Twitter serves as a rich source for exploring the mass attention of millions of its users, reflected in trends that can be extracted from the site.

A trending topic on Twitter is a word, phrase or topic that is posted multiple times. The trends appearing on Twitter are the terms that occur with the highest frequency in the tweets [75]. These trends become popular either through a concerted effort by users, or because of an event that prompts people to talk about one specific topic. They can be Twitter memes, local or global events, or tweets related to the celebrity. Figure 2.3 shows an example of trending topics in Twitter.

A number of researches was made in the area of detecting trending topics in different microblogs data stream. Benhardus *et al.* proposed a method to determine the trending topics using an approach based on TF-IDF scores [18]. This method uses only unigram and bigram word clusters as potential trending topics. This in turn did not provide flexibility in the resulting topics provided.

Mathioudakis and Koudas presented a system, TwitterMonitor, to detect trending topics in Twitter [84]. Its main idea depends on detecting the bursty keywords that appear in tweets at a high rate and then grouping these keywords to form word clusters. The system then discover interesting patterns by extracting additional information from the tweets that belong to these trends.

Cataldi *et al.* presented a technique that can retrieve the real time trending topics expressed by the community. They developed metrics to individually identify each word that might indicate a trending topic. According to these metrics, they then group the words by computing correlations across them [28].

Budak *et al.* presented a different approach in detecting the trending topics in social networks [24]. They presented a new algorithmic tool, Geospace, that can detects geotrends. Their tool depends mainly on detecting correlations between topics and locations, in addition to analyzing topics and locations independently. Mukherjee *et al.* used an approach that identifies trending concepts using the hourly Wikipedia page visitation statistics. They first get lists of trending concepts by processing the text listed on the news aggregation sites through an inverted index. Then by accessing Wikipedia concept visitation statistics, they use a MapReduce framework that analyzes the raw hourly visitation logs and generates a ranked list of trending concepts on a daily basis [89].

Asur and Huberman also presented a different approach in detecting the trends [12]. They used a stochastic model to find the growth of trending topics, along with the factors associated with them. They showed that the trending topics are primarily posts that are reposted by others frequently.

Kannan *et al.* proposed the TrendTracker, which implements real-time visualization system by actively monitoring Twitter feeds and changing window sizes [65]. This allows the user to pick out the most popular trends as they are actively being tweeted.

Kim *et al.* proposed TwitterTrends, a spatio-temporal trend detection and related keyword recommendation scheme for tweets. Their approach analyze the user's tweets and their metadata such as GPS data to identify hot keywords and recommend their related keywords at a given location and time [68].

Related to the relation between the user's interests and the trending topics, Cataldi *et al.* proposed a reranking approach that makes use of the users' activity and the posts' contents to personalize the emerging topics [27]. This is done by monitoring the usage of the keywords appearing in the tweets over time, and comparing its importance with the user's context, in order to highlight the most emerging topics within the user's interests. As the evaluation of a personalized topic detection strategy results is a complex task due to the unavailability of dataset for this, the validation for this approach was implemented by conducting a personnel questionnaire.

AlBawab *et al.* presented a framework that identifies trending local topics by computing the geographic features for the search queries and then using these features to detect local queries [10].

Fiaidhi *et al.* presented an approach for personalizing trending topics through enabling the twitter user to provide RSS feeds that include the personal preferences along with a twitter client that can filter personalized tweets [63]. They used two algorithms to identify tweets that are similar to the RSS Levenshtein Distance algorithm and the LDA.

Some researches studied trends from a temporal view. Leskovec *et al.* studied the temporal properties of information shared in social networks by tracking memes across the blogosphere [79]. Other researches studied the structural nature of the social graph that leads to creating the trends [19].

Other studies focused on studying the dynamics, the growth and the decay of the trending topics [13, 124]. Asur *et al.* studied the trending topics on Twitter, and provided a theoretical basis for the factors affecting the formation, persistence and decay of trends [13].

Limited work has been done to analyze the relation between trends and geography. Kamath *et al.* modeled the social media spread from location to location by trying to predict the top K cities in which a topic will be trending [64]. Ferrara *et al.* investigated the spatial and geographic dynamics that govern trending topics in Twitter. However their goal was different, as they aimed at studying what dynamics underlie the production and consumption of trends in different geographic areas. In other words, they wanted to know if trends travel through the Internet, or by people physically traveling across cities [41].

## 2.4 Topic Modeling

Topic Modeling is a rapidly-growing field of research in the area of text mining, and statistical modeling. As text comprises about 85% of data worldwide [14], topic models have been widely used to address the problem of ‘information overload associated with this huge collection of text and corpuses. It had been applied in different areas, including social networks in general and specifically in Twitter social network [101]. These models can be divided into three categories: topic models for authorship, hypertext, and edges.

The Author-Topic model is the first category in the topic models. It was first introduced in [105]. In this model, each word  $w$  in a document is associated with two latent variables: an author  $x$  and a topic  $z$ . The documents are analyzed with their authors. The basic idea in this model is that it considers a document is created by authors sharing common topics. It then groups documents and authors according to that assumption. Many extensions were then made to the author topic model. Another studies presented the Author-Recipient-Topic (ART) model for social network analysis, which learns topic distributions based on the sender-recipient relationships [85, 100]. Pathak *et al.* modified the ART model and suggested the Community-Author-Recipient-Topic (CART) model for community extraction [95]. This model extracts communities based on communication links and content information. Their assumption is based on the fact that the topics of communication determine the communities. Rajani *et al.* propose another version of the Author-Recipient-Topic model, in which the probabilistic distributions of words are conditioned to the document’s authors and recipients. This model is shown to present better results than LDA when the number of topics is large (over 300). The second category is related to social-network analysis where documents are analyzed according to their citations (hypertext). Chang *et al.* proposed the Relational Topic Model (RTM) which models a citation as a binary random variable [29].

The last category in the topic models only uses linkage (edge) information. Zhang *et al.* dealt with the issues in applying LDA to academic social networks [134]. They proposed edge weighting schemes based on collaboration frequency to convert the co-authorship information into a graph.

Latent Dirichlet Allocation (LDA) is a standard unsupervised machine learning tool that identifies latent topic information in large document collection and has been extended in many ways to be used in identifying topics in social networks and social media [60]. The LDA model treats each document as a bag of words, and according to the frequencies of different words appearing together in each document, the model then determines the most relevant set of words to each topic. After training a topic model, it can be used later to infer the topic(s) available in new documents.

However, their application on microblog contents such as Twitter faces different challenges. 1) The posts are short, 140 characters; 2) The use of informal language and nonstandard abbreviations (e.g. LOL, WOW); and 3) The text contains other context that may act as noise as the URL, Twitter names and tags. To overcome these difficulties, some studies proposed to aggregate all the tweets of a user in a single document [126]. This can be regarded as an application of the author-topic model [116] to tweets, where each document (tweet) has a single author. Another modification to the author-topic model was introduced by Zhao *et al.* [135]. They introduced a model, Twitter-LDA, which assumes a single topic assignment for an entire tweet. The model is based on the following assumptions. There is a set of topics  $T$  in Twitter, each represented by a word distribution. Each user has topic interests modeled by a distribution over the topics. When a user wants to write a tweet, the user first chooses a topic based on the user's topic distribution. Then the user chooses a bag of words one by one based on the chosen topic. However, these treatments assume that the user's interests in topics will not change over time, which contradicts our assumption that the user's interest in topics change over time. We are proposing a complementary approach to the LDA model that can help in extracting better topics from microblogs.

## 2.5 Event Detection

Although the average number of Twitter posts exceeds five-hundred million posts daily, many of them are redundant, or of interest to a limited users [104]. Therefore a need for finding methods to extract interesting information had emerged. Event detection is one of the active fields of research

done on Twitter. Different methods are proposed for analyzing and detecting different kinds of events. These events may range from known ones (such as earthquakes, political news, fires), to smaller-scale ones (as a sale in nearby stores).

Sakaki *et al.* detected the first type of events [106]. They proposed an earthquake reporting system based on semantic analysis of tweets. They first built a classifier for tweets based on important features such as the keywords in a tweet, the number of words, and their context. A probabilistic model is then used for event detection and location estimation. However, the probabilistic model they used support only single event occurrences. Their model can not work on multiple event occurrences (as accidents or traffic jam detection).

An analysis for the twitter based criminal incident prediction was provided by Wang *et al.* [125]. They first collected data from both twitter posts and local law enforcement agencies about the crimes . Semantic role labeling systems are used to extract the events in the tweets, along with the entities involved in the events. Latent Dirichlet allocation (LDA) probabilistic model is used to discover word-based topics and reduce the dimensionality of documents. Prediction of future crimes was then made by using linear modeling. The drawback of this method is that they only assumed that the events contained in the tweet is posted on same day of occurrence. Therefore the posts that are written to describe a distant past event was not included in their search, hence lost their effectiveness.

Ishino *et al.* discussed how to find traffic evacuation routes in case of disasters [90]. They introduced a system that uses tweets that are posted in a disaster time in Japan to extract transportation information and to detect traffic problems. First, tags that describe the transportation information (e.g., From, To, Method) or traffic problem (e.g., road, train line) were defined. Then they used Conditional Random Fields (CRF) machine learning method to extract information like destination or departure place.

Jackoway *et al.* combine both the the news from different news media with information posted in Twitter for future prescheduled events based on first collecting information about new events from news sources [62]. A geotagging system is then used to tag these events. The text in the news is

then processed by tense detectors to determine if the events is in the future or not. Once the future events are identified with locations and keywords, Twitter posts about this event are then queried. The resulting posts are then measured for similarity with the news events. This helps in indicating which post are reliable and which users are near from the event locations.

Detecting crime events were also introduced by Li *et al.*. They introduced TEDAS, a twitter based system that can detect new events and analyze temporal and spatial patterns of the events [83]. Their system is composed of a crawler that crawl twitter for all crime and disaster events. A classifier that uses twitter specific features (e.g. short URL, hashtags), as well as other specific features that are related to the event is then built to determine if the crawled tweets relate to the desired events. After classifying the tweets, Li *et al.* developed a rating model to identify the most important events. The ranking model depends on some features including content, user and usage features.



# Chapter 3

## Problem Description

In recent years, online social networks have experienced exponential growth in the number of users and the amount of information. As an example, Facebook, created in 2004, claims more than 1 billion monthly active users as of June 2015, 50% of which log on to Facebook in any given day [2]. Similarly, the microblogging site Twitter, started in 2006, attracts over 300 million monthly registered users and more than 500 million postings per day as of 2015 [1, 43]. Millions of users visit Facebook, Twitter to chat with friends, make new friends, engage in random chatter, or to share photos, news, and useful tips.

An important characteristic of these social network sites is real-time message streams. Through these message streams, users can broadcast short text messages to their online social network in near realtime. Twitter is considered the first major social network site which is attributed to message streams. Today, message streams have also been integrated into a wide range of social network sites, such as Facebook and LinkedIn.

Because of this information explosion, the users became overwhelmed with the huge amount of information they have to follow, and hence they are spending a lot of time and effort to get just the information they are interested in. So, the key challenge today is for the users to find relevant information based on their interests. It became important for each user to use individual prefer-

ences, and at the same time be involved in the trends that might be of interest to him. This problem has led to the evolution of the recommendation systems that help users find information they need based on their interests.

However, recommendation systems still face many challenges as follows:

1. **Content analysis and data sparsity:** As described in Section 2.1, the algorithms for recommendation systems suffer from the ability to measure item similarity. Content-based methods depends on explicit item descriptions. Such descriptions may be difficult to obtain for abstract items like ideas or opinions. On the other hand, collaborative filtering has a data sparsity problem [8]. In contrast to the huge number of items in recommendation systems, each user normally rates only a few items. It is difficult for recommendation systems to accurately measure user similarities from that limited number of reviews.
2. **Cold Start problem:** Most of these systems suffer the cold-start problem [8]. Cold-start includes users, items, even systems, and it is about new entities entering a new system for the first time. Even for a system that is not particularly sparse, when a user initially joins, the system has no reviews from this user. Therefore, the system cannot accurately interpret this user's preferences.
3. **Trust issues in recommendation systems:** As described in Section 2.1, traditional recommendation systems depend mainly on the user-item rating matrix for making recommendations. However, these recommendations are not evaluated by their information value. Rashmi *et al.* found that if a user is given a choice between recommendations from friends and recommendation systems, friends' recommendations are preferred even though the higher quality of recommendations given by the recommendation systems [112] . This is because the user always prefer recommendation made from trusted friends rather than recommendations made by strangers. However, most of recommendation techniques mentioned in Section 2.1 make recommendations to the user mainly based on other users' preferences. These users have similar rating data with the target user. These recommendations are made

regardless of the trust between these users. Therefore, another challenge for the recommendation systems is how to embed the social elements of the trust relations among users in deciding about new recommendations.

- 4. Challenges to adapting the recommendation system to the dynamical aspects of users and items:** The dynamic nature exists in both users and items. From the user's perspective, the user's preferences or interests change over time. The users' interest in an item and the sensitivity of that item with respect to time also changes. Some items are time-sensitive and expire quickly. Other items are of continuous interest, such as classic story books. Recommendation systems should be able to adapt to these dynamic factors and make effective dynamic recommendations.

Some approaches had been proposed to tackle these problems. One of the approaches is based on clustering users or items according to their latent structure [108]. Unrepresentative users or items are discarded in this approach, and thus the user/item matrix becomes denser. However, this technique does not significantly improve the performance of recommendation systems. Another approach is to utilize some implicit user ratings and exploiting associations among users through their past transactions and feedback to make a denser user/item rating matrix [61]. Hong *et al.* also presented an approach that tackles the problem of content analysis by utilizing temporal features, publishers features beside the content features to predict the popularity of messages measured by the number of future retweets [59].

We are studying the problems of the content analysis and adapting the recommendation system to the dynamical aspects of users and items.

The user's preferences are shaped by his/her personal interests. At the same time, users are also affected by their surrounding environments, which are determined by their geographically located communities. So, the user in social networks is either stating interest in receiving their information feeds from specific sources (e.g., the user's friends in Twitter) or having information recommended based on the user's interests.

Accordingly, our approach takes place on two levels. First, we propose a new dynamic recommendation system model that provides better customized content to the user. The recommendation system enhances the user's interaction by utilizing information in social networks, and studying the effect of the change of the user's interest over time. Our model aims at providing personalized recommendation that will give the user a summary of all received corpuses. Considering the fact that the user interests changes over time, this summary should be based on the user's level of interest in the topic of the corpus at the time of reception. Specifically, we study the user's activities and relationships on Twitter and answer research questions about the individual user's interests: *How can we infer personal interests from the user's Twitter activities and interactions and to what extent do personal interests change over time? What are the other features that can be extracted from the user's Twitter activities and can affect the recommendation made to the user?*

Second, in order to fully customize the user experience, we analyze how the change in the surrounding environment can affect user's experience. Specifically, we study how the change in the geographical community preferences can affect the individual user preferences. Our research focuses on improving the suggestions provided by the social media to the users. Our assumption is that enhancing the trending topics suggested by the social media to user based on their location will reflect positively on their online experience. We approached this point by investigating the interplay between local community interests and public trends. And hence developing a model for predicting localized trends diffusion from one localized community of users to other geographically separated communities of users.

We show that observing the local trends in some locations can be used to predict where these trends will appear next. The topics of interest to the user discovered by the tweets analysis system are used by the trends analysis system to personalize the trends suggested to the user.

The most important aspect of our model is prediction of trends that will appear in a location, before even users in that location start mentioning that topic. This is extremely useful in many cases, such as building a proactive localized recommendation system for topics or for early prediction of social events (e.g., protests).

# Chapter 4

## Tweets Analysis Subsystem

### 4.1 Introduction

In this chapter, we describe how to provide better subscriptions view for the user. This corresponds to the Tweets analysis subsystem shown in Figure 1.6. For this, we are proposing a new model of recommendation systems which can enhance the user's interaction and behavior by utilizing information in social networks and studying the effect of the change of users' interest over time. Specifically, we are addressing the problem of dynamic personalized recommendation systems (specially in Twitter), and studying how to exploit the dynamic patterns in the user's profile to improve the performance of these recommendation systems. There are many important factors that are essential for effective recommendation systems. First, the value of item content may change over time. For example, breaking news are time-sensitive items that can change quickly within short periods of time. On the other side, some items are of continuous interest for a long time (e.g. some movies or reference books) that continue to be referenced for a long time after their initial publication. Recommendation systems should be able to adapt to these dynamic factors and make effective dynamic recommendations.

While some of the related work that mentioned in Section 2.1.3 focused on the study of the behavior

of the users and the factors that affect recommending general items in the social network, our work is focused on the factors at the personalized level and the effect of the change of these factors over time, i.e., building a model that is personalized for each user based on the temporally dynamic features; that is, features that are determined based on the time of publication. The following section describes our subsystem.

## 4.2 Problem Description

In this dissertation, we had used Twitter social network as our case study. Our approach is based on defining a model that recommend the most important tweets to the user according to his past preferences. The model's main idea is to classify a given tweet into important or not important. It will mainly consider the user's level of interest in the tweet's topic at the same time that given tweet is posted. Other features include the authority of the publisher (the number of users following the publisher), the tweet content based features such as the length of the tweet and the retweet count, and the social relation feature which represent the relation between the user and the publisher of the tweet.

When users log on Twitter, they see a stream of tweets sent by friends which composes their timeline. Many of these tweets are conversational tweets and/or are not of personal interest to the user. The goal of our model is to decide for each user the tweets that might be of interest from the user's timeline. Beside being able to post their own tweets, users can also interact with their timeline by replying, retweeting or favoring the tweets. As there are no explicit means to extract the user's level of interest in a tweet from Twitter, we relied on these actions to predict the user's interests. Hence, the retweets, replies and favorites can be used as an indication of the interest of the user in the corresponding tweets. We define a tweet as a tuple  $\langle \mathbf{u}, \mathbf{p}, \mathbf{e}, t, int_{tu} \rangle$  where (vectors are in boldface):

- $\mathbf{u}$  is a vector describing the features of the user  $u$  receiving the tweet.

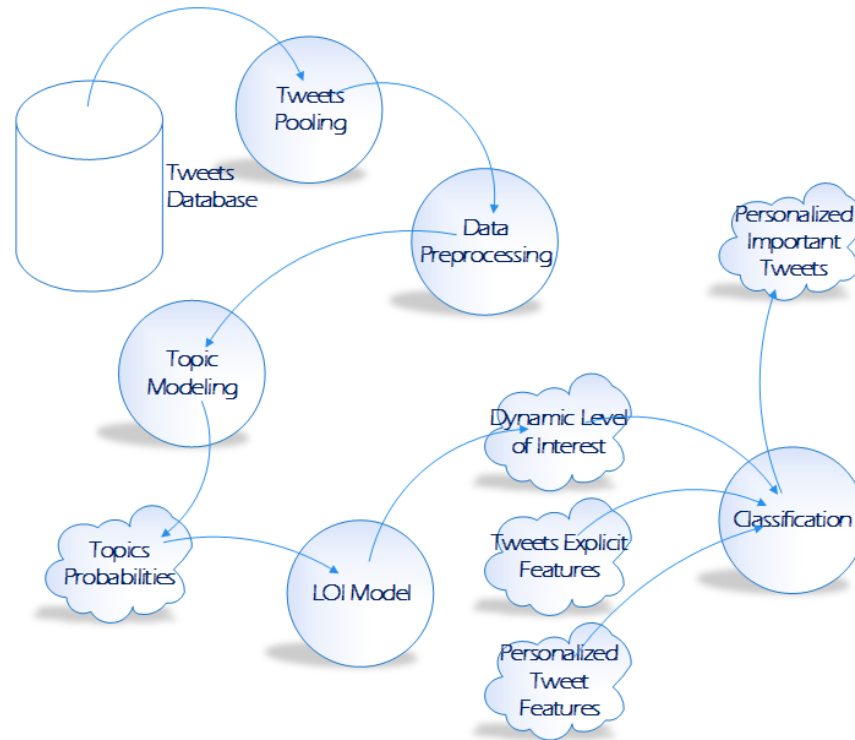


Figure 4.1: Tweets analysis structure

- $\mathbf{p}$  is a vector describing the features of the publisher  $p$  of the tweet.
- $\mathbf{e}$  is a vector describing the features of the tweet  $e$ .
- $\mathbf{o}^e$  is a vector that holds the distribution of probabilities of the tweet text  $e$  across different topics.
- $t$  is the time window in which the tweet is posted.
- $int_{tu}$  is a binary value indicating whether this tweet is of interest to the user  $u$  or not.

Given these tuples for the tweets in the user history, our goal is to predict  $int_{tu}$  for each new tweet.

We now describe in more details our approach and discuss its main components (Figure 4.1).

### 4.3 Topic Extraction

In order to predict the user's interest in a corpus, we based our prediction on the user's interest in the topic(s) covered in that corpus, alongside with other features. Consequently, we needed to build a Topic Model of our tweets. Topic models, such as latent Dirichlet allocation (LDA) [22], are well-known for exploratory and predictive analysis of text. Generally topic models define topics as distributions over the words in a vocabulary and documents as being generated by mixtures of these topics. Topic models represent document words in a bag-of-words format without considering word order to be of any particular importance. According to the frequencies of different words appearing together in each document, the model then determines the most relevant set of words to each topic. After training a topic model, it can be used later to infer the topic(s) available in new documents.

Formally, the Latent Dirichlet Allocation (LDA) Model can be described as follow:

Given: A set of  $e$  posts denoted by  $\mathbf{E} = \{e_1, \dots, e_n\}$ , the LDA algorithm generates a set of  $k$  topics denoted by  $\mathbf{L} = \{l_1, \dots, l_k\}$ . Each topic is a probability distribution over  $m$  words denoted by  $l_i = \{w_1^i, \dots, w_m^i\}$  where  $w_j^i$  is a probability value of word  $j$  assigned to topic  $l_i$ . The post can then be represent as  $\mathbf{o}^e = \{o_1^e, \dots, o_k^e\}$  where  $o_j^e$  is the percentage of topic  $l_j$  in the post  $e$  composition.

