Temporal Topic Embeddings with a Compass

Daniel A. Palamarchuk

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

Masters of Science
in
Computer Science and Applications

Christopher L. North, Chair
Brian B. Mayer
Xuan Wang
Thomas L. Danielson

25 April 2024
Blacksburg, Virginia

Keywords: diachronic, neural network, hci, bureaucracy, legibility

Temporal Topic Embeddings with a Compass

Daniel A. Palamarchuk

(ABSTRACT)

Aligning Word2vec word embeddings using a compass in a system of Compass-aligned Distributional Embeddings (CADE) creates stable and accurate temporal word embeddings. This thesis seeks to expand the CADE framework into the area of dynamic topic modeling (DTM), where temporal word2vec embeddings can be used to describe temporally and unsupervised evolving topics. It also seeks to improve upon the CADE framework through a theoretical and experimental exploration of compass parameters, cluster and topic generation techniques, and topic descriptor creation. This method of Temporal Topic Embeddings with a Compass (TTEC) will be compared to other DTM techniques in the ability to create coherent and diverse clusters and will be shown to be competitive compared to traditional and transformer-aided DTM architectures. In addition to a qualitative discussion of results, there will be a political theoretical overview of the nature of this technique and potential use cases, with interviews from political actors of various backgrounds as to how the technique and machine learning as a whole can be used in the organizational setting.

Temporal Topic Embeddings with a Compass

Daniel A. Palamarchuk

(GENERAL AUDIENCE ABSTRACT)

Diachronic word embeddings look at how the context words appear in evolve over time. Dynamic Topic Modeling (DTM) is the ability to computationally discover topics and how they evolve over time. This thesis creates a DTM technique called Temporal Topic Embeddings with a Compass (TTEC) based off diachronic word embeddings, allowing a user to simultaneously look at word and topic evolution over time. There is also an exploration of the use case of TTEC and similar machine learning models within various political organizational settings through interviews.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Given a large corpus of documents and their creation date, it is useful to understand what topics exist within the corpus and how those topics evolve over time. The temporal dynamics of topics help illustrate global discussion trends in the corpus, such as what topics exist (global topics), topic introduction, end of a topic discussion, and how the discussion surrounding a topic changes. In addition, looking at words in a temporal setting allows one to see how word usage evolves with respect to other words and documents. When combined, dynamic (temporal) topics can be described in terms of dynamically evolving terms, both arising from a corpus of documents.

For example, a user might want to identify significant events on a global scale related to nuclear energy or politics by analyzing a news article corpus containing millions of documents collected over several years. Given the time frame of an event, the user can discover articles related to words associated with that event and link those documents to global topics. Linkage to global topics allows users to look at and analyze articles and topic descriptions leading up to that event. Linkage to temporal word embeddings allows the user to track the usage of that term and see which words, documents, and topics it is closest to over time,

or if it is even relevant to a topic leading up to the event. There are two temporal textual analytical methods at play here: Dynamic word embeddings and dynamic topics, each of which has its own analytical tools.

Dynamic word embeddings allow a user to track the evolution of the usage of terms over time. The association of the word "apple" with the fruit evolves into the association of "apple" with the IT industry. Similarly, the word "lit" evolves from "burning fires" to "glorious and majestic," which is captured in temporal Word2vec methods. These temporal embeddings can also be overlapped to discover temporal analogies of words used in a similar context at different times. The term "Clinton" at one point would be used in a context similar to "Bush" in another context, which in turn would be used in a similar context to "Obama" at a later context depending on which person is president at the time within that temporal part of the text corpus. A temporal word embedding model would capture this relationship by creating "word vectors" to represent these figures based on the temporal context of those figures, and related contexts would yield similar word vectors.

Looking at the task of temporal analytics of textual data from a slightly different angle, one can look at a text corpus as a collection of topics to be described in terms of words. A topic related to computers could initially be talking in terms of "mainframes" and "minicomputers" and then evolve and be described in terms of "the internet" and "laptops." This topic description can then evolve to include "smartphones" and "tablets," showing how popular computing discourse evolves over time and incorporates new technologies. Temporally evolving topics capture an overview of the temporally evolving discourse of a text corpus that can then be separated into documents.

Dynamic word embeddings and topic models have some key ideas in common, but current popular methodology lies squarely in one field or another. Popular topic modeling methods, such as Sequential Latent Dirichlet Allocation [6, 5] and BERTopic [19] capture topic evolu-

tion over time, but remain relatively opaque when it comes to the analysis centered around words themselves. Dynamic word embedding methodologies such as Dynamic Word2vec [51] and Temporal Word Embeddings with a Compass [15] perform well to relate words to each other in a temporally coherent way, but they lack the abstraction and generalizability abilities of dynamic topics. They both have useful abilities that can potentially make up for each other. To create a method combining the two, the following research questions must be asked:

1. How can a single embedding space that combines dynamic words, documents, and topics be created?
2. How can the combined embedding space be visualized to reveal changes over time that aren't captured by dynamic word embeddings or dynamic topics alone?

To address these questions, I propose a new Dynamic Topic Modeling method called Temporal Topic Embeddings with a Compass (TTEC), which places dynamically evolving Word2vec embeddings in relation to Doc2vec document embeddings. TTEC creates a global topic space using document embeddings. Global and locally aligned word embeddings can then describe the documents and topics. This alignment allows for topic-centric and word-centric analyses. My contributions are as follows:

1. An expansion of the compass-aligned temporal Word2vec methodology into dynamic topic modeling (TTEC) using word and document embeddings
2. An overview of possible methods that can take advantage of the presence of dynamic word embeddings and topics
3. A method to improve the time slice creation process of compass-aligned methodologies
4. An analysis of potential audiences that would find the most use out of TTEC.

Chapter 2

Review of Literature

2.1 Word Embeddings

This work combines two approaches, the first of which is Word2vec-based temporal embeddings. Word2vec [32] is a well-known word embedding model built off of the distributional hypothesis of words [22], which holds that a word can be defined by the words that surround it. Similar words will be found in similar surroundings. A hypothetical word embedding algorithm based on this hypothesis has contextually similar words that end up with similar

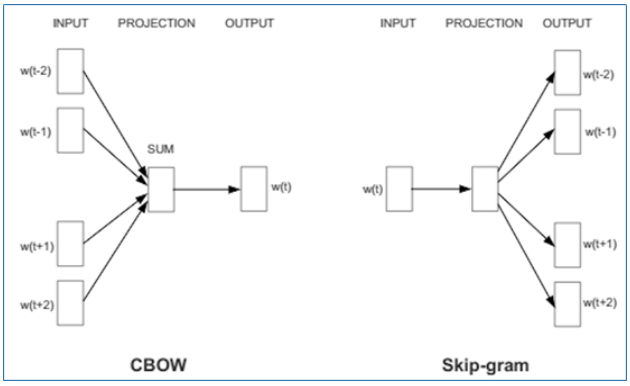

Figure 2.1: A structural representation of Word2vec according to Mikolov et al. [32]

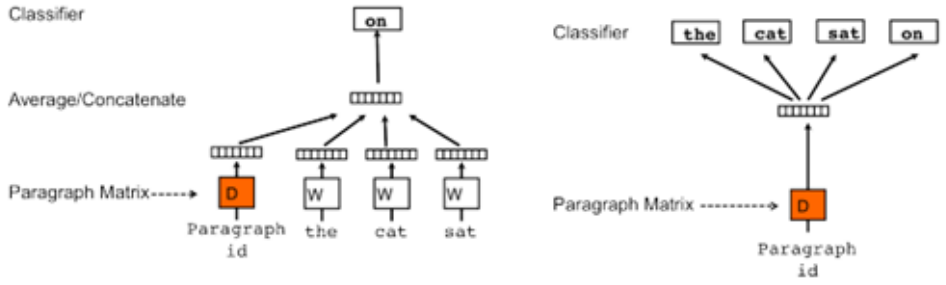

Figure 2.2: Doc2vec representations per Le and Mikolov [28]. Distributed Memory architecture on the left. Distributed Bag of Words on the right

embeddings. Word2vec is the name for a family of unsupervised algorithms that involves training a shallow neural network to predict an output one-hot word out of all the output words using one or more one-hot input words. There exist two Word2vec representations (See Figure 2.1) called the Continuous Bag of Words (CBOW) and Skip-gram (SG) methods. The CBOW method uses a fixed number of surrounding words to predict a center word, usually through summing up the word vectors of the surrounding words to then feed through the neural network. The SG method does the inverse and tries to predict the surrounding words using the word in the center. This neural network is trained through a process of backpropagation that shifts around the weights of the shallow neural network and the contents of the word embedding themselves. Given that all words within a Word2vec model are a possibility, to prevent a backpropagation across all possible words (having the correct output be 1 and everything else as 0), there is usually an alternative process of hierarchical softmax [32] or negative sampling [34] and sub-sampling of popular words to allow for faster and more accurate training. Hierarchical softmax [35] finds the most relevant words in $\log_2(n)$ time rather than looking linearly for the closest word. Negative sampling takes a sample of words that are medium in frequency (i.e. not very rare and not very common), treats those as 0, and performs backpropagation on those rather than all the words being looked at.

The Word2vec model was later expanded to document and general tag-based embeddings by Le and Mikolov [28] with Doc2vec. They work similarly to Word2vec, predicting outcome words given a set of inputs (see Figure 2.2). The Paragraph Vector Distributed Memory uses the document vector for a document and some consecutive words to predict the next word in that document. This model directly creates word and document embeddings that occupy an embedding space, which allows for a comparison between word and document vectors. The Paragraph Vector Distributed Bag of Words model has an identical architecture to the Skip-gram model, except a document vector is used to predict words in that document instead of a word. The model does not generate word embeddings, but it can be combined with a Skip-gram model and share the hidden layer embeddings to produce document and word embeddings in the same embedding space.

2.2 Temporal Word Embeddings

Word2vec is theoretically grounded in the distributional hypothesis of words, so words can have a different embedding representation given a different text corpus. For example, the term "apple" within a corpus of recipes will likely be seen in a similar context to other fruits and will thus have an embedding similar to fruits. However, in a corpus of computational articles, "apple" is more associated with the IT company and will thus have a word embedding similar to terms related to consumer computers. One of the most insightful and practical extensions of this idea of change in corpus usage with Word2vec was seeing how the contexts that words appear in change over time [21]. This can be related to a figure position (the term "Obama" in 2009 will occupy a similar space to "Bush" in 2008 since they would appear in the context of terminology associated with presidential figures) or to word evolution (the word "apple" before 2000 will likely refer to the fruit and will thus appear in

the presence of other food, but will transition over time towards being associated with IT). Given the random nature of training neural networks, there is a goal to make sure that these word embeddings can be compared over time.

A corpus is separated into time slices and context within each time slice is learned. For these slices to be comparable, there needs to be established a notion of slice-wise alignment, since it would be difficult to compare word movements (context changing) between time slices otherwise. There are several methods by which this alignment can take place, but most methods of aligning word embeddings either have a *pairwise* or a *joint* approach [15]. The pairwise approach attempts to align adjacent time slices by initializing the next time slice from the current one, or by applying a linear transformation to ensure pairwise similarity [21, 44]. This can, for example, be done by applying a Procrustes alignment [21] between adjacent time slices so that word embeddings are shifted to be in such a location that maximizes cosine similarity between identical words across time slices. The joint approach creates vector similarity through either global vectors or a smoothing process in the training process itself [51, 42]. In most of these cases, primarily with pairwise methods, there exists a concept of "temporal drift," which describes a phenomenon where, over several time slices, the temporal analogies are predicted less accurately. Usually, this is due to slight shifts in adjacent time slices that result in a larger error when extrapolated to several time slices. A simple Procrustes alignment might work on two adjacent time slices, but there remains a non-zero error and shift of word context that can compound over time to destabilize time slices over time, for instance.

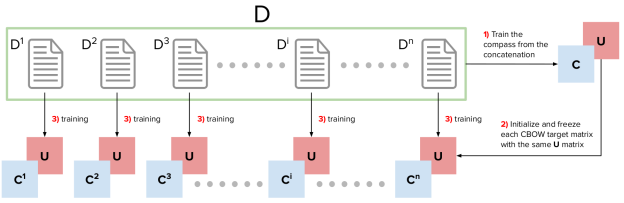

Figure 2.3: Schema for CADE as presented by Di Carlo, Bianchi, and Palmonari [15]

2.2.1 CADE

Compass-Aligned Distributed Embeddings (CADE) [15, 4] are a special alignment method in that there is no reliance on a global vector or any notion of obtaining current aligned word embeddings from previous ones (See Figure 2.3). Instead, CADE initially has an atemporal embedding model trained, obtaining "global" word vectors and hidden weights (the "target embedding"). The word vectors are thrown out and the target embedding is kept and frozen. In other words, each time slice will adopt the hidden layer of the compass before it begins training. For example, if a time slice model has "labor" in its vocabulary, the weights that lead to the output for "labor" will be copied to the time slice from the compass. The hidden layer is copied over and frozen. A frozen layer does not change, so the only parameters that change during the training process are the word vectors. A normal Word2vec training process would have the hidden layer and word vector be learned simultaneously. So, each time slice then trains new word vectors using the frozen target embedding. Since the target embedding is frozen, obtaining similar outcomes from the neural network would require similar inputs, thus creating a notion of alignment across time slices. This theoretically and practically allows for a resistance to temporal drift, since the training process no longer relies on anything temporal, just on the global compass which establishes a general notion of what words mean. This also allows for parallel time slice training once the compass is trained since all the individual time slices will require this compass. Although CADE is the general term, Di Carlo, Bianchi, and Palmonari [15] focused on obtaining aligned word

embeddings, which gave their method the secondary name of Temporal Word Embeddings with a Compass (TWEC).

2.3 Dimension Reduction

Dimension reduction reduces the dimensionality of an embedding space. This is done for several reasons: Visualization, pattern/feature extraction, memory and computational restrictions, and dealing with the curse of dimensionality. Principal Component Analysis (PCA) [18] is a deterministic algorithm that aims to capture as much data variance as possible in components. Classic Multi-Dimensional Scaling (MDS) aims to deterministically minimize the "stress," or pairwise distance, between points [47] through eigenvalue decomposition of absolute distances. The t-based Stochastic Neighbor Embedding algorithm [29] aims to preserve local structure through iteratively updating low dimensional pairwise affinity measures to approximate high dimensional affinities. Finally, Uniform Manifold Approximation and Projection [31] uses algebraic topological representations of points to construct a graph of nearest neighbors (a parameter that can be tweaked to prioritize local or global structure) and iteratively running a cross-entropy minimization to recreate this relationship of nearest neighbors using a minimum distance at low dimensions.

