PROBABILISTIC MODELS FOR SEMANTIC VISUALIZATION AND ITS APPLICATIONS

LE VAN MINH TUAN

SINGAPORE MANAGEMENT UNIVERSITY
2017

Probabilistic Models for Semantic Visualization and Its Applications

by
Le Van Minh Tuan

Submitted to School of Information Systems in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in Information Systems

Dissertation Committee:

Hady W. Lauw (Supervisor/Chair)
Assistant Professor
Singapore Management University

Jing Jiang
Associate Professor
Singapore Management University

David Lo
Associate Professor
Singapore Management University

Lee Wee Sun
Professor
National University of Singapore

Singapore Management University
2017

Probabilistic Models for Semantic Visualization and Its Applications

by Le Van Minh Tuan

Abstract

Visualization of high-dimensional data, such as text documents, is useful to map out the similarities among various data points. In the high-dimensional space, documents are commonly represented as bags of words, with dimensionality equal to the vocabulary size. Classical document visualization directly reduces this into visualizable two or three dimensions. Recent approaches consider an intermediate representation in topic space, between word space and visualization space, which preserves the semantics by topic modeling. These approaches consider the problem of semantic visualization which attempts to jointly model visualization and topics. With semantic visualization, documents with similar topics will be displayed nearby.

This dissertation focuses on building probabilistic models for semantic visualization by modeling other aspects of documents (i.e., document relationships and document representations) in addition to their texts. The objective is to improve the quality of similarity-based document visualization while maintaining topic quality. In addition, we find applications of semantic visualization to various problems. For document collection visualization, we develop a system for navigating a text corpus interactively and topically via browsing and searching. Another application is single document visualization for visual comparison of documents using word clouds.

Contents

Acknowledgments

I would like to thank my supervisor, Prof. Hady W. Lauw. I have learned a lot from his inspiration, insight and feedback. I am also very grateful to him for his thoughtful and diligent advising, which he has shown continually and consistently during my PhD at SMU. I have been given many opportunities to strengthen my skills on research and teaching, which I believe be very useful and stay with me for the rest of my career. I have had a fortunate experience of working with his research group which has talented students and good friends.

I would like to thank my committee members for their valuable feedback on my dissertation.

I would like to thank my parents, my sisters and brothers for their love, invaluable support and encouragement during my long absence from home. I would also like to thank my good friends for keeping me sane, and laughing throughout everything.

Finally, I would like to thank my wife and my daughter for being always with me through the most difficult times of this journey. I am lucky to have her love, confidence, pride, and patience throughout my life.

Thank you, everyone.

For my family

List of Figures

List of Tables

List of Notations

Notation	Description
d_n	a specific document
x_n	latent coordinate of d_n in the visualization space
M_n	number of words in document d_n
θ_n	topic distribution of d_n
ν_n	the observed L^2-normalized word vector of d_n
z	a specific topic
ϕ_z	coordinate of topic z in the visualization space
τ_z	L^2-normalized word vector of topic z
β_z	word distribution of topic z
$\theta_{n,z}$	probability of topic z in document d_n
W	the vocabulary (the set of words in the lexicon)
N	total number of documents in the corpus
Z	total number of topics (user-defined)
χ	the collection of x_n's for all documents
Φ	the collection of ϕ_z's for all topics
$\boldsymbol{\beta}$	the collection of β_z's for all topics
Ψ	the collective set of parameters $\{\chi, \Phi, \boldsymbol{\beta}\}$

Table 1: Notations.

Chapter 1

Introduction

Text documents come in various flavors, such as Web pages, news articles, blog posts, emails, or messages on social media such as Twitter. While much is in English, there are increasing amounts of content in various languages as well. With the backdrop of the growth in volume, diversity, and complexity of various corpora, we need more useful tools to analyze the wealth of text content. One form of analysis which we will look into in this thesis is visualization.

There are two main research directions in document visualization. Research works in the first direction focus on visualization techniques for showing the content of the text documents from different aspects and at different detail levels. Due to this objective, visualization methods in this direction usually discuss more about the effectiveness of visualization forms for displaying the content as well as the interactions that can be performed on that display. The other research direction has a different objective which is to visualize the document similarities. This objective raises research questions such as how to represent the documents and learn their similarities as well as how to map and preserve those similarities into the visualization. Those questions are interesting to the machine learning community. Another research question from the perspective of the visualization community is what visualization form we should use for displaying the similarities. A natural way to encode the similarities is to use the distances. The closer the two documents are,

the more similar they are. Therefore, there are many methods that display documents as points in a 2D/3D scatterplot visualization form and use the distances among them to encode the similarities. We will give a review of these two research directions in Chapter 2.

This thesis follows the second direction where we seek to visualize the document similarities. We are interested in the scatterplot visualization form where we can represent a collection of documents as coordinates on the same low-dimensional space, so as to learn of the similarities and differences among documents based on their distances on the visualization space. The scatterplot visualization form is useful because of two reasons. First, it is convenient for users to perceive the similarities by looking at distances. The second reason comes from the modeling aspect. By representing documents as coordinates with real values, we have a capability of building joint models that can learn the visualization automatically through numerical analysis. More than that, by integrating visualization parameters to the model, the visualization generated will reflect faithfully the information learned from the data and could eventually serve as an interface for tuning the underlying model. These two advantages of scatterplot visualization form are difficult to achieve using other more complex forms of visualization that are more concerned with aesthetics or document content.

We can treat the problem of visualizing document similarities on a scatterplot as a dimensionality reduction problem. From this viewpoint, dimensionality reduction methods can be used for projecting or embedding the documents from high dimensional representation (i.e., a vector of word counts) into lower dimensional 2D (or 3D) space. One pioneering technique is Multidimensional Scaling (MDS) [70]. The goal is to preserve the *distances* in the high-dimensional space in the low-dimensional embedding. When applied to documents, a visualization technique for generic high-dimensional data, e.g., MDS, may not necessarily preserve the topical semantics. Words are often ambiguous, with issues such as *polysemy*, when the same word carries multiple senses, and *synonymy*, when different words carry

the same sense. Because the dimensions in the original representation (which are words) may not accurately capture this ambiguity, this affects the quality of the reduced representation (which is the visualization space) as well.

To model semantics in documents in a way that can resolve some of this ambiguity, the current popular approach is by topic modeling, such as PLSA [57] or LDA [14]. Each document is associated with a probability distribution over a set of topics. Each topic is a probability distribution over words in the vocabulary. In this way, polysemous words can be separated into different topics, and synonymous words can be grouped into the same topic.

Topic modeling itself is another form of dimensionality reduction: from word space to topic space. The word space refers to a document's original representation, which is usually a bag of words. The topic space refers to the simplex of topic distributions. A document's probability distribution over topics is effectively the representation of this document in this topic space. However, a topic model by itself is not designed for visualization. While one possible visualization is to plot documents' topic distributions on a simplex, a 2D visualization space could express only three topics, which is very limiting.

Given its success in modeling semantics in documents, we therefore ask the question of whether and how best to do both forms of dimensionality reductions (visualization and topic modeling) for documents. The end goal is to arrive at a visualization of documents that is consistent with both the semantic representation (topics), as well as the original representation (words). This coupling is a distinct task from topic modeling or visualization respectively, as it enables novel capabilities. For one thing, topic modeling helps to create a richer visualization, as we can now associate each coordinate on the visualization space with both topic and word distributions, providing semantics to the visualization space. For another, the tight integration potentially allows the visualization to serve as a way to explore and tune topic models, allowing users to introduce feedback [59] to the model through a visual interface [28]. These capabilities support several use case scenarios. One

potential use case is a document organizer system. The visualization could potentially help in assigning categories to documents, by showing how closely related documents have been labeled. Another is an augmented retrieval system. Given a query, the results may include not just relevant documents, but also other similar documents (neighbors in the visualization).

As a summary, this thesis looks at the problem of visualizing document similarities on a scatterplot. It seeks to build probabilistic models for jointly modeling topics and visualization, which is referred to as the task of *semantic visualization*. The objective is to improve the quality of the scatterplot visualization while maintaining topic quality. To achieve that, we propose semantic visualization models for modeling document relationships and modeling document representations. The main idea is that by modeling other aspects of documents (i.e., document relationships and document representations) in addition to their texts, we could learn better document similarities which, when being visualized on a scatterplot, help to improve the visualization quality.

1.1 Semantic Visualization

1.1.1 Problem Statement

We refer to the task of jointly modeling topics and visualization as *semantic visualization*. The input is a corpus of documents $\mathcal{D} = \{d_1, \ldots, d_N\}$. Every d_n is a bag of words, and w_{nm} denotes the m^{th} word in d_n. For a specified number of topics Z and visualization dimensionality (assumed to be 2D, without losing any generality), the objective is to learn, for each d_n, a latent distribution over Z topics $\{\mathrm{P}(z|d_n)\}_{z=1}^{Z}$. Each topic z is associated with a parameter β_z, which is a probability distribution $\{\mathrm{P}(w|\beta_z)\}_{w \in W}$ over words in the vocabulary W. The words with the highest probabilities for a given topic capture the semantic of that topic.

In semantic visualization, there is an additional objective for semantic visualization, which is to learn, for each document d_n, its latent coordinate x_n on a low-

dimensionality visualization space. Similarly, each topic z is associated with a latent coordinate ϕ_z on the visualization space. A document d_n's topic distribution is then expressed in terms of the Euclidean distance between its coordinate x_n and the different topic coordinates $\Phi = \{\phi_z\}_{z=1}^{Z}$. Intuitively, the closer is x_n to a topic's ϕ_z, the higher is $\mathrm{P}(z|d_n)$ or the probability of topic z for document d_n. While we focus on documents in our description, the same approach would apply to visualization of other data types for which latent factor modeling, i.e., topic model, makes sense.

1.1.2 Approaches

A straightforward way is to undergo two-step reductions. In the first reduction, the original representation for documents are reduced into topic distributions using topic modeling. In the second reduction, documents' topic distributions are further reduced into visualization coordinates. This approach may have some value compared to direct reduction from word space to visualization space. However, it is not ideal, because the disjoint reductions could mean that errors may propagate from the first to the second reduction, and the resulting visualization may not faithfully capture the original representation.

A better way to solve this problem is to join up the two reductions into a single, joint process that produces both topic distributions and visualization coordinates. This approach was first pioneered by PLSV [62], which also showed that the joint approach outperformed the disjoint approach. PLSV derives the latent parameters by maximizing the likelihood of observing the documents. This goal is concerned with the "error" between the model and the observation.

PLSV is built upon topic modeling technique Probabilistic Latent Semantic Analysis (PLSA) [57] by incorporating visualization coordinates of documents and topics. Similar to PLSA, PLSV makes following assumptions in its generative model:

1. **Documents are generated independently.** This assumption implies that PLSV is not concerned with the relationships among documents, though it

can infer latent semantic relationships among documents by topic modeling of text. However, besides text, documents can appear with other information about their relationships such as neighborhood or network structures. These relationship structures may contain important patterns among documents which should be captured and preserved for a faithful visualization of documents. Therefore, in Part I of this thesis, we focus on modeling document relationships with semantic visualization. The aim is to build models for semantic visualization that preserve both latent semantic relationships learned from topic modeling and the relationship structures exhibited in the data.

2. **Documents are represented as bags of word counts.** PLSV represents documents as bags of word counts. However, as pointed out by Reisinger et al. [101], this type of representation cannot model word absences and it is also sensitive to document lengths. We therefore want to investigate other types of document representation for semantic visualization. In this thesis, we focus on two types of representation. The first is spherical representation where documents are represented as unit vectors (i.e., L^2-normalized vectors). The second is bag of word vectors where each word is embedded as a vector in a high dimensional space and a document is then represented as a bag of these word vectors. These two types have some advantages over the bag of word counts representation as pointed out in the next section.

1.2 Overview

Figure 1.1 shows a graphic overview of this thesis. In general, we propose two approaches of semantic visualization, one for modeling document relationship and the other for modeling document representation. Under each approach, we propose methods for modeling different kinds of document relationship and document representation. Note that the two approaches are orthogonal in the sense that we can combine a method from an approach to a method in the other approach. Next

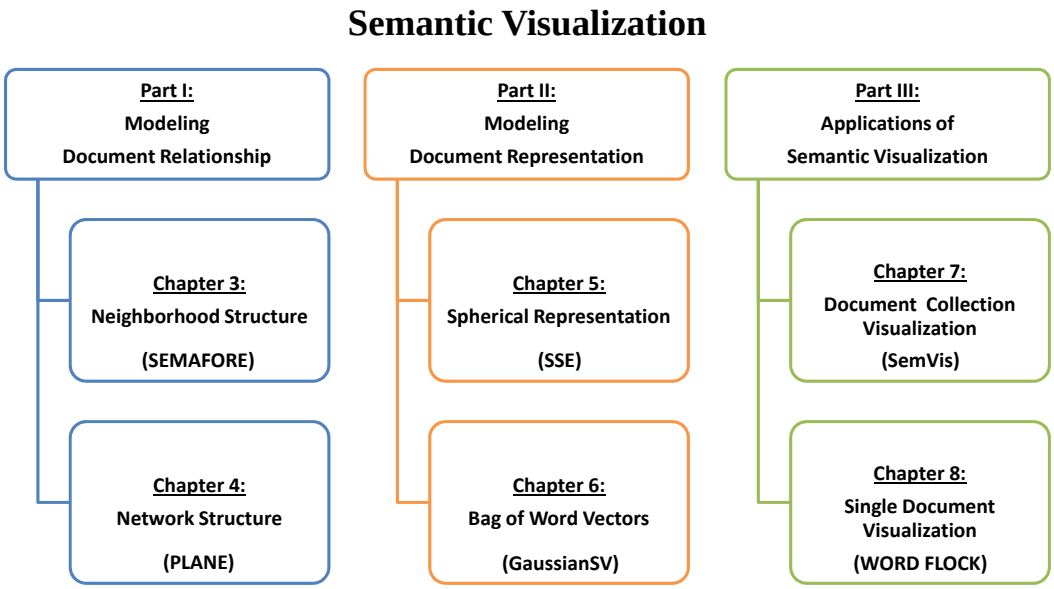


Figure 1.1: A graphic overview of this thesis

sections give an overview of different semantic visualization models and its applications for single document and document collection visualization.

1.2.1 Modeling Document Relationship

Neighborhood Structure. In the literature, it is found that algorithms that ensure "smoothness" tend to perform better at learning tasks [128]. Smoothness concerns preserving the observed proximity between documents. This objective arises naturally from the assumption that the intrinsic geometry of the data is a low-rank, non-linear subspace within the high-dimensional space. Therefore, preserving neighborhood structure is important for learning tasks. This assumption is well-accepted in the machine learning community [71], and finds application in both supervised and unsupervised learning [9, 128, 129]. Recently, there is a preponderance of evidence that this assumption also applies to text data in particular [17, 18, 60]. We therefore propose an unsupervised probabilistic model that jointly derives topic distributions and visualization coordinates while preserving neighborhood structure as well. Our proposed model is called SEMAFORE, which stands for *SE*mantic Visualization with *MA*ni*FO*ld *RE*gularization. We build a neighborhood regularization framework into a semantic visualization model. The framework involves new issues to resolve, in-

cluding the regularization function, and the space in which regularization should take place.

The model is evaluated on a series of real-life, publicly available datasets, which are also benchmark datasets used in document classification task. An advantage of a statistical method, such as ours, is that it is not dependent on a specific language. Two of the datasets are in English, and one is in Brazilian Portuguese. While our model is unsupervised (class label is neither required nor used in learning), to objectively quantify the visualization quality, we leverage on the class label information. It is a common assumption that documents of the same class are expected to be neighbors on the original space [10, 128, 129], which suggests that they should also be close on the visualization space. We investigate the effectiveness of SEMAFORE in placing documents of the same class nearby on the visualization space, and systematically compare it to existing baselines without one or more of our properties, namely: joint modeling of topic and visualization, or neighborhood regularization. This work is presented in Chapter 3.

Network Structure. Besides text, we may observe links such as citations among documents. This network structure may contain important patterns of document relationships that should be discovered and preserved when doing visualization. Network structure, on the other hand, can complement document visualization, by providing more information about how documents are connected and related. To visualize document networks, we can rely on graph embedding techniques such as SPE [108], Fruchterman and Reingold layout [45] or Kamada and Kawai layout [66]. However, these methods does not model text and thus the generated visualization does not express semantic relationships among documents. PLSV, on the other hand, does not model network, which may lose important information about relationships of documents contained in the network. Therefore, we seek to model both text and links in semantic visualization such that it preserves both semantic and network relationships.