### 4.4 Tweets Pooling

The short length of tweets might result in a poor topic model. Thus, to help get around the problems associated with the analysis of numerous small documents, we construct large documents out of the tweets. So, instead of looking at each tweet individually, we group together tweets that are similar in some sense (same semantics, same hashtags, etc.) in a process called *pooling*. In our model, we present some schemes that we used to aggregate tweets into a larger documents from which a better topic model can be trained. These pooling schemes can be described as follows:



1. **Hashtag pooling:** In Twitter, the hashtag is a string of characters preceded by the hash # character. They are used as identifiers for tweets discussing the same topic [76]. By including hashtags in a message, users indicate to which conversations and topic their message is related to. Using these hashtags can be a good indicator for tweets relatedness, and so can be used in the aggregation process of tweets. For the hashtag-based pooling scheme, we aggregated documents sharing each hashtag in one pool, as in [87]. If a tweet has more than one hashtag, this tweet will be added to the tweet-pool of each of those hashtags.
2. **Replies pooling:** We used replies for tweets as another way for aggregation. In general, a reply is a string preceded with the @ character. It is used as a comment on another tweet posted by you or by anybody in the social networks. As the tweets and their replies might share the same topics discussed, aggregating them in one pool can be a good indication for the tweet relatedness.
3. **URLs pooling:** We also aggregated tweets that include the same URLs in their text. Tweets sharing the same URLs might be discussing the same topic, and hence can be aggregated.

In our model we consider each aggregated pool of tweets a document, and the words in the pool the vocabulary. We use these documents and the vocabulary to extract the topics that form the corpus.

## 4.5 Dynamic Level of Interest

In this section, we study how the interests of individual users about a certain topic change over time. Getting the dynamic level of interest in a tweet takes place through some steps:

1. First we get the per topic activity in each day  $d$  for the user, denoted by  $\mathbf{A}_d = \{a_1^d, \dots, a_k^d\}$  where  $a_i^d$  is the level of activity of the user in topic  $l_i$  on day  $d$ .  $\mathbf{A}_d$  is calculated by adding

the vectors  $\mathbf{o}^e$  in that day (Equation 4.1). The details of this step are shown in Algorithm 1.

$$\mathbf{A}_d[i] = a_i^d = \sum_{\forall e \in E: e_{date}=d} \mathbf{o}^e[i] = \sum_{\forall e \in E: e_{date}=d} o_i^e \quad (4.1)$$

2. Given a new tweet  $e_{new}$ , the user's level of interest in the tweet can be calculated using Equation 4.2. Basically, we add the user activity vectors in the window of last seven activity instances prior to the tweet creation day  $d$ . Each of these instances corresponds to user's actions done in one day. For a user who is active (posting a tweet, replying, retweeting or favoring another tweet) every day, this window will span one week period. For less active users (not active every day), this window will be longer to cover the last seven active days in which the user was active. We only consider the last seven instances, as considering intervals longer than seven days will introduce irrelevant noisy tweets as discussed in [102]. This step is illustrated in Algorithm 2.

---

**Algorithm 1** Users Level of Activity Per Topic
 

---

**Procedure** CalculateDailyActivityVectors

**Input** Set of all users  $users$

**begin**

$L \leftarrow$  List of all topics

**for each** User  $u$  **in**  $users$  **do**

$Days \leftarrow$  All Days in which  $u$  was active

**for each** Day  $d$  **in**  $Days$  **do**

$Tweets \leftarrow$  All tweets by  $u$  in  $d$

      // Initialize vector for user  $u$  in day  $d$

$\mathbf{A}_d^u \leftarrow [0, \dots, 0]$

**for each** Tweet  $e$  **in**  $Tweets$  **do**

$\mathbf{o}^e \leftarrow$  Percentages of topics in  $e$

**for each** Topic  $l$  **in**  $L$  **do**

$A_d^u[l] = A_d^u[l] + \mathbf{o}^e[l]$

**end for**

**end for**

**end for**

**end for**

**end**

---

---

**Algorithm 2** Level of Interest in a new Tweet

---

**Function** CalculateLevelOfInterest**Input** User  $u$ Tweet  $e$ **begin** $\mathbf{o}^e \leftarrow$  Percentages of topics in  $e$  from LDA  
model $d \leftarrow e$  posting date $LoI \leftarrow 0$ **for each** Topic  $l$  in  $L$  **do** $val \leftarrow 0$ **for**  $i = 1$  to 7 **do** $val \leftarrow val + \mathbf{A}_{d-i}^u[l]$ **end for** $val \leftarrow val * \mathbf{o}^e[l]$  $LoI \leftarrow LoI + val$ **end for****return**  $LoI$ **end**

---

$$LevelOfInterest(u, e_{new}) = \sum_{l \in L} (\mathbf{o}^{e_{new}}[l] \cdot \sum_{d \rightarrow d-7} \mathbf{A}_d[l]) \sum_{l \in L} (o_l^{e_{new}} \cdot \sum_{d \rightarrow d-7} a_l^d) \quad (4.2)$$

## 4.6 Personalized Tweet Recommendation

In addition to measuring the dynamic level of interest for each user, some other static features can affect his interests. Some of these features represent the personalized interests of the user, others are general features that are related to the tweet's quality or the publisher's authority that can affect the tweet's degree of interest to the user. The following sections describe the personalized features and other explicit features that might affect the user's interests.

### 4.6.1 Personalized Social Features

Social features are the features that represent the social relationship between the user and the publisher. This relation can be friendship, neighborhood who posts tweets about events happening in the neighborhood or celebrities who have interests in common with the user.

User-publisher similarity feature measures the similarity between activity level of the user and the publisher on all topics. This is measured as the cosine similarity between vectors formed by summation of the level of interest in a topic for the user over time (Equation 4.3). Generally, the cosine similarity measure yields a value between -1 and 1. The value of 1 means the exact distribution match, i.e., activities of both users are distributed in the same proportions on different topics, though one of them might be generally more active than the other. The value of 0 means that the users have nothing in common.

$$\text{CosineSimilarity}(U_t, P_t) = \frac{U_t \cdot P_t}{\|U_t\| \cdot \|P_t\|} = \frac{\sum_{t=1}^T U_t \times P_t}{\sqrt{\sum_{t=1}^T U_t^2} \times \sqrt{\sum_{t=1}^T P_t^2}} \quad (4.3)$$

### 4.6.2 Explicit Features

Besides the personalized social features, we analyzed other explicit features that can affect the user's interests. These features appear or can be inferred from the user profile. This includes:

- **Publisher based features:** related to the tweets' publisher. These features are used as an indicator for the activity of the publisher:
  - **Publisher followers:** the number of followers for each publisher. High authoritative publishers are likely to have more followers than others.
  - **Publisher tweets count:** the number of tweets posted by the publisher since the opening of the account. This feature is an indicator for how active the publisher is.

- Mention count: the number of times a publisher's name is mentioned in all the tweets. If a publisher is frequently mentioned, the publisher is more likely to be popular and has more interactions than other publishers.
- Tweet based features: describe the tweets contents as:
  - Retweet count: the number of times the tweet got retweeted. It is a way of estimating the popularity of the tweet. A tweet retweeted more times is more likely to be a useful one.
  - HasURL, HasHashtag: sometimes a publisher includes supplement to their tweets with URL or hashtags. Hashtags can sometimes be an indication of the tweet's topic.
- Location feature: represents the cities or countries found in the publishers profiles. This feature is used to capture the spatial neighborhood effects. If a publisher posted a tweet about local events, and this publisher is the user's neighbor, then most probably the user will be interested in this post.

## 4.7 Experimental Results

In this section, we describe our datasets and the preprocessing steps followed by the experimental results for each step in our model.

### 4.7.1 Data Collection

Twitter provides an API to allow developers to collect tweets programmatically [3]. The API gives access to all public data on the Twitter website. It allows filtering by location, keywords and/or author. There are three types of Twitter API: REST API, Search API and Streaming API [3].

- **REST API:** The REST API allows access to the Twitter main data as the user's timeline,

user's friends and followers, and user's profile data. It also allows posting messages to Twitter by authenticated users.

- **Search API:** The Search API, which is also a part of the REST APIs provides search capabilities for twitter data. It allows the user to search based on keyword or by location. The result is data trends by location and duration. The Search API also allows text-based search. Text-based search allows the user to search all tweets containing a term. Boolean operations like 'OR', 'AND' and negation for terms are also allowed.
- **Streaming API:** Streaming API provides a real-time high-volume access to Twitter data. It provides long-lived connections designed to be open for a long time. We can specify filters for streaming like location, query, user ID and language.

In this work, we relied on the three sets of APIs to collect data used in our experiment. We used the Twitter4J java library to implement the data collection components of our system [128].

One of the main challenges that we faced is that the number of allowed requests to these APIs is rate limited. This means that Twitter allows only a fixed number of requests to each of these APIs per a time window. At the time we were collecting our data, the time window was set by Twitter to 15 minutes. The rate limits for the APIs that we used are shown in Table 4.1. We handled the rate limit constraints by two approaches.

The first approach, which was used in the early development stages, is when our data retrieving application reaches the limit, it sleeps for the remaining part of the rate limit window. Clearly that was a big limitation as the rate was not enough to retrieve reasonable amount of data in a timely fashion.

The second approach, which we adopted later, is relying on the way that Twitter uses to calculate the rate limit. To establish a connection with Twitter, an application need to be acting on behalf of a Twitter user. Users give permissions to applications through the OAuth2 protocol, which allow applications to get access tokens from Twitter, without having the user to give Twitter password to the application. The application then can use these tokens to act on the behalf of that specific user.

Table 4.1: Twitter APIs description and rate limit

<b>Twitter</b>	<b>Description</b>	<b># of objects returned per request</b>	<b>Rate Limit</b>
GET statuses/user_timeline	Returns a collection of the most recent Tweets posted by the user indicated by the screen_name or user_id parameters.	200	15
GET trends/place	Returns the top 10 trending topics for a specific WOEID, if trending information is available for it. This information is cached for 5 minutes. Requesting more frequently than that will not return any more data, and will count against the rate limit usage.	10	15
GET users/show	Returns a variety of information about the user specified by the required user_id or screen_name parameter.	1	180
POST statuses/filter	Returns public statuses that match one or more filter predicates. Multiple parameters may be specified which allows most clients to use a single connection to the Streaming API. Allows filtering by a set of geo-bounding boxes.	N/A	N/A

Twitter calculates the rate limit per user, not per application. Thus an application with  $N$  users, will have  $N$  times the rate limit requests count. We created several Twitter accounts, and let these account give permission to our application to act on their behalf. The application simply switches to another user when it reaches the rate limit of the current user. This allowed us to overcome the rate limit barrier.

### 4.7.2 Dataset and Preprocessing

For our experiments, we created a Twitter data set containing five million tweets and 20 thousands users that were seeded by first selecting 100 active users from the Virtual Town Square blog [5]. We used Twitter REST API [3] to facilitate the data collection. The majority of the tweets collected were published in a three-months period from April 2013 to June 2013. We then expanded the user base by following their followers and friends. We were able to include 20 thousands users with all their posts.

As Twitter APIs does not allow access to the timeline of the user directly without authorization, we build each user's timeline by first getting the posts for each of the base users, and then following the tweets posted by their friends, and consider them the scanned tweets by the user. All the favored tweets by the base users are also retrieved.

We build our model from a repository of more than five million tweets. To eliminate incomplete and noisy data, we preprocessed the tweets by discarding tweets with non-English words. We also removed meaningless words such as stop-words, Twitter specific stop words, user names, and special characters and stemmed the remaining words.

Usually, users do not have time to see all the tweets posted on their timeline. Also, users can be away or inactive (i.e., no posts, retweets or favorites) for long periods of time. Using this period in our dataset will make the number of negative examples much bigger than the number of positive examples.

To overcome this, we filtered the browsing history by considering a window made up of a set of



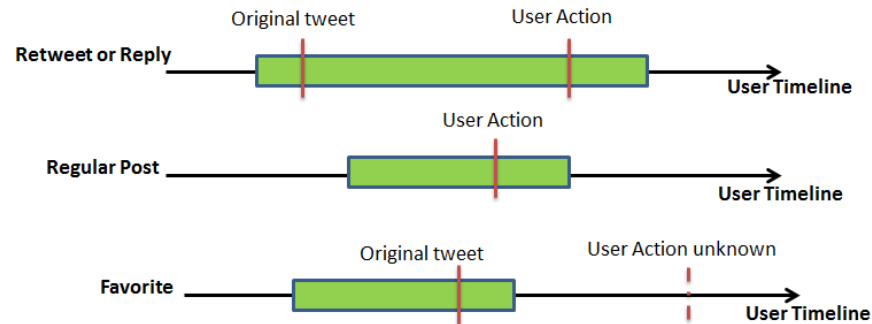


Figure 4.2: Timeline window for the user

20 tweets. The number 20 is chosen due to the fact that Twitter limits the number of tweets to be retrieved to 20 tweets each time the user browse his timeline. The window's sliding scheme depends mainly on the user's action in time of browsing the history tweets. As in the case of retweeting or reply, the window of interest will be 15 tweets before the original tweet till five tweets after the user's action. When the user is just posting a tweet without referencing any history ones, the window of interest will be considered 15 tweets before and five tweets after the user's action, respectively. Twitter API does not reveal the exact date of favoring a tweet. In the case of favoring a certain tweet, the window of interest is chosen to be 15 tweets before and five tweets after the original favored tweet.

Figure 4.2 shows the different cases for choosing the window of interest in the user's timeline. This filtration step is applied to the tweets before the classification step in both training and testing. The filtered out tweets are still used in training the topic model and calculating the user activities.

### 4.7.3 Tweets Pooling

The tweets pooling process aggregates semantically similar tweets into one pool. Each pool is treated as a document. We first began by aggregating each tweet and their replies into one pool. Then, for each hashtag, we aggregated all the tweets that are sharing the same hashtag. Finally, we aggregated the tweets that contain the same URLs in their content. This pooling process decreases the number of documents and increases the document size to be the size of the aggregated tweets.

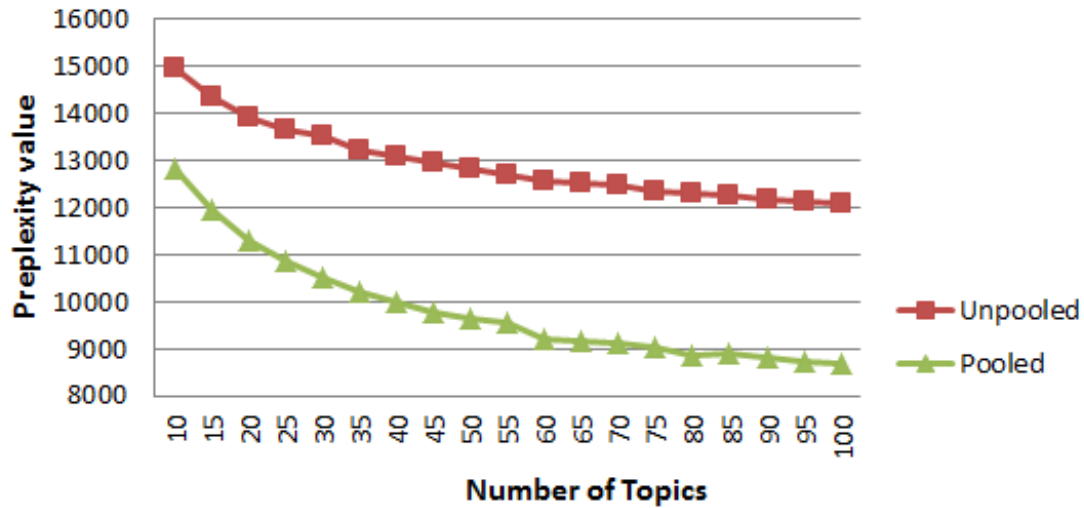


Figure 4.3: Perplexity for LDA and our model

Table 4.2: Characteristics of different pooling scheme

Pooling scheme	Number of pools	Largest pool size
Unpooled	5741434	1 tweet
Replies pooling	5660386	51 tweets
Hashtags pooling	4688744	21483 tweets
URLs pooling	4546896	3364 tweets

This pooling process decreases the number of documents and increases the document size to be the size of the aggregated tweets. Table 4.2 shows the number of pools generated from each pooling schema along with the number of tweets in the largest pool in each scheme.

#### 4.7.4 Evaluating Topic Models

The unsupervised nature of topic modeling methods makes choosing one topic model over another a challenging task. Topic model quality tends to be evaluated by performance in a specific application. Topic models can be evaluated based on perplexity [123] as a quantitative method. Perplexity is a well-known standard metric used in Information Retrieval field. It tries to quantify the accuracy of a model by measuring how well the trained model deals with an unobserved test data as in Equation 4.4. Perplexity is defined as reciprocal geometric mean of per word likelihood

Table 4.3: Example for top ten words for five topics

Topics	Top 10 words
Politics	tcot obama party house gun america vote president romney vote
Technology	app iphone apple google ipad mobile web ios android facebook
Horoscope	libra love capricorn true horoscope virgo cancer money stars sagittarius
Sports	browns game nfl cleveland football coach team eagles win season
Crime	breaking Boston police scene fire sandy shooting victim shot level

of a test corpus. A lower perplexity indicates a better generalization performance.

$$Perplexity(D_{test}|M) = \exp \frac{-\sum_{d \in D_{test}} \log P(w_d|M)}{\sum_{d \in D_{test}} N_d} \quad (4.4)$$

where  $w_d$  represents words of test document  $d$ ,  $M$  is the topic model,  $N_d$  is the number of words in document  $d$ . The perplexity results of LDA with unpooled data and our model are shown in Figure 4.3. The perplexity of the proposed method is better than LDA without pooling the data. We conducted our experiments using 35 topics, as the improvement in perplexity was low compared to the increase in the runtime. More details are provided in Section 4.8.2.

For topic extraction, we used the MACHine Learning for Language Toolkit (MALLET) [86]. MALLET is a Java based package that implements the LDA model. Table 4.3 shows an example for top ten words for five topics (politics, technology, horoscope, sports, crime).

After the model is trained, it can be used to predict the topics in unseen corpuses. Thus we can now predict topics distribution for every corpus in our database.

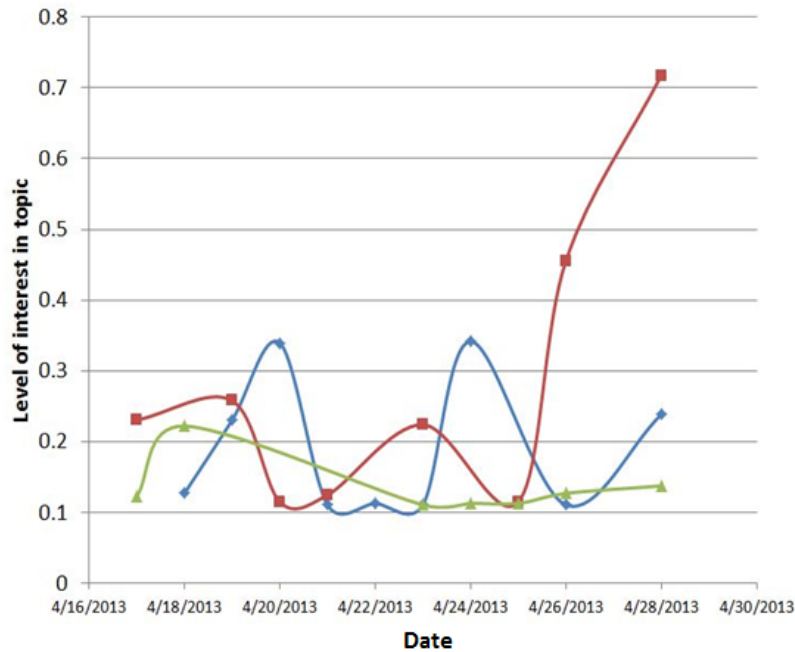


Figure 4.4: User dynamic level of interest in topics

### 4.7.5 Calculating the Dynamic Level of Interest

In real life, the degrees of popularity of the topics are not constant. There are topics that attracts more users than the others. Also, the user's interest in one topic can change from one time to another. Figure 4.4 shows an example of one user's changing levels of activity in some of the topics over time. The dynamic level of interest (LoI) is calculated using Equation 4.2. The user's dynamic LoI is based on the dynamic level of activity of the user in each topic.

### 4.7.6 Personalized Recommender Model

Using the features described above, a feature vector was created for all the tweets in the activity windows of the users, as described in Section 4.5. Each of the feature vectors is augmented by a class value. We considered the only possible class values are *interesting* or *not interesting*. The class value is set to be *interesting* if the user replied, retweeted or favored the corresponding

tweet. Otherwise, the class value is set to be *not interesting*. We used the feature vectors for each user individually to train three classifiers: 1) J48, a Java implementation of the C4.5 tree based classifier [98], 2) supervised Support Vector Machine (SVM), a function based classifier [40], and 3) Naive Bayes Classifiers [88]. The three classifiers are used to predict whether the tweets of the timeline is interesting (the user will most likely interact with) or non interesting.

### 4.7.7 Dynamic LoI and Other Features Effect

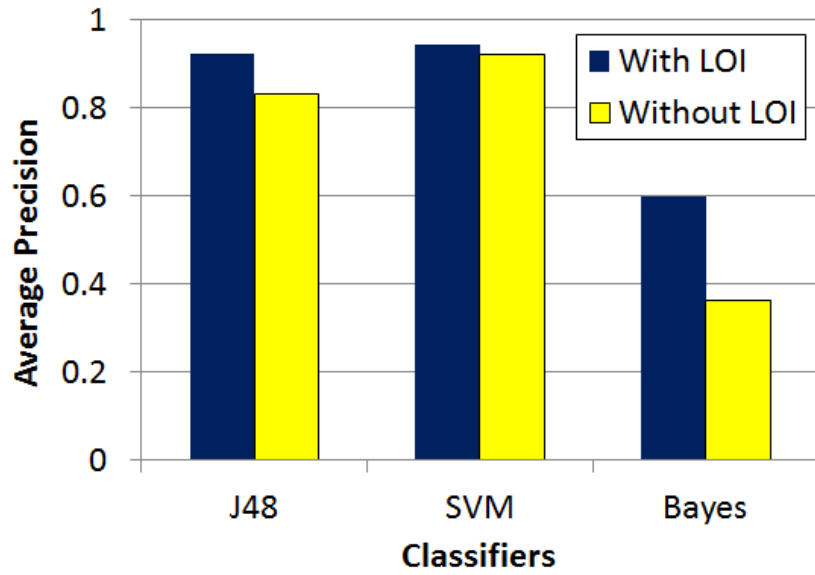
We recorded two quality measures in our experiments: Precision ( $P = TP/(TP + FP)$ ) and Recall ( $R = TP/(TP + FN)$ ) where  $TP$ ,  $FP$  and  $FN$  are the number of true positive, false positive and false negative examples, respectively.

Figure 4.5a shows the average precision values for the three classifiers. Using the *Dynamic LoI* feature improved the average precision of J48 by 8%, and improved that of the SVM and Naive Bayes with about 2% and 36% respectively. SVM is performing better than J48 and Naive Bayes classifiers, when not relying on the *Dynamic LoI*. The best results is achieved when using *Dynamic LoI* feature with either J48 or SVM classifiers. When using the *Dynamic LoI* feature, the J48 and SVM performed equally. The Naive Bayes classifier performed better with *Dynamic LoI*, but not as good as the other two classifiers.