2.4 Clustering Algorithms

Clustering techniques offer a way to classify points into different groups unsupervised. The k-means algorithm [30] attempts to classify points into a fixed cluster count by iteratively finding the mean of each cluster and reclassifying each point based on the closest mean to that cluster. However, it cannot generalize to atypical cluster shapes by assuming a Gaussian

point distribution. Needing to know the number of clusters beforehand is also a potential issue. There are several methods to estimate the ideal best number, such as the elbow method [45], silhouette score [41], the gap score [46], and BIC [37, 25]. Still, other families of clustering algorithms allow for more flexible clusters independent of shape.

Density-based clustering algorithms deterministically find clusters by identifying areas of high density. Perhaps the most well-known of these is Density-based Spatial Clustering of Applications with Noise (DBSCAN) [17]. It takes a minimum number of points and a distance parameter and finds core clusters of points based on whether the amount of points within the distance of a point reaches that minimum. Points that are not "core points" are either classified as "border" points that are reachable by cluster cores or "noise" points that are outside clusters. However, this algorithm has an issue similar to k-means, which has the issue of needing to select an appropriate cluster amount. DBSCAN replaces this with the need to pick a cluster minimum point number **in addition to** pairwise distances.

A third family of clustering is named "hierarchical" clustering [36] algorithms. These deterministically find clusters through a manner of graph construction of closest points. This collection of linked points is then placed into a "hierarchy" where the closest links are linked lower on the hierarchy and bigger clusters are made the higher on the hierarchy one goes. Effectively, smaller clusters are merged into bigger ones as the hierarchy rises. The hierarchical measure is based on cluster similarity, or the distance between clusters, and higher up the hierarchy results in the joining up of more dissimilar clusters into one larger cluster. A cutoff point for similarity is selected for this graph based on minimum cluster size and distance. There are several nuances to this graph construction, such as agglomeration clustering (each observation starts as its own cluster and then joins up) and divisive clustering (starting as one cluster and then breaking it up). HDBSCAN [10] is a "hierarchical" extension of DBSCAN that creates a graph using the minimum distance metric. The cutoff point for the

graph can be algorithmic (using a stability measure), or the leaves of the graph themselves can be selected to be the clusters. This solves one of the issues identified with DBSCAN, making it so only the cluster size is an issue. Cluster size subjectively is easier to pick a parameter for, since a person typically has an idea about how large they want their clusters to be, whereas picking a number of means would require either previous knowledge about a dataset or one of several tests to determine an optimal count.

2.5 Topic Modeling

Topic modeling attempts to discern topics from a set of documents. Initially, this would be done without a neural network, such as LDA [6], though recently there has been a shift towards using embeddings and topics generated, at least in part, by neural networks. The number of clusters can be static and generated using k-means, or it can be dynamically generated using an algorithm like HDBSCAN [10]. These can be combined with a dimension reduction method such as UMAP [31] to obtain more well-defined clusters in lower dimensions. The combination of these methods are used by Angelov [1] and Grootendorst [19] on embeddings generated by Word2vec/Doc2vec [28] and BERT [14] respectively to create topics for document embeddings. Top2vec [1] uses the HDBSCAN topics created using Doc2vec embeddings to create topic vectors, which then are used to generate topic descriptors using the Word2vec embeddings generated in the very same embedding space. BERTopic [19] does something similar with embeddings generated by Sentence-BERT [39], but adds a class-based TF-IDF (c-TF-IDF) to generate topic descriptors in the absence of a way to obtain similar words native to Sentence-BERT.

2.6 Dynamic Topic Modeling

Dynamic topic modeling takes the topic modeling approach and expands it temporally to allow a user to view the evolution of a topic and word usage over time. This was first thought of as a sequential LDA [5] that attempts to create a smooth transition between topic representations of words between time slices. BERTopic expands to dynamic topic modeling [19, pp. 3–4] by initially performing a standard embedding obtaining and topic generation, but then divides up articles based on time stamp, so all the created topics are done so on a global level. C-TF-IDF can then be performed on individual time slices to monitor topic evolution.

Chapter 3

Methodology

First, the further development of the CADE methodology will be discussed. Here, I will propose an enhancement to CADE in general in the form of compass alpha scaling. I will then propose a new CADE architecture including document embeddings, called TDEC. Finally, I will propose a DTM method based on CADE and TDEC, called TTEC.

3.1 CADE Expansion

This first section will build up from the TWEC method [15] towards the dynamic topic model TTEC. The buildup will at its core maintain the frozen hidden layer that will allow for the eventual interrelation between dynamic word embeddings and topics.

3.1.1 Temporal Word Embeddings with a Compass (TWEC)

The inner workings of TWEC were discussed in Subsection 2.2.1. Nonetheless, it is important to stress that the methodological developments will maintain this TWEC training pattern of

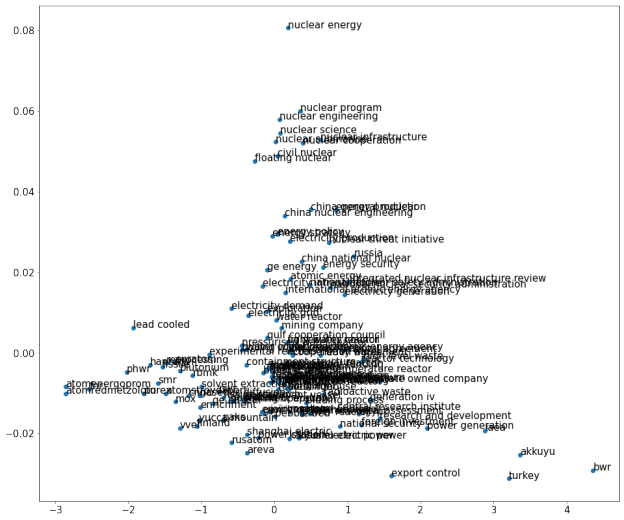

Figure 3.1: Principal Component Analysis of a subset of key terms by a single time slice (composed of 20k articles) trained using compass (composed of 5 million articles). The vast majority of the variance is captured by the first component.

first training a compass using the entire text corpus and then training each time slice using that compass as an alignment tool. The rest of this section will be devoted to developing a method that is applicable to TWEC and further developments.

3.1.2 Alpha Scaling

TWEC by design is good at resisting the dynamic word embedding problem of "temporal drift," which is when word embeddings "drift" to other locations over time periods. There are a few potential causes for this, chief among them being methods that impose alignment between consecutive time slices. If the prime concern is ensuring that adjacent pairs of time slices are closely aligned, then there is potential for degradation in alignment over several time slices since the small adjacent errors can compound. This is especially the case for temporal analogies that are more than a few years. For example, one would expect the word vector for "Obama" in 2009 to be similar to the word vector for "Bush" in 2002 because they are presidents during those time periods. While TWEC, on average in this period of

7 years, guesses such an analogy in the top 2, competing methods get it in the top 10 [15], and the difference in performance becomes more drastic over larger differences in time. A difference in two decades had TWEC degrade to guessing the analogy in the top 3 while the methods more susceptible to temporal drift got it in the top 20.

Given this encouragement, I propose a method to improve the scalability of TWEC in terms of time slice expansion. Given enough time slices and/or an imbalance in data, it is possible that individual time slices might not contain enough data to generate quality embeddings to "fit" the detailed hidden layer created by the compass. This is shown by Figure 3.1, which shows one slice out of 92, with a 250x difference in magnitude between the data that went into compass creation and the data within this particular time slice. There is a suffering in embedding quality. The fact that most variation can be captured within a single principal component means that there is not enough data to generate more meaningful embeddings given a rich compass-based hidden layer. This is true of all neural networks: bigger is better so long as there is enough data.

With this consideration in mind, I propose a scaling of the learning rate parameter alpha for compass training. The goal is to scale the compass learning rate α to be smaller and more attainable by the learning rate of individual time slices. Keeping with the tradition of proposing dual methods, we also propose two methods of scaling alpha. Given a compass alpha α_C, n time slices, time slice alpha α_t, compass document count δ_C, document count for time slice t of δ_t, and document count for the time slice with the smallest number of documents δ_m:

1. $\alpha_C = \alpha_t / n$. This scales the compass alpha down based on the number of time slices
2. $\alpha_C = \alpha_t / \delta_C \cdot \delta_m$. This scales the compass down to the smallest time slice

This is so that the compass process of gradient descent will overall "move" the same amount

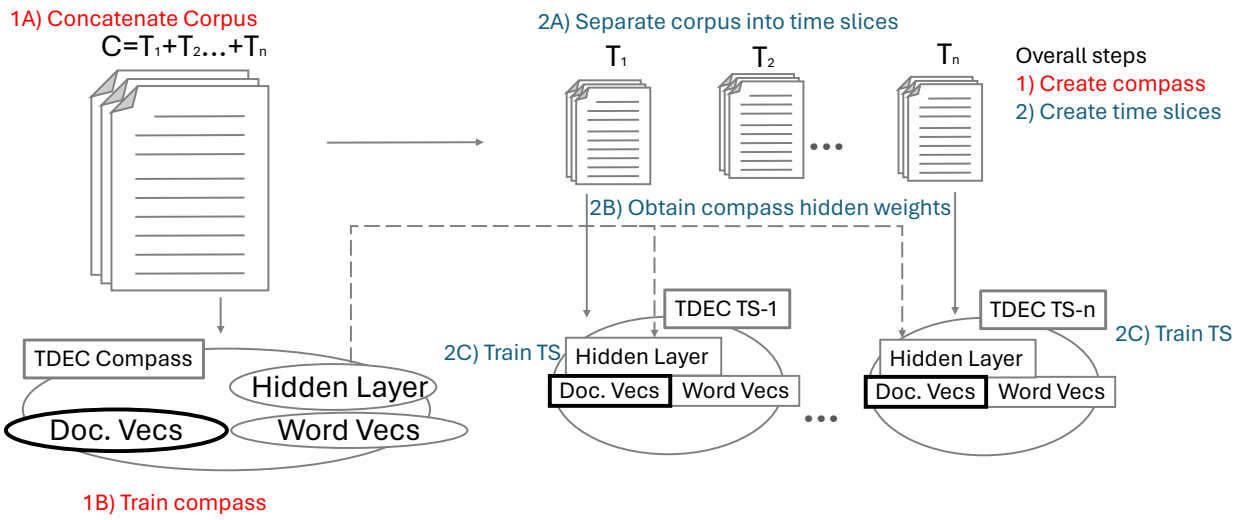

Figure 3.2: A diagram of the TDEC architecture. As shown, the TDEC **compass** is a Doc2vec model, which has word vectors, document vectors, and a hidden layer. Each TDEC **time slice (TS)** is a Doc2vec model composed of word vectors, document vectors, and a frozen hidden layer taken from the compass.

as the word embeddings of the average time slice or the smallest time slice. The hope is to allow this method to scale better to larger time slices. These different methods of alpha scaling will be tested in section 5.1.

3.2 Temporal Document Embeddings with a Compass (TDEC)

To generate a DTM technique using a compass, the compass-aligned methodology must first be expanded to document vectors. By themselves, individual documents generally do not make sense temporally, since a document tends to only appear once and then not change over time, but it is a useful stepping stone toward the creation of TTEC. The training process, as with TWEC, starts by training a compass using the entire text corpus, followed by training

individual time slices. An overview of the training process can be seen in Figure 3.2.

3.2.1 Compass Generation

TDEC generates the compass from the entire text corpus using any of the Doc2vec [28] methods. This compass then serves as the basis for the alignment of each of the time slices. Functionally, this is identical to the TWEC compass creation process, except it substitutes Doc2vec for Word2vec. As a result, global word and document vectors are created in addition to the hidden layer that will be used in time slice training.

Note that, of the two Doc2vec methods, only the paragraph vector-distributed memory (PV-DM) method originally interacts with both word and document vectors simultaneously. The paragraph vector-distributed bag of words (PV-DBOW) method attempts to guess the words contained within a document using only the document vector of that document. However, as was noted by the original authors [28], the PV-DBOW method bears resemblance to the Skip-gram (SG) word vector model, so a user can generate both by combining the methods. This is an option given by the Gensim package [38], which was used for the Word2vec/Doc2vec base of this method. The word and document vectors can then indirectly influence each other through sharing the neural network's hidden layer to achieve an alignment. Afterward, the hidden layer of the compass is frozen and serves as a basis for each time slice. As such, in order to produce a word/document embedding that predicts a certain output, it would have to be similar to the global embedding, as well as the embeddings of similar words and documents across time slices. Given that documents appear once, either the local or the global document can be used. PV-DBOW would theoretically remain identical due to its lack of direct contact with word embeddings. Local PV-DM embeddings might differ from their global embeddings due to being in the presence of different word embeddings, but

such a difference was not tested for. Nonetheless, this paper elects to use locally generated PV-DM embeddings.

3.2.2 Time Slice Training

TDEC trains each time slice similar to TWEC, except it uses the Doc2vec model as the time slice model instead of Word2vec. First, the model initializes its vocabulary, using the contents of the time slice it represents. Next, TDEC initializes the weights of the hidden layer in this Doc2vec model with the corresponding parts in the compass hidden layer. Therefore, if a time slice model has "labor" in its vocabulary, then the weights that lead to the output for "labor" will be copied to the time slice from the compass. It freezes these hidden weights, and they remain unchanged during the time slice training process. Afterward, the time slice word and document vectors go through a standard Doc2vec training process, which can be performed in parallel for every time slice. Because each time slice effectively shares the compass in the hidden layer, this results in aligned word and document embeddings that can be compared across time slices. Thus, for TDEC to produce similar embeddings across different time slices, the inputs must be similar.