We propose an approach based on two key principles. The first principle is to

embed both text and network representations of a document into a single unified low-rank representation such that it preserves both semantic and network relationships. The second principle is to *incorporate both a topic model and an embedding model within a single joint model*. This principle is aligned with the spirit of semantic visualization that aims to infuse the visualization with semantic interpretability. We implement these two principles by proposing a generative model called PLANE, which stands for *P*robabilistic *LA*tent Document *N*etwork *E*mbedding. The generative model explains the process of generation of observable data (text and network) from latent representations (topics and visualization coordinates). We validate this model on four real-life document networks derived from a benchmark collection of academic publications. We compare our model, quantitatively as well as qualitatively, against comparable baselines on both aspects (embedding and topic modeling) on a number of objective evaluation metrics. Chapter 4 presents the work in detail.

1.2.2 Modeling Document Representation

While neighborhood structure and network structure are important for semantic visualization. Another important aspect is how documents are represented in semantic visualization. Similar to LDA, PLSV represents each document as a bag of word counts and relies on multinomial distribution to compute the likelihood of text data. Multinomial modeling is known that it cannot model word absences and it is also sensitive to document lengths [101].

Spherical Representation. The above-mentioned issues of word count representation can be addressed by representing documents as unit vectors (i.e., L^2-normalized vectors) which lie on a unit hypersphere. In this spherical space, relationships between documents are measured as cosine similarity $\in [0, 1]$, which is the angular distance between two directional unit vectors. Firstly, two documents would have higher cosine similarity, not only if some words in common are present, but also if some other words in common are absent. Secondly, the normalization of

all documents to unit vectors effectively neutralizes the impact of document lengths. Moreover, there is indicative evidence from the literature that a spherical approach will be promising in terms of dimensionality reduction. For instance, the spherical topic model SAM [101] performs significantly better than the multinomial topic model LDA [14], when used as a dimensionality reduction technique. Therefore, in Chapter 5, we propose a semantic visualization model for documents with spherical representation.

In our model, documents and topics are represented as L^2-normalized vectors which lie on a unit hypersphere. For each document, its vector is drawn from a von Mises-Fisher (vMF) distribution [84] with the mean equal to the average of topic vectors weighted by the document's topic distribution. Topic distribution of a document is derived from its distances to topics in the visualization space. To combine these, we propose a generative model which implies a mapping from visualization space to topic space to original data space. The model is called SSE, which stands for *S*pherical *S*emantic *E*mbedding. We estimate the model parameters based on variational inference and validate SSE through experiments on publicly available real-life datasets, showing significant gains in visualization quality and topic interpretability.

Bag of Word Vectors. Another type of document representation we are looking at is the bag of word vectors. Word embedding models such as Word2Vec [88] and GloVe [99] learn for each word a vector in an embedding space. They are usually trained from a very large corpora (e.g. Wikipedia or Google News) to derive quality word vectors which encode conceptual similarity of words. For topic modeling, it infers topics based on the co-occurrence of words in the documents. Therefore, it may not perform well for short texts due to the lack of word co-occurrences. Word vectors can be used to alleviate the problem by providing auxiliary information about word similarities. It has been proved that by modeling word vectors, topic modeling can work well with corpus having sparsity problem [37, 58, 93].

In Chapter 6, we address the problem of semantic visualization for short texts.

We present a method, called *Gaussian Semantic Visualization* or GaussianSV, assuming that each topic is characterized by a Gaussian distribution on the word embedding space. Words in a document are then generated by a mixture of these Gaussian distributions weighted by the document's topic distribution. The experiments show that GaussianSV outperforms pipelined baselines that derive topic models and visualization coordinates as disjoint steps, as well as semantic visualization baselines that do not consider word vectors.

1.2.3 Applications of Semantic Visualization

Semantic visualization focuses on dimensionality reduction aspect of visualization which aims to represent documents as points on a 2D/3D scatterplot. This itself is an application where we can see document similarities based on the distances among them. In this section, we present other applications of semantic visualization for single document visualization and document collection visualization. Single document visualization is mostly used to visualize content of a single document for getting an overview. Meanwhile, document collection visualization focuses on discovering and visualizing patterns such as similarities among documents. Below we give an overview of these applications of semantic visualization.

Document Collection Visualization. We build a visualization system, called SemVis, for interactive topical analysis of a document collection. The core of this system is the semantic visualization model which is used to discover topics and learn an embedding of documents in a 2D/3D visualization space. In Chapter 7, we illustrate how SemVis could be used to navigate a text corpus interactively and topically via browsing and searching.

Single Document Visualization. There are many methods which use different kinds of visualization form for visualizing a document. Here, we focus on word clouds visualization. Word clouds display a subset of words within a document, by assigning greater visual prominence to more important words. The importance of words is usually derived by their frequency in the document.

In this work, we seek effective visual comparison of documents via word clouds. Ideally, documents with similar contents have word clouds of similar appearances. Traditional approaches fall short of this ideal, as word clouds of different documents are generated independently using a layout algorithm [106, 121]. Two documents may feature similar words that are placed in different colors and positions within their respective word clouds, placing a burden on the viewer in corroborating their similarities. To overcome these issues, we propose a technique called WORD FLOCK that integrates *two levels* of "synchronization" principles for word clouds: *similar documents* share similar word clouds, and *related words* of the same latent aspects are displayed similarly. This work is presented in detail in Chapter 8.

1.3 Contributions

While visualization and topic modeling are, separately, well-studied problems, the interface between the two, semantic visualization, is a relatively new problem, with very few previous work. In summary, this thesis makes the following main contributions.

1. Modeling Document Relationship:

   (a) Modeling Neighborhood Structure: We propose incorporating neighborhood structure in semantic visualization. In this respect, we propose a probabilistic model SEMAFORE, with two integrated components. One is a kernelized semantic visualization model, enabling the substitution of the kernel functions that relate visualization coordinates to topic distributions (see Section 3.2.2). The other is a neighborhood graph regularization framework for semantic visualization as described in Section 3.3.1.

   This work was published in *AAAI Conference on Artificial Intelligence* (AAAI) in 2014 and earned a honorable mention for outstanding paper award [72]. The extension of this work was published in the award track

of *Journal of Artificial Intelligence Research (JAIR)* in 2016 [75].

(b) Modeling Network Structure: We propose a semantic visualization model called PLANE for embedding a document network. Our novelty arises from the holistic approach to topic-based embedding of document networks. In comparison, previous works, reviewed in Section 4.1, have attempted this as separate segments, namely: embedding of documents, embedding of networks, or topic modeling, but have not recognized the embedding a document network as a distinct problem to be addressed in its own entirety. This work was published in *IEEE International Conference on Data Mining* (ICDM) in 2014 [73].

2. Modeling Document Representation:

   (a) Modeling Spherical Representation: We propose a generative model called SSE, which stands for *S*pherical *S*emantic *E*mbedding to embed documents with spherical representation. To the best of our knowledge, we are the first to address semantic visualization for spherical representation. This work was published in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD) in 2014 [74].

   (b) Modeling Bag of Word Vectors: As far as we are aware, we are the first to propose semantic visualization for short texts. We design a novel semantic visualization model that leverages word vectors. Our model, called *Gaussian Semantic Visualization* or GaussianSV, assumes that each topic is characterized by a Gaussian distribution on the word embedding space. This work has been accepted for publication in *The 26th International Joint Conference on Artificial Intelligence* (IJCAI) in 2017 [77].

3. Application of Semantic Visualization:

   (a) Document Collection Visualization: We build a semantic visualization

system, called SemVis for interactive topical analysis. This is a demonstrable system that is built on, and is generically compatible with PLSV [62], SEMAFORE (Chapter 3), and SSE (Chapter 5). SemVis can be used to navigate a text corpus interactively and topically via browsing and searching.

(b) Single Document Visualization: We propose WORD FLOCK which is the first to integrate *two levels* of "synchronization" principles for word clouds: *similar documents* share similar word clouds, and *related words* of the same latent aspects are displayed similarly. WORD FLOCK is novel in employing latent variable analysis through *joint* usage of *embedding* (synchronized positioning) and *latent aspect modeling* (coloring) among words of similar concepts. This work was published in *Proceedings of the 25th International Joint Conference on Artificial Intelligence* (IJCAI) in 2016 [76].

Chapter 2

Related Work

This chapter covers existing research related to document visualization problem. The content is organized around the taxonomy shown in Figure 2.1. We divide document visualization methods into two categories. The first category pays attention to visualization of a document collection, which aims to display relationship patterns among documents. Meanwhile, the second category focuses on visualization of a single document in order to give an overview of the document content.

For document collection visualization, there is recently greater attention to approaches using topic modeling. Topic modeling is useful to discover abstract topics in a collection of documents which can be used to enrich the visualization of documents with semantic information. From dimensionality reduction view, classic visualization methods that are not based on topic modeling usually reduce the dimension directly from data space to visualization space. Meanwhile, methods based on topic modeling introduce an intermediate topic space to reveal semantic relationship among documents and then preserve it in the visualization space. Next sections give an overview of these visualization methods.

For single document visualization, it can be divided further into two main categories which are independent visualization and coordinated visualization. Independent visualization methods generate independent visualizations for documents (i.e., each document will be visualized independently of others). Coordinated vi-

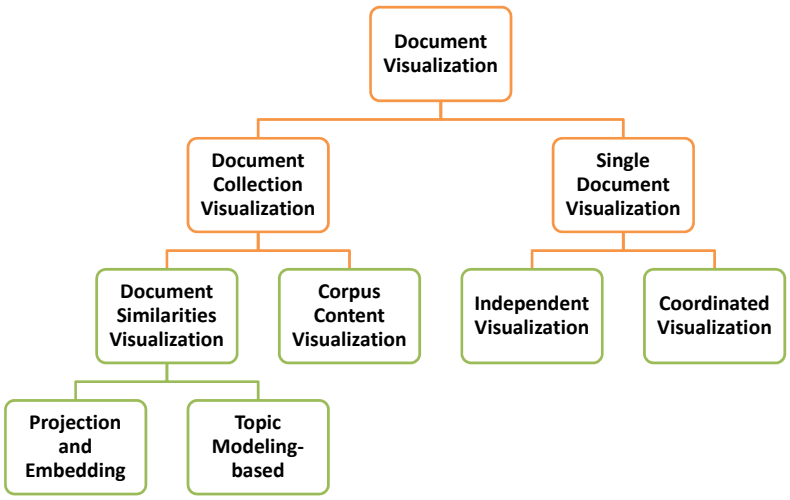


Figure 2.1: Taxonomy for Document Visualization Methods

sualization methods, on the other hand, aim to coordinate visual attributes across the visualizations of different documents. Through coordination, two similar documents will have similar visualizations, which is useful for visual comparisons of documents.

2.1 Document Collection Visualization

In this section, we will give a review of two main research directions in document collection visualization. Research in the first direction focuses on visualizing the document similarities. Meanwhile, the other research direction has a different objective which is to visualize the content of the corpus from different aspects and at different detail levels.

2.1.1 Document Similarities Visualization

The visualization methods reviewed in this section are closely related to us. They focus on visualizing the document similarities by displaying documents as points in 2D/3D visualization space (see Figure 2.2 for examples). The distances among them encode the similarities. There are two main approaches to this problem. The first is to rely on dimensionality reduction methods to project or embed the documents into

the visualization space. The second is to use topic modeling as an intermediate step to learn the representation of documents in the topic space and then embed them into the visualization space. We review these two approaches in the next sections.

Projection and Embedding Approaches

One way to perform visualization is by using a *generic* dimensionality reduction technique. Such techniques come in several flavors, depending on the objective. Principal component analysis (PCA) [65] identifies the components that explain most of the variance in the data. Related to PCA is singular value decomposition (SVD) [48]. Comparatively, independent component analysis (ICA) [32] identifies the components that are independent of one another, whereas linear discriminant analysis (Fisher's LDA) [44] identifies the components that most discriminate between known class labels. Being generic, these techniques are more frequently applied to feature extraction, as they are not optimized for visualization. They focus more on the properties of the components (e.g., orthogonality, independence) rather than on the intrinsic relationship among data instances. Furthermore, as they are based on linear projections, they may not capture non-linearities in the data well.

Another category of techniques, which is more directly related to visualization, is the *embedding* approach. It aims to preserve the high-dimensional similarities or differences in the low-dimensional embedding. One pioneering such work is multidimensional scaling (MDS) [70]. Given a set of pairwise distances δ_{ij} between data points i and j, MDS determines coordinates x_i and x_j respectively, such that the embedded visualization distance $||x_i - x_j||$ approximates δ_{ij} as much as possible. For MDS, the distance to be preserved δ_{ij} is frequently the linear distance, measuring the distance along a straight line between two points in the input space. Instead of this linear distance, Isomap [115] seeks to preserve the geodesic distance, by finding shortest paths in a graph with edges connecting neighboring data points. LLE [104] seeks to preserve linear distances, but only among the neighboring points and avoiding the need to estimate pairwise distances between widely separated data

points.

Recently there are also works applying a similar concept to embedding but using probabilistic modeling, such as GTM [12], PE [61], SNE [55] and t-SNE [117]. Yet others are based on semi-definite programming [107, 108]. Alternatively, several embedding techniques do not aim to preserve relationship among data instances, but rather other properties such as local minima [67]. Importantly, all these techniques are not optimized for *semantic* visualization, as they do not model topics at all. The coordinates do not reflect any semantic meaning, other than reflecting the optimization objective. We will give an overview of GTM, SNE, t-SNE and PE in next paragraphs.

GTM takes as input a set of data coordinates in a high dimensional space and assumes a parameterized function mapping from the embedding space to the original data space. GTM adds a Gaussian noise model to the mapping and estimate the parameters of the mapping function using the EM algorithm. SNE and t-SNE, on the other hand, take as input the pairwise distances between data points and convert these distances into a set of conditional probabilities that represent similarities. In the embedding space, we can also compute a similar set of conditional probabilities based on the distances among embedded points. SNE, t-SNE aim to preserve the similarities among data points in the embedding space by minimizing the sum of Kullback-Leibler divergences between these two set of conditional probabilities.

Different to these methods, PE considers a different kind of embedding problem for a set of points (or objects) $X = x_1, \ldots, x_N$ together with a set of classes $C = c_1, \ldots, c_K$. PE takes as input conditional probabilities $p(c_k|x_n)$ associating each object x_n with each class c_k and seeks to embed both objects and classes in a low-dimensional space such that the distance between object x_n and class c_k is consistent with the probability $p(c_k|x_n)$. We define $Y = \{y_n\}_{n=1}^N$ as embedding coordinates of objects and $\phi = \{\phi_k\}_{k=1}^K$ as coordinates of classes. Based on embedding coordinates, PE computes the conditional probabilities $p(c_k|y_n)$ for each object as in Equation 2.1. The objective of PE is then minimizing the sum of Kullback-

Leibler divergences for each object: $\sum_{n=1}^{N} KL(p(c_k|x_n)||p(c_k|y_n))$.

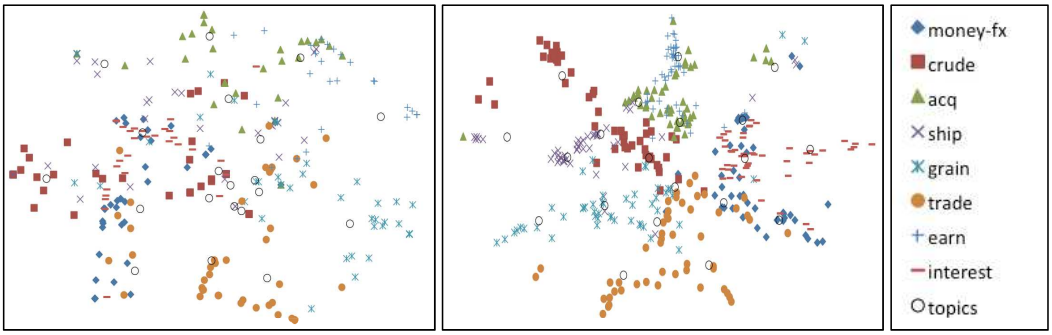
$$p(c_k|y_n) = \frac{p(c_k)\exp(-\frac{1}{2} \parallel y_n - \phi_k \parallel^2)}{\sum_{l=1}^{K} p(c_l)\exp(-\frac{1}{2} \parallel y_n - \phi_l \parallel^2)} \tag{2.1}$$

Topic Modeling-based Approaches

Topic model involves statistical modeling of text (documents and words) in order to discover some abstract concepts or "topics" that occur in a corpus. Beginning with latent semantic indexing [41], topic model evolves into the modern probabilistic treatments, such as Probabilistic Latent Semantic Analysis (PLSA) [57] and Latent Dirichlet Allocation (LDA) [14]. Intuitively, a topic captures a collection of words that tend to co-occur because they describe the same concept. This has the appeal of producing highly interpretable statistical models that let users make semantic sense of the corpus.