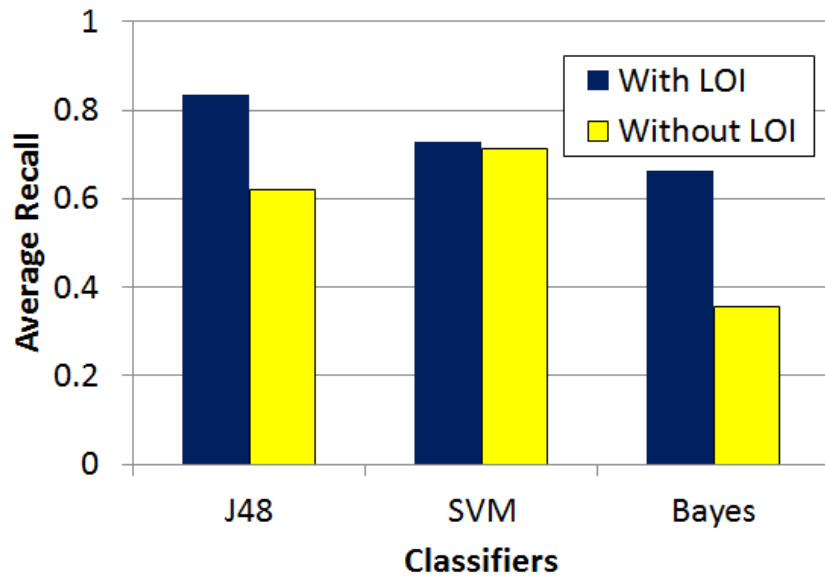
Figure 4.5b shows an improvement in J48, SVM and Naive Bayes average recall by 33%, 3% and 80% respectively. J48 is also out performing SVM and the Naive Bayes classifiers when using *Dynamic LoI*.

## 4.8 Discussion

We had to accurately judge the gain from including the *Dynamic LoI* feature and to determine influence of users who have few tweets, For that we used the concept of ‘active users’ from the traditional media research [82] and focused on those users with some minimum level of activity.

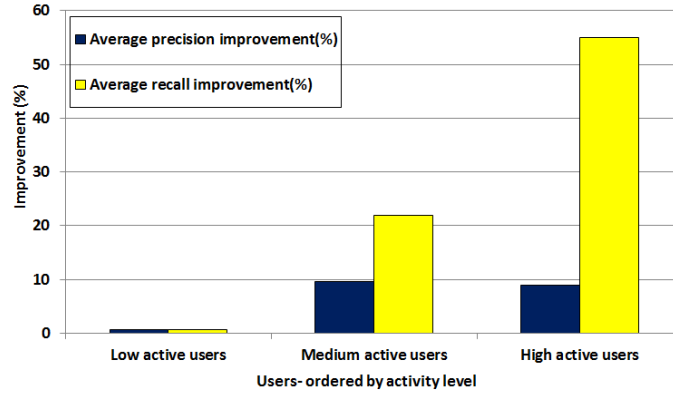


(a) Average precision

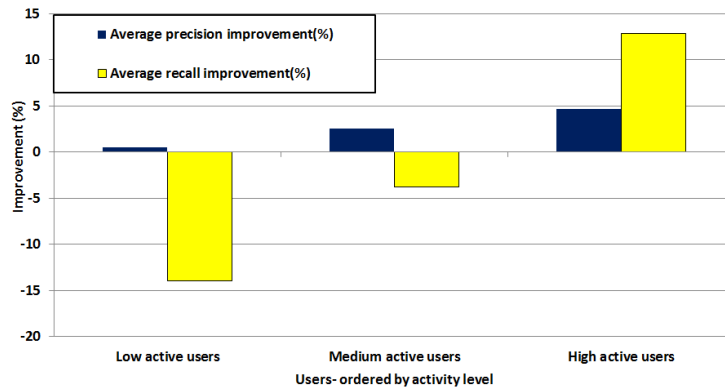


(b) Average recall

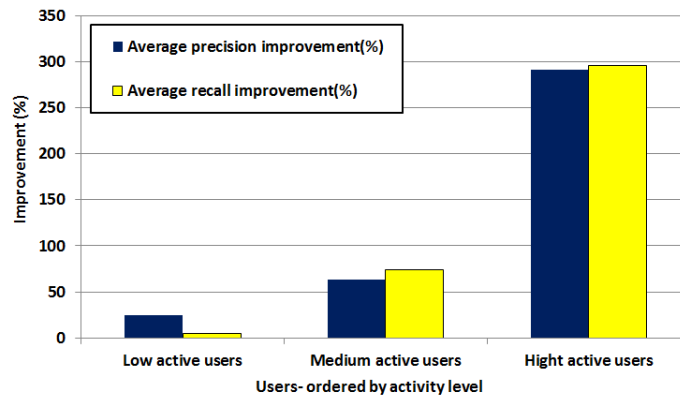
Figure 4.5: Average precision and recall for the three classifiers



(a) J48 — activity level effect.



(b) SVM — activity level effect.



(c) Bayes — activity level effect

Figure 4.6: Average gain in precision and recall with including Dynamic LOI feature

We sorted the users by their activity level in posting, retweeting or favoring the others posts. The users are then divided into three categories according to their activity level (high active, medium active and low active users).

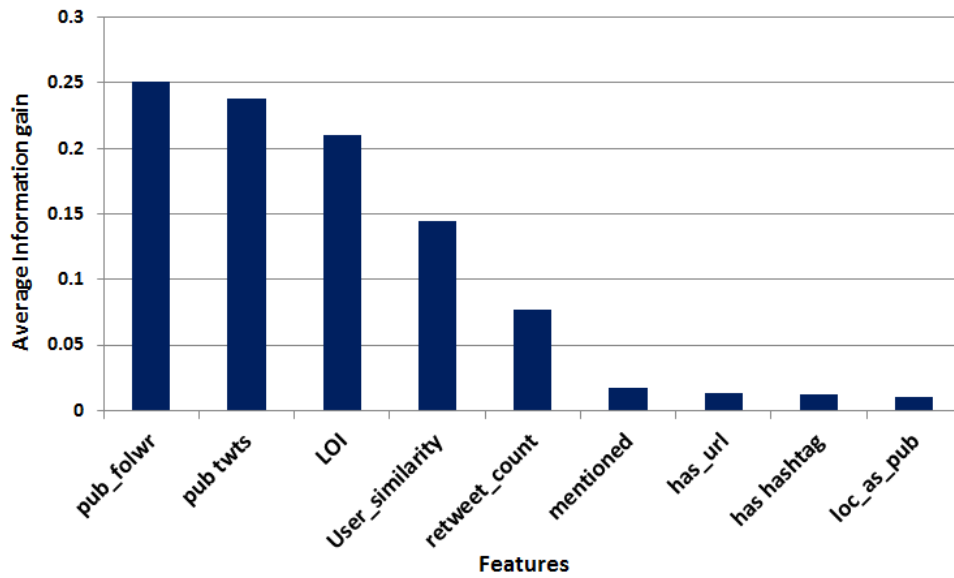
Figures 4.6a, 4.6b and 4.6c show the average gain in precision and recall values in J48, Bayes and SVM classifiers, respectively, when including the *Dynamic LoI* feature. A positive value means that including the feature improved the classification, whereas, a negative value means that including the feature worsen the classification. The gain from including the *Dynamic LoI* feature is higher when considering users with high activity. This makes our model more important for highly active users. The only negative gain is with the SVM classifier for users with less activity. This is intuitive since less active users do not show their interest in the topics as they do not retweet or reply on the tweets.

We had also evaluated the relative importance of other features used in the classification process. We used Information gain feature selection method to measure the dependence between features and the class labels [35]. Figure 4.7a shows the features ranked according to their average information gain. It is clear that the LOI feature is considered one of the relatively highest important feature in the classification process as compared to other features. We analyzed the classification runtime for each classifier. Figure 4.7b shows the runtime for the three classifiers. It is clear that the SVM has the longest runtime compared to J48 and Bayes classifiers (note the logarithmic scale on the Y-axis).

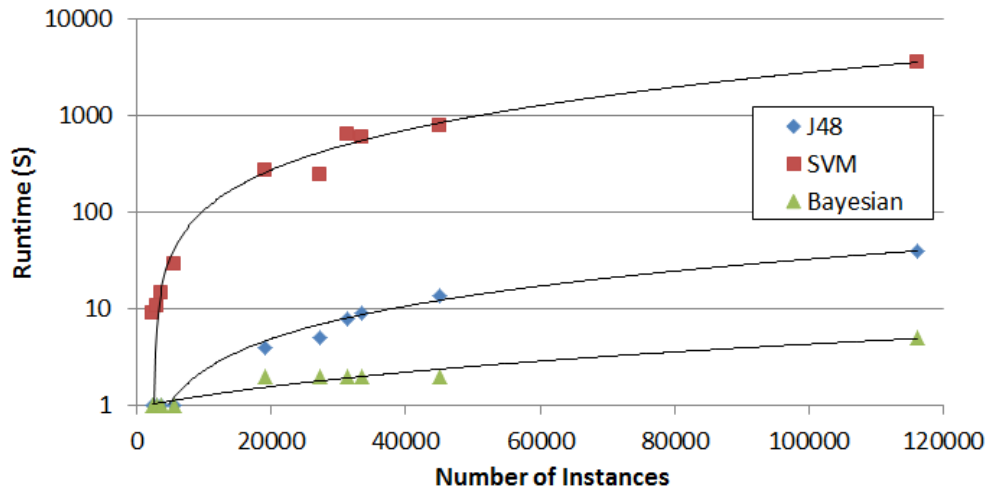
### 4.8.1 Tweets Pooling Effect

Since changing the number of words in the documents can greatly affect the output of the topic modeling step, we repeated the experiments after applying the pooling step. The experiments shows that the average recall was improved by more than 6% without loss in precision.





(a) Average feature information gains across all users.



(b) Classification runtime

Figure 4.7: Classification Analysis

## 4.8.2 Number of Topics Variation Effect

Besides our previous experiments, we analyzed the effect of varying the number of topics on the classification process. For example, a user might be interested in a certain topic, but the classifier only recognize the user's interest in a subtopic. We demonstrated this by re-conducting the experiments with the J48 classifier while varying the number of topics (Figure 4.8). Generally, the small number of topics results in very broad topics. This results in poor classification.

On the other hand, large number of topics will result in many very specific topics, as subtopics become the main categories. This also leads to poor classification in our case. Again, the variation of the number of topics has a minor effect on precision, but the recall was improved by 4% by having around 35 topics. The recall value dropped again, when raising the number of topics to 60. So, although the perplexity value was better at 60 topics than 35 topics, the over specification didn't help in our case.

To clarify, assume that we have tweets about different sports. Ideally, in the generated topics by LDA, there will be a general topic for all sports, such as football, basketball, etc. Using small number of topics will lead to very broad topics. Continuing in the same example, the sports might be merged with other non related topics to form a bigger topic. For example, the topic modeling system (LDA) might merge sports with movies for instance to generate an entertainment topic. So assume there is a user that is only interested in sports. This will make our system mistakenly recommend tweets about movies to the user.

On the other hand, large number of topics means that the sports will be split into more specific topics, for example, football, basketball, etc. According to the user timeline, our system might detect his interest in one of these sports, such as his interest in football, but miss his interest in basketball. This explains the rise and drop of performance of our system when number of topics is varying from small to large.

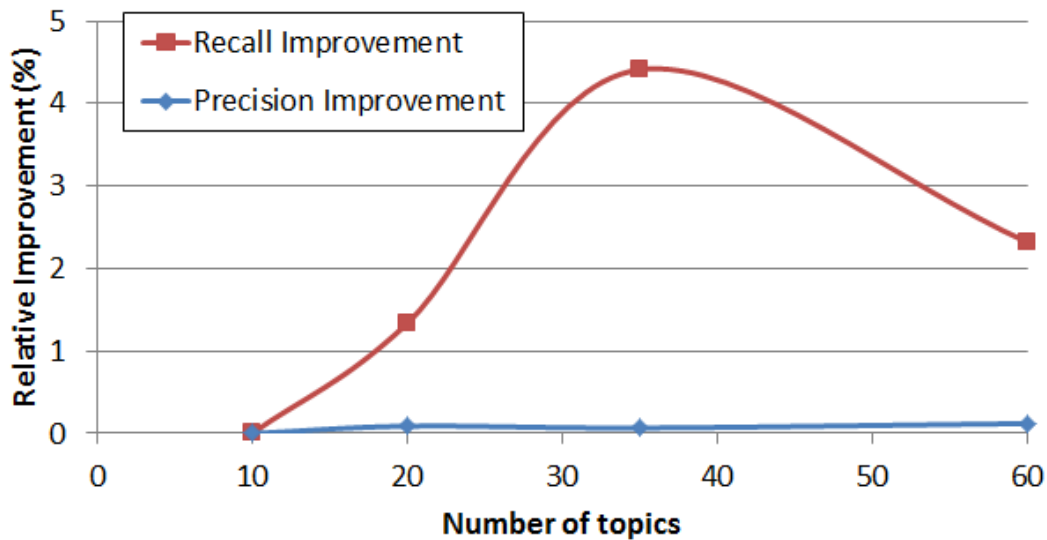


Figure 4.8: Number of topics variation effect effect

## 4.9 Summary

In this chapter, we propose the Tweets analysis subsystem, targeting to provide better subscriptions view for the user. The subsystem studied the user's activities and relationships on Twitter and answered research questions about the individual user's interests:

- How can we infer personal interests from the user's Twitter activities and interactions and to what extent do personal interests change over time?
- What are the other features that can be extracted from the user's Twitter activities and can affect the recommendation made to the user?

Accordingly, we introduced the concept of dynamic level of interest (LoI) for microblogs users. To determine the level of interest of the user in a new corpus, we proposed a model that is based on topics in that corpus and the history of the user activity in each topic. The goal of the model is to identify the important tweets to a user in his/her timeline. We demonstrated the importance of using the Dynamic LoI feature by showing the improvement in the precision and recall of the identified

important tweets. Moreover, the model analysis showed that the model has higher gain for users with high activity level. The subsystem also analyzed the behavior of the LDA topic model to identify the key factors that can affect its performance. We demonstrated that by choosing a proper number of topics and applying pooling techniques to the tweets, an additional improvement to the recommendation can be achieved.

# Chapter 5

## Trends Analysis Subsystem

### 5.1 Introduction

This chapter analyzes how the change in environment can affect the user's experience. Specifically, we study how the change in the geographical community preferences can affect the individual user preferences. Our research focuses on improving the suggestions provided by the social media to the users (trends analysis system in Figure 1.6)

For this, we propose TrendFusion, our model for enhancing the suggestions provided by the social media to the users. The model is used for predicting localized trends' diffusion from one localized community of users (location) to other geographically separated communities of users.

TrendFusion model relies on the information cascade concept to represent the flow of a piece of information, usually called the contagion, through a social network [32]. The cascade is usually represented as a directed acyclic graph (DAG). Figure 5.1 shows an example of information cascade, where:

**Nodes:** The entities (such as users, groups or cities) that represent locations in our model.

**Edges:** Represent the information propagation between entities.

**Seeds:** The vertices that initiate the cascade.

**An activation step (or a step):** Every time a given trend appears at the same time at one or more entities corresponds to an activation *step*, or simply a step, in the cascade.

**A Cascade:** A sequence of activation steps generated by a contagion process. The weights on the edges represent the influence of an active entity on an inactive one. The way to calculate these influences and how an inactive node responds to them are specific to each model.

The following section describes the TrendFusion framework.

## 5.2 TrendFusion Framework

The two main objectives of TrendFusion are:

1. Predict whether a trend will appear for some location based on its diffusion in other locations.
2. Predict when the trend will appear.

The problem we are trying to solve can be defined as:

- Given: A history of spatially and temporally tagged trending topics in a number of locations.
- Processing: Define a model that can extract and capture the dependency relations between locations.
- Output: When a topic is trending in some locations, use the model to predict where and when this topic will be trending next.

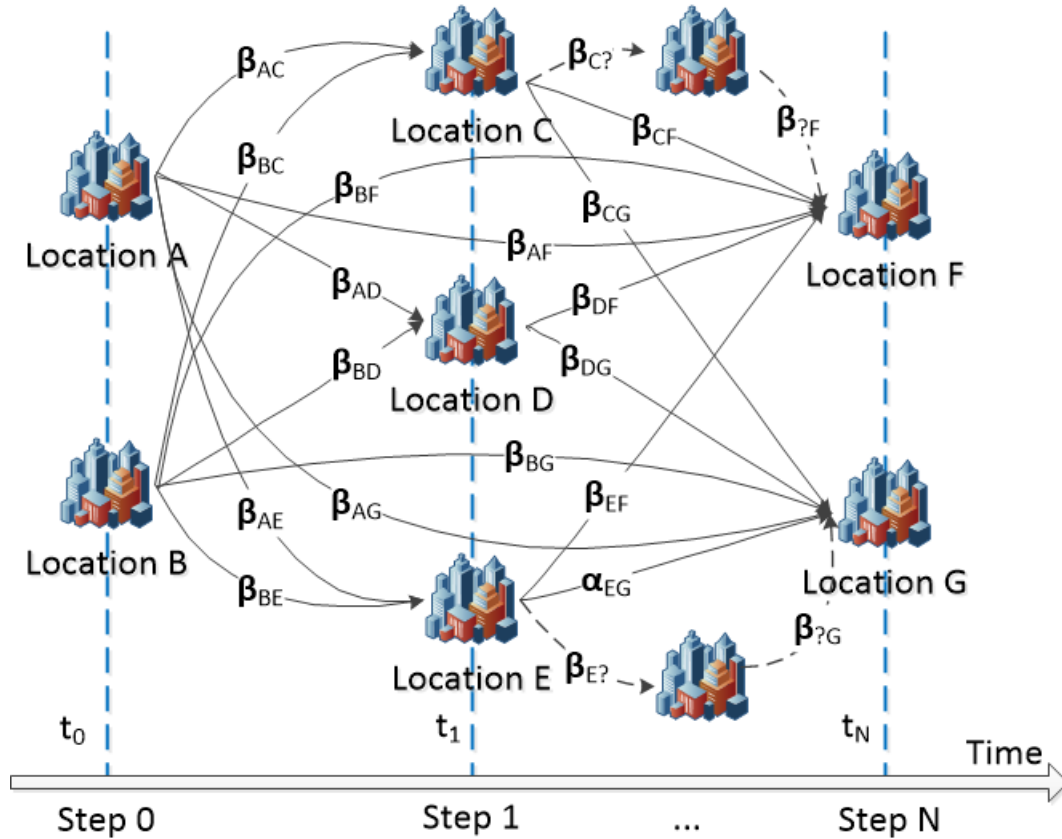


Figure 5.1: An information cascade represented by a Directed Acyclic Graph (DAG).

### 5.3 TrendFusion Model

Generally, most information diffusion models assume that the considered entities (such as users, groups, etc.) are connected by a social graph, and that the graph structure is known beforehand. In our case, there is no such social graph connecting the locations together. Thus, before applying any known diffusion model, we need first to infer the underlying *hypothetical* graph that describes the influence between locations. Fortunately, several network inference models have been developed recently [48, 49, 92]. These algorithms estimate the underlying network structure given past activation times.

In TrendFusion model, we assume a fully connected graph, and estimate the transmission rates along the edges using NetRate algorithm [48]. We based our assumption of the fully connected

graph on the first law of geography by Tobler [56], “*Everything is related to everything else, but near things are more related than distant things*”. We start with a fully connected graph of the locations and estimate the transmission rate between each pair of locations using NetRate. The lowest transmission rates are then omitted reducing the edges (connections) between the locations. NetRate algorithm estimates the transmission rates, not just a binary on/off value. The algorithm takes the input in the form of information cascades. The NetRate algorithm relies on the survival theory and the concept of *hazard rate* that will be explained shortly [50].

## 5.4 Generating the Hazard Rate Graph

After converting the activations to different cascades of trends between locations, we compute the pairwise hazard function between these locations. The hazard rate is mostly related to the survival theory [50], and can be described as the instantaneous activation rate between two locations  $i$  and  $j$  [48], i.e., how likely is it that location  $j$  will adopt a trend at time  $t_j$ , if location  $i$  adopted that trend at time  $t_i$  (Equation 5.1).

$$H(t_j|t_i; \lambda_{i,j}) = \frac{f(t_j|t_i; \lambda_{i,j})}{S(t_j|t_i; \lambda_{i,j})} \quad (5.1)$$

Here  $f(t_j|t_i)$  is the conditional likelihood of transmission from location  $i$  to location  $j$ . Likelihood depends on the activation times  $t_i$  and  $t_j$  (i.e, the time the trend first appears in location  $i$  and location  $j$ ), and a pairwise transmission rate  $\lambda_{i,j}$ . The transmission rate  $\lambda_{i,j}$  models the strength of an edge  $(i, j)$ , and determines how frequently information spreads from location  $i$  to location  $j$ . The most commonly used parametric models for the shape of the conditional transmission likelihood are the exponential, power-law, and Rayleigh distributions models [50].  $S(t_j|t_i)$  in Equation 5.1 refers to the survival function computed for the edge connecting the locations  $i$  and  $j$ . It is computed as the probability that location  $i$  does not cause location  $j$  to activate by time  $t_j$ .



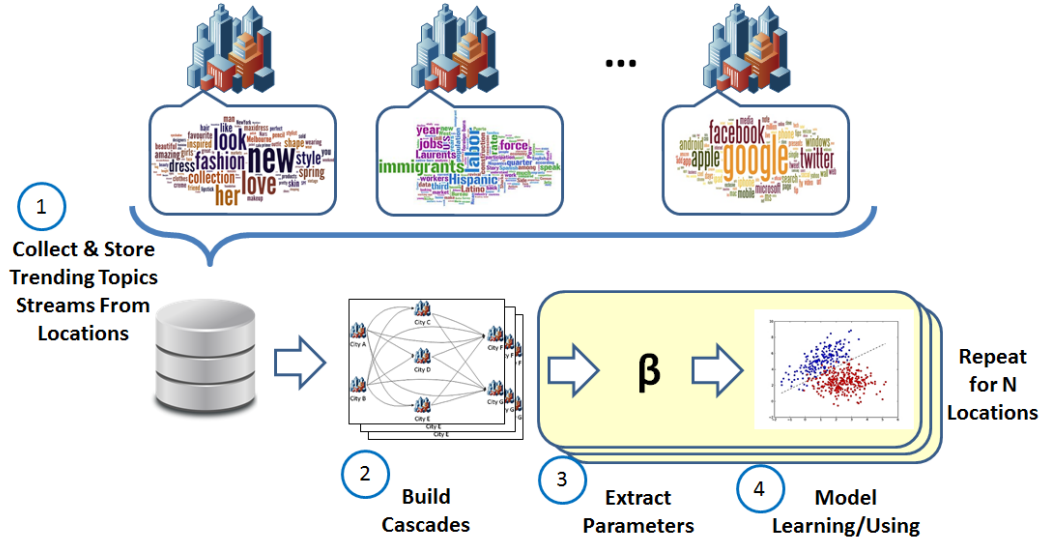


Figure 5.2: The stages of TrendFusion model.

as in Equation 5.2:

$$S(t_j|t_i; \lambda_{i,j}) = 1 - F(t_j|t_i; \lambda_{i,j}) \quad (5.2)$$

where  $F(t_j|t_i)$  denotes the cumulative density function computed from the transmission likelihoods.