Additionally, this training process results in a local and global representation of a document vector. The local representation is created by the time slice, while the global one is created by the compass. Theoretically, randomness inherent to neural networks aside, only the PV-DM representation will be affected by this distinction. The PV-DBOW method only takes the document vector as the input when training document vectors, so there should be no difference between the local and global representation. The PV-DM document vector is created in the context of word vectors, which are different between local and global representations, and can thus affect the creation of document embeddings. An analysis of the difference in

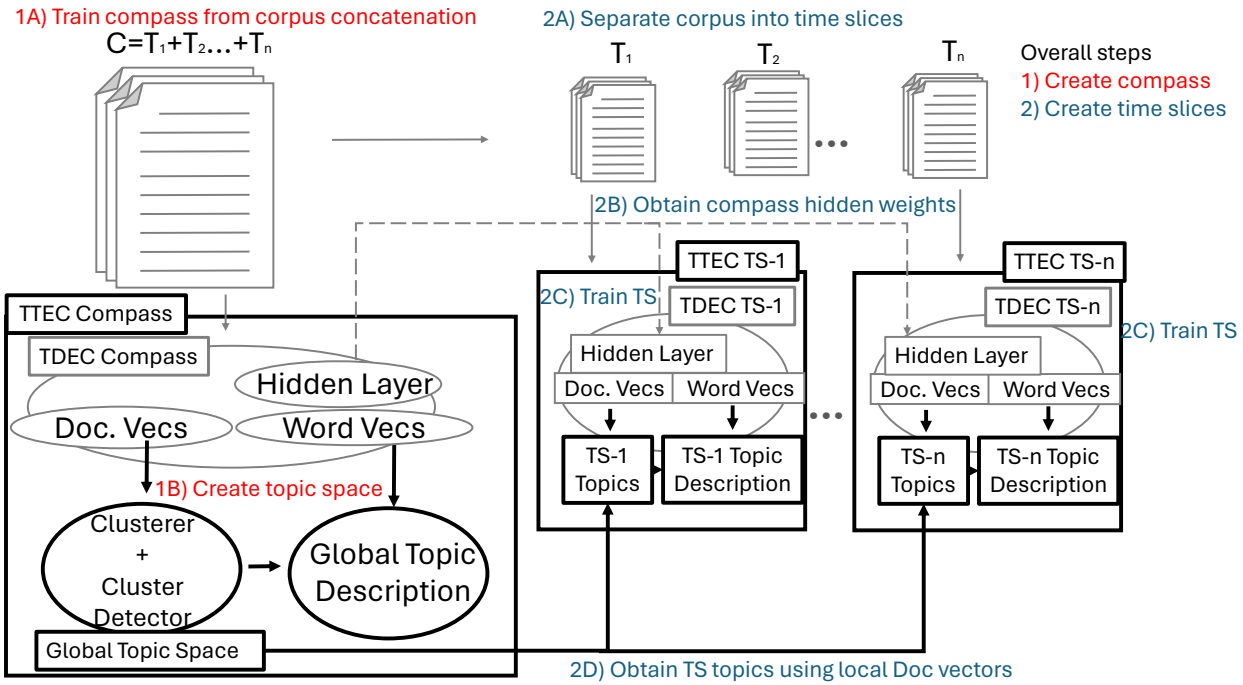

Figure 3.3: A diagram of the TTEC architecture. As shown, the TTEC **compass** has within it a TDEC compass (a Doc2vec model composed of word vectors, document vectors, and a hidden layer), a topic space (composed of a UMAP model trained on the document vectors, and HDBSCAN to identify clusters/topics in the UMAP space), and a global topic description (made by finding the most similar words to the document vectors in each topic). Each TTEC **time slice (TS)** has a TDEC time slice (a Doc2vec model composed of word vectors, document vectors, and a frozen hidden layer taken from the compass), topics specific to that time slice (created by reducing the local document embeddings into the global topic space), and a topic description (made by finding the most similar words to the document vectors in each topic).

quality between local and global representations is out of the scope of this paper. This paper opts to use locally generated document embeddings using the PV-DM architecture to create word and document embeddings.

3.3 Temporal Topic Embeddings with a Compass (TTEC)

After TDEC creates document embeddings with the compass alignment methodology, TTEC can further expand it to DTM. This allows for the fine-grained insights gained through temporally word embeddings to be placed in the context of a global topic space in the presence of documents that influence the word embeddings. The initial training process for this method is identical to that of TDEC, by creating a document-based compass and time slices. Afterward, TTEC creates a global topic representation using UMAP and HDBSCAN that it then shares with each time slice. An overview of the TTEC architecture can be seen in Figure 3.3.

3.3.1 Compass Generation

The creation of this compass follows the TDEC method and can be made using either the PV-DM or the PV-DBOW model. TTEC builds topics on top of the document embedding compass. Topic generation is a two-step process and follows the unsupervised topic generation process of combining UMAP and HDBSCAN. First, TTEC performs UMAP [31] on the global document vectors, using cosine similarity as the closeness metric. This allows for a dimension reduction that preserves local clusters and global patterns while eliminating issues presented by the curse of dimensionality. Afterward, it performs a hierarchical density-based clustering in HDBSCAN [10]. HDBSCAN efficiently performs a density-based cluster discovery using the dimensionally-reduced embeddings, which then become the global topics. In the event of too many topics, TTEC repeatedly merges the smallest topic with the closest topic, as measured by Euclidean distance between topic centroids, until it reaches the desired number of topics.

3.3.2 Time Slice Training

During this step, local time slices obtain their local word and document representations. The frozen TDEC/TTEC compass hidden layer is once again taken as a base for individual time slices. This creates local representations of document and word embeddings. TTEC then performs an additional step to obtain local document topics. Since the local and global word and document vectors all occupy the same embedding space, it is possible to place the new document vectors into the global UMAP space. Afterward, these newly placed points can be assigned a topic within the global UMAP space based on their proximity to the closest topics. As a result, it is possible to link local word and document vectors to the global topic space. Similarly to the point made in 3.2, either the local or the global representation of the document and topic can be used. However, it is possible to create local time slices using just Word2vec, since the documents have a global representation.

3.3.3 Topic Descriptor Selection

Once TTEC generates the individual time slices, there is a question of how to describe topics in the time slices. This was previously done in Doc2vec-based topic modeling [1] by calculating the top `n` most relevant words to a cluster. It is also possible to use a class-based TF-IDF [19]. We present two methods by which these topic descriptors are selected: centroid and voting.

The centroid method averages out all the document vectors in a local topic time slice to obtain a "topic vector." The `n` most similar words based on cosine similarity to this topic vector are then used to describe the topic at that particular time slice. Obtaining the most similar words of a topic at every time slice allows for a temporal adjustment of topic descriptors, as terminology is added and phased out. This method works well for a topic

with points distributed in a Gaussian distribution but does not work for atypical cluster shapes.

The voting method aims to take cluster shape into account. For every document vector of a cluster in a given time slice, the top `n` most similar words are found and kept track of. The top `n` words are then picked based on the overall results of this process. This process results in topics influenced more by the densest parts of a topic space, which complements the density-based method of topic generation. This paper chose to use the voting method due to this increased flexibility.

3.3.4 Method Summary

The following TTEC model training process was used during testing in Chapter 5

1. Preprocessing
    (a) Lowercase
    (b) Remove punctuation
    (c) Remove numbers
    (d) Remove infrequent words (optional, Word2vec does that for you)
    (e) Perform lemmatization
    (f) Separate text into time slices (TTEC wrapper does this automatically, just need to tell the number of time slices)
2. Create Doc2vec compass
    (a) Create a text corpus using all the text in all the time slices

(b) Train a Doc2vec model using text corpus to generate document and word embeddings (apply alpha scaling as needed)

(c) Train UMAP model on document embeddings

(d) Create topic space using HDBSCAN on reduced UMAP space

3. Train each time slice

(a) Create Doc2vec model and create its vocabulary space

(b) Replace the hidden weights of the model with those of the compass

i. Pick word `w` from words present in time slice Doc2vec model

ii. Obtain hidden weights for `w` from compass

iii. Place compass hidden weights for `w` in the place of the time slice hidden weights for `w`

(c) Freeze time slice hidden weights (they can no longer change during the training process)

(d) Train time slice model

(e) Place newly created document embeddings into UMAP space

(f) Use new UMAP points to predict the HDBSCAN topic of each document

(g) Obtain topic descriptors of each topic in a time slice by finding the most similar words in that time slice to the document vectors in that time slice

3.4 Summary

This chapter introduced three methods, two of which will be tested. The first method was Alpha Scaling, which decreases the compass training rate by a factor to improve time

slice embeddings. The second method was TDEC, which is similar to TWEC but replaces Word2vec models with Doc2vec. The third method was TTEC, which starts with TDEC and then builds a topic embedding layer on top of the document embeddings using UMAP and HDBSCAN. TWEC with alpha scaling and TTEC will be tested in Sections 5.1 and 5.2.

Chapter 4

Visualization Use Cases

This chapter will look at the expanded visualization potential of TTEC. It will start with a discussion of the datasets that will be used in this and the following chapter. Afterward, there will be two examples of workflows that TTEC can work with. The first example will be a term-focused analysis, starting with a term-focused analysis and then expanding into the topic space. The second example will be a topic-focused analysis, starting with an event of interest and then expanding into an analysis of word movements. Finally, there will be a section on visualization methods that did not work out.

4.1 Datasets

There will be a handful of datasets used across the following two chapters. This section will summarize each of them.

Examples in this paper were generated using the "Nuclear Corpus" dataset. It was gathered using a keyword search for articles with terms related to domestic and foreign nuclear energy

policy from NewsAPI. The dataset spans between January 2015 and July 2022 and contains 5,637,381 articles with a total of 28,663,811,702 words after pre-processing, which involves the removal of stopwords. In Chapter 5, a random sample of 10% of the original data is used for the purposes of testing (563,739 articles, 92 time slices) due to time constraints. This random sample allows for a preservation of the distribution of articles monthly. Additionally, my lab group received a set of "nuclear energy key terms" composed by a subject-matter-expert in nuclear energy. A list of key terms can be found in Appendix A. This dataset was used in Section 5.2.

The MLPC dataset [42] is a collection of 17,772 machine learning papers from arXiv between 2007 and 2015 broken into 9 time slices. The dataset is shared in a computer-readable format, so it had to be converted back to text data and turned into individual documents using the training and testing files. This dataset was used in Section 5.2 and 4.3.

The UN dataset [3] is a collection of 7507 speeches made during the United Nations General Debate between 1970 and 2015. This paper tests speeches made during the years 2006-2015, resulting in 10 time slices. Because individual speeches might discuss opinions on a variety of topics across different paragraphs, paragraphs of each speech were separated into individual documents, and documents with fewer than 20 characters were removed prior to pre-processing. This dataset was used in Section 5.2.

The New York Times News Corpus [51] was originally a corpus of 100,000 documents between 1987 and 2007. However, since then the authors of the paper gathered a bigger corpus of 2,767,052 articles across the same time period. This dataset was used in Section 5.1.

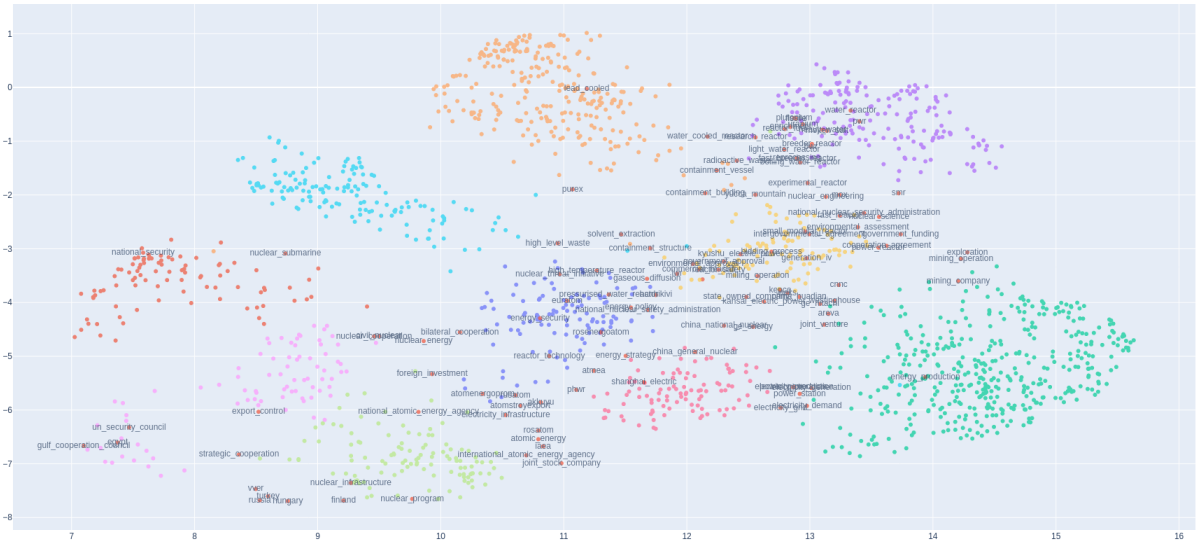

Figure 4.1: Articles in January 2015 for the Nuclear Dataset without noise with 10 topics and a 2D UMAP representation of the documents in the time slice

4.2 Nuclear Analysis

4.2.1 Single Slice Analysis

The base of this method is looking at a single time slice in isolation, which is very reminiscent of Top2vec [1]. This is expected given that Top2Vec also uses Doc2vec, UMAP, and HDBSCAN to generate document projections (although it is atemporal). Figure 4.1 looks at the first time slice of the nuclear dataset and shows words in the context of the entire document space that have associated topics. National security documents go with the term "national_security" in the red cluster on the left, while oil price-related documents occupy the green area in the bottom right corner. Nuclear energy and general energy infrastructure take up most of the center across different topics. This baseline demonstrates that the document and word embeddings generated are sensible in relative position.