Topic model's ability to model semantics in documents makes it a new way to explore and understand text document collections. Meanwhile, a visualization with a rich user interface supports users to conduct discovery and exploratory tasks effectively. Therefore, it is interesting to combine topic model and visualization to take advantage of their strengths. There have been many research works in this direction and they are diverse in terms of which visualization aspects they are focusing on. Some of them focus on effective visualization forms for visualization and building large systems with rich user interface for visual exploration using topic model [40, 50, 123]. A few of them focus on visualizing a topic model itself (i.e. visualizing which topics are important in a corpus, or which words are important in a topic) [23, 29] and may provide ways to interact with the underlying topic model for the tuning purposes [28, 59].

Another line of work in this direction, which is closely related to us, focus on the dimensionality reduction aspect of visualization as well as statistical modeling. Here, we focus on methods which visualize documents in a Euclidean space where each document is represented as a point in a 2D/3D scatterplot. There are two approaches as follows.



a. PE (LDA) b. PLSV

Figure 2.2: Scatterplot Visualizations by PE (LDA) and PLSV for $Reuters8$ dataset

Pipeline Approach. We first conduct topic modeling using a topic model such as Latent Dirichlet Allocation (LDA) [14] and then visualize documents' topic distributions using a visualization method such as PE [61] or a Self-Organizing Map (SOM) [69] as in LDA-SOM [89]. These methods include two separate steps that optimize different objective functions. Therefore, errors from previous step may be accumulated and cannot be fixed in the current step. LDA-SOM separately embeds the documents' topic distributions learned from LDA on a self-organizing map which is a different visualization space than the Euclidean space that we are interested in. Below we give an overview for PE which is in short for Parametric Embedding. In Figure 2.2a, we show an example visualization by PE (LDA) for $Reuters8$ dataset.

PE is a method to embed class conditional probabilities in a Euclidean space. When applying with LDA in a pipeline approach, PE takes topic proportions Λ as input and for each topic z and document n, it finds their coordinates ϕ_z and x_n in the visualization space. To find coordinates, it minimize the following sum of Kullback-Leibler divergences which aim to preserve the input probabilities:

$$\sum_{n=1}^{N}\sum_{z=1}^{Z} \mathrm{P}(z|x_n,\Lambda)\log\frac{\mathrm{P}(z|x_n,\Lambda)}{\mathrm{P}(z|x_n,\Phi)} \tag{2.2}$$

where $\mathrm{P}(z|x_n,\Phi)$ is defined as:

$$\theta_{n,z} = \mathrm{P}(z|x_n,\Phi) = \frac{\exp(-\frac{1}{2}||x_n-\phi_z||^2)}{\sum_{z'=1}^{Z}\exp(-\frac{1}{2}||x_n-\phi_{z'}||^2)} \tag{2.3}$$

Joint Approach. In this approach, we jointly model topics and visualization using

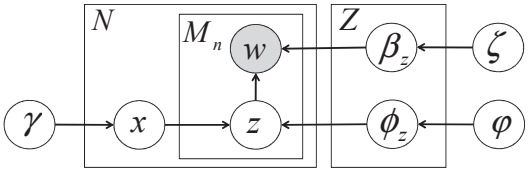


Figure 2.3: Graphical Model of PLSV

a unique objective function. There are only a few related works so far that seek to address the semantic visualization task directly. The closest previous work that does both topic modeling and visualization in a single generative process is Probabilistic Latent Semantic Visualization (PLSV) [62], which also shows that a joint approach outperforms a separate approach. We briefly review PLSV, whose graphical model is shown in Figure 2.3. The generative process of PLSV is as follows. For each topic z, we draw its word distribution β_z from a Dirichlet with parameter ζ, as well as its coordinate ϕ_z from Normal distribution with mean 0 and variance φ^{-1}. In turn, for each document d_n, we draw its coordinate from Normal with mean 0 and variance γ^{-1}. To generate each of the M_n words in d_n, we draw a topic z based on Equation 2.3, and then draw a word from the selected topic's word distribution β_z. Figure 2.2b shows a visualization by PLSV for $Reuters8$ dataset.

Similar to PLSV, we build our model upon the foundation of the topic modeling technique Probabilistic Latent Semantic Analysis (PLSA) [57] by incorporating visualization coordinates. The difference is that we also seek to model document relationships and document representations together with semantic visualization. Document relationships can be represented by neighborhood structure derived by distances among documents or real networks of documents such as citation networks. For document representations, we investigate spherical representation and bag of word vectors which are different from tradition word count representation in terms of their expressiveness.

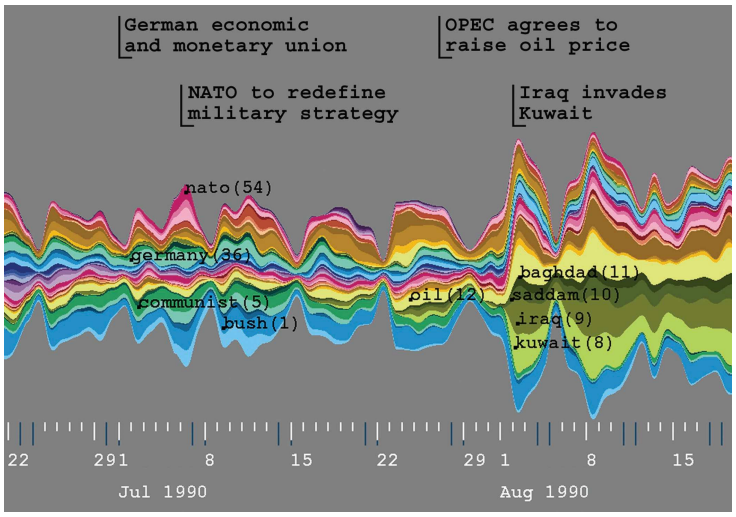


Figure 2.4: A visualization by ThemeRiver for Associated Press news wire stores from July and early August 1990. (The figure is taken from [52])

2.1.2 Corpus Content Visualization

Methods for visualizing corpus content have a common objective that is to give an overview of the corpus by revealing its topics or themes. Some methods also provide ways to track how topics are changing and evolving over time, which is useful for detecting hot topics and discussion trends. Research in this direction often focuses on designing appropriate visualization forms depending on which content information we want to show. For example, for visualizing term distribution information, TileBars [53] displays each document as a rectangle bar. Depending on the query terms, the rectangle bars for retrieved documents will show the relative frequency of the query terms, and how the terms are distributed in a document or across all documents. For visualizing how topics are changing in the corpus, ThemeRiver [52] makes use of river visualization form as shown in Figure 2.4. Each layer represents a theme and the vertical width indicate the strength of the theme at any point in time.

Recently, a large body of methods relying on topic modeling for building complex topic analysis systems. They focus on building effective visualization forms and providing rich user interface for visual exploration using topic model [36, 40, 50, 123]. TextFlow [36] uses a river-like visualization form for detecting the topic

Figure 2.5: A graph of topics by Tiara for over 10,000 research articles crawled from faculty pages at CalIT2. (The figure is taken from [50])

evolution trend, the critical event, and the keyword correlation. ParallelTopics [40] uses a parallel coordinate visualization form to present the topic distributions of documents. TopicNets [50] presents topics in a corpus as a graph. In this approach, documents and topics are treated as connected nodes of different types in an interactive graph. It uses the topic similarity to determine node positions and create visual clusters of documents that are similar topically. Figure 2.5 showcases a visualization by TopicNets for over 10,000 research articles crawled from faculty pages at CalIT2. Tiara [123] provides a visual summary of documents' topics over time. It uses different layers to represent topics as shown in Figure 2.6. Each layer represents a topic described by a set of top words and the height of the layer indicates the strength of that topic.

There are a few methods focus on visualizing a topic model itself (i.e. visualizing which topics are important in a corpus, or which words are important in a topic) [23, 29] and may provide ways to interact with the underlying topic model for the tuning purposes [28, 59].

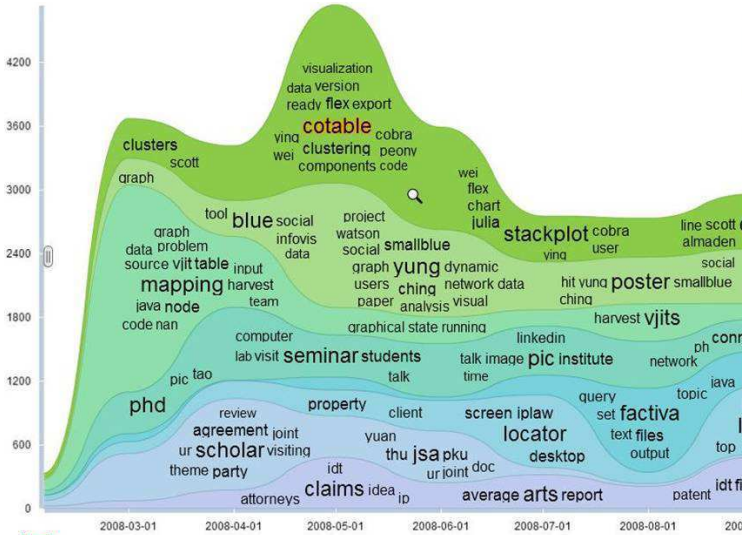


Figure 2.6: A visualization by Tiara for a dataset of 8000 emails. (The figure is taken from [123])

2.2 Single Document Visualization

The objective of single document visualization is to visualize content of the document using some forms of visualization (e.g., word clouds, graph, tree, ...) for overview. Most of the methods generate the visualization for each document independently and there is no consistent view across the documents. This makes the comparison between multiple documents based on their visualizations inefficient. To overcome this problem, some methods propose ways for coordinating the visual attributes across the visualizations. The purpose of coordinating is to make a more consistent view across the documents for easier visual comparison. We will review these methods in the next sections.

2.2.1 Independent Visualization of Document Content

Independent visualization methods generate visualization for every single document independently. They focus on giving an overview of the document content and do not generate a consistent view for visualizations across documents. Therefore, it is difficult for visual comparison to discover patterns among documents. Some popular methods falling into this category are Tag Clouds [120], Word Clouds (such

Figure 2.7: An example word cloud generated by Wordle

as those generated by Wordle [121]), TextArc [96], WordTree [122], and Phrase Nets [119].

Tag Clouds, Word Clouds, and TextArc display prominent words in the document based on the word frequency. A word with high frequency is likely an important word. Therefore displaying these important words can give a first glance of what document is about. In Wordle [121], a document is visualized as a cloud of prominent or important words and each word is displayed according to a number of visual attributes. Figure 2.7 shows an example word cloud generated by Wordle. There are various visual variables within a word cloud such as font size, color, position, and orientation. To express different important weight of words, some visual attributes such as font size and colors are used. For example, more important words will have bigger font size and may be assigned more contrast colors. For the word positions, Wordle uses spiral algorithm (Algorithm 1) which works in a greedy and incremental manner to layout the words. Basically, the algorithm has two main steps. For each word in a document, first we will initialize the word with a position (line 3) and then move the word along a spiral path until there is no intersection with previous displayed words (line 4 & 5). We refer the reader to [110] for a thorough discussion on how to initialize the word positions as well as how to check the intersection efficiently. For the word orientation, it is usually random and sometimes it is used to resolve the word intersection in the layout algorithm.

Different to previous methods, WordTree and Phrase Nets, while visualizing

Algorithm 1 Spiral Algorithm

Require: M_n words that are to be displayed in cloud C_n for each document d_n, and optionally initial positions $\mathbf{p}_n$ of M_n words.

Ensure: For each document d_n, positions $\mathbf{p}_n$ of M_n words in the word cloud of d_n.

```
for all document d_n, n ∈ {1, ..., N} do
    for all words w ∈ {w_1, ..., w_{M_n}} do
        Initialize p_w (e.g. sample from Gaussian) if initial position unsupplied
        while p_w intersects any previous words do
            Move p_w one step along a spiral path
        end while
    end for
end for
```

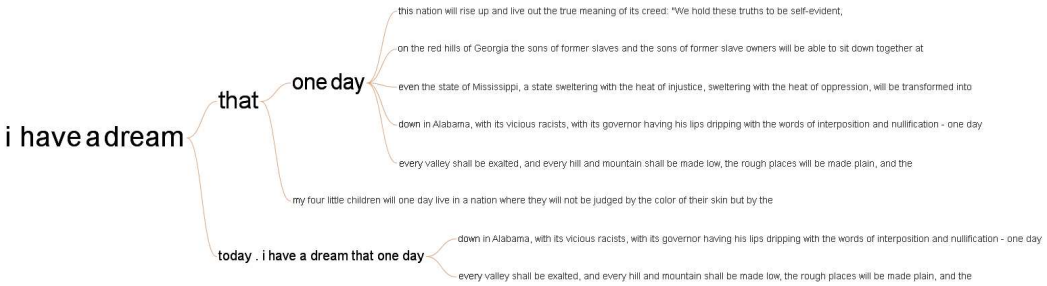


Figure 2.8: An example of word tree rooted at the phrase "i have a dream" generated by WordTree for Martin Luther Kings historical speech. (The figure is taken from [122])

word frequency, they also take into account the word context. To display the context, WordTree use a tree rooted at a word or a phrase to show all sentences that contains that word or phrase. As an example, Figure 2.8 is the word tree rooted at the phrase "i have a dream" drawn by WordTree for Martin Luther Kings historical speech. Phrase Nets, on the other hand, use a graph whose nodes are words and whose edges indicate that there is a context relationship between two words. The relationship is characterized by a "phrase" pattern such as "Word_A and Word_B" or "Word_A at Word_B" which can be found using either simple pattern matching or syntactic analysis. Figure 2.9 shows phrase nets generated by the method when the user selects the pattern "...A and B..." for James Joyces Portrait of the Artist as a Young Man.

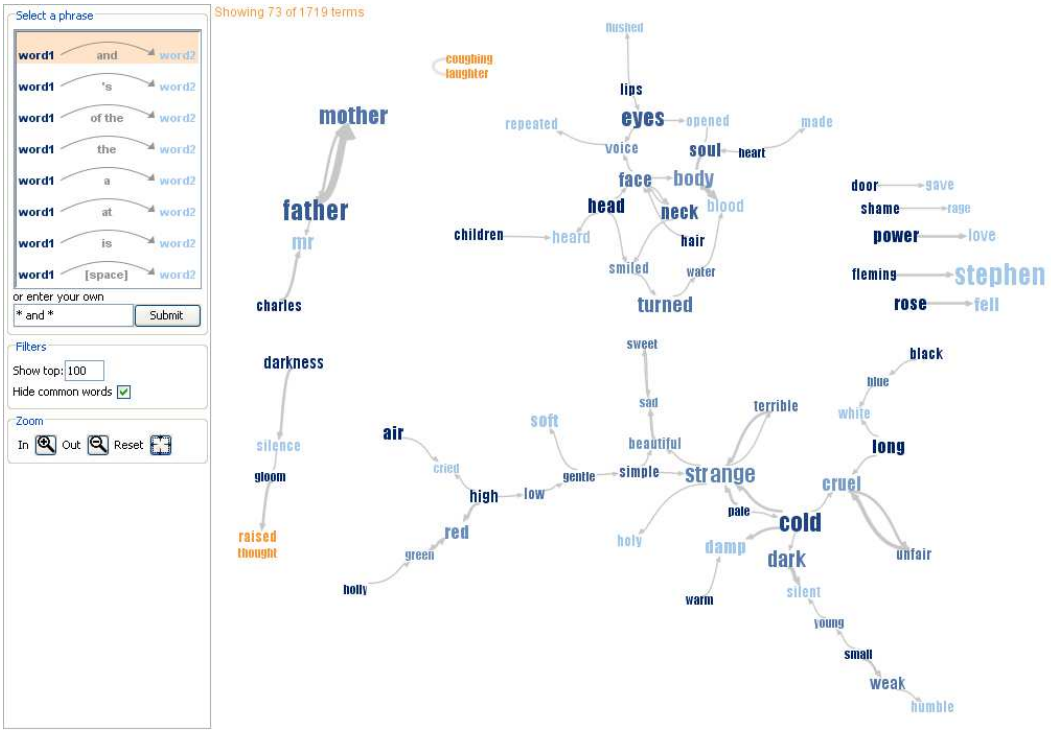
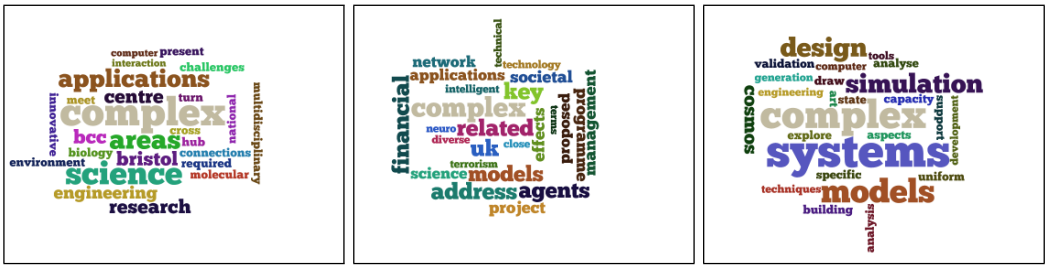


Figure 2.9: An example of phrase nets generated when the user selects the pattern "...A and B..." for James Joyces Portrait of the Artist as a Young Man. (The figure is taken from [119])

2.2.2 Coordinated Visualization for Visual Comparison of Documents

Coordinated visualization methods, while visualizing the document content, also aim to coordinate visual attributes across documents to make sure that similar documents will have similar visualizations. Therefore, based on the visualizations, we can perform visual comparison of documents efficiently. Some of these methods are Word Storms [22], Docuburst [31] and Gist Icons [38]. Each of these focuses on different types of visualization forms.