## 5.5 TrendFusion Stages

TrendFusion consists of four stages (Figure 5.2). The first two stages can be shared across the locations of interest. Stages three and four should be repeated for each location.

### 5.5.1 Stage 1: Collect and Store Trending Topics Stream from Locations

Trends should be collected from all the locations of interest. The trends are collected every  $\Delta t$  time units. If social media does not reveal the localized trending topics, an extra step of monitoring user activities and extracting the trending topics is needed.

As the stream of the trending topics is received, they are labeled by the location/time they were received from/at. The trending topics are stored for further analysis.

### 5.5.2 Stage 2: Build Cascades

Since trending topics are continuously polled every fixed time step, it is not always clear if a trend is a beginning of a new cascade or a continuation of an old one. Therefore, a process is needed to build cascades from trending topics that are retrieved every  $\Delta t$ . Algorithm 3 provides the details of the cascades building process. The process begins by chronological ordering of all received spatially and temporally tagged trends (Activations List), where one activation represents a record of (*trend, location and time*). The algorithm first determines if an activation should be part of an earlier cascade or it should be considered as a seed for a new cascade. Ferrara *et al.* [41] state that the life time of almost all trends does not exceed 24 hours [41]. Thus we consider a trend to be a seed for a new cascade if it was not trending for more than 24 hours. The algorithm then determines whether or not to consider this activation as a new step. If the location did not appear before in the cascade, then this is a new step. Otherwise, this is considered an update to the location activity times.

### 5.5.3 Stage 3: Extract Parameters

This stage is done for each location. In a given cascade, every location that appears in that cascade will have a distinctive set of parameters. The parameters are calculated mainly based on the diffusion model used as will be explained in Section 5.6. For example, an average distance param-

---

**Algorithm 3** Build Cascades

---

**Procedure** BuildCascadesFromActivations**Input** ActivationsList  $al$ **begin**// An activation  $a$  is a record  $a = (trend, location, time)$ ActivationsList  $alo \leftarrow$  Order  $al$  by time**for all** Activation  $a$  **in**  $alo$  **do**  **if**  $a.trend$  appeared in  $(a.time - 24 \text{ hours})$  **then**     $cas \leftarrow$  last cascade of  $a.trend$     **if**  $a.location$  appeared in  $cas$  **then**      Add  $a.time$  to instances of  $a.location$  in  $cas$     **else if**  $a.time$  equals time of last step in  $cas$  **then**      Add  $a$  to last step of  $cas$     **else**      Add new step to  $cas$  containing  $a.location$     **end if**  **else**    Create new cascade  $cas$     Add new step to  $cas$  containing  $a.location$   **end if****end for****end**

---

eter will be calculated between a given location and all its parents or ancestors depending on the diffusion model. There are four main classes of the parameters:

- Diffusion Parameters (Hazard rate): The value representing the activation rate between any two locations calculated over all cascades.
  - Maximum hazard ( $max\_hazard$ ).
  - Sum of hazards ( $sum\_hazard$ ).
- Geographical Parameters: It is used to examine the geospatial properties of the trending topics spread.
  - Geographical distance between locations ( $shrt\_dist$ ): indicates the shortest distance between locations and whether these distances affect the appearance of trends in these

locations. For this, we have used the Haversine distance, which is commonly used to measure the distance between locations based on the spherical shape of the Earth (as compared to Euclidian distance) [113]. Average distance between locations (*avg\_dist*) are also calculated.

- Coverage (*cvr*): a spread over geographical area of a trend  $S$  at time  $t$ . The area which the trend covers is determined by getting the area of bounding box in which the trend appeared. For the bounding box area, we determined the bounding locations (north east, north west, south east, south west) in which each trend appear. We then calculated the area using the Haversine distance between the boundaries
- Historical Parameters: these parameters describe the path characteristic of each trend through all locations. Their values are based on previous cascades.
  - Trending topics similarity between locations (*sim<sub>tt</sub>*) [64]: the similarity parameter is used to measure the trending topics similarity between locations. We used the Jaccard coefficient between the sets of trends observed at each location, as shown in Equation 5.3:

$$sim_{tt}(location_i, location_j) = \frac{|M_{location_i} \cap M_{location_j}|}{|M_{location_i} \cup M_{location_j}|} \quad (5.3)$$

where  $M_{location_i}$  is the set of trends appeared in  $location_i$ . A similarity score of 1 means that all trends are common between the two locations. A score of 0 means that no trends are in common between the two locations. Average similarity is calculated over all trends.

- Average gap (*avg\_gap*): for each trend appearing between two locations, the gap is calculated as the time difference between the end time in location  $i$  and its appearance in location  $j$ . It is calculated over all trends.
- Overlap time: for two locations  $i$  and  $j$  the overlap time is calculated as the differ-

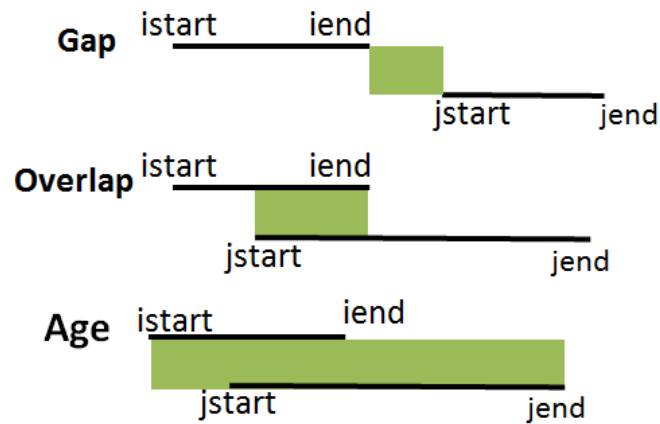


Figure 5.3: Time tracking of trends' appearances in locations  $i, j$

ence between trend's end time in location  $i$  and its appearance in location  $j$ , given that ( $t_{i.end} > t_{j.start}$ ). Average overlap time is generated over all cascades.

- Average trend age (*avg\_age*): the average time of trend's presence in the social network.

Figure 5.3 illustrates calculating time differences between trends' appearances in locations.

- Trend Parameters: Information about the relationship between locations based on the current cascade.
  - Trend's rank (*sum\_rank*): As Twitter provides a trends box that contains the top 10 trending topics, ranked according to their popularity. The trend's rank differ when the trend list is updated every 5 minutes.
    - \* Maximum rank (*sum\_rank*): The highest rank reached by each trend in each cascade. The sum of trend's ranks over all cascades is also computed.
    - \* Weighted sum of trend's rank (*weighted\_sum\_rank*): It indicates whether or not the trend's rank has effect on the transmission rate. It is calculated as a sum of trend's ranks multiplied by the hazard rate between two locations.
  - Number of parents / ancestors (*num\_parents / num\_ancestors*): The number of parents and ancestors' locations for each location/cascade.

#### 5.5.4 Stage 4: Model Learning/Using

As locations are different, a distinct predictive model is needed for each location. The model should learn the parameters extracted from the previous stage and should be used to predict if a new cascade will appear in that location. For this, we utilized two diffusion models. We first present our information diffusion model, the Snowball Cascade (SC) model. We then use the widely used General Threshold model (GT) as our baseline. The differences between the two models will be described in details.

### 5.6 Snowball Cascade Model

The central part of TrendFusion is a new cascade model, Snowball Cascade (SC) Model. Conceptually, as any other information diffusion cascade model, the SC model tries to predict whether or not a certain piece of information will get adopted by different nodes in a social network. Generally, there are three types of nodes: active, contagious, and inactive. Given a piece of information, *inactive* nodes are those nodes that did not adopt that information yet, *active* nodes are the nodes that adopted it already, and the *contagious* nodes are the nodes that are trying to influence other nodes of adopting it. Initially, other than the seed nodes, all the other nodes are considered inactive. The seed nodes are those nodes that initially introduce that information to the network. At the beginning of the cascade, seed nodes are activated. Once a set of nodes is activated, they become contagious, and will always be contagious, i.e., it will keep trying to influence other nodes. The rationale behind the continuous influence is simple: as long as a topic is trending in a location, this interest can affect other locations. Thus in the SC model, the number of active nodes in the system that are trying to spread the influence will grow over time. Active nodes try to influence other nodes which, if activated, become contagious and try to influence other nodes, and so on. This snowball effect is the reason behind the model name.

The SC model is different from the widely used Generalized Threshold cascade model (GT) [67, 94]. In GT model, contagious nodes try to collectively influence other inactive nodes. But once

they are done, they are no longer contagious, i.e., they will no longer try to influence other nodes.

Yet another difference between the two models is that in SC model, the edge weights are vectors rather than scalars. The vector values change from one activation to the other. This is different from the GT model, where the edge weights are required only to be fixed scalars. The vectors on the edges represent the set of parameters that might affect the influence between a contagious location and inactive location at a given step of a cascade.

Figure 5.4 shows an example of two steps for four nodes in SC and GT model respectively. In SC model (Figure 5.4a), two nodes are contagious, both trying to influence the two inactive nodes. The  $\beta$  values on the edges represent vectors containing the influence rates along with other parameters that are described in 5.5.3. The function box in the SC model acts as a binary classifier that takes the  $\beta$  vector values as an input. In SC second step (Figure 5.4b), the contagious nodes remain contagious, and keep on trying to influence other inactive nodes till the end of the cascade.

However, in GT model (Figure 5.4c), two nodes start as contagious nodes, both trying to influence the two inactive nodes. The second step (Figure 5.4d) shows that one of the inactive nodes got infected and became contagious itself, and the other one was not affected. The two contagious nodes in step one, became active in step two. This means that they are already infected but will not try to influence other node anymore. Another difference between the SC and the GT model lies in calculating the influence rate, where in GT the  $\beta$ s on the edges are scalar values representing the influence rates between the corresponding nodes. The function box in the GT model is just a summation operation followed by a condition to check that the sum is below a certain threshold. The threshold is a specific property of each node, i.e. the threshold is different from node to the other. If the sum exceeds that threshold, the node becomes contagious; as in the second step shown in Figure 5.4d.

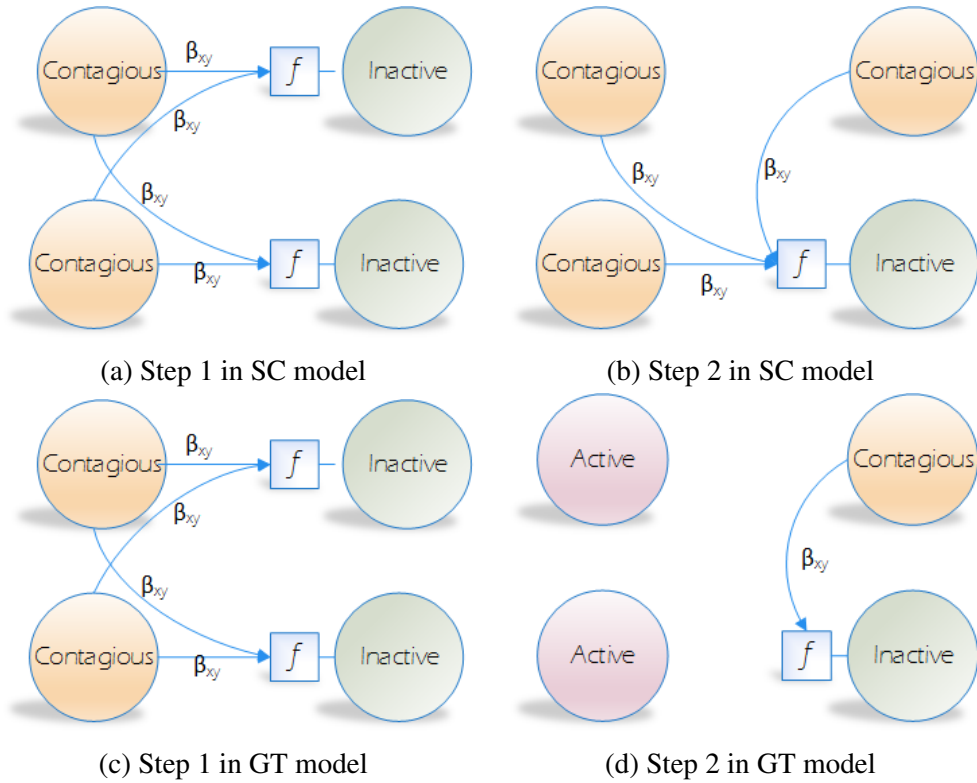


Figure 5.4: Steps of the Snowball and General Threshold model

### 5.6.1 SC Model Definition

Consider a directed graph  $G = (V, E)$ , where  $V$  is the set of vertices representing locations, and  $E$  is the set of weighted edges, with weights  $\beta_{uv}^t$  of edge  $e_{uv} \in E$  representing the influence rate from location  $u$  to location  $v$  at time step  $t$ . Let  $N_v$  be the set of vertices with edges going into  $v$ , and  $S_t$  be a subset of  $N_v$  with the vertices that are active on or before time  $t$ . For every vertex  $v$  there is an activation function  $f()$ , such that at time  $t$ , if  $f(\beta_{u_0v}^t, \beta_{u_1v}^t, \dots, \beta_{u_nv}^t) > \theta_v \forall u_i \in S_t$ , vertex  $v$  becomes active at time  $t + 1$ . The value of  $\theta_v$  can be learned for each location by a binary classifier.



### 5.6.2 GT Model Definition

Consider a directed graph  $G = (V, E)$ , where  $V$  is the set of vertices representing locations, and  $E$  is the set of weighted edges, with weights  $w_{uv}$  representing the influence rate of the edge  $e_{uv} \in E$  from location  $u$  to location  $v$ . Let  $N_v$  be the set of vertices with edges going into  $v$ , and  $S_t$  the subset of  $N_v$  active at time  $t$ . For every vertex  $v$  there is an activation function  $f()$ , such that at time  $t$ , if  $f(S_t) > \theta_v$ , vertex  $v$  becomes active at time  $t + 1$ . In the original model, the value of  $\theta_v$  is randomly chosen from a uniform distribution in the interval  $[0, 1]$ . In our evaluation, we rely on statistical classifiers to estimate the likelihood value of  $\theta_v$ .

The GT model can be considered as a special case of the SC model, where the  $\beta$  vectors are reduced to a fixed scalar (influence rate), and the  $\beta = 0$ , for all nodes that are already active before time  $t$ .

## 5.7 Evaluation

We now describe the methodology used to generate our dataset, and then we describe in details the results of every stage in our model.

### 5.7.1 Trending Topics Dataset

We used Twitter APIs [4] to collect all trending topics appearing on Twitter for a period of 30 days, starting from August 2014 until September 2014, in 48 US locations (cities). Twitter provides a trends box that contains the top 10 trending hashtags or phrases at any given moment, ranked according to their popularity. These trending topics, along with their rank, are updated every 5 minutes. Each user can monitor the trends at the worldwide, country, or city level.

We deployed a crawler to get the trends every 5 minutes for the 48 cities. At the time of the experiment, Twitter updates the trends every five minutes. Consequently, the data retrieval should make

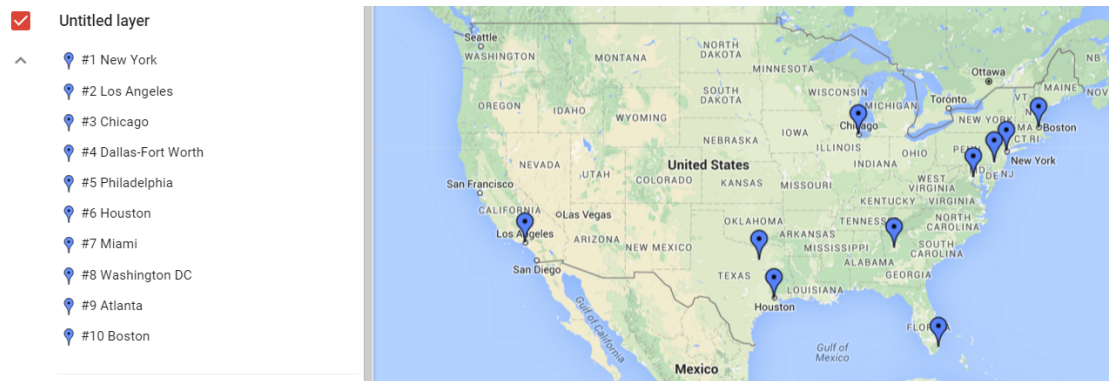


Figure 5.5: 10 Major US cities according to population

this request every five minutes at least. Otherwise, we will be retrieving unnecessary duplicate data.

At the same time, the rate limit for the trending topics API is 15 request per window. To retrieve all the trending topics, we need to issue 3 requests per place per window. Thus we need to make 30 requests for the ten places per window. Since the rate limit for one user is not enough, our trends retriever switches between two users accounts to make these requests. We also collected all trends reported by Twitter for the United States and the whole world. To mask the effect of global trends in our experiments, we filtered out the trends for the cities, that appeared in the U.S. trends or the global trends. We ended up collecting more than 400K different trends.

The data is stored as tuples of the form:  $(woeid, trend_0, trend_1, \dots, trend_9, date/time)$  where *woeid* is Yahoo Where On Earth ID (WOEID) [127] and  $trend_0, \dots, trend_9$  are the top 10 trends. Table 5.1 shows the first 10 US cities used ranked according to their population. The table shows the WOEIDs we used for data retrieval purposes. Figure 5.5 also shows the first 10 out of the 48 major US cities used in our experiments. Figure 5.6 shows the histogram of the distances between the 48 cities, where the x-axis represent the upper limit of each distance bin in miles. Figure 5.5 also shows the major US cities used in our experiments.

As described before, we are also retrieving tweets for the specified locations, in case we need to extract the trending topics from the raw tweets. For this, we are relying on the streaming APIs from Twitter. The major advantage of the streaming APIs from Twitter, is that they are not bounded by

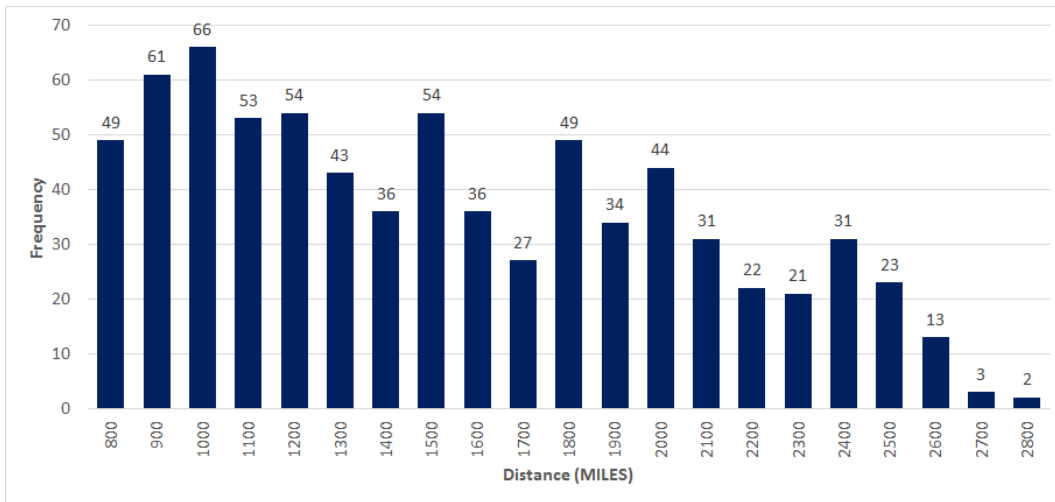


Figure 5.6: Histogram of the distances between the 48 cities

Table 5.1: WOEIDs of 10 major US cities

City Name	State	WOEID
New York City	NY	2459115
Los Angeles	CA	2442047
Chicago	IL	2379574
Houston	TX	2424766
Philadelphia	PA	2471217
Phoenix	AZ	2471390
Washington	D.C.	2514815
Miami	FL	2450022
Boston	MA	2367105
Atlanta	GA	2357024

rate limits. The streaming APIs accepts a list of geo bounding boxes, represented by longitude and latitude of the North East point and the South West point. The query key word field of the API is left blank, thus the API will forward all tweets that are sent within these bounding boxes. The matching first is done using the geolocation, if the tweet is geo tagged. If not, the matching is done based on the location stored in the user profile. Table 5.2 shows the geo bounding boxes used for data retrieval in our data set.

Table 5.2: Geo Bounding Boxes of 10 major US cities

City Name	State	Geo Bounding Box
New York City	NY	(-74, 40), (-73, 41)
Los Angeles	CA	(-118.95, 32.8), (-117.65, 34.82)
Chicago	IL	(-88.26, 41.47), (-87.52, 42.15)
Houston	TX	(-95.78, 30.93), (-94.96, 31.59)
Philadelphia	PA	(-75.28, 39.87), (-74.96, 40.14)
Phoenix	AZ	(-113.33, 32.51), (-111.0, 34.5)
Washington	D.C.	(-77.12, 38.8), (-76.91, 38.99)
Miami	FL	(-80.87, 25.14), (-80.12, 25.98)
Boston	MA	(-71.19, 42.23), (-70.81, 42.45)
Atlanta	GA	(-84.85, 33.5), (-84.1, 34.19)

### 5.7.2 Using TrendFusion Framework

As mentioned earlier, the steps presented in Algorithm 3 are used to convert the data collected from previous step into cascades. We then use the MATLAB® implementation of NetRate algorithm [47] to build the influence graph for all locations. This implementation assumes linear DAG for cascades, i.e., it assumes that each step in the cascade consists only of one location. However, the Snowball Cascade model allows multiple locations per cascade step. So the algorithm was modified slightly to account for this difference. The modified NetRate is used to generate three graphs, one for each assumed distribution for the hazard rate. The graphs from NetRate are then used with the cascades to generate the training and testing examples for each location.