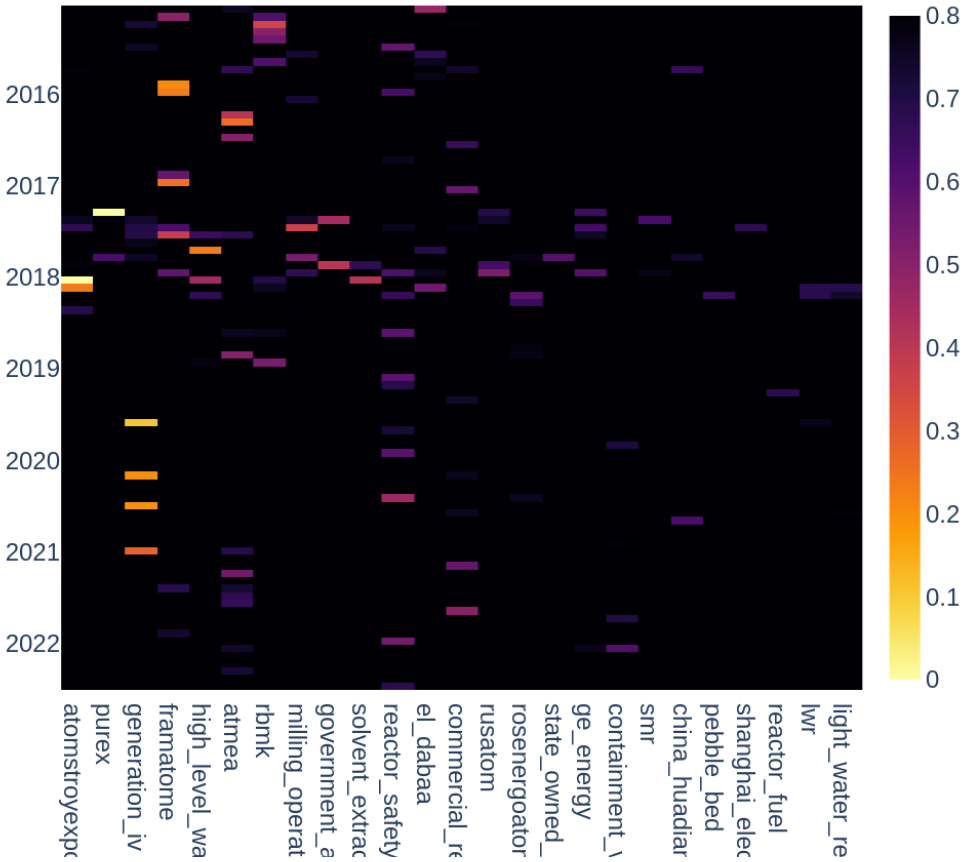

Figure 4.2: Heatmap represents the total change in cosine similarities for each term for temporal word embeddings trained using TWEC. It can be seen that the term "purex" moved significantly from April to May 2017.

4.2.2 Word-centered analysis

The base case visualization methods for the CADE family are those used in the diachronic word embedding setting. These visualizations look at word vector movements across time slices and attempt to find significant shifts. The list of keywords was derived from a list provided by a subject-matter-expert, shown in Appendix A. These shifts might include discovering gradual changes (i.e. The transition of "gay" from referring to happiness to being related to the LGBT community) or sudden shifts. Figure 4.2 shows a heatmap of select term cosine similarity between time slices. The terms have been arranged by smallest

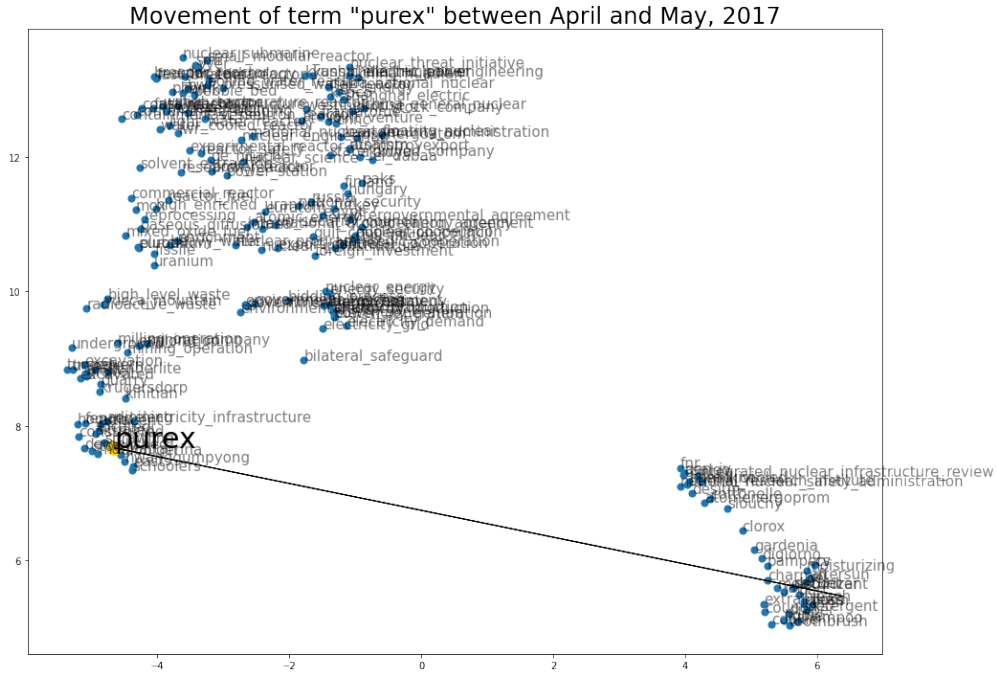

Figure 4.3: Scatterplot of UMAP representation of term "purex" between April and May in relation to other key terms.

cosine similarity within that term movement during the 92 time slice window. Note that lower cosine similarity between two time windows indicates larger movement, thereby indicating points of interest during times in which context shifts abruptly. This movement can then be visualized through a scatterplot of UMAP-projected key terms, as shown in Figure 4.3. The compass alignment approach allows us to plot two embeddings from different time slices in the same dimensionally reduced space to identify how the position (and surroundings) change between two time windows. The change in the term "purex," for instance, can be traced in the actual embedding space from closeness to cleaning products towards closeness to mining terminology. This corresponds to a mining incident at the "Plutonium Uranium Extraction Plant" (PUrEx) that occurred at the start of May 2017[1]. The identification of these abrupt changes can act as indicators of events and shifts in discussion surrounding a specific term, but context is limited given that documents and/or topics are not currently

[1]See: https://www.hanford.gov/page.cfm/purex

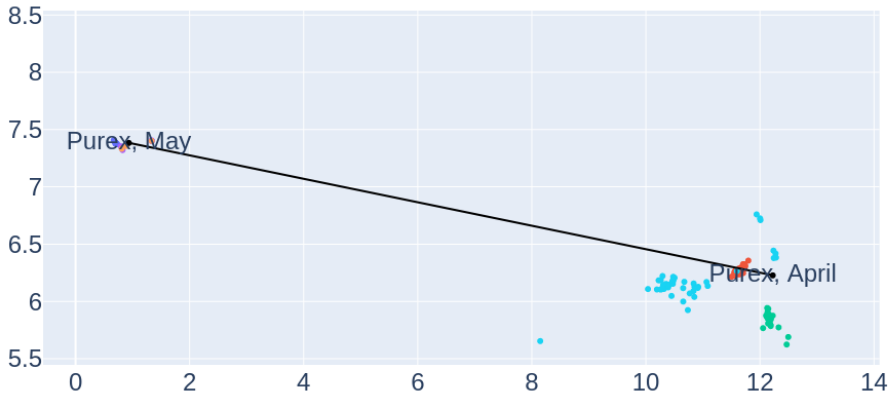

Figure 4.4: Baseline Purex comparison with documents and DTM. In April, Purex is near topics related to health (aqua), nutrition (red), and home appliances (green). In May, Purex is near geology/radioactive material (purple), archaeology (blue), and radioactive material/miscellaneous (orange)

shown. This also loses out on the topics that these documents encompass.

TTEC allows an additional layer of analysis that includes document embeddings with topics as an overall abstraction as shown by Figure 4.4. Given that the nuclear dataset is millions of articles, there are upwards of thousands of topics. Therefore, only select topics are shown in order to highlight an example of a key change of interest. Purex in April is surrounded by topics and documents relating to domestic cleaning items and is more spread out, whereas May shows a shift toward documents concerning geology, radioactive material, and archaeology. The tightness in May corresponds to articles specifically hovering around the word "purex" rather than before, where articles would be more spread out since they would be discussing different matters relating to shopping.

Looking at the articles that are closest to "purex" before and after the mine collapse (see Table 4.1), the difference in term density reflects the accuracy of the articles. Before, the articles related to "purex" were a bit scattered due to a lack of central events occurring. There

April, 2017	**May, 2017**
The eco guide to mainstream organics	Tunnel collapses at nuclear waste site in Washington state
Announcement: Kim Jong-un No Patsy	Tunnel collapses at Hanford nuclear waste site in Washington state
Iodine protection from nuclear fallout? Not always, warns the Health Ranger in science article debunking iodine myths	Emergency Declared At Nuclear-Contaminated Site In Washington State
A patch of edible delights	Tunnel collapses at Hanford nuclear waste site in Washington state, reports say
News story: Review of methods for coffee bean authenticity testing	Hanford tunnel breach confirmed in emergency at nuclear reservation

Table 4.1: Title of articles related to "purex" before and after purex mine collapse

was a humor tabloid article and a few culinary/health pieces, which are to be expected from a household commodity. The May articles, on the other hand, all explicitly mention the mine collapse. The months leading up to the collapse remain focused on health and nutrition, while afterward the news transitions to cleaning up and addressing the collapsed tunnel and then moves on to other nuclear reactor cleanup events. It is worth mentioning that the tunnel is mentioned again several months after (November/December 2017) to report on the finishing of the cleanup, though the topics dissipated by then.

4.3 Machine Learning Paper Analysis

The machine learning paper corpus will highlight the topic-centric visual capabilities of TTEC. These capabilities will start with the generated topic space and hone in on particular events. Figure 4.5 provides an overview of all the documents as they appear in the compass. The topics are semantically close to similar topics. For instance, topic 4 (natural language

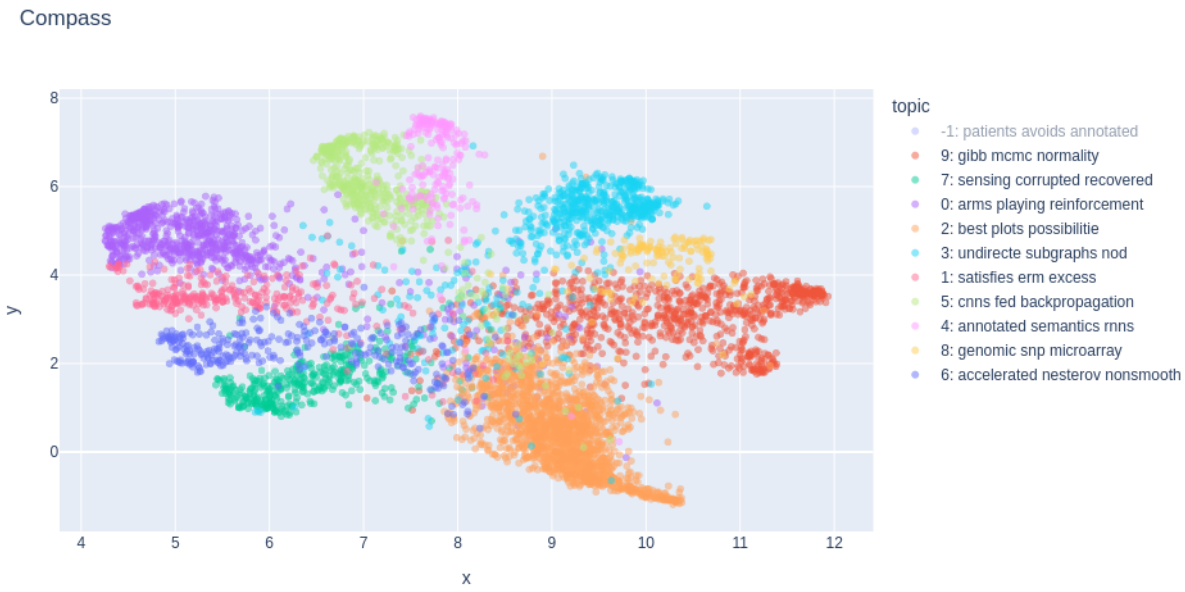

Figure 4.5: Machine learning paper corpus with ten total derived topics. The graph is a UMAP representation of the 100-dimensional document vectors. The top three words of those topics are listed. Noise was excluded.

processing [NLP]) and topic 5 (neural networks) have similar, overlapping methodologies and appear close to each other in the diagram. Relatively, in the field of machine learning, topic 3 (graph theory) is far from topic 6 (numerical optimization). This can be separated into separate time slices and viewed via frequency plot (see Figure 4.6). Topic 2 dominates, which is an unfortunate consequence of HDBSCAN where at times there ends up one large cluster. As a result of topic 2 being too large to hold a specific classification, it will be considered noise for the purpose of this section. The interactive graph shows how topic descriptors change over time, which corresponds to the change in how that topic is discussed in that particular time period.

Once more zooming is done (see Figure 4.7), noteworthy characteristics can be noticed about topics by highlighting time periods over time. In particular, topic 4 (NLP) suddenly gained popularity in 2013 and 2014 at a much more rapid pace than all other topics. A highlight over

Figure 4.6: Line graphs listing topic frequency per year. The second graph excludes cluster 1, which otherwise dominates. Hovering over a point tells the local topic descriptors at the time