Word Storms [22] advances word clouds for visual comparison of documents by introducing the coordination of similarity principle. Word clouds are coordinated so that similar documents will have similar clouds, and dissimilar documents will have visually different clouds. For example, words that appear in multiple clouds will have similar font, similar colors and approximately the same position across the clouds. Based on this, by quickly scanning the clouds, the viewer can visually



(a) (b) (c)

Figure 2.10: A word storm contains three grants from Complexity Programme in EPSRC Scientific Programs. Each cloud represents a grant abstract. (The figure is taken from [22])

check which clouds do not contain shared words and assess the document similarity. Figure 2.10 shows a word storm containing three grants from the Complexity Programme in EPSRC Scientific Programs. There is a similarity between these three clouds because they represent for grant abstracts from the same programme.

To coordinate the layout of different clouds, Word Storms uses an iterative algorithm as shown in Algorithm 2. Here a word cloud C_n for document d_n is a tuple $C_n = (W_n, \{p_{nw}\},$

$\{l_{nw}\}, \{s_{nw}\})$, where W_n is the set of words that are to be displayed in cloud C_n, and for any word $w \in W_n$, define $p_{nw} = (x_{nw}, y_{nw})$ as the position of w in the cloud C_n, l_{nw} as the color, and s_{nw} as the font size. $p_n = \{p_{nw} | w \in W_n\}$ is the set of all word positions in C_n and V_w is the set of clouds that contain word w. The idea of the iterative layout algorithm is that first we call the spiral algorithm (Algorithm 1) with the word positions are set randomly. Subsequently, for each shared word, derive a new desired position by averaging the previous positions of the word in the different clouds, i.e., the new desired position for word w is $p'_w \leftarrow \frac{1}{|V_w|} \sum_{v_j \in V_w} p_{jw}$. This average ensures that word w has the same position across all clouds $v_j \in V_w$. After having new desired positions, the spiral algorithm is called again to resolve any intersection among words. This process is repeated until we reach a terminate condition.

Other approaches for document comparison deviate from the traditional word cloud format, e.g., graph visualization [27], or topographic map [46]. Some show

Algorithm 2 Word Storms: Iterative Layout Algorithm

Require: Storm $v_n = (W_n, \{l_{nw}\}, \{s_{iw}\})$ without positions
Ensure: Word storm $\{v_1, \ldots, v_N\}$ with positions.

for all document d_n, $n \in \{1, \ldots, N\}$ **do**
  $p_n \leftarrow SpiralAlgorithm(W_n)$
end for
while Not Converged && count $<$ Max Iteration **do**
  for all document d_n, $n \in \{1, \ldots, N\}$ **do**
    $p'_{nw} \leftarrow \frac{1}{|V_w|} \sum_{v_j \in V_w} p_{jw}, \forall w \in W_n$
    $p_n \leftarrow SpiralAlgorithm(W_n, p_n)$
  end for
  count = count + 1
end while

comparisons in a modified format, e.g., intersecting or common words [33, 81], different topics [95] or corpora [98, 103].

In the work presented in Chapter 8, we build a method called Word Flock for visual comparison of documents using word clouds. Different to Word Storms, our work models latent aspects of words and increases the coherence of word clouds by displaying related words with similar colors and positions. Word similarity was previously considered only in the context of an individual word cloud [11, 51, 68], and based on similarity measures such as cosine [5, 35, 124]. In contrast, we model the coherence of *synchronized* (rather than individual) word clouds for comparison of documents. Moreover, we employ latent variable analysis to learn the probability distribution over k latent aspects (rather than similarity).

Docuburst [31] shows the word lexical hyponymy relationship (i.e., IS-A relation) which are derived from a lexical database such as WordNet [43]. The word lexical relationships are displayed by a radial tree layout. Given a document, Docuburst draws a radial tree rooted at a word of interest. For example, Figure 2.11 shows a visualization by Docuburst for a science textbook rooted at "thought". In this radial tree, a node is a synset. The angular width of a node is proportional to the occurrence count of the document words in the subtree rooted at that node. By using radial tree layout and sharing a lexical database, Docuburst can generate visualizations which have a consistent view across documents for document comparison.

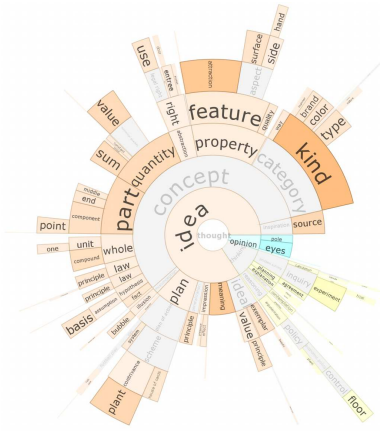


Figure 2.11: DocuBurst of a science textbook rooted at “thought”. Node hue distinguishes the WordNet synsets containing “thought”. (The figure is taken from [31])

Gist Icons [38] displays the histograms of words inside a document in a circular pattern. The words are ordered to form a list of loosely defined concepts. The shape of the histograms will show concepts described in the document and we can compare documents by comparing these shapes. DocuBurst and Gist Icons both provide word clustering into higher concepts (synsets). However, Gist Icons are only one level deep while DocuBurst can have as many levels as the depth of the synset tree.

Part I

Modeling Document Relationship

Chapter 3

Modeling Neighborhood Structure

In this chapter, we propose a semantic visualization model that incorporates neighborhood structure of documents. Different to previous approaches, our model aims at generating a visualization that preserves both the semantics learned from topic modeling and the local neighborhood structure on the document manifold. We achieve this by jointly modeling topics and visualization on the intrinsic document manifold, modeled using a neighborhood graph. Each document has both a topic distribution and visualization coordinate. Specifically, we propose an unsupervised probabilistic model, called SEMAFORE, which aims to preserve the manifold in the lower-dimensional spaces through a neighborhood regularization framework designed for the semantic visualization task. To validate the efficacy of SEMAFORE, our comprehensive experiments on a number of real-life text datasets of news articles and Web pages show that the proposed methods outperform the state-of-the-art baselines on objective evaluation metrics.

3.1 Introduction

We focus on the task of semantic visualization which is formulated in Section 1.1. The joint approach was attempted by PLSV [62], which derives the latent parameters by maximizing the likelihood of observing the documents. This objective is known as *global consistency*, which is concerned with the "error" between the

model and the observation.

Crucially, PLSV has not cared to meet the *local consistency* objective [128], which is concerned with preserving the observed proximity or distances between documents. This shortcoming is related to PLSVs assumption that the document space is Euclidean (a geometrically flat space), by sampling documents coordinates independently in a Euclidean space. The local consistency objective arises naturally from the assumption that the intrinsic geometry of the data is a low-rank, non-linear manifold within the high-dimensional space. This manifold assumption is well-accepted in the machine learning community [71], and finds application in both supervised and unsupervised learning [9, 128, 129]. Recently, there is a preponderance of evidence that manifold assumption also applies to text data in particular [17, 18, 60]. We therefore propose to incorporate this manifold assumption into a new unsupervised, semantic visualization model by using a neighborhood regularization framework.

Regularization as a technique to realize this assumption has a long history [10]. The specific form of the regularization function varies among applications. The study of this assumption for unsupervised topic models begins with LapPLSI [17], which introduces regularization to PLSA [57], by minimizing the Euclidean distance between neighboring documents' topic distributions. Follow-up work introduce other distance functions [18, 125]. While these previous work focus on maintaining proximity of similar documents, DTM [60] adds a new criterion to also maintain the distance among different documents. Our work is different in that we also need to contend with the visualization aspects, and not just topic modeling.

3.1.1 Overview

We propose an unsupervised probabilistic model that jointly derives topic distributions and visualization coordinates on the intrinsic geometry of the data. Our proposed model is called SEMAFORE, which stands for *SE*mantic visualization with *MA*ni*FO*ld *RE*gularization. We build a neighborhood regularization framework into

a semantic visualization model. The framework involves new issues to resolve, including the regularization function, and the space in which regularization should take place.

The model is evaluated on a series of real-life, publicly available datasets, which are also benchmark datasets used in document classification task. An advantage of a statistical method, such as ours, is that it is not dependent on a specific language. Two of the datasets are in English, and one is in Brazilian Portuguese. While our model is unsupervised (class label is neither required nor used in learning), to objectively quantify the visualization quality, we leverage on the class label information. It is a common assumption that documents of the same class are expected to be neighbors on the original space [10, 128, 129], which suggests that they should also be close on the visualization space. We investigate the effectiveness of SEMAFORE in placing documents of the same class nearby on the visualization space, and systematically compare it to existing baselines without one or more of our properties, namely: joint modeling of topic and visualization, or neighborhood regularization.

3.1.2 Contributions

While visualization and topic modeling are, separately, well-studied problems, the interface between the two, semantic visualization, is a relatively new problem, with very few previous work. In this work, we make the following contributions.

- We propose incorporating neighborhood structure in semantic visualization. In this respect, we propose a probabilistic model SEMAFORE, with two integrated components. One is a kernelized semantic visualization model, enabling the substitution of the kernel functions that relate visualization coordinates to topic distributions (see Section 3.2.2). The other is a neighborhood graph regularization framework for semantic visualization as described in Section 3.3.1.
- Realizing the neighborhood graph regularization involves an exploration of how to incorporate the appropriate forms of the neighborhood structure. In

this respect, we investigate the effects of neighborhood graph construction techniques such as k-nearest neighbors (k-NN), ϵ-ball, and disjoint minimum spanning trees (DMST), as well as different edge weight estimations such as heat-kernel (see Section 3.3.2) in the context of semantic visualization.

- In Section 3.4, we describe the requisite learning algorithms based on maximum a posteriori (MAP) estimation using expectation-maximization (EM), in order to fit the parameters for the various regularization functions and kernels that we propose.
- Our final contribution is the evaluation of SEMAFORE's effectiveness on a series of real-life, public datasets described in Section 3.5, which shows that SEMAFORE outperforms existing baselines on a well-established and objective visualization metric.

3.2 Semantic Visualization

Our focus in this paper is on the effects of the neighborhood graph structure on the semantic visualization task. We figure that the clearest way to showcase these effects is to design a neighborhood preservation framework over and above an existing generative process, such as PLSV [62], which we will review in Section 3.2.1. In Section 3.2.2, we describe an innovation over the semantic visualization model, which is an abstraction of the mapping between the topic space and the visualization space using radial basis function (RBF) kernels. This allows the exploration of various kernels, of which we identify two for further exploration. In the following Section 3.3, we discuss how to incorporate neighborhood structure into semantic visualization.

3.2.1 Generative Process

We now describe the generative process of documents based on both topics and visualization coordinates. Below we review PLSV whose graphical model is shown in

Figure 2.3. Our eventual complete model is a generalization of this model, involving enhancements through kernelization (Section 3.2.2) and neighborhood structure preservation (Section 3.3).

The generative process is as follows:

1. For each topic $z = 1, \ldots, Z$:
   (a) Draw z's word distribution: $\beta_z \sim \text{Dirichlet}(\zeta)$
   (b) Draw z's coordinate: $\phi_z \sim \text{Normal}(0, \varphi^{-1} I)$
2. For each document d_n, where $n = 1, \ldots, N$:
   (a) Draw d_n's coordinate: $x_n \sim \text{Normal}(0, \gamma^{-1} I)$
   (b) For each word $w_{nm} \in d_n$:
      i. Draw a topic: $z \sim \text{Multi}(\{\text{P}(z|x_n, \Phi)\}_{z=1}^{Z})$
      ii. Draw a word: $w_{nm} \sim \text{Multi}(\beta_z)$

Here, ζ is a Dirichlet prior, I is the identity matrix, φ and γ control the variance of the Normal distributions. The parameters $\chi = \{x_n\}_{n=1}^{N}$, $\Phi = \{\phi_z\}_{z=1}^{Z}$, $\boldsymbol{\beta} = \{\beta_z\}_{z=1}^{Z}$, collectively denoted as $\Psi = \langle \chi, \Phi, \boldsymbol{\beta} \rangle$, are learned from documents $\mathcal{D}$ based on maximum a posteriori estimation. The log likelihood function is shown in Equation 3.1.

$$\mathcal{L}(\Psi|\mathcal{D}) = \sum_{n=1}^{N} \sum_{m=1}^{M_n} \log \sum_{z=1}^{Z} \text{P}(z|x_n, \Phi) \text{P}(w_{nm}|\beta_z) \tag{3.1}$$

We reiterate that our focus here is on incorporating neighborhood graph structure into semantic visualization. By building a neighborhood graph regularization framework into an existing generative process, i.e., PLSV, we can clearly observe that any improvement over PLSV arises from the neighborhood graph regularization. In this sense, our work is in the tradition of introducing neighborhood graph regularization to probabilistic topic modeling [17, 18, 60], where the contributions relate to the neighborhood graph regularization, rather than the generative process. That said, there is one significant difference to PLSV, which is our flexibility in allowing various kernel functions, which we will discuss next.

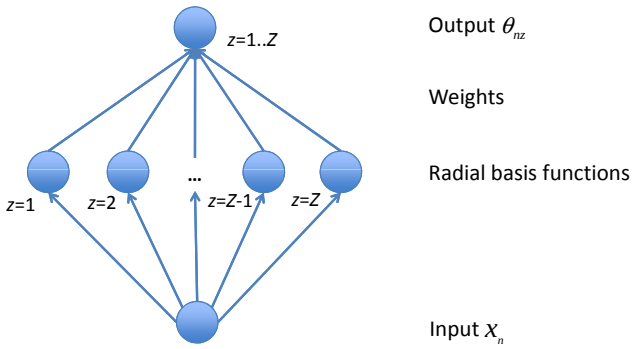


Figure 3.1: Topic distribution is expressed as a function of visualization coordinates using Radial Basis Function (RBF) network.

3.2.2 RBF Kernels

In the Step 2(b)i of the above generative process, the topic z of a word is drawn from the distribution $\{\mathrm{P}(z|x_n, \Phi)\}_{z=1}^{Z}$. This distribution relates the coordinates of topics in the visualization space $\Phi = \{\phi_z\}_{z=1}^{Z}$ and the coordinate x_n of a document d_n with the document's topic distribution $\{\mathrm{P}(z|d_n)\}_{z=1}^{Z}$.

This relationship can be formulated as a mapping problem where we want to find a function $\mathcal{G}$ which maps a point in visualization space to a point in the topic space. However, the form of $\mathcal{G}$ cannot be known exactly because both visualization space and topic space are latent spaces and $\mathcal{G}$ may be different across different domains. Therefore, to compute the topic distributions, we need a way to approximate $\mathcal{G}$.

To build a function approximation of the unknown function $\mathcal{G}$, we use the abstraction of Radial Basis Function (RBF) neural networks [13] because feedforward multilayered RBF neural networks with one hidden layer can serve as a universal approximator to arbitrary continuous functions [97]. This property provides the confidence that the model would have the ability to approximate any existing relationship between visualization space and topic space with arbitrary precision. Unlike PLSV [62] that defined a specific mapping function, our approach generalizes the semantic visualization model by defining the mapping problem in terms of kernelization, which admits several mapping functions within the family of RBF kernels.

In our context, Radial Basis Function [16] will relate coordinate variables based

on distances which defines a kernel function $\Lambda(||x_n - \phi_z||)$ in terms of how far a data point (e.g., x_n) is from a center (e.g., ϕ_z). The kernel function Λ may take on various forms, e.g., Gaussian, multi-quadric, inverse quadratic, polyharmonic spline. To express $\mathrm{P}(z|d_n)$ as a function of x_n, we consider the normalized architecture of RBF network, with three layers as shown in Figure 3.1. The input layer consists of one input node (x_n). The hidden layer consists of Z number of normalized RBF activation functions. Each is centered at ϕ_z and computes $\frac{\Lambda(||x_n - \phi_z||)}{\sum_{z'=1}^{Z} \Lambda(||x_n - \phi_{z'}||)}$. The linear output layer consists of Z output nodes. Each output node $y_z(x_n)$ corresponds to $\mathrm{P}(z|d_n)$, which is a linear combination of the RBF functions, as shown in Equation 3.2. Here, $w_{z,z'}$ is the weight of influence of the RBF function of z' on the $\mathrm{P}(z|d_n)$, with the constraint $\sum_{z'=1}^{Z} w_{z,z'} = 1$.

$$\mathrm{P}(z|d_n) = y_z(x_n) = \frac{\sum_{z'=1}^{Z} w_{z,z'} \cdot \Lambda(||x_n - \phi_{z'}||)}{\sum_{z'=1}^{Z} \Lambda(||x_n - \phi_{z'}||)} \tag{3.2}$$

While Equation 3.2 is the general form, to instantiate a specific mapping function, we need to determine both the assignment of $w_{z,z'}$ and the form of the function Λ. For $w_{z,z'}$, we will experiment with a special case $w_{z,z'} = 1$ when $z = z'$ and 0 otherwise.