For each location, we generate the training file containing the examples for the first 22 days of the data and a testing file containing the remaining data. The extracted parameter was based on

the Snowball Cascade model. We also used the GT model as a baseline, so training and testing data was also generated for it. Each of the parameter vectors is augmented by one class and one dependent variable.

Given a cascade, when generating the training examples for a location, an example is generated for each step in the cascade before that location appears in it. For example, if a location appeared in step  $n$ , we generate  $n - 1$  examples for each step before that location appeared. If the location doesn't appear in the cascade, then the number of examples generated will be equal to the number of steps in the cascade. The class values are set to be the *appearing* or *not appearing*, depending on whether or not the location appeared in the cascade. If the class value is appearing, then the dependent variable value is set based on the lag value between the time at the cascade step and the time the trend appeared in the location.

We used the parameter vectors for each cascade for individual trends to train three classifiers:

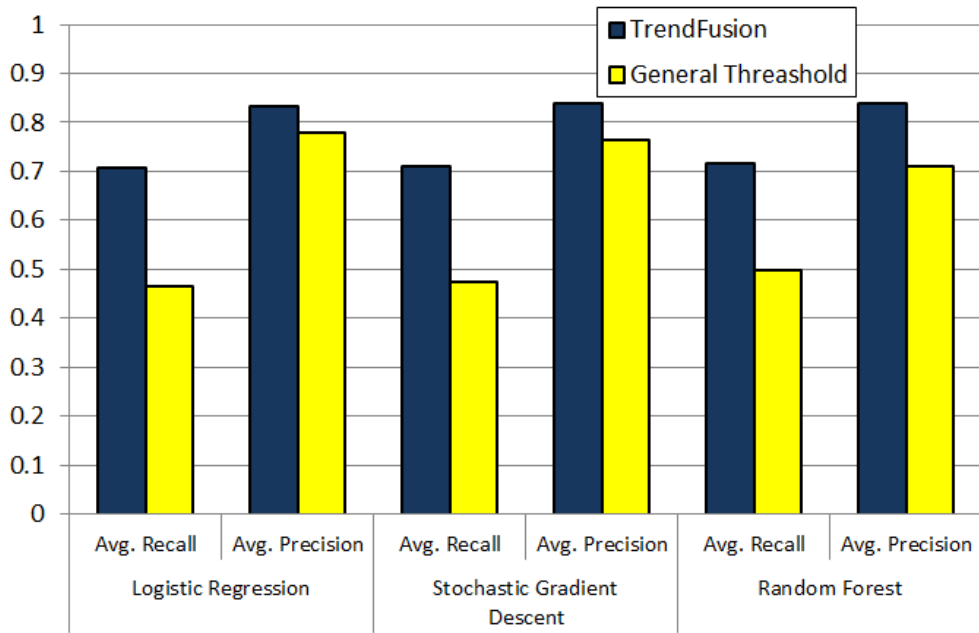
- Logistic Regression (LR), a probabilistic statistical classification model [88].
- Stochastic Gradient Descent (SGD) classifier [45].
- Random Forest (RF), ensemble learning method classifier [20].

We used Weka [52] and R [99] statistical packages to train the three classifiers and afterwards use them to predict whether or not the trend will appear in the designated location.

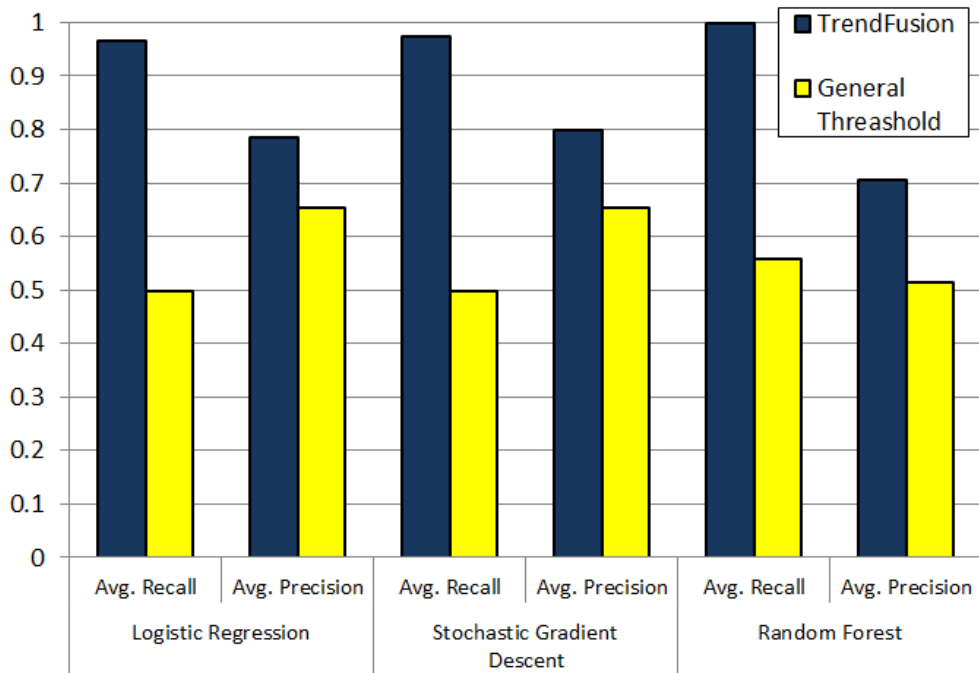
### 5.7.3 Experiments

The evaluation includes five experiments:

- Predict trends based on individual steps.
- Predict trends considering each cascade as a whole.
- Determine the effect of each parameter on the classification process.



(a) precision and recall- cascade steps



(b) Precision and recall- all cascades

Figure 5.7: Average precision and recall for TrendFusion and GT models considering cascade steps and all cascades respectively

- Determine the average time a topic can be predicted to be trending before it actually does.
- Predict when a trend will appear.

#### 5.7.4 Results and Discussion

We evaluated the performance of TrendFusion by running our training and testing examples through the three classifiers. Each example represents a step in a cascade. We used the widely adopted GT model as a baseline to compare its performance with TrendFusion. We recorded two quality measures in our experiments, *recall* and *precision*.

Here *recall* is the ratio of the number trends we were able to predict to the total number of actual trends. Similarly, *precision* is the quality of our prediction, i.e., the ratio of the number of topics that actually become trending in our predictions to the total number of topics we predicted will be trending.

In the first experiment, we considered the output from each individual example. This means that at each step, we take a decision regardless of other steps in the cascades. Figure 5.7a shows the recall and precision values obtained by TrendFusion and the GT models using the three classifiers. It is clear that TrendFusion was giving the same performance across the different classifiers with a recall value of around 0.71 and precision around 0.84 (84% of the predicted trend will be actually trending).

On the other hand, the GT model recall values were in the range between 0.47, 0.48 and 0.5 for the LR, SGD and RF classifiers respectively, which means that it misses around half of the trends. The precision values ranges from 0.71 to 0.78, which means that the slight increase in the recall was accompanied with more false positive predictions. The shows that the GT model is not suitable for modeling the diffusion of trending topics between locations.

In the second experiment, we evaluated each cascade as a whole, getting one decision for the whole cascade. For a given location, we set the class value to be *appearing* for the cascades in which the

location appeared, and *not appearing* for the cascades in which the location didn't appear. The classification is performed on each step, then the predicted values are reduced to one value for the whole cascade. If the predicted value at any of the steps is *appearing*, we consider the combined prediction as *appearing*, as if doing a *logical OR*. The reason behind this way of classification is that the *class* is assigned at each step based on the fact whether or not the location appeared later in the cascade. So at an early step in reality, that might not have any influence on a given location that appeared later in the cascade, the class is still assigned as *appearing*. This is due to the fact that we do not have ground truth data.

A false positive prediction is made in a cascade where a given location didn't appear, if at any step an *appearing* class is predicted. The logic of this classification process is detailed in Algorithm 4.

Figure 5.7b shows the average recall and average precision values for the TrendFusion and GT models for the second experiment with the same three classifiers as before. The average recall values for TrendFusion improved greatly. The wrong *not appearing* predictions made in the first experiment, at the beginning of the cascades that are neutralized in this experiment by a later correct *appearing* prediction. Values for *recall* are 0.96, 0.98 and 0.99 for LR, SGD and RF classifiers, respectively.

On the other hand, *precision* dropped slightly to around 0.8 for the LR and SGD classifiers and to 0.71 for the RF classifier. This also means that one wrong *appearing* prediction at any step of cascade in which a given location did not appear, will cause the overall prediction to be considered wrong. Although, the average recall is slightly improved for the GT model, it still in the range of 0.5 to 0.56 for the three classifiers. The average precision also dropped as expected to the values of 0.65, 0.65 and 0.51 for LR, SGD and RF classifiers, respectively. This still point out that even though that the GT model was good in modeling information diffusion in a social graph at the users level, it is not suitable to model the trending topics diffusion between locations.

These two experiments were conducted using the transmission rates generated by the modified Ne-tRate algorithm assuming exponential distribution. We also examined the two distribution models (power-law and Rayleigh) to decide the shape of the conditional transmission likelihood, and to



---

**Algorithm 4** Classify Cascade

---

**Procedure** ClassifyCascade**Input** Location  $l$ Cascade  $cas$ **begin**

// Determine the class for the whole cascade

 $count_{true\_positive} \leftarrow 0$  $count_{false\_positive} \leftarrow 0$  $count_{true\_negative} \leftarrow 0$  $count_{false\_negative} \leftarrow 0$ **if**  $l$  appears in  $cas$  **then then** $class \leftarrow appearing$ **else** $class \leftarrow notappearing$ **end if**

// Collective classification for all steps

**for all** step  $s$  in  $cas$  **do** $prediction \leftarrow classify\_at(s)$ **if**  $prediction$  is  $appearing$  **then****if**  $class$  is  $appearing$  **then** $count_{true\_positive} \leftarrow count_{true\_positive} + 1$ **else** $count_{false\_positive} \leftarrow count_{false\_positive} + 1$ **end if****return****end if****end for**//  $prediction$  should be  $not$   $appearing$ **if**  $class$  is  $not$   $appearing$  **then** $count_{true\_negative} \leftarrow count_{true\_negative} + 1$ **else** $count_{false\_negative} \leftarrow count_{false\_negative} + 1$ **end if****end**

---

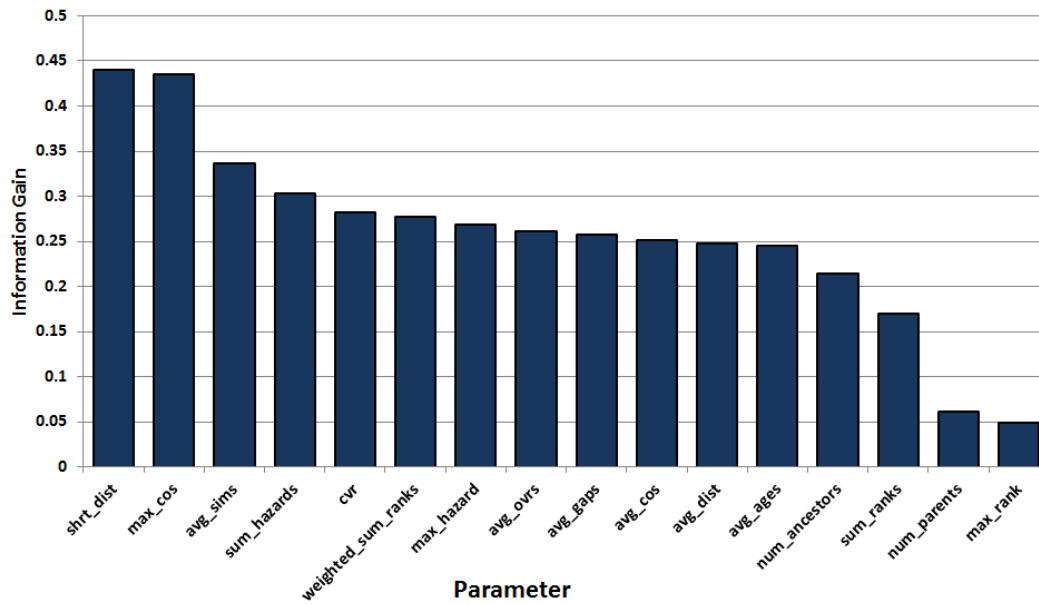


Figure 5.8: Rank of each parameter used in the classification process

analyze the effect of changing them on the classification process. The experiments were repeated using the other two distributions. The results were very consistent with the results obtained for exponential distribution. The variation in the results obtained in all experiments did not exceed 1%.

The third experiment was conducted to measure the effect of each parameter on the classification process. This is achieved by ranking all the parameters according to their average information gain. Figure 5.8 shows the rank of each parameter used in the classification process. We observed that:

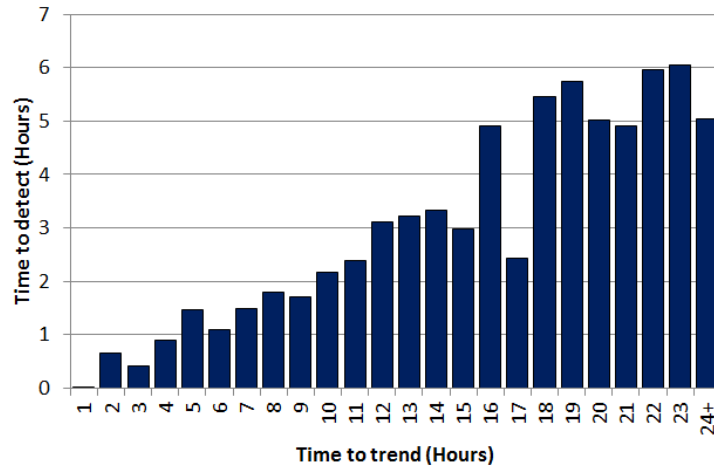
- **Geography matters:** It is clear from Figure 5.8 that locations that are geographically near each other are most likely to influence each other in the social context.
- **The similarity in interests and diffusion parameters are of high importance:** The locations similar in the trending topics in the past, are more likely to have the same trends later on. The locations with high combined diffusion rate to a given location, will have high probability to affect it.

- **Trend parameters are the least important:** Although locations may be influencing each other, the rank of the trending topic in one location is not affecting its rank in the other location. This might be due to the fact that each location has different interests in topics. This also means that it does not really matter in how many locations did a topic appear in, to be influential to other locations, it might just give an indication of how globally important is that topic.
- The remaining parameters were nearly equal with average importance.

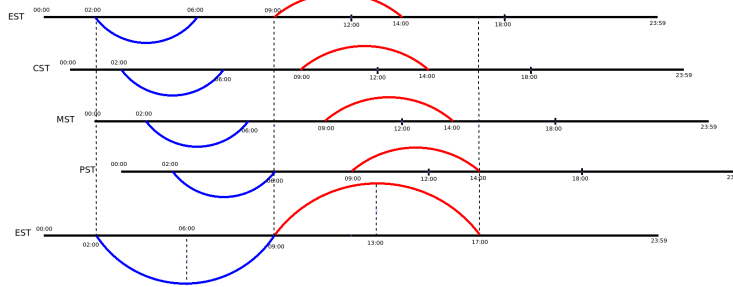
The fourth experiment explored the average time a topic can be predicted to be trending before it actually becomes trending. Figure 5.9a shows the average time before a trend can appear. The  $x$ -axis represents the lag time between the beginning of the cascade and the time a trend will occur. The  $y$ -axis represent the time before a trend is predicted as trending. This shows that we are able to predict the topics on average three hours before they actually trend.

We noticed a drop at hour 17 of time to trend (Figure 5.9a). We investigated the possible reasons for this drop. We found that the number of trends that appeared in new locations after 17 hours are relatively much less than different hours. To find out the reason for that, we used the facts presented by Upbin [121] that shows the average Twitter activity by hour. Upbin showed that the user activity is highest between 9 AM and 2 PM, and lowest between 1 AM and 6 AM. Based on this, we assume that most trends are formed during the high activity intervals. The first four horizontal lines in Figure 5.9b represent different time zones in the US. The upper represent Eastern time, then Central, Mountain, and finally Pacific. The red peaks represent high activity time at each timezone. The blue troughs represent low activity intervals. The lower line represent the combined activities, and it shows that the highest activity in the US happens around 1 PM Eastern, and the lowest activity happens around 6 AM Eastern. The difference between these numbers is 17 hours, thus the trends will not be trended within this gap, hence the drop in number of trends that happen after 17 hours.

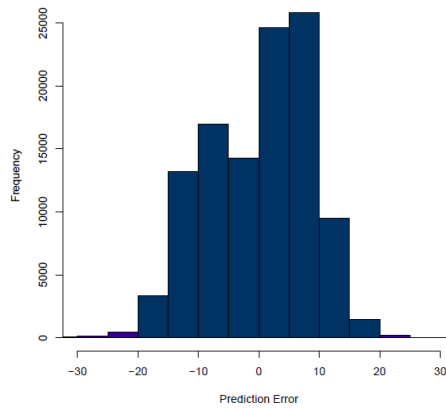
In the fifth experiment we tried to predict when a trend will appear. The training and testing examples in this case are labeled by the time lag between each step and the step at which the trend



(a) Lag analysis for predicted trends



(b) Activity times over 24 hours. Red: highly active window, Blue: low active window



(c) Prediction error histogram

Figure 5.9: Predicted trends analysis

appeared in a given city. We trained a linear regression model and used it to try to predict when will the trend happen. Figure 5.9c shows a histogram where the bins ( $x$ -axis) represent the error

in prediction in hours. The results shows that most of the predictions were around zero error. The bimodal peaks is probably due to the activity windows described in Figure 5.9b, where the high activity interval makes the trends travel faster, and the low activity window makes the trends be delayed in traveling.

## 5.8 Summary

In this chapter, we proposed a trends analysis subsystem, TrendFusion, to provide better suggestions to the user. The model is used to predict the localized trends diffusion in social networks. The developed model allow us to predict whether a trend will be appearing on some location in the future, and if it will appear, when it would appear. We showed that the diffusion models designed for modeling information spread between users are not suitable for modeling trends diffusion across locations, where no real friendship relations exist. The main aspect of TrendFusion is a new information cascade model, Snowball Cascade (SC) model. The model assumes that an activated node in a graph will always be contagious.

We applied our proposed models on trending topics obtained from Twitter for 48 of biggest US cities. We demonstrated the effectiveness and the capability of our model in predicting the time at which the trend will appear. TrendFusion successfully predicted trends before they actually become trending by up to 24 hours.

# Chapter 6

## Personalized Recommendation

### 6.1 Introduction

When an important topic is discussed on Twitter and the tweet is tagged with #, a trending topic is created. Twitter attempts to help users discover what is popular currently by periodically declaring a set of trending topics, keywords/phrases that are being most discussed by users in Twitter [6]. However, Twitter takes into account the global popularity of the topic without taking into consideration the user's personal interests. Among all the tweets within a trending topic, not all the contents are of interest to the user. Therefore, personalization plays an important role in filtering the contents that may not be of interest to the user.

In this chapter, we describe how the recommendation is personalized for the user using his personal preferences, along with the community trends of his interest. Figure 6.1 illustrates the usage of different information sources in our personalized recommendation model.

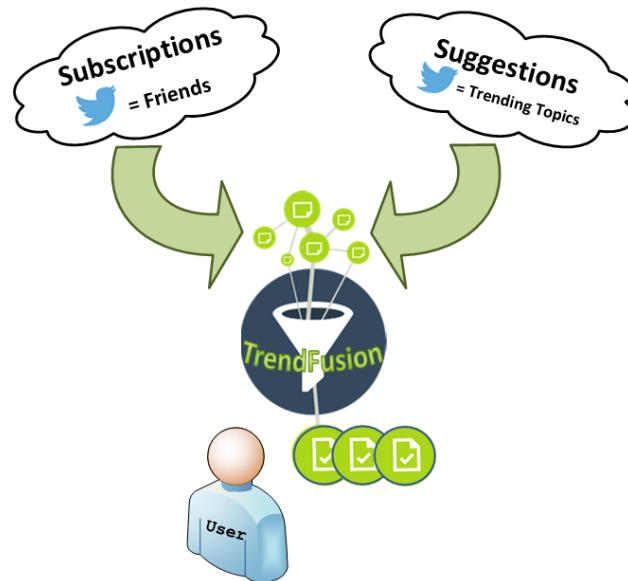


Figure 6.1: Trendfusion sources of information

## 6.2 Multilevel Trends Filtering

As Twitter takes into account the global popularity of the topic without taking into consideration the user's personal interests, our system proposed a multilevel filtering for trends in order to provide the user with personalized trends, along with their related tweets. The multilevel trends filtering stages are as follows:

1. **Geo-Located Community trends filtering:** Different communities have different collective topical interests. So the influence of a trend on a community is affected by the trend's topic. It is expected that a trend related to a topic favored by the community will likely spread in that community. Our model applies trends filtering through finding the trends that are within the topics of interest in the Geo-located community.
2. **User personalized trends filtering:** Similarly, the user's interest in trending topics may differ than those of his Geo-located community. Our model personalizes the trends presented to the user by considering his topics of interest. The model then displays the tweets that are related to the trends of his interest.

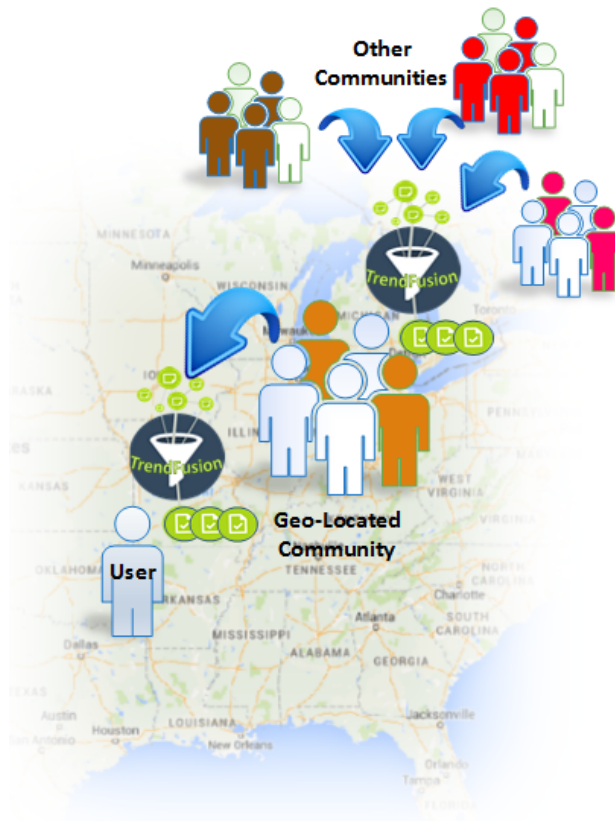


Figure 6.2: Multilevel trends filtering

Figure 6.2 illustrates the multilevel trends filtering stages, where first TrendFusion provides trends of related topics interest to the user's Geo-located community. Then, the user gets his personalized trends and related tweets through TrendFusion's filtering of his Geo-located community trends.