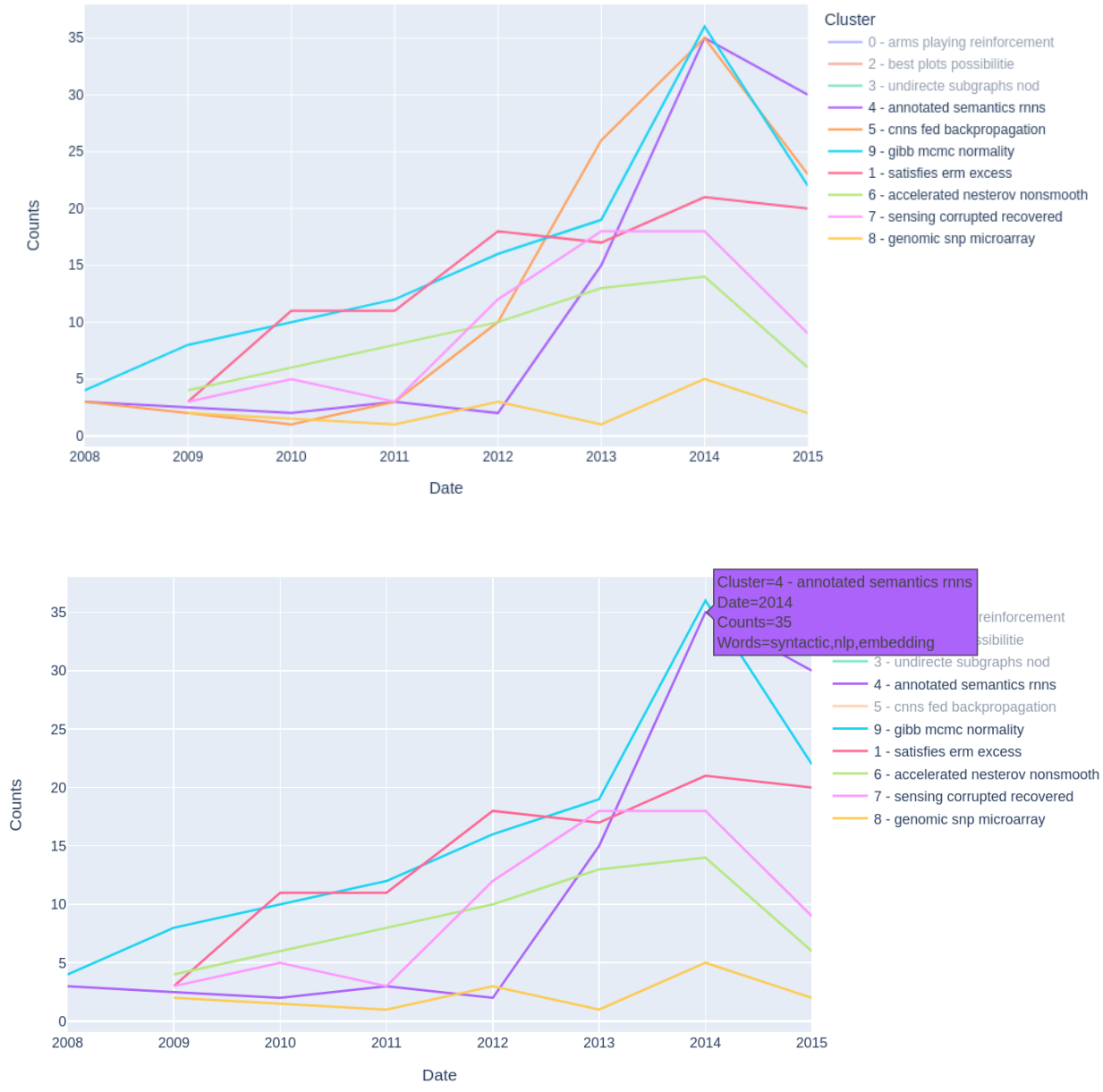

Figure 4.7: Line graph with the larger topics removed. Of note, **topic 4**, which is related to NLP, jumped drastically between 2013 and 2014, suggesting a gain in popularity at that particular time period.

the topic descriptors during that time period yields the term "embedding," which corresponds to the interest generated by word embedding architectures such as Word2vec that came out around that time [32, 34, 33, 28]. Topic 5 (neural networks) received the exact same amount of articles in 2014, though the rise is steadier and starts closer to 2012 when Krizhevsky, Sutskever, and Hinton [27] published state-of-the-art results on image classification using convolution. This rise is accompanied by an appearance of terms related to convolutional neural networks in topic descriptors for topic 5 (such as "convolution" and "convnet"). These increases in topic usage line up with the "ground knowledge" of when influential papers and architectures were published to influence a paper. This much is possible through any topic modeling architecture that generates document embeddings.

Where TTEC has the potential to gain insight is through its incorporation of a common embedding space that contains **both** documents (which have topics) and words. This allows for tracking of word vector movement through temporally aligned time slices, which contain aligned document embeddings. Thus, temporal word embeddings can be placed in relation to document embeddings into a global UMAP dimension reduction, for example, and obtain an approximate path that the term takes over time. Figure 4.8 shows this path-creating ability for the terms "embedding" and "convolutional." Both are considered noise prior to the publication of their "flagship papers," after which the term is discussed more in the fields of those papers. The terms then become identified with those particular topics, as shown by their physical manifestation inside their topic space within the globally reduced UMAP embedding space.

Figure 4.8: Path of word vector movements through the global topic space over time. It can be seen that "embedding" and "convolutional" are both considered noise prior to their rise to relevance in their fields of neural networks and NLP.

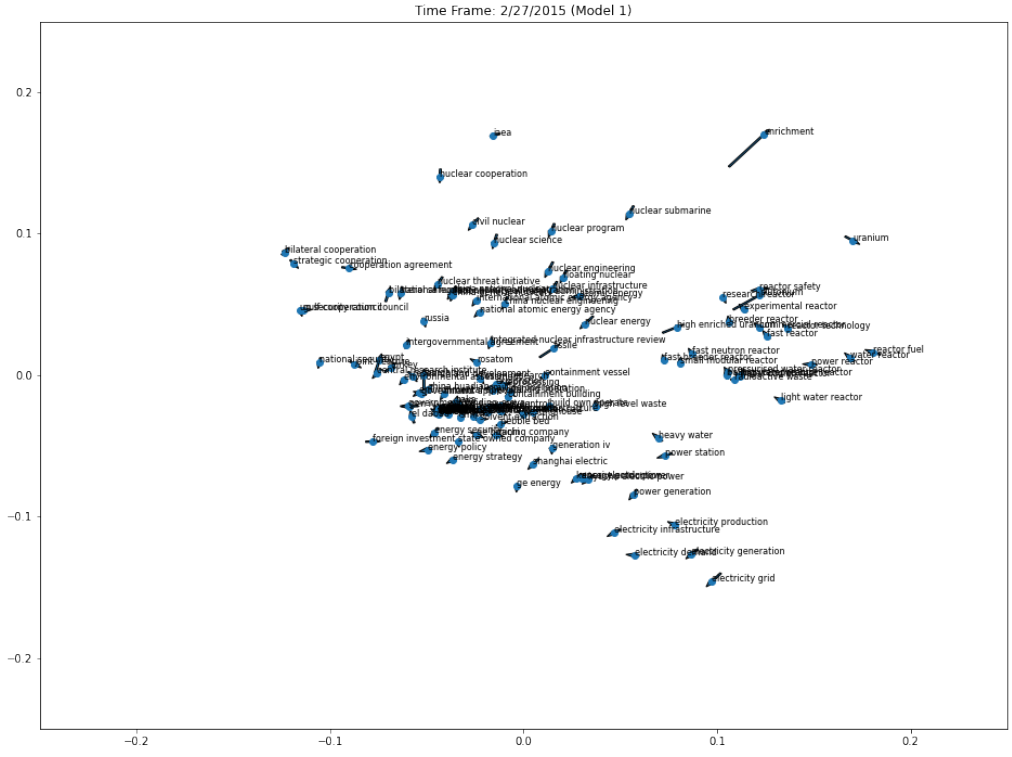

Figure 4.9: Principal Component dimension reduction plot of word vectors in time slice 0 for nuclear corpus

4.4 PCA Plots

Principal component analysis was initially tested, but was proven to have less than desirable outcomes for several reasons. The first was scalability issues. The size of Word2vec embeddings is proportional to how frequently they are mentioned, so the most frequently mentioned terms ("uranium" and "enrichment" in the case of the nuclear corpus) would end up on the edges and would have the highest effect on the generation of the principal components. The terms on the edge would also have the largest Euclidean distance movement due to the large vector magnitude exaggerating small changes in term usage. Figure 4.9 shows the difference in movement, with "enrichment" jumping the most out of all the terms. However, there is preserved a sensible structure within this plot. Terms related to electricity would be in the bottom right corner, while reactor terms would be on the right in the middle. Foreign policy terms would generally be on the left.

A second problem with using PCA plots occurred when document embeddings were added to the plot. Individual documents are given far less attention than words because a single document is one out of thousands in a particular time slice. As a result, word vectors ended up being far larger than document vectors (see Figure 4.10), since words are mentioned in several articles. While the angle at which the articles are pointing is consistent with the words they represent, the magnitude is wildly inconsistent even after hundreds of epochs of training on the documents.

The limitations of principal component analysis for visualizing combined embedding spaces ultimately led to the decision of selecting UMAP as the dimensionality reduction and visualization technique. Ultimately, UMAP inherently overcomes the challenges presented above while preserving logical structure of embeddings. Specifically, UMAP with the "cosine similarity" metric allows for a more accurate comparison between word embeddings. Such a

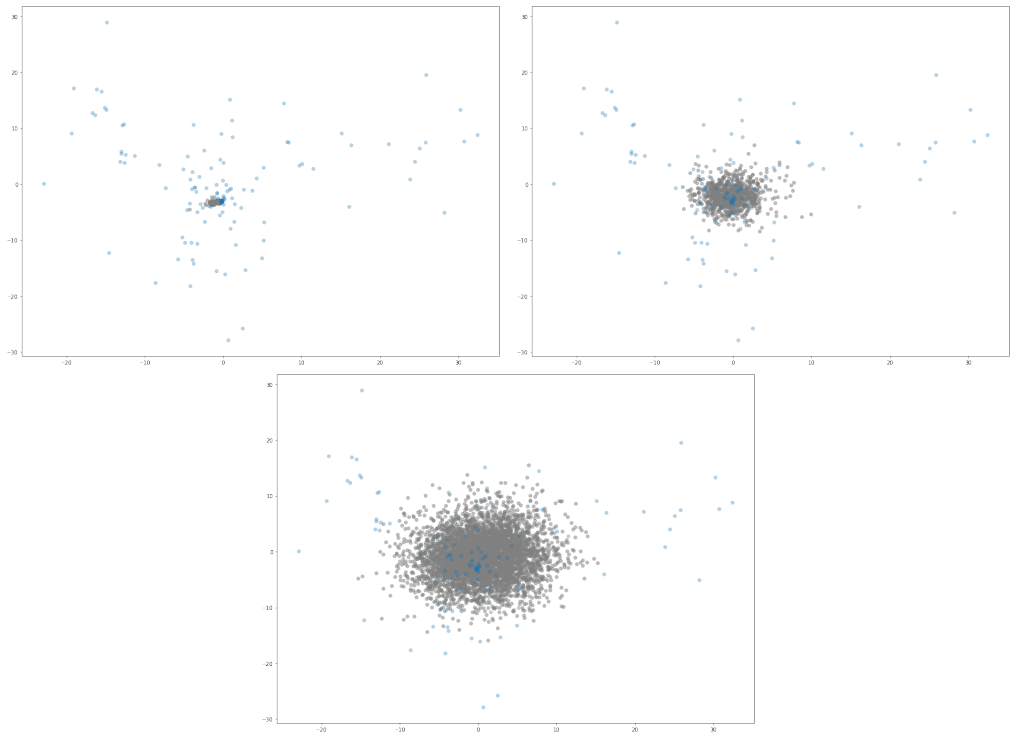

Figure 4.10: PCA plot of words and documents, with the documents being trained on frozen word embeddings (which were trained for 5 epochs) for 20, 100, and 400 epochs. Words are light blue and documents are in gray. The sizing is inconsistent even after 400 epochs for documents.

setup results in a reduced embedding space akin to Figure 4.1 where word embeddings are placed in logical locations.

Chapter 5

Quantitative Results

This chapter will look at quantitative tests surrounding my proposed methodology. I will answer the following two research questions:

1. Is "alpha scaling" (see Subsection 3.1.2) of TWEC effective in producing more accurate time slices?

2. How coherent and diverse is TTEC as a DTM method compared to other DTM methods?

5.1 TWEC Testing

There was initially testing performed on alpha scaling (introduced in Subsection 3.1.2) comparing the standard compass to the min-scale method (scale the compass to the smallest time slice) and n-scale method (scale compass based on number of time slices). This is because the time slices for the nuclear corpus could not be created otherwise (see Figure 3.1) with meaningful embeddings. As a result, there was testing performed on smaller data to see how

effective this method of parameter tuning would be.

5.1.1 Method

The test was performed using the New York Times article dataset and temporal analogies test put forth by Yao et al. [51]. The idea behind this test is to ensure that the word vector for "Obama" in the 2009 time slice is similar to the "Bush" word vector in 2002. This comparison is performed using a variety of positions that change at least yearly (President, Secretary of State, New York governor, etc.). A pairwise comparison is then made on the words within those positions on each 2-permutation of years between 1990 and 2016, inclusive.

The metrics for this test are Mean Reciprocal Rank (MRR) and Mean Precision at K (MP@K). MRR discovers the rank of the correct most similar result (r) and then assigns a score $\frac{1}{r}$ for that comparison. So, if the most similar word for vector "Bush" in 2002 in the 2009 Word2vec model is "Obama," a score of 1 is assigned. If "Obama" is in 2nd place, a score of $\frac{1}{2}$ is assigned. The result is then the mean of these scores. MP@K looks at whether the correct prediction is in the top K results. If "Obama" is in 7th place for most similar, MP@3 = 0, MP@5 = 0, and MP@10 = 1 because the term does not appear in the top 3 or the top 5, but does appear in the top 10.

The comparisons can be separated into "static" and "dynamic" comparisons. A static set has the same example across two years (Obama, 2009 $\rightarrow$ Obama, 2010). A dynamic set has different examples across two years (Bush, 2008 $\rightarrow$ Obama, 2009) This separation is useful to show how different versions of TWEC (or any other temporal word embedding model) handle comparisons when they involve the same or different terms across time slices. In theory, static comparisons should be easier to guess than dynamic due to there being no difference in person, but both are useful to note.

All	Generic	Scale-n	Scale-min
MRR	0.508	**0.552**	0.549
MP1	0.445	**0.493**	0.491
MP3	0.553	**0.598**	0.597
MP5	0.590	**0.629**	0.627
MP10	0.631	**0.658**	0.654

Table 5.1: Test set 2 results on regular TWEC, and TWEC with the compass alpha adjusted by the two proposed methods. The scaled variations of TWEC perform better.

Stat.	Generic	Scale-n	Scale-min
MRR	**0.916**	0.887	0.885
MP1	**0.902**	0.869	0.869
MP3	**0.927**	0.902	0.899
MP5	**0.931**	0.910	0.908
MP10	**0.939**	0.919	0.913

Table 5.2: TWEC results for static comparisons. The generic TWEC model performs slightly better.

5.1.2 Test Results

Looking at the results, scaling does have an effect on the alignment quality of time slices in general (see Table 5.1) and for dynamic comparisons (see Table 5.3). This translates into the most relevant result generally being closer to the top for scaled methods. For static comparisons, the results (see Table 5.2) were similar, though generic TWEC ended up being slightly better.