For the kernel function Λ, one variation we consider is Gaussian, which yields the function in Equation 3.3, where Φ refers to the collective set of ϕ_z's. Note that here we set variance of Gaussian to 1. However, its true value is not really important because a different variance value just produces a re-scaled visualization with the scaling factor equal to that variance.

$$\mathrm{P}(z|d_n)_{Gaussian} = \mathrm{P}(z|x_n, \Phi)_{Gaussian} = \frac{\exp(-\frac{1}{2}||x_n - \phi_z||^2)}{\sum_{z'=1}^{Z} \exp(-\frac{1}{2}||x_n - \phi_{z'}||^2)} \tag{3.3}$$

Another variation of Λ being considered is Student-t. This distribution is also used by t-SNE [117] in the context of non-semantic, direct embedding to mitigate the effects of crowding. Due to mismatched dimensionalities, the points are crunched together in the center of the visualization, which prevents gaps from forming between the clusters. Therefore, we hypothesize that using Student-t as radial

basis function, which yields the function in Equation 3.4, can help to improve the performance of our model if crowding becomes an issue. Note that the Student-t distribution with one degree of freedom yields a radial basis function having the form similar to the inverse quadratic.

$$\mathrm{P}(z|d_n)_{Student-t} = \mathrm{P}(z|x_n, \Phi)_{Student-t} = \frac{(1 + ||x_n - \phi_z||^2)^{-1}}{\sum_{z'=1}^{Z}(1 + ||x_n - \phi_{z'}||^2)^{-1}} \tag{3.4}$$

The Gaussian function (Equation 3.3) was also used previously in the baseline PLSV [62] that we will compare to. Its inclusion helps to establish parity for comparative purposes, both to investigate the effectiveness of the alternative Student-t kernel (described above), as well as that of the neighborhood regularization (described in the next section).

3.3 Neighborhood Graph Regularization Framework

There are recent works [17, 18, 60] trying to preserve the local neighborhood structure when learning low-dimensional topic representations of documents. These works assume that documents are sampled from a nonlinear low-dimensional subspace that are embedded in a high-dimensional space. Therefore, the local neighborhood structure is important for revealing the hidden topics of documents and should be preserved when learning topic representations of documents [3]. In the generative process for semantic visualization described in Section 3.2, the document parameters are sampled independently, and may not necessarily reflect the underlying local neighborhood structure. We therefore seek to realize this assumption for semantic visualization. In particular, we assume that when two documents d_i and d_j are close in the original space, then their parameters ψ_i and ψ_j of the low-rank representation are similar as well. Coupled with the kernelized semantic visualization model described in Section 3.2, the neighborhood preservation approach described in this section constitutes our proposed model, SEMAFORE, which stands for *SE*mantic visualization with *MA*ni*FO*ld *RE*gularization.

3.3.1 Neighborhood Regularization

The neighborhood structure can be represented by a neighborhood graph. Given a set of data points in the Euclidean space, a neighborhood graph is constructed with the input data points as vertices. By definition, edges are symmetric, i.e., $\omega_{ij} = \omega_{ji}$, and weighted. The collection of edge weights are collectively denoted as $\Omega = \{\omega_{ij}\}$.

For the moment, we will assume that we have the neighborhood graph, and address the issue of how this neighborhood graph may be incorporated into our semantic visualization framework. In actuality, the neighborhood graph construction itself is an important component, whose construction is described in detail in Section 3.3.2.

One effective means to incorporate a neighborhood structure into a learning model is through a regularization framework [10]. This leads to a re-design of the log-likelihood function in Equation 3.1 into a new *regularized* function $\mathbf{L}$ (Equation 3.5), where Ψ consists of the parameters (visualization coordinates and topic distributions), and $\mathcal{D}$ and Ω are the documents and neighborhood structure.

$$\mathbf{L}(\Psi|\mathcal{D}, \Omega) = \mathcal{L}(\Psi|\mathcal{D}) + \lambda \cdot \mathcal{R}(\Psi|\Omega) \tag{3.5}$$

The first component $\mathcal{L}$ is the log-likelihood function in Equation 3.1, which reflects the fit between the latent parameters Ψ and the observation $\mathcal{D}$. The second component $\mathcal{R}$ is a regularization function, which reflects the consistency between the latent parameters Ψ of neighboring documents in the neighborhood structure Ω. λ is the regularization parameter, commonly found in neighborhood based algorithms [10, 17, 18], which controls the extent of regularization (we will experiment with different λ's in experiments).

Proposed Regularization Function

We now turn to the definition of the $\mathcal{R}$ function. The intuition is that the data points that are close in the high-dimensional space, should also be close in their low-rank representations, i.e., local consistency, also known as smoothness. One function

that satisfies this is $\mathcal{R}_+$ in Equation 3.6. Here, $\mathcal{F}$ is a distance function that operates on the low-rank space. Minimizing $\mathcal{R}_+$ leads to minimizing the distance $\mathcal{F}(\psi_i, \psi_j)$ between neighbors ($\omega_{ij} = 1$).

$$\mathcal{R}_+(\Psi|\Omega) = \sum_{i,j=1;i\neq j}^{N} \omega_{ij} \cdot \mathcal{F}(\psi_i, \psi_j) \tag{3.6}$$

The above level of local consistency is still insufficient, because it does not regulate how *non*-neighbors (i.e., $\omega_{ij} = 0$) behave. For instance, it does not prevent *non*-neighbors from having similar low-rank representations. Another valid objective in visualization is to keep *non*-neighbors apart, which is satisfied by another objective function $\mathcal{R}_-$ in Equation 3.7. $\mathcal{R}_-$ is minimized when two *non*-neighbors d_i and d_j (i.e., $\omega_{ij} = 0$) are distant in their low-rank representations. The addition of 1 to $\mathcal{F}$ is to prevent division-by-zero error.

$$\mathcal{R}_-(\Psi|\Omega) = \sum_{i,j=1;i\neq j}^{N} \frac{1 - \omega_{ij}}{\mathcal{F}(\psi_i, \psi_j) + 1} \tag{3.7}$$

We hypothesize that neither objective is effective on its own. A more complete objective would capture the spirits of both keeping neighbors close, and keeping *non*-neighbors apart. Therefore, we put Equation 3.6 and Equation 3.7 together using summation and maximize the objective function as shown in Equation 3.8. Note that the coefficient $\frac{1}{2}$ in Equation 3.8 is for simplifying the formula of the derivative of $\mathcal{R}_*(\Psi|\Omega)$.

$$\mathcal{R}_*(\Psi|\Omega) = -\frac{1}{2}(\mathcal{R}_+(\Psi|\Omega) + \mathcal{R}_-(\Psi|\Omega)) \tag{3.8}$$

Summation preserves the absolute magnitude of the distance, and helps to improve the visualization task by keeping *non*-neighbors separated on a visualizable Euclidean space. Taking the product is unsuitable, because it constrains the *ratio* of distances between neighbors to distances between *non*-neighbors. This may result in the crowding effect, where many documents are clustered together, because the relative ratio may be maintained, but the absolute distances on the visualization space could be too small.

Other than the proposed regularization function above, it is also possible to con-

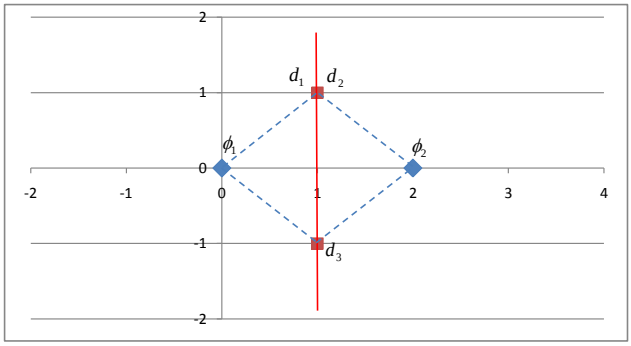


Figure 3.2: Example of how the same topic distribution may have different visualization coordinates. Any points on the red line have same topic distributions.

sider other regularization functions. For instance, we have also experimented with modifying the regularization function adapted from Discriminative Topic Model (DTM) [60], which addressed topic modeling but not semantic visualization. Note that while in the original DTM formulation, the distance function $\mathcal{F}(\psi_i, \psi_j)$ operates in the topic space, we adapt it for semantic visualization by redefining the distance function $\mathcal{F}(\psi_i, \psi_j)$ so that it will operate in the visualization space instead. This modified DTM formulation is shown to underperform the proposed regularization function above [72].

Enforcing Neighborhood Structure: Visualization vs. Topic Space

We now turn to the definition of $\mathcal{F}(\psi_1, \psi_2)$. In neighborhood-based models [10, 17, 18], there is only one low-rank representative space. For semantic visualization, there are two: topic and visualization spaces. We look into where and how to enforce the neighborhood graph structure.

At first glance, they seem equivalent. After all, they are representations of the same documents. However, this is not necessarily the case. Consider a simple example of two topics z_1 and z_2 with visualization coordinates $\phi_1 = (0, 0)$ and $\phi_2 = (2, 0)$ respectively. Meanwhile, there are three documents $\{d_1, d_2, d_3\}$ with coordinates $x_1 = (1, 1)$, $x_2 = (1, 1)$, and $x_3 = (1, -1)$. If two documents have the same coordinates, they will also have the same topic distributions. In this example, x_1 and x_2 are both equidistant from ϕ_1 and ϕ_2, and therefore according to

Equation 3.3, they have the same topic distribution $\mathrm{P}(z_1|d_1) = \mathrm{P}(z_1|d_2) = 0.5$, and $\mathrm{P}(z_2|d_1) = \mathrm{P}(z_2|d_2) = 0.5$. If two documents have the same topic distributions, they may not necessarily have the same coordinates. d_3 also has the same topic distribution as d_1 and d_2, but a different coordinate. In fact, any coordinate of the form $(1, ?)$ will have the same topic distribution. This example is illustrated in Figure 3.2.

This suggests that enforcing neighborhood structure on the topic space may not necessarily lead to having data points closer on the visualization space. We postulate that regularizing the visualization space is more effective. There are also advantages in computational efficiency to doing so, which we will describe further shortly. Therefore, we define $\mathcal{F}(\psi_i, \psi_j)$ as the squared Euclidean distance $||x_i - x_j||^2$ between the corresponding visualization coordinates.

3.3.2 Neighborhood Graph

We discuss how the neighborhood graph may be approximated, which concerns the two issues of how the graph edges are defined, as well as how they are weighted. The neighborhood graph is constructed in the original data space where we represent each document as a tf-idf vector [82]. We also experiment with different vector representations, including word counts and term frequencies, and find tf-idf to give the best results. The distance between two document vectors is measured using Euclidean distance.

Graph Construction

There have been research studies on the properties and methods for construction of neighborhood graphs [21, 127]. Since the construction of neighborhood graph is a critical step that may affect the performance of various graph-based algorithms, this problem itself is a research issue of independent interest. Our scope is in exploring how some well-established graph construction techniques may apply to the case of semantic visualization. We will investigate these various graph construction

methods empirically in Section 3.5.

In the following, we briefly review two categories of graph construction methods.

1. *Neighborhood-based Graphs.* In this formulation, edges are formed between data points that are deemed to be sufficiently close to each other. This admits different definitions of "sufficient closeness". The most common definitions found in the literature include the two below.

   (a) ϵ-ball: The neighborhood graph contains an edge connecting two documents d_i and d_j, if d_i and d_j have a distance less than a threshold ϵ.

   (b) k-nearest neighbors (k-NN) graph: The neighborhood graph contains an edge connecting two documents d_i and d_j, if d_i is in the set $\mathcal{N}_k(d_j)$ of the $k-$nearest neighbors of d_j, or d_j is in the set $\mathcal{N}_k(d_i)$.

   ϵ-ball and k-NN both have strongly data-dependent parameters (i.e., ϵ and k) and it is not straightforward to choose the best value for these parameters. Neither guarantees that the graph would be connected. They also need to be carefully selected or tuned, as to some extent they also affect the "balance" between the contribution of neighbors $\mathcal{R}_+$ and non-neighbors $\mathcal{R}_-$ to the neighborhood regularization $\mathcal{R}_*$ in Equation 3.8.

   ϵ-ball suffers from another issue that it tends to produce many edges for the points located at high-density regions, and thus has little restriction on the maximum degree of a vertex. k-NN does not suffer from that problem and is one of the most commonly used types of graphs.

   In our subsequent development and experiments, we will experiment with both ϵ-ball and k-NN graph as there may be some variance in the performance of different graph construction techniques for different datasets [30, 54, 116].

2. *Minimum Spanning Tree-based Graphs.* While ϵ-ball and k-NN are quite sensitive to noise and sparsity, graph construction based on combining multiple minimum spanning trees can help to reduce sensitivity to noise of the output

graph [127]. There are two variations based on this approach.

(a) Perturbed Minimum Spanning Trees (PMST): PMST builds a neighborhood graph by generating $T > 1$ perturbed copies of the whole dataset according to the local noise model and fitting an MST to each perturbed copy. A weight $e_{ij} \in [0, 1]$ will be assigned to the edge between points x_i and x_j equal to the average number of times that edge appears on the trees.

(b) Disjoint Minimum Spanning Trees (DMST): DMST produces a neighborhood graph by finding a deterministic collection of r minimum spanning trees that satisfies the property that no tree in the collection uses any edge of other trees. The neighborhood graph is the union of all edges of trees and contains $r(N-1)$ edges.

As the representative of this category, we use DMST, which is deterministic and easier to construct than PMST while showing similar efficacies.

Graph Weighting

The next issue is how to assign weights to the edges in the neighborhood graph. In this respect, we consider two variations of edge weights.

1. *Simple Minded*:

$$\omega_{ij} = \begin{cases} 1, & \text{if only if } d_i \text{ and } d_j \text{ are connected,} \\ 0, & \text{otherwise.} \end{cases} \tag{3.9}$$

This is the simplest approach where we use binary weighting to assign the weights to the edges. However, this approach to assign uniform weights to edges can be sensitive to errors, because of the "cliff effect" from 1 immediately to 0. Moreover, since the weights are not smoothed, it could result in some loss of information. We hypothesize that among the connected nodes, there may still be some differences in terms of degrees of similarity, which are expressed by their mutual distances. This motivates the second approach

below.

2. *Heat Kernel*:

$$\omega_{ij} = \begin{cases} \exp(-\frac{||d_i - d_j||^2}{\tau}), & \text{if only if } d_i \text{ and } d_j \text{ are connected,} \\ 0, & \text{otherwise.} \end{cases} \tag{3.10}$$

An alternative approach is using the Heat Kernel function [8] [63]. Heat Kernel has the advantage over Simple Minded by allowing smoother weights for the edges, which helps address the issues of sensitivity and loss of information. However, while Simple Minded is not parameterized, Heat Kernel has one parameter that needs to be determined (i.e., τ). Note that for $\tau = \infty$, Heat Kernel degenerates into Simple Minded, i.e., the former is the more general formulation. The exact value of τ is not important in our model because it would effectively be absorbed by the regularization parameter. For simplicity, we set $\tau = 2$.

3.4 Model Fitting

We now discuss how the parameters of the model described in Sections 3.2 and 3.3 can be learned. One well-accepted framework to learn model parameters using maximum a posteriori (MAP) estimation is the Expectation-Maximization or EM algorithm [39].

For our model, the regularized conditional expectation of the complete-data log likelihood in MAP estimation with priors is:

$$\begin{aligned} \mathcal{Q}(\Psi|\hat{\Psi}) = & \sum_{n=1}^{N}\sum_{m=1}^{M_n}\sum_{z=1}^{Z} \mathrm{P}(z|n,m,\hat{\Psi}) \log\left[\mathrm{P}(z|x_n,\Phi)\mathrm{P}(w_{nm}|\beta_z)\right] \\ & + \sum_{n=1}^{N} \log(\mathrm{P}(x_n)) + \sum_{z=1}^{Z} \log(\mathrm{P}(\phi_z)) + \sum_{z=1}^{Z} \log(\mathrm{P}(\beta_z)) \\ & + \lambda \cdot \mathcal{R}(\Psi|\Omega), \end{aligned}$$

where $\hat{\Psi}$ is the current estimate. $\mathrm{P}(z|n,m,\hat{\Psi})$ is the class posterior probability of the n^{th} document and the m^{th} word in the current estimate. $\mathrm{P}(\beta_z)$ is a sym-

metric Dirichlet prior with parameter ζ for word probability β_z. $\mathrm{P}(x_n)$ and $\mathrm{P}(\phi_z)$ are Gaussian priors with a zero mean and a spherical covariance for the document coordinates x_n and topic coordinates ϕ_z. We set the hyper-parameters to $\zeta = 0.01$, $\varphi = 0.1N$ and $\gamma = 0.1Z$ following PLSV [62].