### 6.3 Tweets and Trends Recommendation

In order to personalize the recommendation made to the user according to the tweets and trends of his interest, we relied on the approaches presented in Chapters 4 and 5 to recommend interesting tweets and predict interesting trends. We then integrated these approach to provide personalized recommendation for the user. The following steps describes the integration process as illustrated in Figure 6.3.



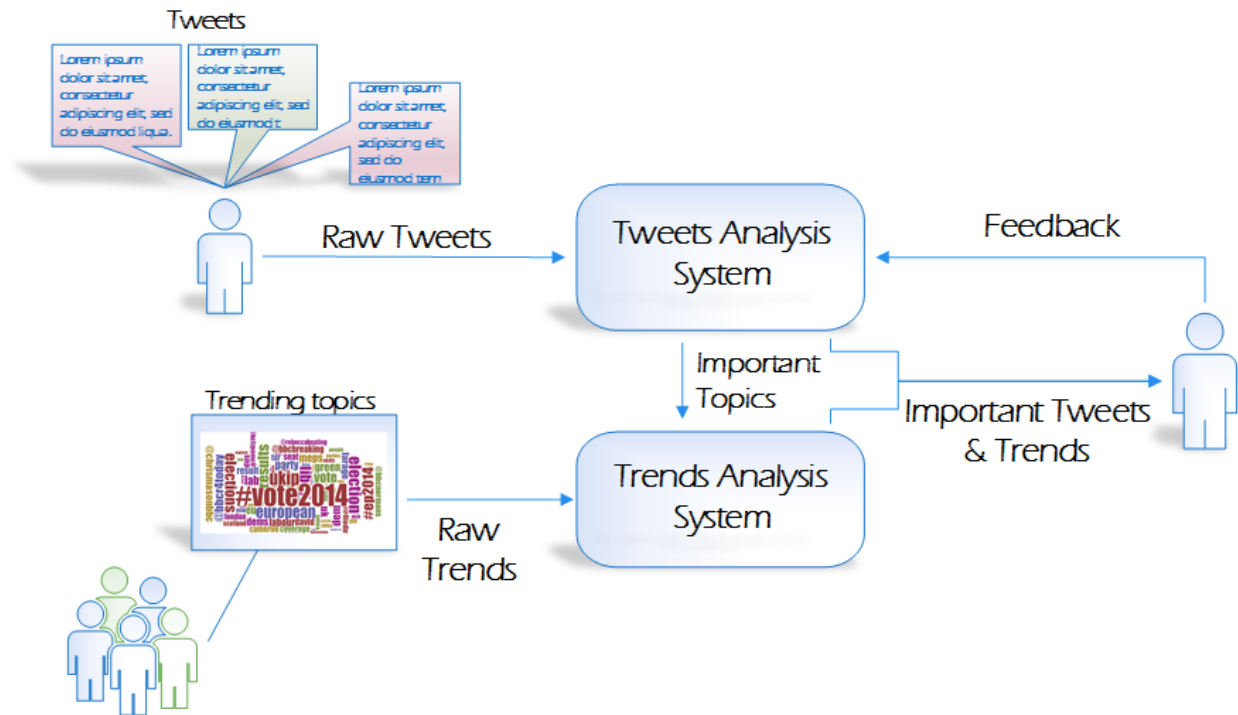


Figure 6.3: The general framework

1. Collecting raw tweets: for each location, all tweets posted for users in this location and during the examined period were collected.
2. Collecting trends: we used Twitter APIs [4] to collect all trending topics appearing in the user's locations of interest.
3. Tweets analysis system: all tweets are passed through our tweet analysis system discussed in Chapter 4. Tweets are first pooled to construct large documents of Tweets. The pooled Tweets are then used to extract topics of interest.
4. Applying topic modeling on tweets: using the algorithm discussed in Section 4.3, we get the distributions of topics in each tweet.
5. Trends Analysis system: the input for the trend analysis system is the raw trends collected from different locations of interest. The trends are passed through the trends analysis system described in Chapter 5. The output is a prediction of important trends and when will it likely

to occur.

6. Getting related tweets for trends: for each trend appearing, we get all the tweets that includes the trend's hashtag or word.
7. Getting related topic for trends: using the previous two steps, we measure the topic interest in each location by getting topic distribution all over the trends.

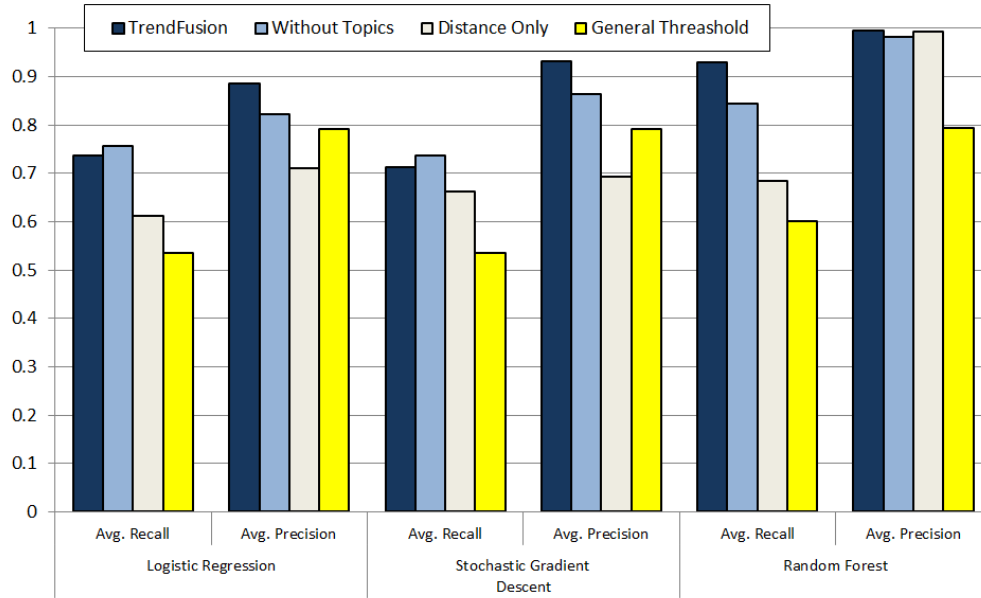
## 6.4 Effect of Topic Modeling on Recommendation

As the trending topic names may or may not be indicative of the kind of information people are tweeting about, so we wanted to measure the effect of applying topic modeling on the trends, and how this can affect the flow of trends between different locations. So we conducted an experiment to measure the effect of similarity of users' topics of interest on the quality of prediction.

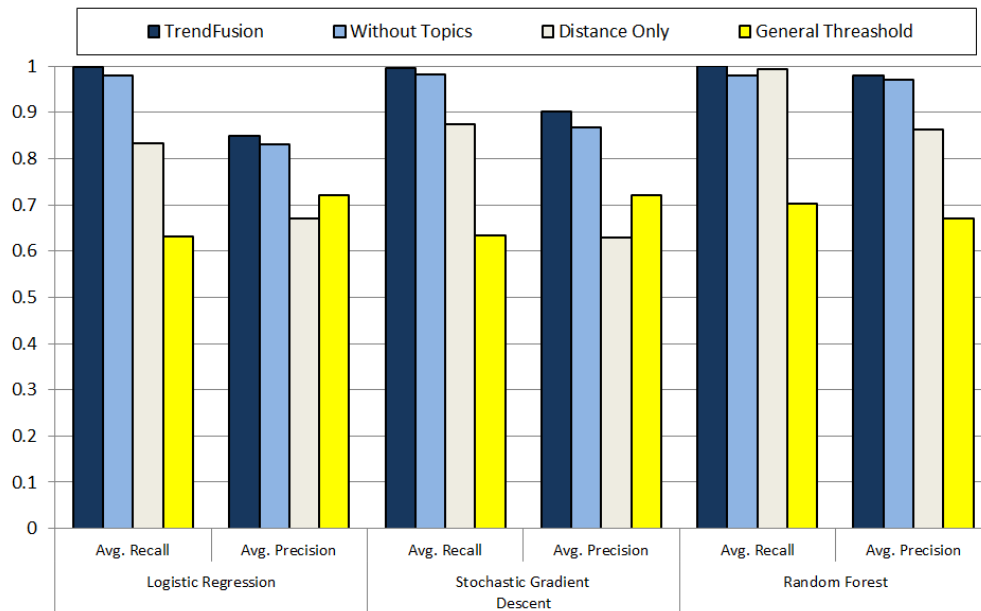
After measuring the topic interest in each location, we included this as a parameter in the prediction model (Section 5.5.3). Figures 6.4a and 6.4b show the recall and precision values obtained by TrendFusion after including the topics features, TrendFusion after including distance features only, and the General Threshold models using the three classifiers considering the cascade steps and all the cascades, respectively. The results show that including the topics as parameter helped in improving the average recall and precision in all of the cases when considering cascades as a whole. Including the topics also helped the precision in all cases when considering individual steps, almost without affecting the recall. Also, the results for using the distance as the only parameter to the models show that the distance is not the only factor that impacts the propagation of trends.

## 6.5 TrendFusion System

To further show the applicability of the concepts presented, we created "TrendFusion" web application (<http://vt.trendfusion.org/>). It provides the user with personalized timeline



(a) Precision and recall– all cascades



(b) precision and recall– cascade steps

Figure 6.4: Average precision and recall for TrendFusion with and without adding topics, with considering distance features only and the General Threshold model

that suits his/her interests. The site analyzes the user's Twitter feed, according to the techniques and theories presented in this work, and predicts the tweets interesting to the user. Figure 6.5 shows the parts of our general framework and how they map into the TrendFusion system.

After registration, users are redirected to Twitter to give permission to TrendFusion to access (read only) their Twitter accounts. This is essential for retrieving the timeline information of the user. When given the permission from the user, TrendFusion retrieves up to 800 of the past tweets in the user's timeline (the maximum Twitter allows to be retrieved from a user timeline). TrendFusion extracts the tweets with user's actions using the window system described in Section 4.7.2.

As described earlier in Chapter 4, the system relies on the user's actions, such as retweets, replies, favorites and posts, to assume the user's interest in a past tweet. The extracted tweets are thus used to train a classifier. TrendFusion builds a unique personalized model for each registered user. The implementation is still relying on Weka package [52] to train an J48 classifier. The choice of J48 is based on its relatively high accuracy and at the same time small overhead, according to the results presented in Section 4.8.

After signing into TrendFusion, the user is given the option to view all the timeline, or to view the interesting tweets only. To keep the user timeline as current as possible, TrendFusion will try to pull new tweets from Twitter every five minutes. When a new tweet is retrieved, TrendFusion will extract all the relevant features as described in Section 4.6. TrendFusion will then load the user classifier model to predict if the tweet is important to the user or not.

The user can also choose to set a specific tweet as important to him or not by clicking on the check box that exists at the right of each tweet. A checked box means that the tweet is important to the user, and an empty box means that the tweet is not important. TrendFusion will update the user models once every day. The user can thus guide the system by checking the important tweets that the system didn't recognize as important, and unchecking the unimportant tweets that the system recognized as important.

The system allows the user to view the Twitter trending topics for 48 US cities and to view the predicted trending topics that will be appearing in the user's chosen location. The suggested trends are

also personalized according to the user's interests discovered from the tweets marked as important by the tweets analysis system. So for each user in the system, his/her interesting topics are passed from the tweets analysis system to the trends analysis system. As we also build a topic model for the trends, we use the user interesting topics to filter and rank the suggested trends by the trends analysis system. For example, in Figure 6.5, an important tweet to the user is reflected in a suggest trending topic that is predicted to appear at the user's city, as marked by the blue arrow.

Before launching TrendFusion web application, trends were retrieved for each of the 48 cities for about a month. These trends are then used to create the hazard rate graph using NetRate algorithm [48] as described in Section 5.4. The collected history trends were also use to build a localized classifier for every city in the 48 cities. The model was build using a SGD classifier with the features described in Section 5.5.3. The features is extracted based on the snowball cascade model described in Section 5.6.

TrendFusion will retrieve Twitter trends for the 48 locations (cities) every five minutes. This is because Twitter caches trends for five minutes. If a shorter interval is used, duplicated sets of trends will be retrieved. Once a set of trends is retrieved for a given location, they are added to the cascades following the steps in Algorithm 3. For the cities that didn't appear in a cascade, run the location localized classifier on the features extracted from that cascade. The user can choose to view the worldwide trending topics, with no predicted trends, or can select one city to see Twitter trends and predicted trends according to TrendFusion.

## 6.6 Summary

In this chapter, we proposed a model that provides personalized tweets and trends recommendation for the user. The model uses the concepts presented in Chapter 4 for tweets personalization. For trends personalization, trends analysis concepts presented in Chapter 5 along with multilevel trends filtering are used. We measured the applicability of our concepts through presenting "TrendFusion" web application system. The system recommends for the user the tweets and trends of his interests.

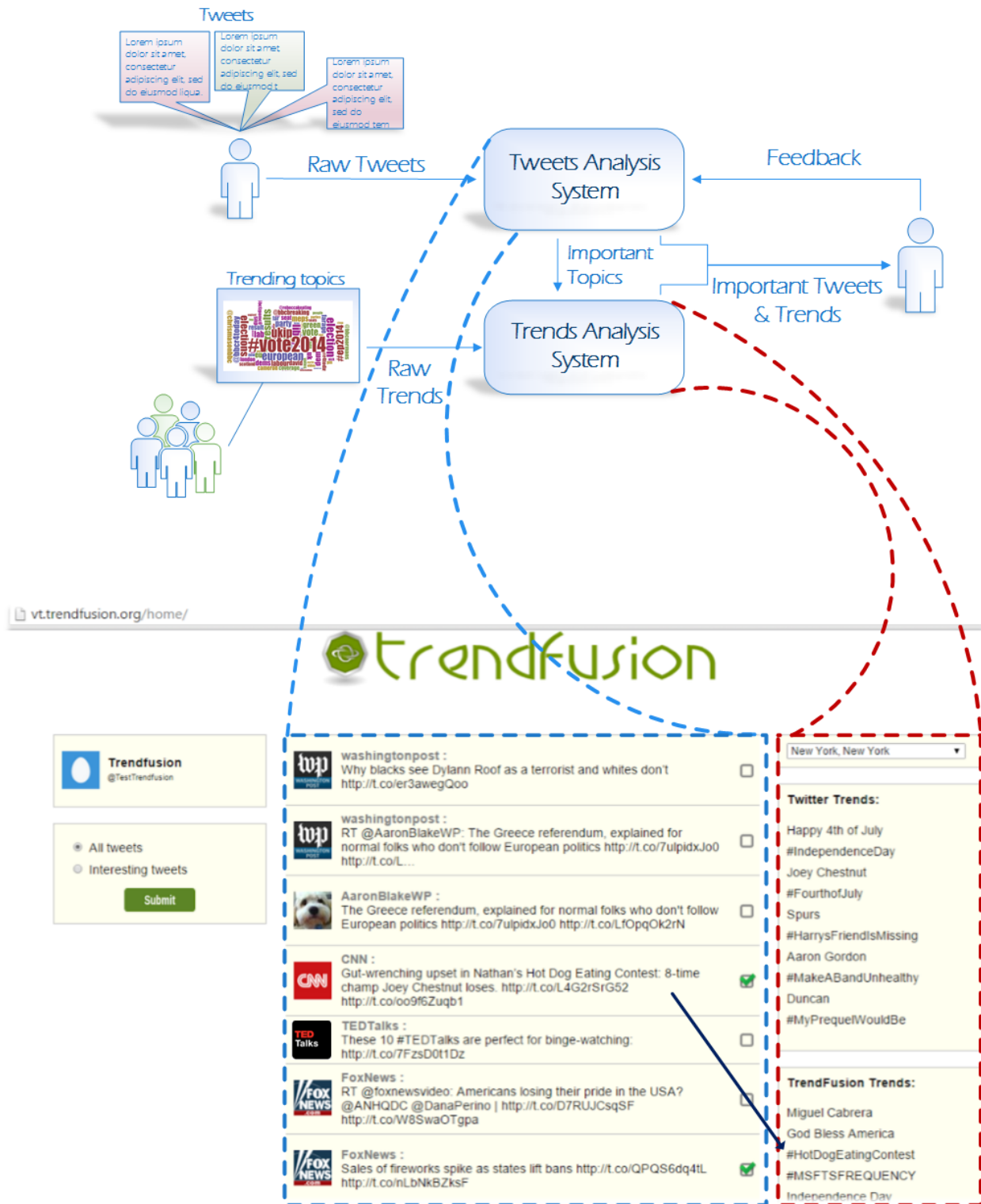


Figure 6.5: TrendFusion system

# Chapter 7

## TrendFusion User Study

### 7.1 Purpose

To assess the applicability of our proposed methods and techniques and to evaluate the accuracy of the TrendFusion system, we also conducted a user study to measure the following:

- How well can the system identify important tweets for a user?
- How precise can this prediction be?
- Is the system able to predict the trending topics in a user selected location?

### 7.2 Method

#### 7.2.1 Participants

We posted several requests on different online research groups asking for volunteers to participate in our study. We requested that the participants have some activity on Twitter. We received the acceptance to participate in the questionnaire from ten graduate students. The ten participants

(four females, six males) were in the age ranging between 25-35 years old. Four out of the ten users reported that they were not so active users (active users are the users who perform actions on Twitter as posting/retweeting/favoriting).

### **7.2.2 Procedure**

The participants (users) were asked to complete the steps presented below in Section 7.3. The users are then asked to heavily use TrendFusion web application for at least three days before providing their feedback. The users provide the feedback through completing a questionnaire related to the TrendFusion performance. This time period during which the user is asked to use the system is important for TrendFusion system to learn the user preferences. The TrendFusion system relies on the users interactions on Twitter such as tweeting, retweeting, replying and favoriting to establish the user's important topics. While TrendFusion system interface does not provide the means to do these actions, it provides the user a way to indicate whether a tweet is important to him/her or not.

During the study, the users were asked to refrain from using other applications or website to access their Twitter accounts. While they were in the system training phase, the users were asked to log into their TrendFusion accounts every hour and go over newly posted tweets. The user was asked to identify the tweets that are important to them, and mark them as important, by placing a check mark next to them using the TrendFusion interface. Additionally, if the user found a tweet that is marked as important by the system, but was not really important, the user marked that tweet as unimportant by removing the check sign next to it.

## **7.3 Users Tasks**

Guided by the design requirements for the TrendFusion system that were presented in Chapter 5, TrendFusion web application system was developed and presented to the Twitter users to obtain a feedback. At the beginning of using the system, the users were asked to follow some steps to allow



them to register into the TrendFusion system. These steps are described in details in Appendix A.1. The system's performance was evaluated according to two basic tasks:

- **Task 1: Check important tweets**

Description: Filter out unimportant tweets and only view tweets that are predicted important.

1. Check the Important Tweets radio button on the left hand side.
2. Click on the Submit button on the left hand side.
3. To mark a tweet on your timeline as important tick on the check box on the right end of the desired tweet.

- **Task 2: Predict TrendFusion trends**

Description: View the localized trending topics prediction in TrendFusion.

1. To view the trending topics in a certain location.
2. Find the closest city to where you live in the list.
3. Click the list box appearing on the right side of the page to choose the location where you want the trend to appear.

Inquiries included in the questionnaire presented to the user are listed in Appendix A.2

## 7.4 Results and Discussion

The study measured the performance and the accuracy of the TrendFusion system. Specifically, during the user study, the observations were taken to answer the following questions:

1. How well can the system identify important tweets for a user?
2. Is the system able to predict the trending topics in a user selected location?

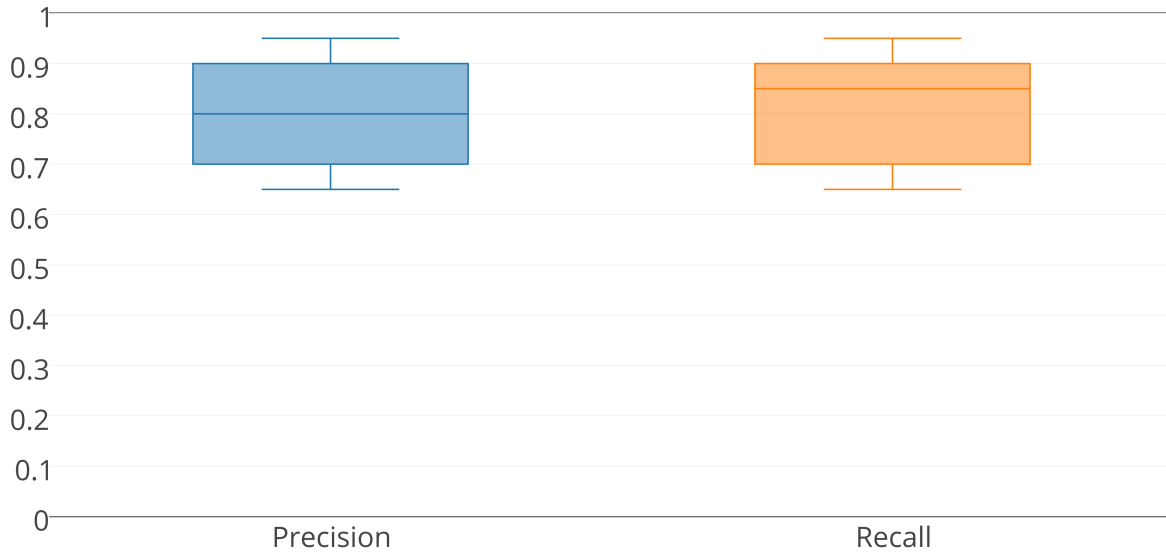


Figure 7.1: Range bars representing precision and recall values as reported by TrendFusion users

### 3. What was the quality of the localized trending topics according to the users?

From the results of the questionnaire conducted, it was observed that most of the users are using other online social networks other than Twitter. The user's activity on Twitter (logging into Twitter) was ranging between several times a week and several times a day. We also observed that most of the users are not enabling the Geolocation settings for privacy reasons. In terms of followers and friends, High active users (posting, retweeting of favoriting tweets) were observed to have more followers and friends than less active users. Figure 7.1 represents precision and recall values as reported by TrendFusion users.