Dyn.	Generic	Scale-n	Scale-min
MRR	0.359	**0.430**	0.427
MP1	0.280	**0.357**	0.353
MP3	0.418	**0.487**	0.486
MP5	0.466	**0.527**	0.524
MP10	0.519	**0.562**	0.560

Table 5.3: TWEC results for dynamic comparisons. Both scaled methods dominated

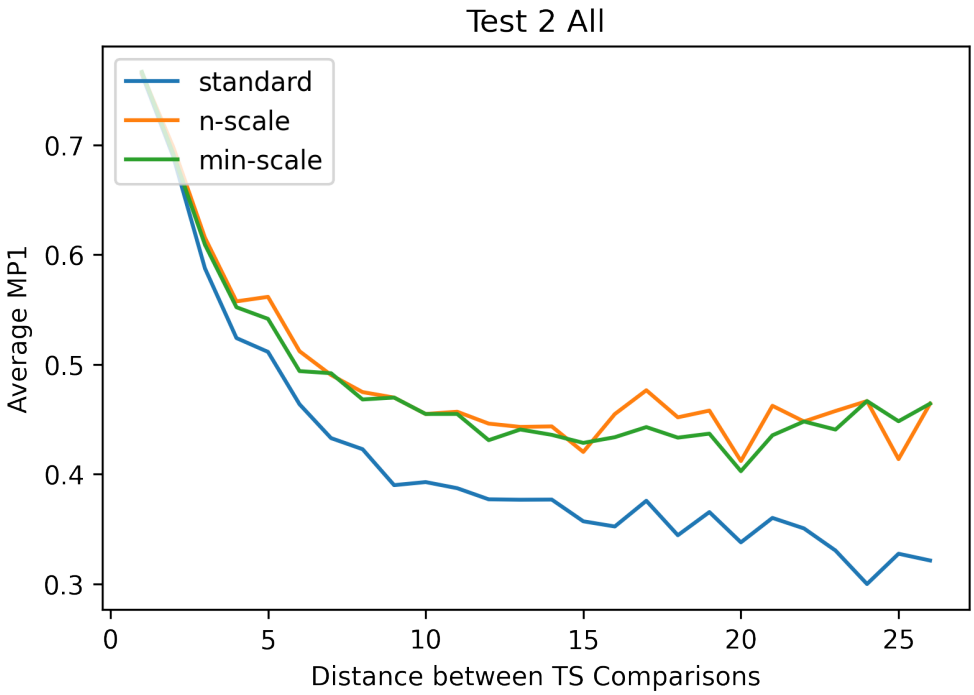

Figure 5.1: Line graphs of TWEC MP1 results overall (including both static and dynamic results) with respect to the difference in years between time slices. Scale-n performs best, with Scale-min following close behind and standard lagging after a time slice difference of five.

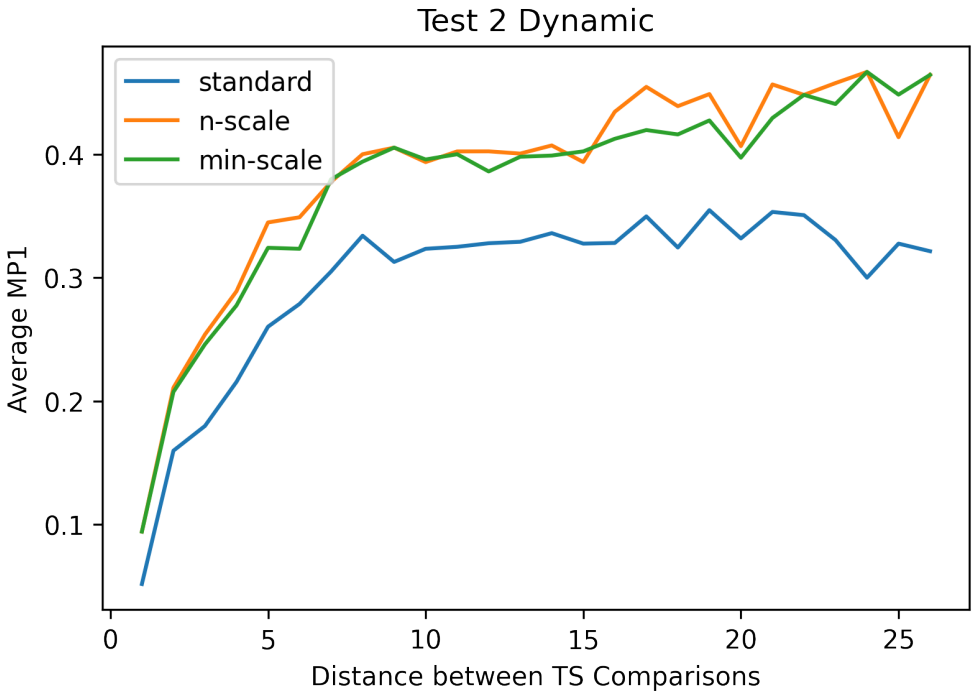

Figure 5.2: Line graphs of TWEC MP1 results for dynamic results with respect to the difference in years between time slices. Scale-n performs best, with Scale-min following close behind and standard lagging after a time slice difference of five.

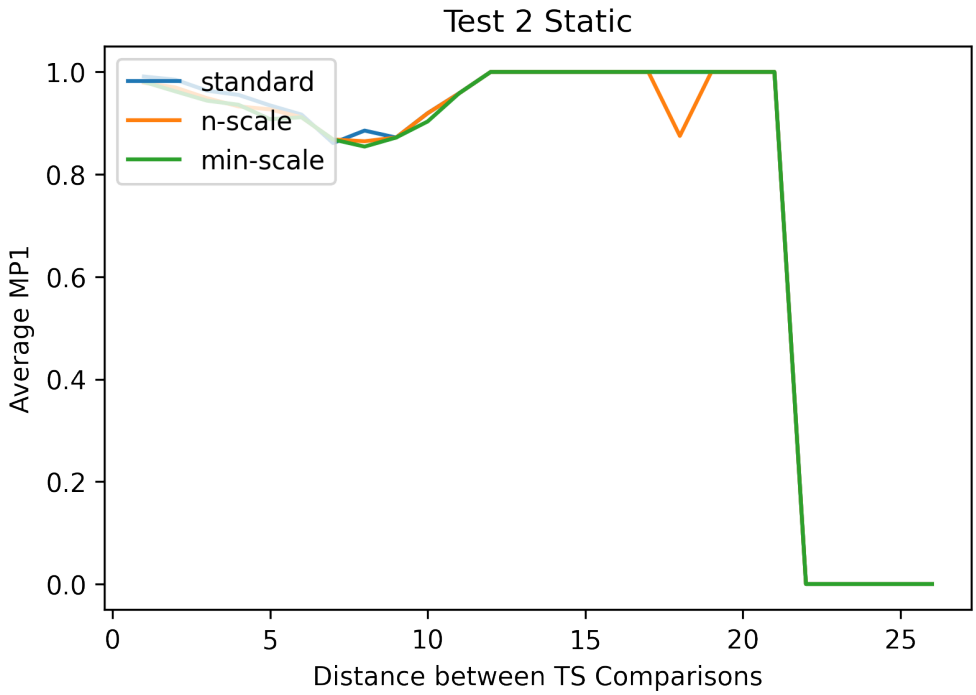

Figure 5.3: Line graphs of TWEC MP1 results for static results with respect to the difference in years between time slices. Scale-n performs best, with Scale-min following close behind and standard lagging after a time slice difference of five.

When taking the temporal distance of pairwise comparisons into account (see Figure 5.1), comparisons made a further time apart are less likely to be correct. This is to be expected and is the case with all other temporal word embedding methods, although temporal drift makes them less accurate over time [15]. It can be seen that the three versions of TWEC are initially fairly similar, though there can be a performance difference between the unscaled and scaled versions after a difference of 5 years (see Figure 5.1). This is because many more comparisons become dynamic around that period of time, so dynamic results end up being the major factor the further out the temporal difference is. The scaled performance gains in dynamic testing (See Figure 5.2) eventually make up far more comparisons as the difference in years between two time slices increases. This reasoning is why the pure dynamic (Figure 5.2) and static (Figure 5.3) line graphs look a bit strange. Most static results are in a temporal difference between 1 and 5 years, so everything after that point is strange for static comparisons due to a lack of data. Similarly, dynamic results are strangest in the first few years due to a lack of data, while later on there is more dynamic data, so the results become less chaotic.

Overall, there appears to be a large difference in the quality of TWEC results when the scale of the compass' learning rate is taken into account. This is especially important when scaling to higher time periods that potentially have more dynamic comparisons. N-scaling tends to perform better, though that will largely depend on how drastic time slice data imbalance is.

5.2 TTEC Testing

This part of the testing looks at TTEC as a DTM method in relation to other DTM methods. The goal is to compare the coherence and diversity of topics produced by each method to see how competitive TTEC is as a purely DTM method. This would contribute TTEC to a

growing DTM literature as a potential candidate for traditional DTM methodology.

5.2.1 Models

TTEC was compared to the original dynamic topic model, called here Sequential Latent Dirichlet Allocation (S-LDA) [5]. It was also compared to the transformer-based Dynamic BERTopic (D-BTopic) [19] model.

5.2.2 Datasets

TTEC was tested against the current state of the art in DTM using the Machine Learning Paper Corpus (MLPC), United Nations General Corpus (UN), and Nuclear Corpus (NC) datasets. Preprocessing was standard across all three datasets, with fixed casing, removal of numbers and punctuation, and lemmatization.

5.2.3 Test Methods

Comparison was made using Topic Coherence (TC) and Topic Diversity (TD) on the top 10 topic descriptors per topic. TC was measured using a Normalized Pairwise Mutual Information (NPMI) [7, 40] metric to see how likely words are to appear together and apart. A coherent pair of topic words will have an NPMI closer to 1, 0 implies an independence of terms, while -1 means that a pair of words never appear together in the same document. Coherence closer to one suggests that the topics are sensible. TD is a measure of the proportion of unique topic words among all topics [16]. A TD closer to 1 implies distinct topics, while a TD closer to 0 implies redundancy within topics and their descriptions. The aim is to have topics be coherent and diverse. TD and TC are measured at every time slice with

DTM Type	MLPC		UN		NC	
	TC	TD	TC	TD	TC	TD
S-LDA	-.035	.938	**.078**	.785	DNF	DNF
D-BTopic	**.006**	.957	-.104	.877	**.099**	**.935**
TTEC	-0.072	**.966**	-.307	**.985**	-.145	.912

Table 5.4: Topic Coherence and Topic Diversity of different DTM methods across three datasets. The goal of each test is to achieve a value as close to one as possible. TTEC is comparable given a dataset with large enough bodies of text (which arose with the UN dataset) and enough preprocessing to remove rare words (which was an issue with the NC dataset). S-LDA did not have enough time to finish the NC corpus despite sampling to create a smaller dataset. This is likely due to there being 92 time slices.

the average of 10, 20, 30, 40, and 50 topics and then averaged across all time slices.

5.2.4 Results

Looking at Table 5.4, it is apparent that TTEC performs in certain types of data better than others. In particular, TTEC has trouble capturing smaller pieces of text, as was the case with the UN dataset, which was broken up into paragraphs. With the UN dataset, it was noticed that there was difficulty in capturing terms that would appear together in the same documents, with almost 70% of pairwise comparisons resulting in no documents containing the two terms (and thus a coherence of -1). This is partially due to the pre-processing breaking up text into smaller paragraphs that made capturing pairwise comparisons difficult. There is no guarantee of terms existing in the text of the topic corpus as with tf-idf. This was also the case with MLPC, although to a smaller extent, with 33% of pairwise comparisons yielding no documents that shared the pair of terms. With NC, there was an issue of rarer words and non-words that made it past the frequency filter to ruin the quality of some topics, while other topics depicted desirable topics of interest (domestic and international topics of nuclear energy production). More preprocessing to remove rare words can help improve the

topic descriptor quality.

5.3 Summary

Alpha scaling is an effective way of training a compass with a large number of time slices. Scaling by a factor of `n` was slightly more effective at discovering pairwise relationships, though additional testing across varying data disparities could prove `min` scaling to be better in certain circumstances. As a DTM method, at best TTEC performs on par with the state of the art in DTM. This requires the conditions of long enough texts and sufficient preprocessing (particularly with the removal of rare words and non-words) to be present. Otherwise, there is the risk of poorly constructed document vectors and topic descriptors. Looking beyond coherence and diversity measurements, TTEC offers more granularity that is impossible to replicate with other DTM methods. S-LDA only allows for a word overview without the chance for an embedding space. BERTopic only displays document embeddings and leaves out the potential for visualizations that include word embeddings due to its reliance on c-tf-idf to create topic descriptions. TTEC offers a unique visualization method in interacting with words and documents in the same space while being comparable in topic generation capability.

Chapter 6

TTEC Users

This chapter will look at which organizations would be most likely to use and most likely to benefit from insights gained by TTEC. To aid, there will be interviews with members of various political organizations to gain insight as to how those organizations operate, what their incentives are, and what resources they have access to. Given these insights and a discussion on organizational legibility [43], the question of who the audience of TTEC is can be grasped.

6.1 Interview Description

To investigate how TTEC can be harnessed in organized structures, three interviews were conducted across a diverse set of organizations. The participants were asked a series of questions that were related to how they perceive machine learning within their organizational spaces. A list of interview questions can be found in Appendix B. In no particular order, the first participant (Participant-D) is a scientist in the United States Department of Energy with expertise in machine learning and data science. The second participant (Participant-A)

is a university student with nearly a decade of environmental activist organizing experience across a national and local level. The third participant (Participant-U) was a former staffer with the United Nations working with primarily Latin American countries to construct economic and water policy. Participant-D was the only one with extensive machine learning experience, so a brief explanation of the field and applications was provided to the other participants.

The questions asked largely fit into three categories, the results of which will be discussed in the appropriate subsections. The first category of questions asked about the use of machine learning in an interviewee's organization. The second asked about collaboration efforts within the interviewee's organization. The third asked about the challenges and limitations of machine learning within the interviewee's organization. The results of these insights will aid in figuring out the place of TTEC within an organizational structure.

6.1.1 Machine Learning Use in Organization

Participant-D was the only participant to utilize machine learning daily and mostly focused on Natural Language Processing (NLP) applications. NLP was noted as particularly useful with wide applications due to its potential for knowledge management in relation to the knowledge digitization effort and with discovering collaboration opportunities across various offices that would otherwise be handling too many individual departmental tasks to discover such an opportunity. The Department of Energy in general is able to utilize vast government resources to gather big data from sensors in various ecological contexts, including environmental regulation and nuclear material waste monitoring. These can then be fed into models to extract generalizable insights and potential warnings. Overall, there were recurring themes of using machine learning to help grapple with organizational scalability issues

inherent within such a large bureaucratic body containing millions of people and organizing so much data.