In the E-step, $\mathrm{P}(z|n, m, \hat{\Psi})$ is updated as follows:

$$\mathrm{P}(z|n, m, \hat{\Psi}) = \frac{\mathrm{P}(z|\hat{x}_n, \hat{\Phi})\mathrm{P}(w_{nm}|\hat{\beta}_z)}{\sum_{z'=1}^{Z} \mathrm{P}(z'|\hat{x}_n, \hat{\Phi})\mathrm{P}(w_{nm}|\hat{\beta}_{z'})}.$$

In the M-step, by maximizing $\mathcal{Q}(\Psi|\hat{\Psi})$ w.r.t β_{zw}, the next estimate of word probability β_{zw} is as follows:

$$\beta_{zw} = \frac{\sum_{n=1}^{N}\sum_{m=1}^{M_n} \mathrm{I}(w_{nm} = w)\mathrm{P}(z|n, m, \hat{\Psi}) + \zeta}{\sum_{w'=1}^{W}\sum_{n=1}^{N}\sum_{m=1}^{M_n} \mathrm{I}(w_{nm} = w')\mathrm{P}(z|n, m, \hat{\Psi}) + \zeta W},$$

where $\mathrm{I}(.)$ is the indicator function. ϕ_z and x_n cannot be solved in a closed form, and are estimated by maximizing $\mathcal{Q}(\Psi|\hat{\Psi})$ using quasi-Newton [79].

The computation of the gradients of $\mathcal{Q}(\Psi|\hat{\Psi})$ w.r.t ϕ_z and x_n depend on the specific kernel used (see Section 3.2.2).

- For the Gaussian kernel, we have the following gradients:

$$\begin{aligned}
\frac{\partial \mathcal{Q}(\Psi|\hat{\Psi})}{\partial \phi_z} &= \sum_{n=1}^{N}\sum_{m=1}^{M_n} \big(\mathrm{P}(z|x_n, \Phi) - \mathrm{P}(z|n, m, \hat{\Psi})\big)(\phi_z - x_n) - \beta\phi_z, \\
\frac{\partial \mathcal{Q}(\Psi|\hat{\Psi})}{\partial x_n} &= \sum_{m=1}^{M_n}\sum_{z=1}^{Z} \big(\mathrm{P}(z|x_n, \Phi) - \mathrm{P}(z|n, m, \hat{\Psi})\big)(x_n - \phi_z) - \gamma x_n \\
&\quad + \lambda \cdot \frac{\partial \mathcal{R}(\Psi|\Omega)}{\partial x_n}.
\end{aligned}$$

- For the Student-t kernel, we have the following gradients:

$$\frac{\partial \mathcal{Q}(\Psi|\hat{\Psi})}{\partial \phi_z} = \sum_{n=1}^{N}\sum_{m=1}^{M_n} \frac{2\big(\mathrm{P}(z|x_n,\Phi) - \mathrm{P}(z|n,m,\hat{\Psi})\big)(\phi_z - x_n)}{1+||x_n-\phi_z||^2} - \beta\phi_z,$$

$$\frac{\partial \mathcal{Q}(\Psi|\hat{\Psi})}{\partial x_n} = \sum_{m=1}^{M_n}\sum_{z=1}^{Z} \frac{2\big(\mathrm{P}(z|x_n,\Phi) - \mathrm{P}(z|n,m,\hat{\Psi})\big)(x_n - \phi_z)}{1+||x_n-\phi_z||^2} - \gamma x_n + \lambda \cdot \frac{\partial \mathcal{R}(\Psi|\Omega)}{\partial x_n}.$$

The gradient of $\mathcal{R}(\Psi|\Omega)$ w.r.t. x_n is computed depending on the form of the regularization function $\mathcal{R}(\Psi|\Omega)$. When we use the proposed regularization function $\mathcal{R}_*(\Psi|\Omega)$ described in Section 3.3.1, we have the following gradient:

$$\frac{\partial \mathcal{R}(\Psi|\Omega)}{\partial x_n} = \frac{\partial \mathcal{R}_*(\Psi|\Omega)}{\partial x_n} = -\frac{1}{2}\sum_{j=1;j\neq n}\big(4\omega_{nj}(x_n - x_j)\big) - \sum_{j=1;j\neq n}\big(4(1-\omega_{nj})\frac{(x_n-x_j)}{(\mathcal{F}(\psi_n,\psi_j)+1)^2}\big).$$

As mentioned earlier, there is an efficiency advantage to regularizing on the visualization space. $\mathcal{R}(\Psi|\Omega)$ does not contain the variable ϕ_z if we do regularization on visualization space. The complexity of computing all $\frac{\partial \mathcal{R}(\Psi|\Omega)}{\partial x_n}$ is $O(N^2)$. In contrast, if we do regularization on topic space, we have to take the gradient of $\mathcal{R}(\Psi|\Omega)$ w.r.t to ϕ_z. That contributes towards a greater complexity of $O(Z^2 \times N^2)$ to compute all $\frac{\partial \mathcal{R}(\Psi|\Omega)}{\partial \beta_z}$. Therefore, regularization on topic space would run much slower than on visualization space.

3.5 Experiments

The main objective of our experiments is to evaluate the effectiveness of neighborhood regularization for semantic visualization model. After describing the experimental setup, we first examine the different design choices of the model relating to kernel, graph construction, and regularization function. Thereafter, we compare SEMAFORE against the baseline methods that also aim to address both visualization and topic modeling, quantitatively and qualitatively, first in terms of visualization

and then in terms of topic modeling.

3.5.1 Experimental Setup

In this section, we give a description of benchmark datasets as well as suitable metrics that are used for evaluation.

Datasets. We use three real-life, publicly available datasets [19] for evaluation.

- $20News$ contains newsgroup articles (in English) from 20 classes.
- $Reuters8$ contains newswire articles (in English) from 8 classes.
- $Cade12$ contains web pages (in Brazilian Portuguese) classified into 12 classes.

These are benchmark datasets used for document classification. While our task is fully unsupervised, the ground-truth class labels are useful for an objective evaluation. We create balanced classes by sampling fifty documents from each class, following the practice in PLSV [62]. This results in, for one sample, 1000 documents for $20News$, 400 for $Reuters8$, and 600 for $Cade12$. The vocabulary sizes are 5.4K for $20News$, 1.9K for $Reuters8$, 7.6K for $Cade12$. As the algorithms are probabilistic, we generate five samples for each dataset. For each sample, we conduct five independent runs. Therefore, the result reported for each setting is the average over a total of 25 runs.

Metrics. For a suitable metric, we return to the fundamental principle that a good visualization should preserve the relationship between documents (in high-dimensional space) in the lower-dimensional visualization space. For an objective evaluation, we rely on two types of quantitative analysis:

- *Classification*: This evaluation relies on the ground-truth class labels found in the datasets.

  The basis for this evaluation is the reasonable assumption that documents of the same class are more related than documents of different classes. Therefore a good visualization would place documents of the same class as neighbors on the visualization.

For each document d_n, we hide its true class c_n, and generate a prediction for its class $\hat{\mathcal{C}}_t(n)$ by taking the majority class among its t-nearest neighbors, as determined by Euclidean distance on the visualization space. Classification accuracy $Classification\_Acc(t)$ is defined as the fraction of documents whose predicted class $\hat{\mathcal{C}}_t(n)$ matches the true class c_n. More specifically, we have:

$$Classification\_Acc(t) = \frac{1}{N}\sum_{n=1}^{N}\delta(\hat{\mathcal{C}}_t(n) = c_n),$$

where δ is the delta function that equals 1 if the prediction matches and 0 otherwise.

This metric can be considered as an approximation for human evaluation when class labels are assigned by human. The essence of visualization is to show document similarity which is reflected by distances between documents in the visualization. To evaluate this, we need some ground truth that reflects human evaluation of document similarity. Here we assume that the ground truth similarity is provided by sharing the same class labels. Since class labels are often assigned by human, this metric which is based on distances indirectly approximates the degree to which document similarity is evaluated by human. This metric is also well-accepted and heavily used in the literature for the same evaluation [62],[114], [118].

While accuracy is computed based on documents' coordinates, the same trends will be produced if computed based on topic distributions (due to their coupling through the kernels described in Section 3.2.2).

- *Neighborhood Preservation*: This evaluation does not rely on the ground-truth class labels but on the local neighborhood structure in the input data. The assumption is that a good visualization would be able to preserve the local structure in the input data as much as possible. If two documents are neighbors in the input data, they should still be neighbors in the visualization

space.

For every document d_n, we compute sets of t-nearest neighbors $\mathcal{Y}_t(n)$ and $\mathcal{X}_t(n)$ of document d_n in the input data and the visualization respectively. The neighborhood preservation accuracy $Preservation\_Acc(t)$ is then defined as the average fraction of the overlap size of $\mathcal{Y}_t(n)$ and $\mathcal{X}_t(n)$ over the size of $\mathcal{Y}_t(n)$ (i.e. t), where $n = 1, \ldots, N$. More specifically, we have:

$$Preservation\_Acc(t) = \frac{1}{N} \sum_{n=1}^{N} \frac{|\mathcal{Y}_t(n) \cap \mathcal{X}_t(n)|}{t},$$

where $|\mathcal{Y}_t(n) \cap \mathcal{X}_t(n)|$ is the size of the overlap set $\mathcal{Y}_t(n) \cap \mathcal{X}_t(n)$.

A similar measure can be found in the literature [2], where it is called the "rate of agreement in local structure" or "agreement rate" and is used to measure how well the local structure is preserved between the input data and the low dimensional embedding. It is also used for tuning the parameters of a non-linear dimensionality reduction method [26].

In the subsequent experiments, we let t vary in the range $[5, 50]$ with the step size 5 and report the accuracies. Since different methods may behave differently at different t's, choosing a specific t for comparison may be unfair for some methods. Moreover, a method that consistently does well for different t's would also have a "smoother" local structure. Therefore, when comparing various methods, we present the preservation or classification accuracies averaged across $t \in [5, 50]$, denoted $Preservation\_Acc(Avg)$ and $Classification\_Acc(Avg)$ respectively.

3.5.2 Parameter Study

In this section, we study the effects of graph parameters on our model. Specifically, the parameters concern the graph construction, including the number of neighbors k in k-NN graph, the distance threshold ϵ in ϵ-ball graph, and the number of minimum spanning trees r in DMST. For each type of graph, we use the Simple Minded weight. For the following figures, the regularization function is $\mathcal{R}_*$ with $\lambda = 10$

and the number of topics $Z = 20$. We use neighborhood preservation accuracy $Preservation\_Acc(t)$ to show the effects of graph parameters because this metric does not need ground-truth class labels, which are not always available for tuning these graph parameters.

In Figure 3.3, we show the performance of our model with different neighborhood size k in k-NN graph for different datasets. For every k, we vary t and plot the $Preservation\_Acc(t)$. Figure 3.3 shows that the optimum k for $20News$, $Reuters8$, and $Cade12$ is 10, 10, and 5 respectively. We compute the average accuracy $Preservation\_Acc(Avg)$ and it confirms that the optima are indeed at those k values. From now on, we will use k=10 for $20News$ and $Reuters8$, and k=5 for $Cade12$ when k-NN graph is used.

For DMST graph, we plot the $Preservation\_Acc(t)$ for different number of minimum spanning trees r with different datasets in Figure 3.4. It is difficult to see which r is the best in the figure because the differences between them are not much. The $Preservation\_Acc(Avg)$ is computed and it shows that for all three datasets, the optimum is about at r=5,6,7. Subsequently, we will use r=6 for DMST graphs for all three datasets.

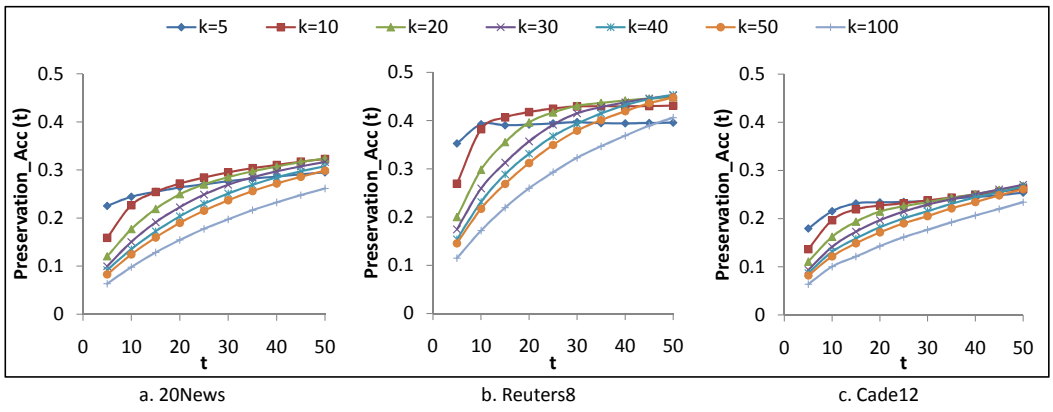


Figure 3.3: Preservation accuracy of SEMAFORE when using k-NN graph with different neighborhood size k for (a) $20News$, (b) $Reuters8$, and (c) $Cade12$.

For ϵ-ball graph, in Figure 3.5 we plot the $Preservation\_Acc(t)$ for different values of ϵ in the range $[1.32, 1.40]$. We choose that range because ϵ=1.32 and ϵ=1.40 roughly give an average number of neighbors of 5 and 100 respectively. The

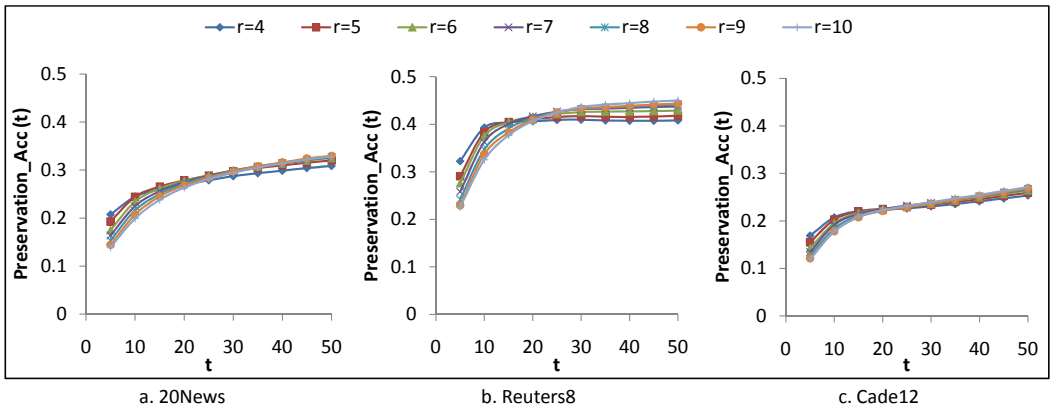


Figure 3.4: Preservation accuracy of SEMAFORE when using DMST graph with different number of minimum spanning trees r for (a) $20News$, (b) $Reuters8$, and (c) $Cade12$.

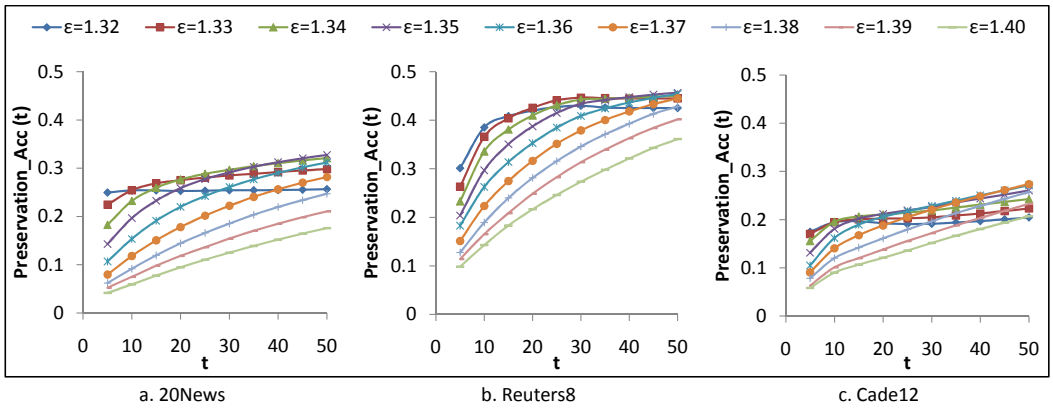


Figure 3.5: Preservation accuracy of SEMAFORE when using ϵ-ball graph with different values of distance threshold ϵ for (a) $20News$, (b) $Reuters8$, and (c) $Cade12$.