TrendFusion was able to identify on average 80% of the important tweets to the user, with reported precision more than 70%. In general, we noticed from the findings that this percentage of identifying the important tweets is greatly affected by the amount of user activity. In other words, the system will be able to identify important tweets better for users that reported higher number of followers, friends, and with higher activity level (posting tweets, favorites, retweets).

Another finding is the effect of using other languages other than English. We noticed that the

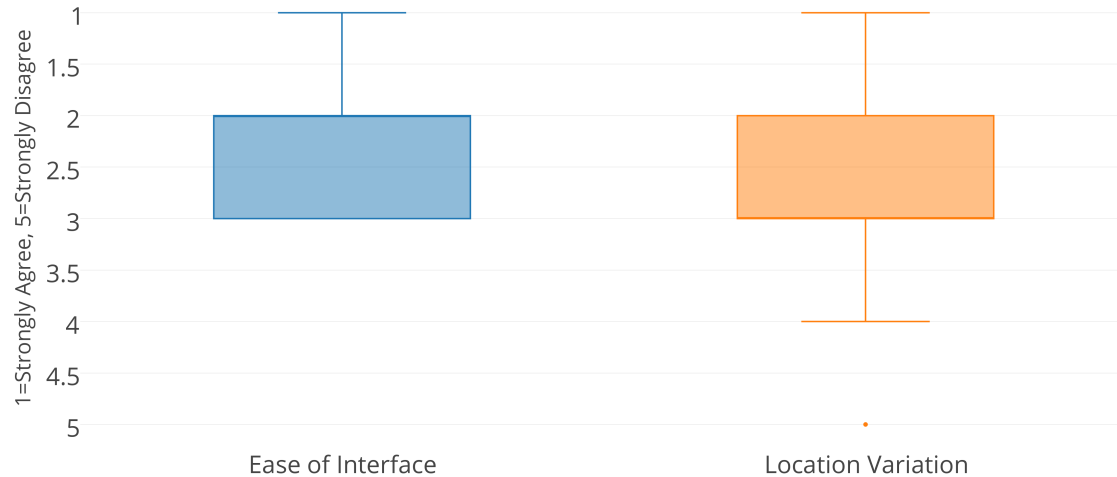


Figure 7.2: Range bars representing the ease of interface use and location variation as reported by users (lower value is better).

ability to identify important tweets was greatly affected by the language used in the user posts and/or the followers.

The next question of interest was to measure the system ability to predict the trending topics in a user selected location. The answer to this question was affected by different factors. As the scope of our study were restricted to only the 48 cities in the US and English language, the users were instructed to choose the cities that are closest to them. When the user geolocation perfectly coincides with one of the 48 chosen cities, the user reported a higher quality of the suggested trending topics by TrendFusion. On the other hand, when the user location is not one of these cities, the user seemed to be less content of the suggested trending topics. Figure 7.2 represents the range bars representing the ease of interface and location variation selection as reported by users. The mean for the ease of interface is 2.2, which means that users found the interface was mostly easy. Also, the mean for location variation is 2.7, which means that users found cities close to their location, but was looking for more variations in choosing the locations they want.

# Chapter 8

## Conclusion

In this research, we introduced the concept of dynamic level of interest (LoI) for microblogs users. To determine the level of interest of the user in a new corpus, we proposed a novel model that is based on topics in that corpus and the history of the user activity in each topic. The goal of the model is to identify the important tweets to a user in his/her timeline.

To illustrate the effectiveness of our model, we used a Twitter APIs to build a dataset with more than five million tweets, and more than 20 thousands users. We demonstrated the importance of using the *Dynamic LoI* feature, by showing the improvement of the average precision and the average recall for the three classifiers used (J48, Naive Bayes, and SVM). Using our approach, we were able to improve the precision and recall of identifying important tweets by up to 36% and 80% respectively. The model analysis showed that the model has higher gain for users with high activity level.

We analyzed the behavior of the LDA topic model to identify the key factors that can affect its performance. We demonstrated that by choosing a proper number of topics and applying pooling techniques to the tweets, an additional 10% improvement can be achieved.

We also proposed TrendFusion, a model for predicting the localized trends diffusion in social networks. Our goal was to develop a model that will allow us to predict whether a trend will

be appearing in a certain city in the future, and if it will appear, when it would appear. We also demonstrated that the diffusion models that are designed for modeling information spread between users, are not suitable for modeling trends diffusion across cities, where no real friendship relations exist. The main aspect of TrendFusion is a new information cascade model, Snowball Cascade (SC) model. The model assumes that an activated node in a graph will always be contagious.

To illustrate the effectiveness of our model, we applied our proposed models on trending topics obtained from Twitter for 48 of biggest US cities. We demonstrated the effectiveness of our model by comparing it to the General Threshold (GT) model, a widely accepted diffusion model. TrendFusion out performed GT model by achieving the recall and precision of prediction of trends by 98% and 80% respectively.

TrendFusion is also capable of predicting the time at which the trend will appear. TrendFusion successfully predicted trends before they actually become trending by up to 24 hours. The root mean squared error (RMSE) in TrendFusion time prediction is less than 6 hours.

To further assess the applicability of our proposed methods and techniques and to evaluate the accuracy of the TrendFusion system, we also conducted a user usability study. This study is used to measure the quality of the system's performance in recommending and predicting the personalized tweets and trending topics.

The research findings provide a foundation to help us understand how to identify locations that can be influential for the spread of a given topic.

Some points learned during our research and that highlights possible future directions include:

- **Data scalability:** With the exponential growth of geospatial and social media data, it is important to address the problem of scalable and high performance computing for big data analytics because many research activities are constrained by the inability of software or tool handle the data volume and computational complexity. Also, heterogeneous geospatial data integration and analytics tremendously magnify the complexity of the problem. Many large-scale geospatial problems are such that most computer systems do not have sufficient

memory or computational power. Nowadays, different computer architectures, such as Intels Many Integrated Core (MIC) Architecture and Graphics Processing Unit (GPU), provide solutions to achieve scalability and high performance for data intensive computing over large spatiotemporal and social media data. Given this, exploring different algorithms to achieve the capability for scalable data processing and analytics over large-scale, complex social media data is needed to process the huge amount of tweets , and to apply our models.

- **Multilingual topic models:** Social scientists, especially sociolinguists, have long been interested in the role language plays in the formation of social networks and in how structures of social networks impact on language practices. Relatively little is known about the role multilingualism plays in forming these networks and how the virtual networks impact on multilingual practices. Understanding the pattern of connections between monolingual and bilingual speakers would not only offer a new perspective on multilingualism on the social media, but also provide new insights into the societal structures and human relations in multilingual societies. Based on this, including multilingual topic models will allow us to better better understand users interests and at the same time allow us to better study information diffusion patterns through different countries.
- **Data collection limitation:** Researchers trying to get data from Twitter are constrained by Twittter API limited data retrieval rates. When trying to access tweets for research purposes, researchers are subject to a limit of 180 API calls every 15 minutes. Researchers trying to gather their own datasets would use Twitters streaming APIs, which returns a real-time feed of tweets posted to Twitter. However, the publicly-available API for that is limited to only a small fraction of total tweets to the service, around 1 percent. There is an API that allows all tweets to the service to be collected, the “firehose”, but Twitter limits access to it and charges a fee that is well outside the budget of most academic research. This in turn leads to extended period of time just to collect a reasonable amount of tweets. Other methods should be allowed for researchers to get around this problem.

Finally, the conducted research has lead to the following publications:

- **Shaymaa Khater, Hicham G. Elmongui, and Denis Gračanin.** Personalized Microblogs Corpus Recommendation based on Dynamic Users Interests. In Proceedings of the 2013 International Conference on Social Computing (SocialCom), Washington, D.C., September 2013 (acceptance rate 9.9%).
- **Shaymaa Khater, Hicham G. Elmongui, and Denis Gračanin.** Tweets You Like: Personalized Tweets Recommendation based on Dynamic Users Interests In Proceedings of the Third ASE International Conference on Social Informatics (SocialInformatics 2014), Cambridge, MA, USA, December 2014 (acceptance rate 14.6%).
- **Shaymaa Khater, Denis Gračanin, and Hicham G. Elmongui.** TrendFusion: Trends and Social Influences Between Geographically Separated Large User Communities. In Proceedings of the 2015 ASE Eighth International Conference on Social Computing (acceptance rate 14.8%).

A journal paper is submitted

- **Shaymaa Khater, Denis Gračanin, and Hicham G. Elmongui.** Enhancing User's Interaction and Experience in Online Social Networks. Targeting IEEE Transactions on Computational Social Systems. In this journal, we are explaining in more depth our overall objectives, our proposed approaches and our major findings.

# Bibliography

- [1] About Twitter. <https://about.twitter.com/company>. [Online; last accessed 13-Oct-2015].
- [2] Facebook companyinfo. <http://newsroom.fb.com/company-info/>. [Online; last accessed 15-Oct-2015].
- [3] Twitter documentation. <https://dev.twitter.com/overview/documentation>. [Online; last accessed 10-Oct-2015].
- [4] Twitter homepage. <http://twitter.com>. [Online; last accessed 10-Oct-2015].
- [5] Virtual Time Square. <http://vts.cs.vt.edu>. [Online; last accessed 10-March-2014].
- [6] Twitter blog: Trend or not trend, 2010.
- [7] F. Abel, Q. Gao, G.-J. Houben, and K. Tao. Analyzing user modeling on Twitter for personalized news recommendations. In *Proceedings of the 19th International Conference on User Modeling, Adaption, and Personalization (UMAP'11)*, 2011.
- [8] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734–749, June 2005.



- [9] N. Agarwal, H. Liu, L. Tang, and P. S. Yu. Identifying the influential bloggers in a community. In *Proceedings of the 2008 International Conference on Web Search and Data Mining (WSDM'08)*.
- [10] Z. Al Bawab, G. H. Mills, and J.-F. Crespo. Finding trending local topics in search queries for personalization of a recommendation system. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12*, pages 397–405, New York, NY, USA, 2012. ACM.
- [11] S. Aral and D. Walker. Identifying influential and susceptible members of social networks. *Science*, 337(6092):337–341, 2012.
- [12] S. Asur and B. Huberman. Predicting the future with social media. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, volume 1, pages 492–499, 2010.
- [13] S. Asur, B. A. Huberman, G. Szabó, and C. Wang. Trends in social media : Persistence and decay. *CoRR*, abs/1102.1402, 2011.
- [14] H. Baars and H.-G. Kemper. Management support with structured and unstructured data-an integrated business intelligence framework. *Information Systems Management*, 25(2):132–148, Mar. 2008.
- [15] L. Backstrom, E. Sun, and C. Marlow. Find me if you can: improving geographical prediction with social and spatial proximity. In *Proceedings of the 19th international conference on World wide web, WWW '10*, pages 61–70, New York, NY, USA, 2010. ACM.
- [16] P. Bao, H.-W. Shen, J. Huang, and X.-Q. Cheng. Popularity prediction in microblogging network: A case study on sina weibo. *WWW '13 Companion*, pages 177–178, 2013.
- [17] C. Basu, H. Hirsh, and W. Cohen. Recommendation as classification: Using social and content-based information in recommendation. In *In Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 714–720. AAAI Press, 1998.

- [18] J. Benhardus and J. Kalita. Streaming trend detection in twitter. *Int. J. Web Based Communities*, 9(1):122–139, Jan. 2013.
- [19] S. Bharathi, D. Kempe, and M. Salek. Competitive influence maximization in social networks. In *Proceedings of the 3rd International Conference on Internet and Network Economics*, WINE’07, Berlin, Heidelberg, 2007. Springer-Verlag.
- [20] G. Biau. Analysis of a random forests model. *J. Mach. Learn. Res.*, 13(1):1063–1095, Apr. 2012.
- [21] D. Billsus and M. J. Pazzani. User modeling for adaptive news access. *User Modeling and User-Adapted Interaction*, 10(2-3):147–180, Feb. 2000.
- [22] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, Mar. 2003.
- [23] S. Borzsony, D. Kossmann, and K. Stocker. The skyline operator. In *Data Engineering, 2001. Proceedings. 17th International Conference on*, pages 421–430, 2001.
- [24] C. Budak, T. Georgiou, D. Agrawal, and A. El Abbadi. Geoscope: Online detection of geo-correlated information trends in social networks. *PVLDB*, 7(4):229–240, 2013.
- [25] F. Cairncross. *The death of distance: How the communications revolution is changing our lives*. Harvard Business Press, 2001.
- [26] E. Casetti. Innovation Diffusion as a Spatial Process, by Torsten Hägerstrand. *Geographical Analysis*, 1:318–320, 1969.
- [27] M. Cataldi, L. D. Caro, and C. Schifanella. Personalized emerging topic detection based on a term aging model. *ACM Trans. Intell. Syst. Technol.*, 5(1):7:1–7:27, Jan. 2014.
- [28] M. Cataldi, L. Di Caro, and C. Schifanella. Emerging topic detection on twitter based on temporal and social terms evaluation. In *Proceedings of the Tenth International Workshop*

- on Multimedia Data Mining*, MDMKDD '10, pages 4:1–4:10, New York, NY, USA, 2010. ACM.
- [29] J. Chang and D. M. Blei. Relational topic models for document networks. In *AISTATS*, volume 5 of *JMLR Proceedings*. JMLR.org, 2009.
- [30] J. Chen, W. Geyer, C. Dugan, M. J. Muller, and I. Guy. Make new friends, but keep the old: recommending people on social networking sites. In *Computer Human Interaction*, pages 201–210, 2009.
- [31] J. Chen, R. Nairn, L. Nelson, M. Bernstein, and E. Chi. Short and tweet: experiments on recommending content from information streams. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'10)*, 2010.
- [32] W. Chen, L. V. Lakshmanan, and C. Castillo. Information and influence propagation in social networks. *Synthesis Lectures on Data Management*, 5(4):1–177, 2013.
- [33] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin. Combining content-based and collaborative filters in an online newspaper, 1999.
- [34] J. Cranshaw, E. Toch, J. Hong, A. Kittur, and N. Sadeh. Bridging the gap between physical location and online social networks. In *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, UbiComp '10, pages 119–128, New York, NY, USA, 2010. ACM.
- [35] M. Dash and H. Liu. Feature selection for classification. *Intelligent Data Analysis*, 1(1–4):131–156, 1997.
- [36] G. De Francisci Morales, A. Gionis, and C. Lucchese. From chatter to headlines: harnessing the real-time web for personalized news recommendation. In *Proceedings of the fifth ACM international conference on Web search and data mining*, WSDM '12, pages 153–162, New York, NY, USA, 2012. ACM.
- [37] M. Dolliver. Social networking: A waste of time?, 2010.

- [38] Y. Duan, L. Jiang, T. Qin, M. Zhou, and H.-Y. Shum. An empirical study on learning to rank of tweets. In *Proceedings of the 23rd International Conference on Computational Linguistics, COLING '10*, pages 295–303, 2010.
- [39] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan. Collaborative filtering recommender systems. *Found. Trends Hum.-Comput. Interact.*, 4(2):81–173, Feb. 2011.
- [40] Y. EL-Manzalawy and V. Honavar. *WLSVM: Integrating LibSVM into Weka Environment*, 2005. Software available at <http://www.cs.iastate.edu/yasser/wlsvm/>.
- [41] E. Ferrara, O. Varol, F. Menczer, and A. Flammini. Traveling trends: Social butterflies or frequent fliers? *CoRR*, abs/1310.2671, 2013.
- [42] W. Galuba, K. Aberer, D. Chakraborty, Z. Despotovic, and W. Kellerer. Outtweeting the twitterers - predicting information cascades in microblogs. In *Proceedings of the 3rd Workshop on Online Social Networks (WOSN'10)*.
- [43] A. X. Garcia, R. Weighted content based methods for recommending connections in online social networks. In *In: The 2nd ACM Workshop on Recommendation Systems and the Social Web, Barcelona, Spain, June 2010*.
- [44] M. T. Gastner and M. E. Newman. The spatial structure of networks. *The European Physical Journal B - Condensed Matter and Complex Systems*, 49(2):247–252, 2006.
- [45] R. Gemulla, E. Nijkamp, P. J. Haas, and Y. Sismanis. Large-scale matrix factorization with distributed stochastic gradient descent. *KDD '11*, pages 69–77, New York, NY, USA, 2011. ACM.
- [46] M. A. Y. S. B. D. Golder, S.A. A structural approach to contact recommendations in online social networks. In *Workshop on Search in Social Media, In conjunction with ACM SIGIR Conference on Information Retrieval, 2009*.
- [47] M. Gomez-Rodriguez. Netrate algorithm. <http://people.tuebingen.mpg.de/manuelgr/netrate/>, note = "[Online; accessed October 2015]", 2014.

- [48] M. Gomez-Rodriguez, D. Balduzzi, and B. Schölkopf. Uncovering the temporal dynamics of diffusion networks. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, pages 561–568, 2011.
- [49] M. Gomez-Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. *ACM Trans. Knowl. Discov. Data*, 5(4):21:1–21:37, Feb. 2012.
- [50] M. Gomez-Rodriguez, J. Leskovec, and B. Schölkopf. Modeling information propagation with survival theory. *CoRR*, abs/1305.3616, 2013.
- [51] A. Goyal, F. Bonchi, and L. V. Lakshmanan. Learning influence probabilities in social networks. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining (WSDM'10)*, 2010.
- [52] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, Nov. 2009.
- [53] J. Hannon, M. Bennett, and B. Smyth. Recommending twitter users to follow using content and collaborative filtering approaches. In *Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys '10*, pages 199–206, New York, NY, USA, 2010. ACM.
- [54] J. Hannon, K. McCarthy, and B. Smyth. Finding useful users on twitter: twittomender the followee recommender. In *Proceedings of the 33rd European conference on Advances in information retrieval, ECIR'11*, pages 784–787, 2011.
- [55] J. He and W. Chu. A social network-based recommender system (snrs). In *Data Mining for Social Network Data*, volume 12 of *Annals of Information Systems*, pages 47–74. Springer US, 2010.
- [56] B. Hecht and E. Moxley. Terabytes of tobler: Evaluating the first law in a massive, domain-neutral representation of world knowledge. In *Spatial Information Theory*, volume 5756 of *Lecture Notes in Computer Science*. Springer, 2009.

- [57] G. Hobgen. Security issues and recommendations for online social networks. Technical report, European Network and Information Security Agency, 2007.
- [58] L. Hong, A. Ahmed, S. Gurumurthy, A. J. Smola, and K. Tsioutsoulouklis. Discovering geographical topics in the twitter stream. *WWW '12*, pages 769–778, New York, NY, USA, 2012. ACM.
- [59] L. Hong, O. Dan, and B. D. Davison. Predicting popular messages in twitter. In *Proceedings of the 20th international conference companion on World wide web*, *WWW '11*, pages 57–58, New York, USA, 2011. ACM.
- [60] L. Hong and B. D. Davison. Empirical study of topic modeling in twitter. In *Proceedings of the First Workshop on Social Media Analytics*, *SOMA '10*, pages 80–88, New York, USA, 2010. ACM.
- [61] Z. Huang, H. Chen, and D. Zeng. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Trans. Inf. Syst.*, 22(1):116–142, Jan. 2004.
- [62] A. Jackoway, H. Samet, and J. Sankaranarayanan. Identification of live news events using twitter. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks*, *LBSN '11*, pages 25–32, New York, NY, USA, 2011. ACM.
- [63] A. I. Jinan Fiaidhi, Sabah Mohammed. Towards identifying personalized twitter trending topics using the twitter client rss feeds. *Journal of Emerging Technologies in Web Intelligence*, 4(3):221–226, 2012.
- [64] K. Y. Kamath, J. Caverlee, Z. Cheng, and D. Z. Sui. Spatial influence vs. community influence: Modeling the global spread of social media. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, *CIKM '12*, pages 962–971, New York, USA, 2012. ACM.
- [65] J. Kannan A., Patzer and B. Avital. Trendtracker: Trending topics on twitter, 2010.

- [66] P. Kazienko and P. Koodziejki. Windows-adaptive system for the integration of recommendation methods in e-commerce. volume 3528 of *Lecture Notes in Computer Science*, pages 218–224. Springer Berlin Heidelberg, 2005.
- [67] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. *KDD '03*, pages 137–146, New York, NY, USA, 2003. ACM.
- [68] D. Kim, D. Kim, E. Hwang, and S. Rho. Twitertrends: A spatio-temporal trend detection and related keywords recommendation scheme. *Multimedia Syst.*, 21(1):73–86, Feb. 2015.
- [69] S.-C. Kim, C.-S. Park, and S. Kim. A hybrid recommendation system using trust scores in a social network. In J. J. J. H. Park, Y.-S. Jeong, S. O. Park, and H.-C. Chen, editors, *Embedded and Multimedia Computing Technology and Service*, volume 181 of *Lecture Notes in Electrical Engineering*, pages 107–112. Springer Netherlands, 2012.
- [70] Y. Kim and K. Shim. Twitobi: A recommendation system for twitter using probabilistic modeling. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 340–349, Dec 2011.
- [71] Y. Kim and K. Shim. Twilite: A recommendation system for twitter using a probabilistic model based on latent dirichlet allocation. *Information Systems*, 42:59 – 77, 2014.
- [72] K. Kodama, Y. Iijima, X. Guo, and Y. Ishikawa. Skyline queries based on user locations and preferences for making location-based recommendations. In *Proceedings of the 2009 International Workshop on Location Based Social Networks, LBSN '09*, pages 9–16, New York, NY, USA, 2009. ACM.
- [73] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. Grouplens: Applying collaborative filtering to usenet news. *Commun. ACM*, 40(3):77–87, Mar. 1997.
- [74] W. Krutkam, K. Saikew, and A. Chaosakul. Twitter accounts recommendation based on followers and lists. *proceeding JICTEE*, 2010.