Participant-U themselves rarely used machine and statistical learning methods, though they did discuss their role in the data-gathering process for their department. Aside from using it to perform statistical tests and performing translation tasks, machine learning is not utilized much within the UN. For translation, it is especially useful due to the role of the UN as a mediator between nations and due to it being one of the core problems within machine learning [50]. Usually, it can be relied on for the task, though for more common languages such as Spanish there is usually a followup with a native speaker to ensure correct translation. The role of the UN in relation to machine learning was described to be more about advising individual governments on the best and responsible use of it. There were identified potential use cases for emergency planning and tracking political events within countries to be able to quickly identify tasks and parties of interest.

Participant-A also has not found themselves using machine learning on a daily within the activist setting. A large reason for that is that grassroots-level organizing tends to concentrate on the immediate areas of an organization. There remains open space for nationwide collaboration efforts within organizations, but the focus within individual chapters tends to be on the local environmental and political scene. As such, there is no need for massive analytical tools within these spaces. Potential identified use cases would be to help individual activists make various forms of media such as social media posts, emails, and websites. These are either mundane tasks (ensuring professional tone, ensuring smooth transition) that a tool like Chat GPT [8] can work with little risk for error or tasks to help extend an activist's current capabilities to a skillset they are unfamiliar with (code completion). There is room for machine learning tools, but there is less need to use them for organizational and analytical tasks.

6.1.2 Organization Collaborators

Looking at the ground level, the progressive activist groups of Participant-A primarily collaborate with communities on a personal level in individual neighborhoods and households. The local issues are discovered through the questioning of locals and being informed about the local political scene. There is also a cross-pollination of individuals among progressive grassroots organizations, and a collaboration effort within these organizations on a more national level, despite the dominant focus on the local scene. Looking at larger institutions, Participant-A expressed wariness when looking at business institutions, given that they have a profit motive that makes an activist uncomfortable. It is more likely that there could be a collaboration effort from a governmental research perspective, but there would remain this issue of trying to figure out how genuine the collaborator is and what the potential agenda could be in collaborating with the organization of Participant-A.

Looking at a governmental level, Participant-D emphasized the ability to collaborate between laboratories within the Department of Energy and within other departments. There is also enjoyed access to academic laboratories where the collaboration effort can produce new methods, such as the one this thesis is based on. The objectives and regulations given for the Department of Energy to accomplish are created and handed down by the legislative and bureaucratic apparatus. An effort is also made to work with industry personnel who are interested in researching technologies related to, in the case of the Department of Energy, energy technologies, and environmental cleanup.

The UN, as was mentioned, primarily collaborates with governments around the world and imposes suggestions on what governments can do on various policy matters. There is also work done with various NGOs, primarily when it comes to African countries. When an issue arises, it is important for the UN to identify the issue and assign it to the relevant govern-

ments and NGOs to effectively deal with issues. It is very much a mediating international body that way.

6.1.3 Challenges of Machine Learning Integration in Organization

There was common agreement among interviewees to remember and try to account for biases that appear in AI technologies, particularly those that are trained off internet data that is woven in with the hierarchical biases and institutions of present society. This was particularly emphasized by Participant-U, whose work in the UN with South American countries makes this issue particularly important. There exist directives within the UN for ethical AI use and training for the participants dealing in policymaking related to machine learning to ensure that this risk of bias is accounted for when advising other countries on their AI policy and usage. Participant-U also identified the current pace at which AI technologies are evolving, expressing concern about being unable to regulate or take advantage of the technologies quickly enough. There are donors to please, so when productivity increases created by AI become more commonplace, that raising of productivity will become more expected of other bodies. This concern is true "not just for the UN, it's for everyone," whether it be an economic body or a "strictly political" body not beholden to a profit motive.

Participant-D had more political and technical concerns and hopes. There are issues with data privacy that make some types of models and knowledge sharing difficult or impossible to accomplish, particularly when it comes to sensitive national security data. There is also a "problem" of needing to justify the use of machine learning models, whether the reason for such justification is sound or not. On the more sound end, there is an expectation that these models will maintain or increase safety while lowering costs for the taxpayer. On the more questionable end, there is an issue of technological illiteracy within the present legislative

bodies[1] that prevents reasonable technical regulatory legislation from being discussed. This makes policies specific to AI [12] difficult to discuss. The issue in this case has less to do with the potentiality to take advantage of technologies and more to do with institutional challenges of various degrees of legitimacy.

Looking at the grassroots level, there tend to be much fewer bureaucratic challenges and the need to appease a global donor community. The "issue" is in the interpersonal nature that these grassroots movements have. There is a certain "value [put] into individual connection and personal relationship" (Participant-A) that would be harmed if AI became a mediator. There exists such a wariness of AI within technologically literate countries such as the US [26], and even though it is becoming harder to tell AI apart from non-AI [11], there is something to be said here emotionally. Tight communities are formed at this level by greater social bonds than a simple linguistic choice that can be replicated by AI. Perhaps these attitudes will change as a new generation of kids more used to using AI in their everyday lives (such as using Chat GPT or Photomath to work on homework) becomes of "activism age." Participant-A still expressed that, despite this normalcy, it would be difficult to break apart the social bond of legwork-based organizing that grassroots movements rely on.

6.2 Legibility

My analysis will be based on the idea of "legibility" [43], also known as "bureaucratic capacity" in common discourse. It involves the idea that an effective bureaucratic apparatus needs to create categorizations and abstractions to effectively interact with the people "under it." This often results in clashing with local customs, which cannot be represented in their full complexity to an efficiently run bureaucracy. This observation is present in the bureaucracies

[1] See: https://www.cnn.com/2023/03/25/tech/tiktok-user-reaction-hearing/index.html

of modern nation-states and corporations.

The modern state can historically be traced through its development of a bureaucratic apparatus capable of effectively taxing and keeping track of its citizens. A common way to account for both is through the census, which provides the state with information on the citizens within their territory. This tool was employed since the Roman Republic and was noteworthy enough to be mentioned biblically (Luke 2:1-5). This census was administered and kept track of through a large administrative effort on the part of the Roman state and was aided in part by extensive road networks. A similar strategy of census-making was employed by more modern states such as France, which prioritized transportation networks linking to Paris [43]. To ensure that a census was doable back then, only certain rigid statistics were accounted for, which ignored customary and cultural naming conventions among individual villages and ethnic groups. Last names could be based on occupation or the stage of a person's development, but these were overridden of in favor of universal last names to make it easier for the state to keep track of family ties. Later, as technological innovation developed, census processing would be automated by machines affordable for the most part only by the US government starting as far back as 1890 [9, pp. 13–18]. Standard, machine/computer-readable forms would become commonplace to facilitate this state process for a larger population. This automation allowed for additional flexibility for minority groups, such as keeping track of tribal origins for American Natives. At the end of it all, this is a process meant to better inform the state apparatus through citizen legibility.

A similar machine-aided increase in legibility can be seen in other areas as well. To look at environmental monitoring, for instance, it was initially performed on a mass scale through a process of simplification known as "scientific forestry." Diversity of biolife would be reduced to ensure simplicity in tracking, and the one or two remaining species of tree would be categorized 1-5 depending on size [43], allowing for a small group of foresters to keep track of

forest grown over centuries as early as the 18th century. This is in contrast to the people who lived in the context of this biodiversity and were able to interact with it without the need for ecologically destructive simplification. Nowadays, the surveying efforts of individual foresters are replaced by "sensor data" which can be used to "make predictions about things that are happening in the environment" (Participant-D) through machine learning models. Legibility through categorization and damaging simplification is replaced with legibility through massive and automatic data gathering affordable only by the state apparatus. There is no longer such a blatant disregard for local ecology characteristic of early industrialists, though there remains the idea of monitoring the ecosystem.

6.3 Most Likely User

After looking at three different political contexts, a most likely organizational user of TTEC can be identified.

To address the resource requirements, TTEC requires a temporal text corpus that the user wants to analyze and discover the topics and word evolution of. This data likely is large, otherwise, the user could find out what the topics of the articles are. Such an analysis would create a general overview of dynamic words and topics. Given this, TTEC would be appropriate within bureaucratic functionality, where there is a need to effectively be able to compress large data. An analysis of all the news related to a general topic (such as nuclear energy) would require the computing power and abstraction ability of an algorithm like TTEC to be made legible. Otherwise, there is a limit to individual keyword searches, which is reasonable for finding articles relating to a particular event, but does not yield a general insight of what is going on in a text corpus. Something similar can be said of the para-governmental UN, where there is a need to discover topics from temporal text data

about the ongoings of individual countries. The United Nations corpus [3] is itself a corpus of speeches where country representatives summarize various political and economic affairs and stances within their countries. So, this sort of analysis of text data generated by countries would create a similar summary that allows an increase in legibility and response timing within this international, inter-governmental organization.

Within the context of a grassroots activist setting, there is a smaller reliance on larger algorithmic tools to gain general insight. This is due to a large focus on obtaining public grievances on policy through the actual members of the local communities that Participant-A lives in. There is no need to create a generalized topic space when the user is informed about the policy ongoings within their immediate setting. What machine learning was used was described in a convivial [24] sense, where the tool acts as an "augmentation" to increase the individual capacity of a user in their existing setting (such as enable the user to more easily create a website), unalienated from their environment. This is contrary to the analytical nature of TTEC, which is less an augment to existing capabilities, but rather a way to compress a corpus of temporal text data.

There is an interesting relationship to be shown when academia is analyzed as an organization and compared to the other bodies. An interview with an academic was not conducted, so this paragraph will rely on personal research experience across three departments to comment. The goal of academia being scientific pursuit coincides with the analytical nature of TTEC. In addition, there are often resources available for performing such an analysis, assuming that there is consistent grant funding and/or a relationship with outside government and corporate entities to provide computational and monetary resources. The results of this funding go into developing methods such as TTEC that can then be used within a bureaucratic setting. TTEC can also be applied in such a context to generate insights for collaborating bodies. The results of analysis performed in this setting can be used within an activistic setting to gain

a general insight as to the temporal literature pertaining to a topic, though that deliverable is secondary to satisfying requirements provided by the funding client. The results of this analysis would create legibility for within the academic sphere and the organizations funding the analysis.

TTEC would squarely fit in with the organizational capacity and needs of the governmental/para-governmental bureaucracy and academia. It works best on a corpus of text data that a user is not already familiar with (an activist is probably familiar with the relevant local paper coverage), and the topics that need to be discovered and labeled (which, again, an activist is probably familiar with the topics of the papers they are looking at). The task is one of legibility, which involves compressing large data to extract meaningful topic descriptions that are useful to someone trying to gain an overview of corporal evolution over time. Of course, this analytical methodology can be performed on the individual or grassroots organizational level, but those will not gain as much useful insight as a hierarchical bureaucracy that aims to understand the news ongoings in their territories.

Chapter 7

Conclusions

7.1 Performance and Scalability

TTEC performs well with respect to the state of the art in DTM. Notably, TTEC is trained using CPU resources and is parallelizable once the compass is trained due to a lack of reliance on previous time slices. Similar parallelism was not possible to achieve with S-LDA, as shown by the DNF result. If a GPU is available and analyzing the evolution of temporal word vectors is unnecessary, using BERTopic [19] could be beneficial for obtaining sentence embeddings because it does not need to be preprocessed. Compass parallelization is more difficult because of programming language limitations when editing shared data and training is memory-efficient because the actual text does not need to be kept in RAM.

7.2 Further Computational Evaluation

The present quantitative evaluation does not have a temporal aspect to it. Instead, atemporal measurements were taken at individual time slices and averaged to find an overall coherence and diversity score. Future work would present measures that emphasize the temporal aspect akin to those presented in temporal word embedding literature[51, 15]. Further work would also experiment with different hyperparameters to achieve optimal metric performance.

7.3 Further Interviews

All the interview results presented support the conclusion of TTEC serving most usefully as a tool for legibility within bureaucratic bodies. However, the limited number and scope of interviews leaves a few things to be desired. There were no interviews with people from academia, corporate entities, or citizens that would use machine learning recreationally. There was only a single interview per organization, and so additional interviews from members of those organizations could have yielded additional insights, particularly if those people had differing experiences with machine learning. Additional interviews would serve to enhance the understanding of the use of TTEC within these settings, even if the initial conclusion holds.

7.4 TDEC

The TDEC method was introduced as an intermediary between TWEC and TTEC with no methodology developed from it. Its lack of topic details misses out on a bit of abstraction, but there could be potentially a way to analyze this through the visualization of which space

document embeddings take up overall in a temporal context. Potentially, if an event has a significant enough impact, it can carve out its own area in the global document space, and this area can then be used to inform about other features in the global space. It can potentially inform about how the topics should be created. This is an area for potential future methodological development.

7.5 Transformer Embedding Representation

There exist transformer models that generate comparable document embeddings [39], and research has gone into discovering temporal and spacial properties within transformers [20] such as LLaMA 1 or 2 [48, 49]. However, these models by themselves are incapable of generating temporally changing word embeddings, resulting in a reliance on alternative methods of generating topic descriptors [19]. A mimicking of visualization potential within TTEC using transformers would require the creation of a way to generate temporal word embeddings using the transformer. Paralleling the global word vector approach of Bamman, Dyer, and Smith [2], a global representation of embeddings can be created using the generic transformer. Then, fine-tuning using the text in individual time slices can create local time slice model representations from which local word and document embeddings can be generated. The spatial and computational requirements per time slice can be reduced using Low-Rank weights [23, 13], so the local weights can be stored using low-rank differences from a single global model, rather than creating a separate model per time slice. Inspiration could be taken from CADE [4] by freezing one or more attention layers to ensure more stable alignment. Experimentation would have to be performed to analyze how robust the temporal word embeddings that arise from these fine-tuned time slices are to a changing context. If successful, transformer-backed temporal word embeddings can be used for dynamic word

embedding tasks (assuming a method for "most similar words" can be developed within this architecture) and for creating visualizations akin to TTEC.

7.6 Summary

I introduced a novel method called TTEC for Dynamic Topic Modeling based on the Compass-aligned Dynamic Word Embedding methodology. This method and TDEC are contributions to the CADE methodology. I also introduced alpha scaling as a way to improve the Compass alignment in the presence of many time slices. Alongside these methodological developments, I presented an analysis based on a history of legibility on why TTEC would be most useful within a bureaucratic governmental, para-governmental, and research setting. Through this study, TTEC presents itself as a method that combines the visual and analytical capabilities of dynamic word and topic spaces, presenting new opportunities for informative graphics that allow for various degrees of focus and granularity in discovering temporal change.