$Preservation\_Acc(Avg)$ shows that the optimum ϵ for $20News$, $Reuters8$, and $Cade12$ is 1.34, 1.35, and 1.33 respectively.

3.5.3 Model Analysis

In this section, we study the various design choices involved in designing the SEMAFORE model, before finally concluding on the eventual synthesis of design choices to be used for comparison against the baselines. To keep the discussion focused and organized, in each of the following sub-section, we vary a single design choice, in order to isolate its effects. When unvaried, the model has the following setup by

default: the number of topics is $Z = 20$, the graph construction method is k-NN, the graph weighting method is simple minded, the RBF kernel is Gaussian, and the regularization function is $\mathcal{R}_*$ with $\lambda = 10$.

Neighborhood Graph Construction

We investigate three graph construction methods: k-NN, ϵ-ball and DMST, which are representatives of neighborhood-based and minimum spanning tree-based methods respectively. For each graph, its parameter is tuned as shown in Section 3.5.2. For the regularization parameter λ, we try different settings of λ on each dataset. It so happens that $\lambda = 10$ performs the best for all the graph construction methods across the three datasets.

In Figure 3.6, we run SEMAFORE with different types of graph on the three datasets and report the $Preservation\_Acc(Avg)$ at different number of topics Z. The results show that different types of graph behave differently with different datasets. In $20News$, ϵ-ball and DMST give our model highest performance. Since the difference between the two are not statistically significant, we choose to use DMST for subsequent experiments on $20News$. For $Reuters8$, since ϵ-ball outperforms the others (significant at 0.05 level), it is going to be the default choice for subsequent experiments. For $Cade12$, the choice is DMST, which is slightly better than k-NN (statistically significant for $Z = 10, 40, 50$).

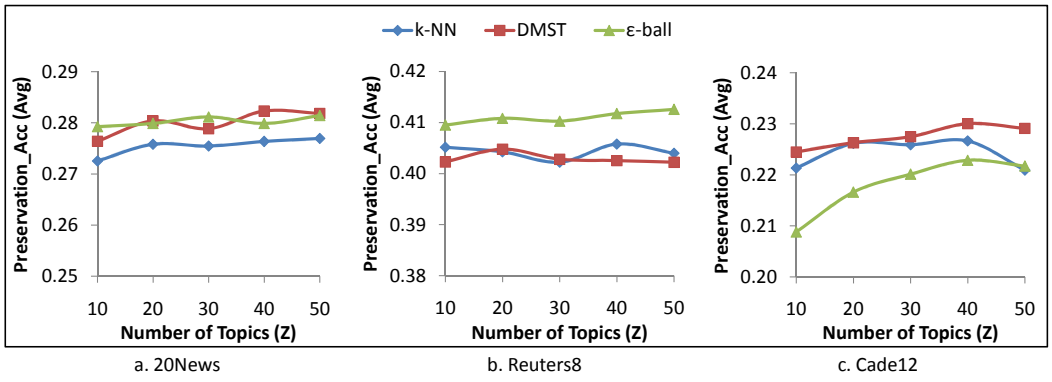


Figure 3.6: The effects of different graph construction methods on our model's performance.

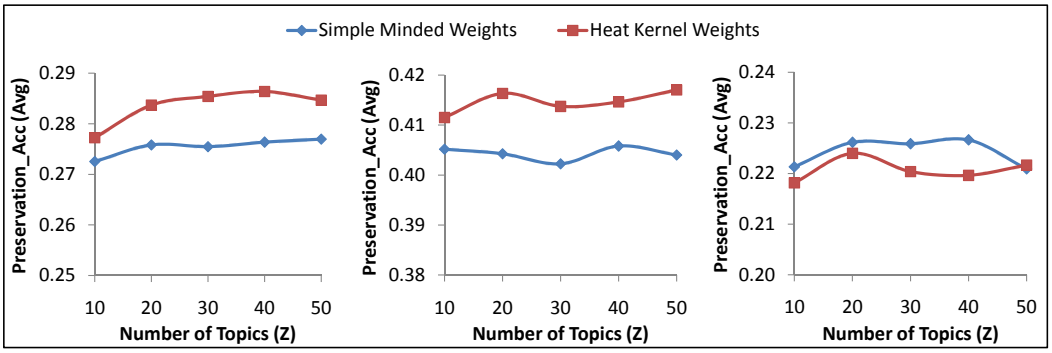


Figure 3.7: The effects of different graph weighting schemes on our model's performance. The graph used in this experiment is k-NN graph with specific k's for different datasets as studied in Section 3.5.2.

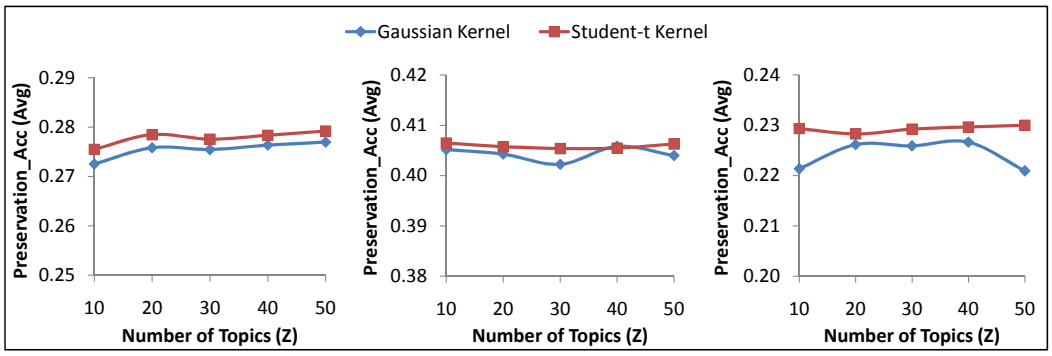


Figure 3.8: The effects of Gaussian and Student-t RBF kernels on our model's performance.

Neighborhood Graph Weighting

We now compare two variations of graph weighting methods, namely: Simple Minded and Heat Kernel methods. In this experiment, we use k-NN graph with specific k's for different datasets as studied in Section 3.5.2. The regularization parameter λ is set to 10 after trying various settings and picking the best one.

In Figure 3.7, we compare Simple Minded method and Heat Kernel method to see their influences on our model at different number of topics Z. We observe that Heat Kernel is significantly and consistently better than Simple Minded method across all the cases in $20News$ and $Reuters8$. The difference is statistically significant at 0.01 level. One explanation is that Heat Kernel assigns smoother weights to

the graph edges, and thus is more robust than Simple Minded. For $Cade12$, Simple Minded is slightly better, though the differences are statistically significant at 0.05 level only for $Z = 40$. Subsequently, we will use Heat Kernel for $20News$ and $Reuters8$, and Simple Minded for $Cade12$ as part of the final synthesis.

RBF Kernel

As described in Section 3.2.2, we express topic distributions as a function of visualization coordinates using RBF network as an abstraction. In this section, we show how different RBF kernels affect our model's performance. The two kernels we are exploring are Gaussian (Equation 3.3) and Student-t (Equation 3.4). We tune the regularization term λ for each kernel and see that the best one for the two kernels are $\lambda = 10$.

Figure 3.8 shows the results for different number of topics Z. Student-t kernel has a slight edge over Gaussian kernel consistently across different number of topics. The difference is small, but is statistically significant (at 0.05 level) in a majority of the cases (for $20News$ at $Z = 10, 20, 30, 50$, for $Reuters8$ at $Z = 30$, and for $Cade12$ at $Z = 10, 30, 50$). The slight improvement could be a sign that crowding problem does exist in the model. Student-t kernel would be even more useful when there is more extreme crowding issues, such as when the number of documents to be visualized is even larger. Subsequently, due to its slight edge, we will use Student-t as part of the final synthesis. As we will see shortly, using Student-t within the synthesized model results in a significant improvement overall.

Synthesised SEMAFORE Model

Based on the model analysis in the preceding paragraphs, we combine the design choices into a final synthesis model called SEMAFORE. The synthesized model is slightly different for different datasets, as listed in Table 3.1. We will use these synthesized models in the comparisons against the baseline methods in the following section.

	$20News$	$Reuters8$	$Cade12$
Regularization function	$\mathcal{R}_*$	$\mathcal{R}_*$	$\mathcal{R}_*$
Graph construction	DMST	ϵ-ball	DMST
Graph weighting	Heat Kernel	Heat Kernel	Simple Minded
RBF kernel	Student-t	Student-t	Student-t

Table 3.1: Synthesized Model for Each Dataset.

	Visualization	Topic model	Joint model	Neighborhood
SEMAFORE	✓	✓	✓	✓
PLSV	✓	✓	✓	
PE (LDA)	✓	✓		

Table 3.2: Comparative Methods.

3.5.4 Comparison of Visualizations

We now compare our proposed model with several baselines. First, we outline the set of comparative methods. Thereafter, we discuss quantitative evaluation (in terms of accuracy), as well as qualitative evaluation (in terms of example visualizations). Finally, we will show that the gains in visualization quality does not come at the expense of topic modeling.

As semantic visualization seeks to ensure consistency between topic model and visualization, the comparison focuses on methods producing *both* topics and visualization coordinates which are listed in Table 3.2.

- SEMAFORE is our proposed method that incorporates neighborhood structure into semantic visualization.
- PLSV [62] is the state-of-the-art, representing the joint approach without neighborhood structure preservation.
- PE (LDA) represents the pipeline approach involving topic modeling with LDA [14], followed by visualizing documents' topic distributions with PE [61]. This pipeline is better than the LDA/MDS that appeared in our earlier work [72]. There are other pipeline methods, shown inferior to PLSV [62], which are not reproduced here to avoid duplication.

Accuracy

In this section, we compare our model with several baselines in terms of classification accuracy (Figure 3.9) and neighborhood preservation accuracy (Figure 3.10). In the two figures, only the standard deviations for SEMAFORE are shown.

Classfication Accuracy. Figure 3.9(a), 3.9(c) and 3.9(e) show the $Classfication\_Acc(t)$ at different t's for $Z = 20$ for $20News$, $Reuters8$, and $Cade12$ respectively. At any t, the comparison shows outperformance by SEMAFORE over the baselines consistently. All four methods show the same behavior that their performances decrease when t increases. As t increases, they may lose accuracy in predicting labels for documents near to the border of each "cluster".

Now, we vary the number of topics Z. In Figure 3.9(b), we show the performance in $Classfication\_Acc(Avg)$ on $20News$. Figure 3.9(d) and 3.9(f) show the same for $Reuters8$ and $Cade12$ respectively. From these figures, we draw the following observations about the comparative methods:

- SEMAFORE performs the best on all datasets across various numbers of topics (Z). SEMAFORE beats PLSV by 25% to 51% on $20News$, by 6–13% on $Reuters8$, and by 22–32% on $Cade12$. These margins of performance with respect to PLSV are statistically significant at 0.01 significant level or lower in all cases. This effectively showcases the utility of neighborhood regularization in enhancing the quality of visualization. By preserving local consistency, SEMAFORE achieves a good accuracy even at small number of topics (e.g., 10).
- PLSV performs better than PE (LDA) , which shows that there is utility to having a *joint*, instead of separate, modeling of topics and visualization. PE (LDA) is worse than PLSV because it embeds documents by using two-step reductions that optimize separately two different objective functions. Therefore, the errors from the previous step may propagate to the next, without an opportunity for correction. This may cause distortions in the visualization.

- In some cases, PLSV, PE (LDA) tend to have decreasing accuracies when the number of topics increases. This may be because when number of topic increases, the topic distributions and the word probabilities may overfit the data and thus the accuracy is reduced. In contrast, SEMAFORE shows a quite stable performance across different numbers of topics. This may be explained by the utility of neighborhood regularization, which helps to prevent overfitting when the number of topics increases.

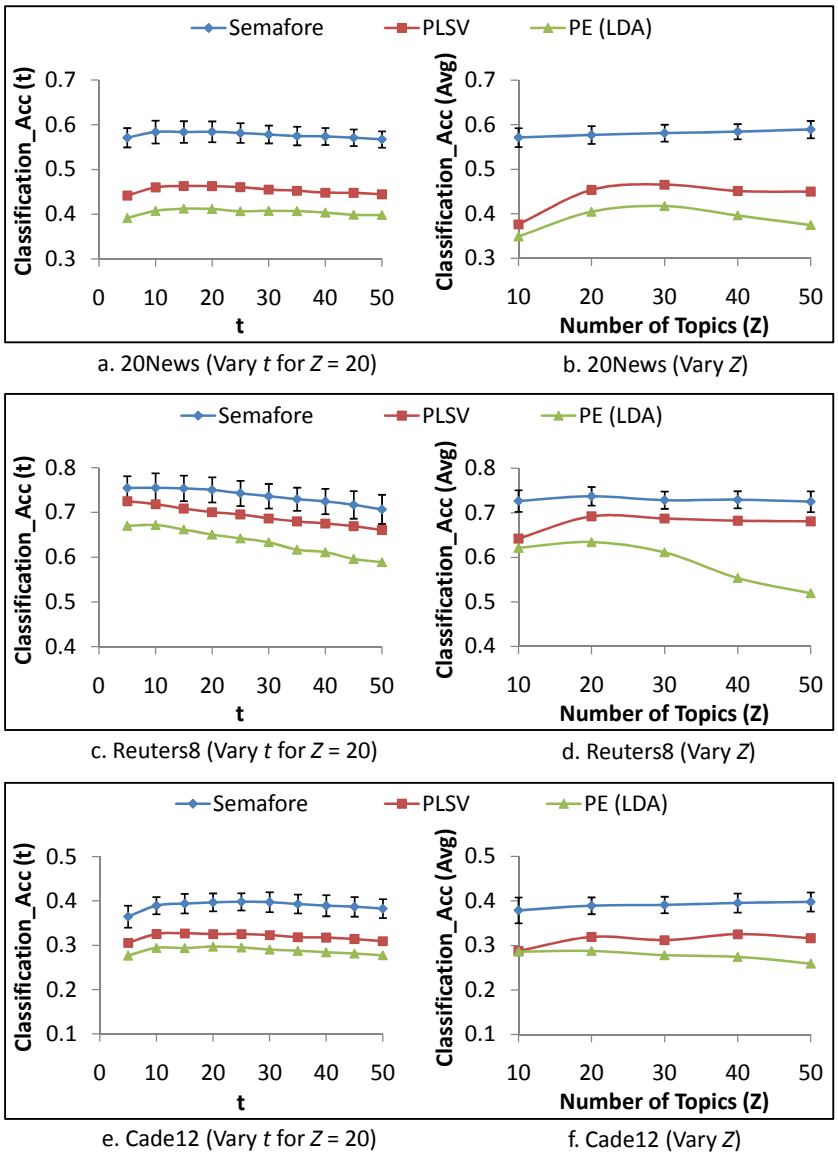


Figure 3.9: Classification Accuracy Comparison.

Neighborhood Preservation Accuracy. While having better classification accuracy, SEMAFORE also preserves well the local structure of the input data in the

visualization space. The $Preservation\_Acc(t)$ results in Figure 3.10(a),3.10(c) and 3.10(e) show that SEMAFORE is consistently better than the other baselines in terms of neighborhood preservation across different t's and different datasets. In Figure 3.10(b), 3.10(d) and 3.10(f), we vary the number of topics Z and report the $Preservation\_Acc(Avg)$ results. SEMAFORE beats PLSV by 41% to 76% on $20News$, by 24–36% on $Reuters8$, and by 29–45% on $Cade12$ in terms of neighborhood preservation accuracy. The improvements of SEMAFORE over PLSV are statistically significant at 0.01 significant level or lower in all cases.

The above accuracy results are based on visualization coordinates. We have also computed accuracies based on topic distributions, which have similar trends.

Visualizations

To provide an intuitive appreciation, we briefly describe a qualitative comparison of visualizations. For each method on each dataset, a visualization is shown as a scatterplot (best seen in color). Each document has a coordinate, and is assigned a shape and color based on its class. Each topic also has a coordinate, drawn as a black, hollow circle. A legend is provided, mapping each symbol to the corresponding class label.

Note that this is an illustrative, rather than a comparative discussion, as an objective evaluation should not rely on eyeballing alone. However, as we have shown the quantitative results in the preceding section, in this section, we focus on the qualitative study of the output visualizations.

20News. Figure 3.11 shows a visualization of $20News$ dataset. SEMAFORE's Figure 3.11(a) shows that the different classes are well separated. There are distinct clusters of blue squares and purple diamonds at the top for hockey and baseball classes respectively, clusters of orange triangles and pink asterisks at the bottom for cryptography and medicine, etc. Beyond individual classes, the visualization also places related classes nearby. Computer-related classes are found on the lower left. Politics and religion are on the lower right.