- [75] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web, WWW '10*, pages 591–600, New York, 2010. ACM.
- [76] D. Laniado and P. Mika. Making sense of Twitter. In *Proceedings of the 9th International Semantic Web Conference on The Semantic Web - Volume Part I, ISWC'10*, pages 470–485, Berlin, Heidelberg, 2010. Springer-Verlag.
- [77] W.-J. Lee, K.-J. Oh, C.-G. Lim, and H.-J. Choi. User profile extraction from twitter for personalized news recommendation. In *Advanced Communication Technology (ICACT), 2014 16th International Conference on*, pages 779–783, Feb 2014.
- [78] W. S. Lee. Collaborative learning for recommender systems. In *Proceedings of the 18th International Conference on Machine Learning*, pages 314–321. Morgan Kaufmann, 2001.
- [79] J. Leskovec, L. Backstrom, and J. Kleinberg. Meme-tracking and the dynamics of the news cycle. *KDD '09*, pages 497–506, New York, NY, USA, 2009. ACM.
- [80] J. Leskovec, M. McGlohon, C. Faloutsos, N. Glance, and M. Hurst. Cascading behavior in large blog graphs: Patterns and a model. In *Society of Applied and Industrial Mathematics: Data Mining (SDM07)*, 2007.
- [81] K. W.-T. Leung, D. L. Lee, and W.-C. Lee. Clr: a collaborative location recommendation framework based on co-clustering. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, SIGIR '11*, pages 305–314, New York, NY, USA, 2011. ACM.
- [82] M. R. Levy and S. Windahl. The concept of audience activity. In K. E. Rosengren, L. A. Wenner, and P. Palmgreen, editors, *Media gratifications research: Current perspectives*, pages 109–122. Sage Publications, Beverly Hills, CA, 1985.



- [83] R. Li, K. H. Lei, R. Khadiwala, and K.-C. Chang. Tedas: A twitter-based event detection and analysis system. In *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*, pages 1273–1276, 2012.
- [84] M. Mathioudakis and N. Koudas. Twittermonitor: Trend detection over the twitter stream. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, SIGMOD '10, pages 1155–1158, New York, NY, USA, 2010. ACM.
- [85] A. McCallum, X. Wang, and A. Corrada-Emmanuel. Topic and role discovery in social networks with experiments on enron and academic email. *J. Artif. Int. Res.*, 30(1):249–272, Oct. 2007.
- [86] A. K. McCallum. Mallet: A machine learning for language toolkit. <http://mallet.cs.umass.edu>, 2002.
- [87] R. Mehrotra, S. Sanner, W. Buntine, and L. Xie. Improving lda topic models for microblogs via tweet pooling and automatic labeling. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 889–892, New York, NY, USA, 2013. ACM.
- [88] T. M. Mitchell. *Machine learning*. McGraw Hill series in computer science. McGraw-Hill, Boston, 1997.
- [89] S. Mukherjee, R. Sujithan, and P. Subasic. Detecting trending topics using page visitation statistics. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14 Companion, pages 347–348, Republic and Canton of Geneva, Switzerland, 2014. International World Wide Web Conferences Steering Committee.
- [90] O. Nanba Ishino and Takezawa. Extracting transportation information and traffic problems from tweets during a disaster: Where do you evacuate to? In *Proceedings of the Second International Conference on Advances in Information Mining and Management*. IMMM, 2012.

- [91] N. Naveed, T. Gottron, J. Kunegis, and A. C. Alhadi. Bad news travel fast: a content-based analysis of interestingness on twitter. In *Proceedings of the ACM WebSci'11*, pages 1–7, 2011.
- [92] P. Netrapalli and S. Sanghavi. Learning the graph of epidemic cascades. *SIGMETRICS Perform. Eval. Rev.*, 40(1):211–222, June 2012.
- [93] M.-H. Park, J.-H. Hong, and S.-B. Cho. Location-based recommendation system using bayesian user preference model in mobile devices. In *Ubiquitous Intelligence and Computing*, volume 4611 of *Lecture Notes in Computer Science*, pages 1130–1139. Springer, 2007.
- [94] N. Pathak, A. Banerjee, and J. Srivastava. A generalized linear threshold model for multiple cascades. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 965–970, Dec 2010.
- [95] N. Pathak, C. Delong, A. Banerjee, and K. Erickson. Social Topic Models for Community Extraction. In *The 2nd SNA-KDD Workshop '08 (SNA-KDD'08)*, Aug. 2008.
- [96] M. Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13(5-6):393–408, 1999.
- [97] A. Pozdnoukhov and C. Kaiser. Space-time dynamics of topics in streaming text. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks, LBSN '11*, pages 1–8, New York, NY, USA, 2011. ACM.
- [98] J. R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.
- [99] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013.
- [100] N. F. N. Rajani, K. McArdle, and J. Baldridge. Extracting topics based on authors, recipients and content in microblogs. In *Proceedings of the 37th International ACM SIGIR Conference*

- on Research &#38; Development in Information Retrieval*, SIGIR '14, pages 1171–1174, New York, NY, USA, 2014. ACM.
- [101] D. Ramage, S. Dumais, and D. Liebling. Characterizing microblogs with topic models. In *ICWSM*, 2010.
- [102] Z. Ren, S. Liang, E. Meij, and M. de Rijke. Personalized time-aware tweets summarization. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, 2013.
- [103] E. Rich. User modeling via stereotypes. *Cognitive Science*, 3:329–354, 1979.
- [104] A. Ritter, Mausam, O. Etzioni, and S. Clark. Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pages 1104–1112, New York, NY, USA, 2012. ACM.
- [105] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, UAI '04, pages 487–494, Arlington, Virginia, United States, 2004. AUAI Press.
- [106] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 851–860, New York, NY, USA, 2010. ACM.
- [107] G. Salton, editor. *Automatic Text Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1988.
- [108] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. T. Riedl. Application of dimensionality reduction in recommender system - a case study. In *In ACM WebKDD Workshop*, 2000.
- [109] J. B. Schafer, J. A. Konstan, and J. Riedl. E-commerce recommendation applications. *Data Min. Knowl. Discov.*, 5(1-2):115–153, Jan. 2001.

- [110] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '02*, pages 253–260, New York, NY, USA, 2002. ACM.
- [111] U. Shardanand and P. Maes. Social information filtering: Algorithms for automating “word of mouth”. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '95*, pages 210–217, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.
- [112] R. R. Sinha and K. Swearingen. Comparing recommendations made by online systems and friends. In *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*, 2001.
- [113] R. W. Sinnott. Virtues of the Haversine. *Sky and Telescope*, 68(2):159+, 1984.
- [114] B. Smyth and P. Cotter. A personalised TV listings service for the digital TV age. *Knowledge-Based Systems*, 13(23):53 – 59, 2000.
- [115] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas. Short text classification in Twitter to improve information filtering. In *Proceedings of ACM SIGIR Conference*, 2010.
- [116] M. Steyvers, P. Smyth, M. Rosen-Zvi, and T. Griffiths. Probabilistic author-topic models for information discovery. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.
- [117] B. Suh, L. Hong, P. Pirolli, and E. H. Chi. Want to be retweeted? large scale analytics on factors impacting retweet in twitter network. In *Social Computing (SocialCom), 2010 IEEE Second International Conference on*, pages 177–184, 2010.
- [118] W. R. Tobler. A computer movie simulating urban growth in the detroit region. *Economic Geography*, 46:pp. 234–240, 1970.

- [119] F. Tönnies. *Community and society (Gemeinschaft und Gesellschaft) Translated and edited by Charles P. Loomis*. 1957.
- [120] J. Ugander, B. Karrer, L. Backstrom, and C. Marlow. The anatomy of the facebook social graph. *CoRR*, abs/1111.4503, 2011.
- [121] B. Upbin. What are the best times to share on facebook and twitter? <http://www.forbes.com/sites/bruceupbin/2012/05/09/when-to-make-stuff-go-viral-online> [Online; accessed June 2015].
- [122] I. Uysal and W. B. Croft. User oriented tweet ranking: a filtering approach to microblogs. In *Proceedings of the 20th ACM international conference on Information and knowledge management (CIKM'11)*, 2011.
- [123] H. M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09*, pages 1105–1112, New York, 2009. ACM.
- [124] C. Wang and B. Huberman. Long trend dynamics in social media. *EPJ Data Science*, 1(1), 2012.
- [125] X. Wang, M. Gerber, and D. Brown. Automatic crime prediction using events extracted from twitter posts. In S. Yang, A. Greenberg, and M. Endsley, editors, *Social Computing, Behavioral - Cultural Modeling and Prediction*, volume 7227 of *Lecture Notes in Computer Science*, pages 231–238. Springer Berlin Heidelberg, 2012.
- [126] J. Weng, E.-P. Lim, J. Jiang, and Q. He. Twiterrank: Finding topic-sensitive influential twitterers. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining (WSDM'10)*, 2010.
- [127] Yahoo!-GeoPlanet. [developer.yahoo.com/geo/geoplanet/](http://developer.yahoo.com/geo/geoplanet/). [Online; accessed September 2014].
- [128] Y. Yamamoto. Twitter4j: Java library for the twitter api, 2014.

- [129] M.-C. Yang, J.-T. Lee, S.-W. Lee, and H.-C. Rim. Finding interesting posts in twitter based on retweet graph analysis. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '12, pages 1073–1074, New York, USA, 2012. ACM.
- [130] X. Yang, Y. Guo, Y. Liu, and H. Steck. A survey of collaborative filtering based social recommender systems. *Computer Communications*, (0):–, 2013.
- [131] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '11, pages 325–334, New York, NY, USA, 2011. ACM.
- [132] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann. Time-aware point-of-interest recommendation. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '13, pages 363–372, New York, NY, USA, 2013. ACM.
- [133] E. Zangerle, W. Gassler, and G. Specht. Recommending #-tags in Twitter. In *Proceedings of the Workshop on Semantic Adaptive Social Web 2011 in connection with the 19th International Conference on User Modeling, Adaptation and Personalization, UMAP 2011*, pages 67–78, Gerona, Spain, 2011. CEUR-WS.org, ISSN 1613-0073, Vol. 730, available online at <http://ceur-ws.org/Vol-730/paper7.pdf>, urn:nbn:de:0074-581-7.
- [134] Y. Zhang, J. Callan, and T. Minka. Novelty and redundancy detection in adaptive filtering. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, pages 81–88, New York, NY, USA, 2002. ACM.
- [135] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li. Comparing twitter and traditional media using topic models. In *Proceedings of the 33rd European Conference on Advances in Information Retrieval (ECIR'11)*, 2011.

- [136] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang. Collaborative location and activity recommendations with gps history data. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 1029–1038, New York, NY, USA, 2010. ACM.

# Appendix A

## User Study

### A.1 TrendFusion System Setup

When TrendFusion system was first introduced to the users, they were requested to follow some steps in order to register to our system, and link their Twitter account to TrendFusion to allow them to access their timelines. These steps are as follows:

#### **Step 1: (Register a new TrendFusion user)**

Description: This step is used to create a profile for a new TrendFusion user. The new user should be having a Twitter account in advance.

1. Type `vt.trendfusion.org` in the location field. You should see a web page as in Figure A.1.
2. Click on the Register button on the left hand side.
3. Enter username that you would like to use on TrendFusion in the username text field.
4. Enter a password in the Password text field.
5. Click on the Submit button at the end of the page.



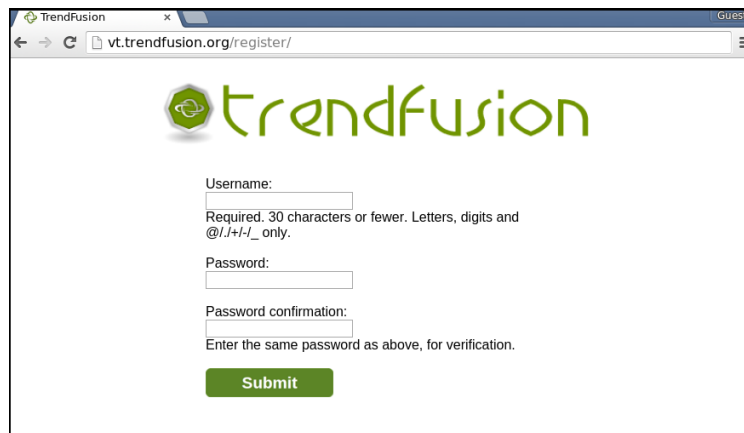
A screenshot of a web browser showing the registration page for TrendFusion. The browser's address bar displays "vt.trendfusion.org/register/". The page features the TrendFusion logo at the top, which consists of a green circular icon with a stylized 't' and the word "trendfusion" in a green, lowercase, sans-serif font. Below the logo, there are three text input fields: "Username:", "Password:", and "Password confirmation:". The "Username:" field has a small text requirement below it: "Required, 30 characters or fewer. Letters, digits and @,.,+,-/\_ only." The "Password confirmation:" field has a small text requirement below it: "Enter the same password as above, for verification." At the bottom of the form is a green "Submit" button.

Figure A.1: Register a new TrendFusion user

### Step 2: Authorize TrendFusion to use Twitter account

Description: After the user is registered in TrendFusion, the system prompts the user to authorize TrendFusion to use the user's Twitter account.

1. Enter username or email of your Twitter account in the Username or email text field as it appears in Figure A.2.
2. Enter a password in the Password text field.
3. Click on the Authorize app button at the end of the page.

### Step 3: check all tweets on your timeline

Description: After the user signs in/registers to TrendFusion, TrendFusion is now connected to the users Twitter account. Figure A.3 shows the webpage appearing to show the user's timeline. The user can now view all the tweets appearing on the user's timeline, along with the trends appearing at this time.

1. Check the All tweets radio button on the left hand side
2. Click on the Submit button on the left hand side

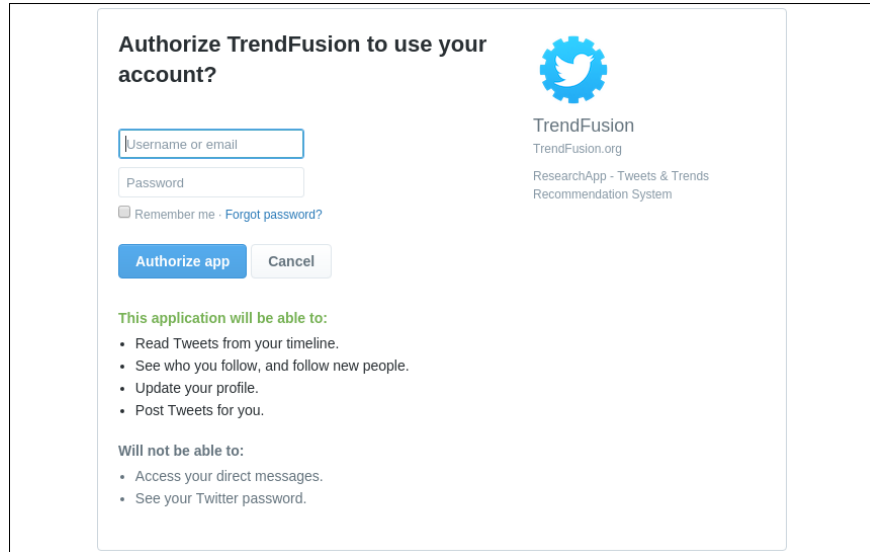


Figure A.2: Authorize TrendFusion user

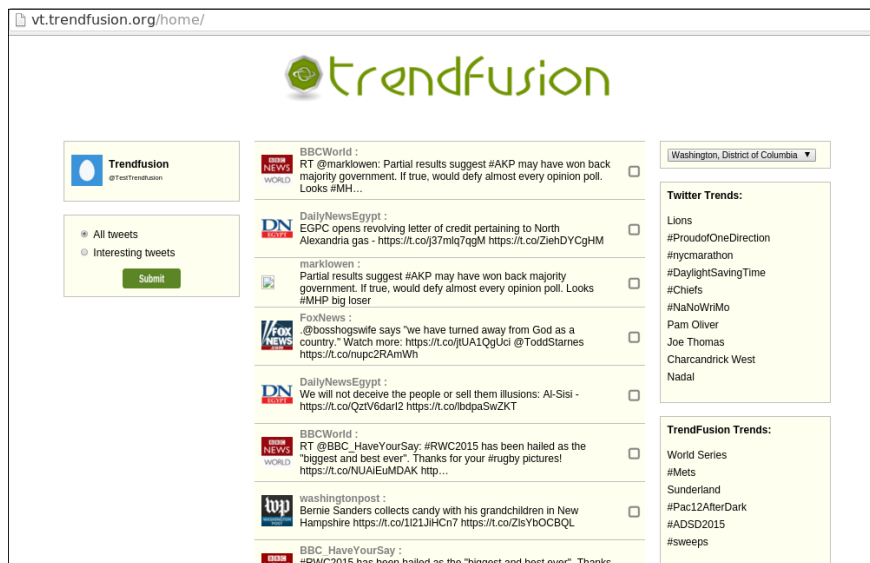


Figure A.3: Check user timeline

3. To mark a tweet on the timeline as important, the user checks the check box to the right of the desired tweet.
4. To view the trending topics in a certain location, click the list box appearing on the right side of the page to choose the location where you want the trend to appear.

#### Step 4: Sign in to TrendFusion site

Description: This step is used to sign in for an already existing TrendFusion user (the user already has a TrendFusion account).

1. Type `vt.trendfusion.org` in the location field. A login page will appear for signup as in Figure A.4
2. Enter username of the TrendFusion user in the username text field.
3. Enter a password in the Password text field.
4. Click on the Submit button at the end of the page.

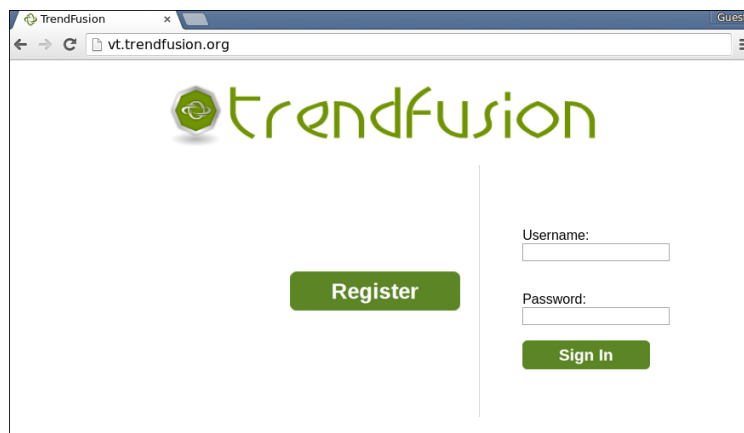


Figure A.4: User login

## A.2 User Questionnaire

Based on the steps we identified for the TrendFusion system, we designed a user questionnaire, which was presented to the study participants. The goal was to evaluate the systems ability to recommend tweets of interest to the user, along with predicting trending topics that will likely to occur in a user's place of interest and also under a topic of interest to the user. This Appendix contains the list of study questions.

### A.2.1 Part I: General usability questions

**Q1: Are you a member of any of the online social networking services**

- Yes
- No

**If yes, please choose all that apply**

- Facebook
- Twitter
- Google+
- Myspace
- (please specify)

**Q2: How often do you visit Twitter site?**

- Several times a day
- Once a day

- Several times a week
- Once a week
- Once a month

**Q3: When logging in to Twitter, what actions do you usually do?(check all that apply)**

- Post tweets
- Read tweets
- Retweet/favorite
- All

**Q4: How do you usually access Twitter? (check all that apply)**

- Through Twitter site
- Through Twitter mobile application
- Through other applications
- If others(please specify)

**Q5: When using Twitter, do you enable the "Geo-location" settings?**

- Yes
- No

**Q6: How many followers do you have on Twitter?**

- Less than 10

- Between 10 and 50
- Between 50 and 100
- More than 100
- More than 500

**Q7: How many friends are you following (Followee) on Twitter?**

- Less than 10
- Between 10-50
- Between 50-100
- More than 100

**Q8: From the tweets you are posting, what is the average percentage of tweets posted in English language?**

- 0%
- 10%-30%
- 30%-50%
- 50%-70%
- 70%-90%
- 100%

**Q9: Out of your total retweets/favorited posts, what is the average percentage of tweets posted in English language?**

- 0%
- 10%-30%
- 30%-50%
- 50%-70%
- 70%-90%
- 100%

### **A.2.2 Part II: TrendFusion application usability**

**Q10: Based on TrendFusion application usage, how would you rate the system ability to provide the following (1= very easy, 5= very difficult)**

**a) Ease of Interface**

1    2    3    4    5

**b) Variation in choosing your location of interest when choosing to display trends of certain location**

1    2    3    4    5

**Q11: From the tweets recommended by TrendFusion application, how many of them do you think they are of interest to you? (an average percentage)**

please specify:

**Q12: From the total number of interesting tweets on your timeline, how many interesting tweets were successfully detected by the TrendFusion application?**

please specify an average percentage:

**Q13: a) How old are you?**

please specify

**b) Gender:**

- male
- female

**c) Educational level:**

please specify

**Q14: If you live in one of the US cities included in TrendFusion trending topics suggestion, how would you rate the quality of the suggested trending topics by TrendFusion? (1=Very interesting, 5=Not interesting)**

1    2    3    4    5    "I do not live in the US"



# **Appendix B**

## **Questionnaire Results**

User	Q1 a	Q1b	Q2	Q3	Q4	Q5
1	Yes	Twitter	Once a day	Read Tweets, Retweet/favorite	Twitter mobile application	Yes
2	Yes	Facebook, Twitter, Google+	Several times a day	All	Twitter site, Twitter mobile app	No
3	Yes	Twitter, Google+	Several times a week	Read Tweets	Twitter site, Twitter mobile app	No
4	Yes	Facebook, Twitter	Several times a week	Read Tweets, Retweet/favorite	Twitter mobile app	No
5	Yes	Facebook, Twitter	Several times a day	All	Twitter mobile app	Yes
6	Yes	Twitter, Google+	Once a day	Post Tweets, Read Tweets	Twitter site	No
7	Yes	Twitter, Google+	Several times a day	All	Twitter mobile app	No
8	Yes	Facebook, Twitter, Google+	Several times a week	Read Tweets	Twitter site	No
9	Yes	Facebook, Twitter	Once a day	All	Twitter site, Twitter mobile app	Yes
10	Yes	Twitter	Several times a day	All	Twitter site, Twitter mobile app	No

User	Q6	Q7	Q8	Q9	Q10: a)	Q10(b)	Q11	Q12	Q13	Q14
1	Between 10 and 50	Between 50-100	70-90%	70-90%	3	2	65%	65%	30	1
2	Between 50 and 100	More than 100	All in English	100%	1	1	90%	90%	29	1
3	Less than 10	Between 10-50	All in English	100%	3	1	70%	70%	29	Not in US
4	Between 10 and 50	Between 10-50	All in English	70-90%	2	3	75%	75%	32	4
5	Between 50 and 100	Between 50-100	All in English	100%	2	2	95%	95%	31	2
6	Less than 10	Less than 10	All in English	70-90%	2	3	80%	90%	30	2
7	Between 50 and 100	More than 100	All in English	100%	2	3	90%	90%	26	2
8	Less than 10	Between 50-100	70-90%	70-90%	3	4	70%	65%	28	3
9	Between 10 and 50	More than 100	70-90%	70-90%	2	5	80%	80%	34	3
10	More than 100	More than 100	All in English	100%	2	3	90%	95%	35	2