Bibliography

[1] Dimo Angelov. "Top2Vec: Distributed Representations of Topics." In: *ArXiv* abs/2008.09470 (2020). URL: https://api.semanticscholar.org/CorpusID:221246303.

[2] David Bamman, Chris Dyer, and Noah A. Smith. "Distributed Representations of Geographically Situated Language." In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Baltimore, Maryland: Association for Computational Linguistics, June 2014, pp. 828–834. DOI: 10.3115/v1/P14-2134. URL: https://aclanthology.org/P14-2134.

[3] Alexander Baturo, Niheer Dasandi, and Slava J. Mikhaylov. "Understanding state preferences with text as data: Introducing the UN General Debate corpus." In: *Research & Politics* 4.2 (2017), p. 2053168017712821. DOI: 10.1177/2053168017712821. eprint: https://doi.org/10.1177/2053168017712821. URL: https://doi.org/10.1177/2053168017712821.

[4] Federico Bianchi, Valerio Di Carlo, Paolo Nicoli, and Matteo Palmonari. "Compass-aligned Distributional Embeddings for Studying Semantic Differences across Corpora." In: *arXiv preprint arXiv:2004.06519* (2020).

[5] David M. Blei and John D. Lafferty. "Dynamic Topic Models." In: *Proceedings of the 23rd International Conference on Machine Learning*. ICML '06. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 2006, pp. 113–120. ISBN: 1595933832.

DOI: 10.1145/1143844.1143859. URL: https://doi.org/10.1145/1143844.1143859.

[6] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. "Latent Dirichlet Allocation." In: *J. Mach. Learn. Res.* 3.null (Mar. 2003), pp. 993–1022. ISSN: 1532-4435.

[7] Gerlof J. Bouma. "Normalized (pointwise) mutual information in collocation extraction." In: 2009. URL: https://api.semanticscholar.org/CorpusID:2762657.

[8] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. "Language Models are Few-Shot Learners." In: *CoRR* abs/2005.14165 (2020). arXiv: 2005.14165. URL: https://arxiv.org/abs/2005.14165.

[9] Martin Campbell-Kelly, William Aspray, Nathan Ensmenger, and Jeffrey R. Yost. *Computer. A History of the Information Machine*. 3rd ed. NY: Routledge, 2018, pp. 3–19.

[10] Ricardo J. G. B. Campello, Davoud Moulavi, and Joerg Sander. "Density-Based Clustering Based on Hierarchical Density Estimates." In: *Advances in Knowledge Discovery and Data Mining*. Ed. by Jian Pei, Vincent S. Tseng, Longbing Cao, Hiroshi Motoda, and Guandong Xu. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 160–172. ISBN: 978-3-642-37456-2.

[11] Elizabeth Clark, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah A. Smith. *All That's 'Human' Is Not Gold: Evaluating Human Evaluation of Generated Text*. 2021. arXiv: 2107.00061 [cs.CL].

[12] H.R.6580 117th Congress. *Algorithmic Accountability Act of 2022*. Feb. 2022. URL: https://www.congress.gov/bill/117th-congress/house-bill/6580.

[13] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. *QLoRA: Efficient Finetuning of Quantized LLMs*. 2023. arXiv: 2305.14314 [cs.LG].

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." In: *CoRR* abs/1810.04805 (2018). arXiv: 1810.04805. URL: http://arxiv.org/abs/1810.04805.

[15] Valerio Di Carlo, Federico Bianchi, and Matteo Palmonari. "Training temporal word embeddings with a compass." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 6326–6334. URL: https://aaai.org/papers/06326-training-temporal-word-embeddings-with-a-compass/.

[16] Adji B. Dieng, Francisco J. R. Ruiz, and David M. Blei. "Topic Modeling in Embedding Spaces." In: *Transactions of the Association for Computational Linguistics* 8 (July 2020), pp. 439–453. ISSN: 2307-387X. DOI: 10.1162/tacl_a_00325. eprint: https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00325/1923074/tacl_a_00325.pdf. URL: https://doi.org/10.1162/tacl%5C_a%5C_00325.

[17] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. "A density-based algorithm for discovering clusters in large spatial databases with noise." In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. KDD'96. Portland, Oregon: AAAI Press, 1996, pp. 226–231.

[18] Karl Pearson F.R.S. "LIII. On lines and planes of closest fit to systems of points in space." In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (1901), pp. 559–572. DOI: 10.1080/14786440109462720.

[19] Maarten Grootendorst. *BERTopic: Neural topic modeling with a class-based TF-IDF procedure*. 2022. arXiv: 2203.05794 [cs.CL].

[20] Wes Gurnee and Max Tegmark. "Language Models Represent Space and Time." In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=jE8xbmvFin.

[21] William L. Hamilton, Jure Leskovec, and Dan Jurafsky. *Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change*. 2016. URL: https://api.semanticscholar.org/CorpusID:5480561.

[22] Zellig S. Harris. "Distributional Structure." In: *WORD* 10.2-3 (1954), pp. 146–162. DOI: 10.1080/00437956.1954.11659520. eprint: https://doi.org/10.1080/00437956.1954.11659520. URL: https://doi.org/10.1080/00437956.1954.11659520.

[23] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. "LoRA: Low-Rank Adaptation of Large Language Models." In: (2021). arXiv: 2106.09685 [cs.CL].

[24] Ivan Illich. *Tools for Conviviality*. Great Britain: Fontana/Collins, 1973, pp. 9–114.

[25] Tsunenori Ishioka. "Extended K-means with an Efficient Estimation of the Number of Clusters." In: *Intelligent Data Engineering and Automated Learning — IDEAL 2000. Data Mining, Financial Engineering, and Intelligent Agents*. Ed. by Kwong Sak Leung, Lai-Wan Chan, and Helen Meng. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 17–22. ISBN: 978-3-540-44491-6.

[26] Shivani Kapania, Oliver Siy, Gabe Clapper, Azhagu Meena SP, and Nithya Sambasivan. ""Because AI is 100% Right and Safe": User Attitudes and Sources of AI Authority in India." In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. CHI '22. New Orleans, LA, USA: Association for Comput-

ing Machinery, 2022. ISBN: 9781450391573. DOI: 10.1145/3491102.3517533. URL: https://doi.org/10.1145/3491102.3517533.

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks." In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[28] Quoc V. Le and Tomas Mikolov. *Distributed Representations of Sentences and Documents*. 2014. arXiv: 1405.4053 [cs.CL].

[29] Laurens van der Maaten and Geoffrey Hinton. "Visualizing Data using t-SNE." In: *Journal of Machine Learning Research* 9.86 (2008), pp. 2579–2605. URL: http://jmlr.org/papers/v9/vandermaaten08a.html.

[30] J. MacQueen. "Some methods for classification and analysis of multivariate observations." In: 1967. URL: https://api.semanticscholar.org/CorpusID:6278891.

[31] Leland McInnes, John Healy, and James Melville. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. 2020. arXiv: 1802.03426 [stat.ML].

[32] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. *Efficient Estimation of Word Representations in Vector Space*. 2013. arXiv: 1301.3781 [cs.CL].

[33] Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. *Exploiting Similarities among Languages for Machine Translation*. 2013. arXiv: 1309.4168 [cs.CL].

[34] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. *Distributed Representations of Words and Phrases and their Compositionality*. 2013. arXiv: 1310.4546 [cs.CL].

[35] Frederic Morin and Yoshua Bengio. "Hierarchical Probabilistic Neural Network Language Model." In: *International Conference on Artificial Intelligence and Statistics*. 2005. URL: https://api.semanticscholar.org/CorpusID:1326925.

[36] Frank Nielsen. "Hierarchical Clustering." In: Feb. 2016, pp. 195–211. ISBN: 978-3-319-21902-8. DOI: 10.1007/978-3-319-21903-5_8.

[37] Dan Pelleg and Andrew W. Moore. "X-means: Extending K-means with Efficient Estimation of the Number of Clusters." In: *Proceedings of the Seventeenth International Conference on Machine Learning*. ICML '00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 727–734. ISBN: 1558607072.

[38] Radim Řehůřek and Petr Sojka. "Software Framework for Topic Modelling with Large Corpora." English. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. http://is.muni.cz/publication/884893/en. Valletta, Malta: ELRA, May 2010, pp. 45–50.

[39] Nils Reimers and Iryna Gurevych. *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*. 2019. arXiv: 1908.10084 [cs.CL].

[40] Michael Röder, Andreas Both, and Alexander Hinneburg. "Exploring the Space of Topic Coherence Measures." In: *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. WSDM '15. Shanghai, China: Association for Computing Machinery, 2015, pp. 399–408. ISBN: 9781450333177. DOI: 10.1145/2684822.2685324. URL: https://doi.org/10.1145/2684822.2685324.

[41] Peter J. Rousseeuw. "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis." In: *Journal of Computational and Applied Mathematics* 20 (1987), pp. 53–65. ISSN: 0377-0427. DOI: https://doi.org/10.1016/0377-0427(87)90125-7. URL: https://www.sciencedirect.com/science/article/pii/0377042787901257.

[42] Maja Rudolph and David Blei. "Dynamic Embeddings for Language Evolution." In: *Proceedings of the 2018 World Wide Web Conference*. WWW '18. Lyon, France: International World Wide Web Conferences Steering Committee, 2018, pp. 1003–1011. ISBN: 9781450356398. DOI: 10.1145/3178876.3185999. URL: https://doi.org/10.1145/3178876.3185999.

[43] James C. Scott. *Seeing Like a State. How Certain Schemes to Improve the Human Condition have Failed*. Yale: Yale University Press, 1998.

[44] Terrence Szymanski. "Temporal Word Analogies: Identifying Lexical Replacement with Diachronic Word Embeddings." In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Ed. by Regina Barzilay and Min-Yen Kan. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 448–453. DOI: 10.18653/v1/P17-2071. URL: https://aclanthology.org/P17-2071.

[45] Robert L. Thorndike. "Who belongs in the family?" In: *Psychometrika* 18.4 (Dec. 1953), pp. 267–276. ISSN: 1860-0980. DOI: 10.1007/BF02289263. URL: https://doi.org/10.1007/BF02289263.

[46] Robert Tibshirani, Guenther Walther, and Trevor Hastie. "Estimating the Number of Clusters in a Data Set Via the Gap Statistic." In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 63.2 (Jan. 2002), pp. 411–423. ISSN: 1369-7412. DOI: 10.1111/1467-9868.00293. eprint: https://academic.oup.com/jrsssb/article-pdf/63/2/411/49590410/jrsssb_63_2_411.pdf. URL: https://doi.org/10.1111/1467-9868.00293.

[47] Warren S. Torgerson. "Multidimensional scaling: I. Theory and method." In: *Psychometrika* 17 (1952), pp. 401–419.

[48] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. "LLaMA. Open and Efficient Foundation Language Models." In: (2023). arXiv: 2302.13971 [cs.CL].

[49] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. *Llama 2. Open Foundation and Fine-Tuned Chat Models*. 2023. arXiv: 2307.09288 [cs.CL].

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. *Attention is All You Need*. Long Beach, California, USA, 2017.

[51] Zijun Yao, Yifan Sun, Weicong Ding, Nikhil Rao, and Hui Xiong. "Dynamic Word Embeddings for Evolving Semantic Discovery." In: *Proceedings of the Eleventh ACM In-*

ternational Conference on Web Search and Data Mining. WSDM '18. Marina Del Rey, CA, USA: Association for Computing Machinery, 2018, pp. 673–681. ISBN: 9781450355810. DOI: 10.1145/3159652.3159703. URL: https://doi.org/10.1145/3159652.3159703.

Appendices

Appendix A

Appendix - Nuclear energy terms

akkuyu, areva, atmea, atomic energy, atomenergoprom, atomredmetzoloto, atomspetstrans, atomstroyexport, atomtechexport, bidding process, bilateral cooperation, bilateral safeguard, boiling water reactor, breeder reactor, build own operate, bwr, central research institute, china general nuclear, china huadian, china national nuclear, china nuclear engineering, civil nuclear, commercial reactor, cnnc, containment building, containment structure, containment vessel, cooperation agreement, electricity demand, electricity generation, electricity grid, electricity infrastructure, electricity production, el dabaa, energy production, energy policy, energy security, energy strategy, enrichment, environmental approval, environmental assessment, euratom, eurodif, experimental reactor, exploration, export control, fast breeder reactor, fast neutron reactor, fast reactor, fissile, fission research, floating nuclear, fnr, foreign investment, framatome, gaseous diffusion, ge energy, ge hitachi, gen iv, generation iv, government approval, government funding, gulf cooperation council, hanhikivi, heavy water, high temperature reactor, high enriched uranium, high level waste, iaea, integrated nuclear infrastructure review, intergovernmental agreement, international atomic energy agency, joint stock company, joint venture, kansai electric power, kepco, kyushu elec-

tric power, lead cooled, light water reactor, lwr, milling operation, mining company, mining operation, molten salt, mox, national atomic energy agency, national nuclear safety administration, national nuclear security administration, national security, non-proliferation treaty, nuclear energy, nuclear engineering, nuclear cooperation, nuclear infrastructure, nuclear program, nuclear science, nuclear submarine, nuclear threat initiative, paks, pebble bed, phwr, plutonium, power generation, power reactor, power station, pressurised water reactor, pwr, purex, radioactive waste, rbmk, research and development, reactor fuel, reactor safety, reactor technology, reprocessing, research reactor, rosatom, rusatom, rosenergoatom, shanghai electric, small modular reactor, solvenske elektrarne, smr, solvent extraction, state owned company, strategic cooperation, un security council, uranium, urex process, vver, water reactor, water cooled reactor, westinghouse, yucca mountain, turkey, egypt, hungary, finland, russia

Appendix B

Appendix - Interview questions

1. Introduction + Context
    - I introduce myself and the purpose of the interview
    - The purpose is to examine the role of machine learning in the organizational setting
2. Background questions
    - Learn about the interviewee's organization and their role within it
    - Learn about the interviewee's experience with machine learning
3. Explain machine learning (if necessary)
4. Potential applications
    (a) Ask about how machine learning is applied in the interviewee's organization
    (b) Ask about how machine learning can be applied in the interviewee's organization
5. Concerns and limitations

- Ask about concerns and potential limitations and liabilities of machine learning

6. Opportunities for collaboration

   - Ask about which organizations the interviewee's organization interacts with
   - Ask about how machine learning has/can be applied within that collaborative context

7. Future potential

   - Ask about what the interviewee thinks the role of machine learning is in the future of their organization