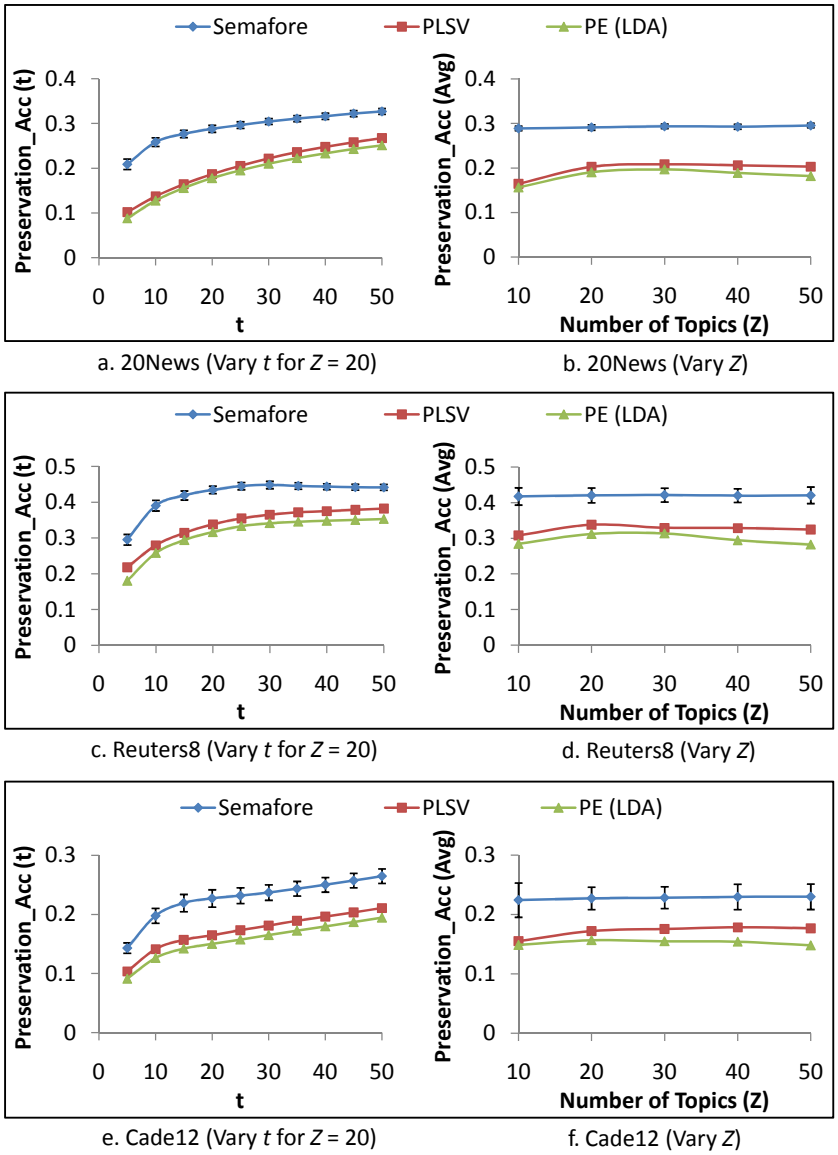


Figure 3.10: Preservation Accuracy Comparison.

Comparatively, Figure 3.11(b) by PLSV shows crowding at the center. For instance, motorcycle (green dashes) and autos (red dashes) are mixed at the center without a good separation. Figure 3.11(c) by PE (LDA) is worse. PE (LDA) does not give good separation for not similar classes. It mixes autos (red dashes) and space (green circles) together at the center. Medicine (pink asterisks) is also mixed with other classes in PE (LDA) while SEMAFORE and PLSV give a good separation for it.

Reuters8. Figure 3.12 shows the visualization outputs for $Reuters8$ dataset. SEMAFORE in Figure 3.12(a) is better at separating the eight classes into distinct

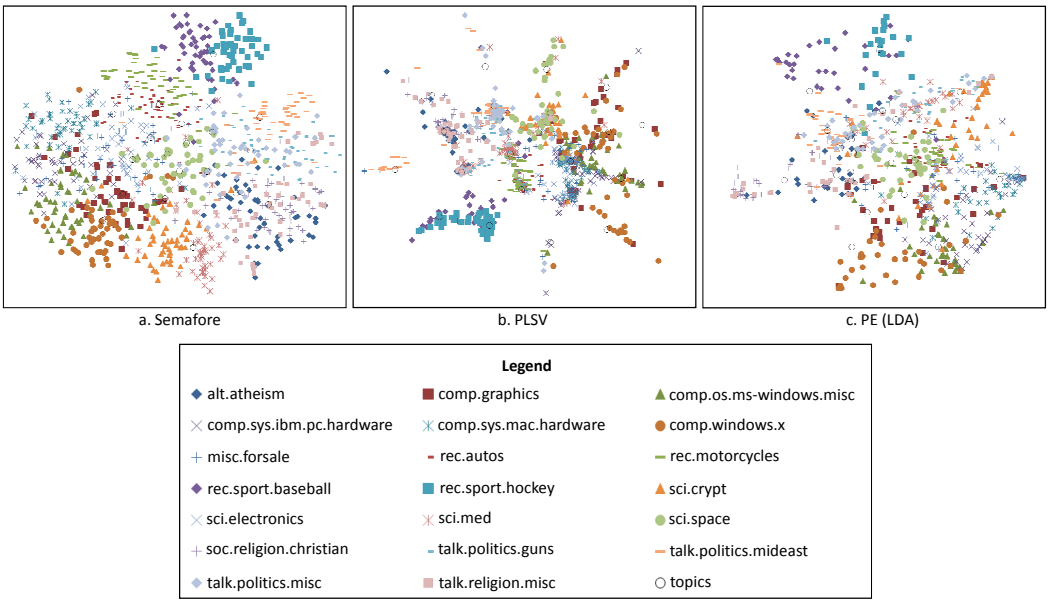


Figure 3.11: Visualization of documents in $20News$ for number of topics $Z = 20$. Each point represents a document and the shape and color represent document class. Each topic is drawn as a black, hollow circle.

clusters. In an anti-clockwise direction from the top, we have navy blue diamonds (*money-fx*), red dashes (*interest*), red squares (*crude*), light blue pluses (*earn*), green triangles (*acq*), purple crosses (*ship*), blue asterisks (*grain*), and finally orange circles (*trade*).

In comparison, PLSV in Figure 3.12(b) shows that several classes are intermixed at the center, including red dashes (*interest*), orange circles (*trade*), and navy blue diamonds (*money-fx*). PE (LDA) in Figure 3.12(c) is also worse when it mixes differentiated classes such as red dashes (*interest*) and navy blue diamonds (*money-fx*) together.

Cade12. Figure 3.13 shows the visualization outputs for $Cade12$. This is the most challenging dataset. Even so, SEMAFORE in Figure 3.13(a) still achieves a better separation between the classes, as compared to PLSV in Figure 3.13(b). Particularly, SEMAFORE gives better separation for esportes (green triangles) as well as compras-on-line (orange circles) than PLSV and PE (LDA).

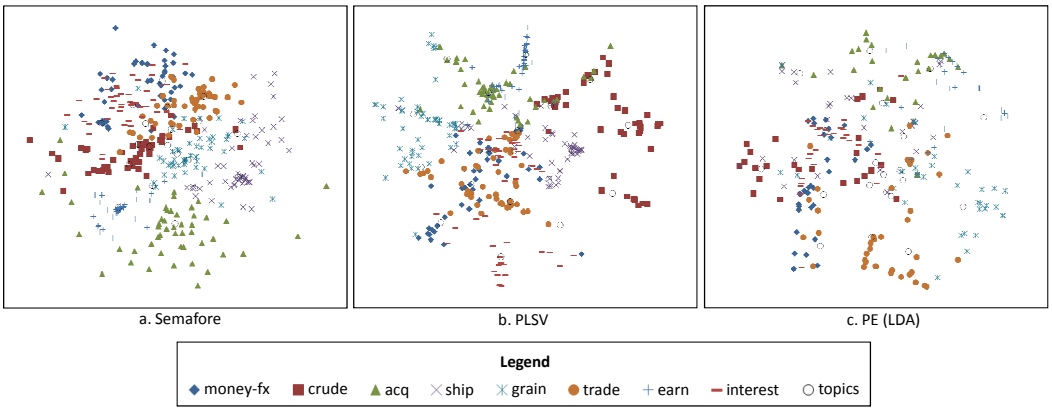


Figure 3.12: Visualization of documents in $Reuters8$ for number of topics $Z = 20$. Each point represents a document and the shape and color represent document class. Each topic is drawn as a black, hollow circle.

3.5.5 Comparison of Topic Models

One question is whether SEMAFORE's gain in visualization quality over the closest baseline PLSV is at the expense of the quality of its topic model. To investigate this, we will compare the topic models of SEMAFORE and PLSV, which share a core generative process. For parity, in this comparison, we only include the joint models, whereby the visualization coordinates affect the topic models as well.

The metric we use to measure the quality of topic models is pairwise mutual information or PMI. It measures topic interpretability, based on coocurrence frequencies of the top words in each topic in a large external corpus. Although other metrics such as perplexity or held-out likelihood can show the generalization ability of a learned topic model on unseen test data, these traditional metrics do not capture whether topics are coherent [25]. Therefore, in this comparison, we rely on PMI, which can measure the quality of topic words in terms of their interpretability to a human. To human subjects, interpretability is closely related to coherence [91], i.e., how much the top keywords in each topic are "associated" with each other. After an extensive study of evaluation methods for coherence, Newman et al. 2010 identify Pointwise Mutual Information (PMI) as the best measure, in terms of having the greatest correlation with human judgments.

PMI is based on term cooccurrences. For a pair of words w_i and w_j, PMI is

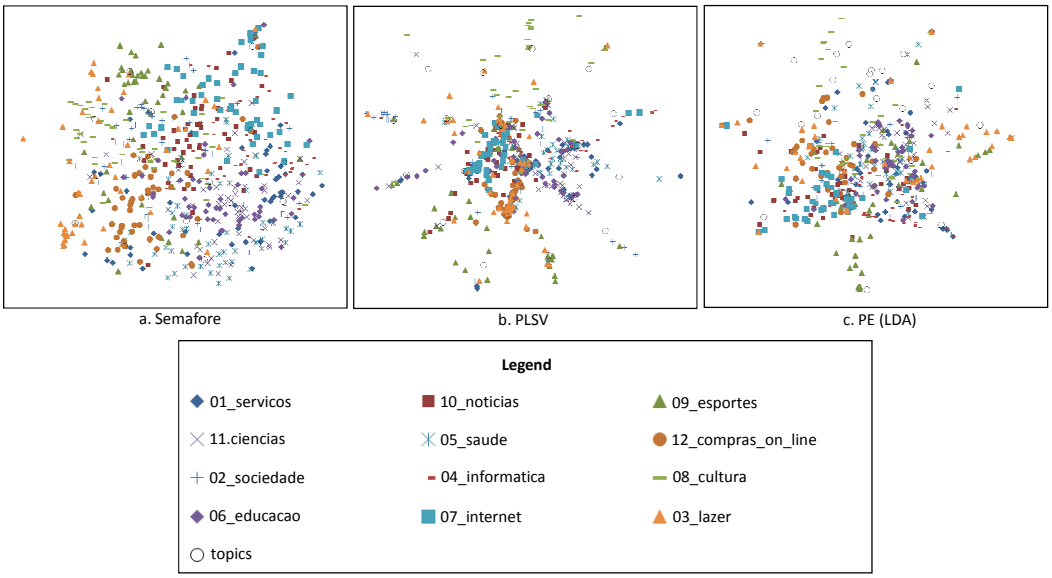


Figure 3.13: Visualization of documents in $Cade12$ for number of topics $Z = 20$. Each point represents a document and the shape and color represent document class. Each topic is drawn as a black, hollow circle.

defined as $\log \frac{p(w_i,w_j)}{p(w_i)p(w_j)}$. For a topic, we average the pairwise PMI's among the top 10 words of that topic. For a topic model, we average PMI across the topics. Intuitively, PMI is higher (better), if each topic features words that are highly correlated with one another.

Key to PMI is the use of an external corpus to estimate $p(w_i, w_j)$ and $p(w_i)$. Following Newman et al. 2009, we use *Google Web 1T 5-gram Version 1* [15], a huge corpus of n-grams generated from 1 trillion word tokens. $p(w_i)$ is estimated from the frequencies of 1-grams. As recommended by Newman et al., $p(w_i, w_j)$ is estimated from the frequencies of 5-grams. We obtain PMI for the English-based $20News$ and $Reuters8$, but not for $Cade12$ because we do not possess a large-scale n-gram corpus specifically for Brazilian Portuguese.

In Figure 3.14, we plot the PMI score for various number of topics Z. SEMAFORE performs better than PLSV across most of the topics settings. In Figure 3.14(a) for 20News, except for the case at $Z = 10$, all cases of SEMAFORE's outperformance are significant at 0.05 level or lower. In Figure 3.14(b) for Reuters8, all cases of SEMAFORE's outperformance are significant at 0.05 level or lower ex-

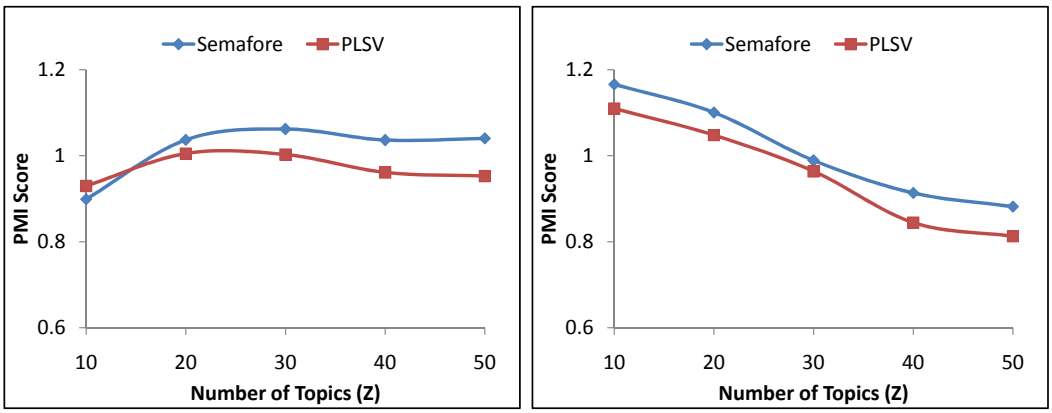


Figure 3.14: Topic Interpretability of SEMAFORE and PLSV in terms of PMI Score (higher is better).

cept for the $Z = 30$. These results show that SEMAFORE improves visualization while not sacrificing the topic interpretability of learned topics.

3.6 Conclusion

In this paper, we address the semantic visualization problem, which jointly conducts topic modeling and visualization of documents. We propose a new framework to incorporate neighborhood structure within a probabilistic semantic visualization model called SEMAFORE. The model is carefully designed to reflect the context of semantic visualization, leading to a number of design choices related to the RBF kernel for mapping topic and visualization spaces, the approximation of neighborhood graph through construction and weighting, as well as the appropriate regularization functions and spaces. Experiments on real-life datasets show that SEMAFORE significantly outperforms the baselines in terms of visualization quality and accuracy, while having a similar, if not slightly better topic model. This provides evidence that neighborhood structure, together with joint modeling of topics and visualization, is important for semantic visualization.

Chapter 4

Modeling Network Structure

A document network refers to a data type that can be represented as a graph of vertices, where each vertex is associated with a text document. Examples of such a data type include hyperlinked Web pages, academic publications with citations, and user profiles in social networks. Such data have very high-dimensional representations, in terms of text as well as network connectivity. In this chapter, we study the problem of embedding, or finding a low-dimensional representation of a document network that "preserves" the data as much as possible. These embedded representations are useful for various applications driven by dimensionality reduction, such as visualization or feature selection. While previous works in embedding have mostly focused on either the textual aspect or the network aspect, we advocate a holistic approach by finding a unified low-rank representation for both aspects. Moreover, to lend semantic interpretability to the low-rank representation, we further propose to integrate topic modeling and embedding within a joint model. The gist is to join the various representations of a document (words, links, topics, and coordinates) within a generative model, and to estimate the hidden representations through MAP estimation. We validate our model on real-life document networks, showing that it outperforms comparable baselines comprehensively on objective evaluation metrics.

4.1 Introduction

Due to their importance and wide applicability, document networks have been an intensive subject of research, particularly in information retrieval and link analysis. Relatively less attention has been paid to much-needed methods for conducting *exploratory analysis* on document networks. Analyzing a document network is very challenging because of the high-dimensional nature of the data. In one sense, a document can be expressed in terms of the occurrences of words (i.e., the dimensionality of text). In another sense, a document can also be expressed in terms of its connectivity to the other documents (i.e., the dimensionality of network).

Problem. In this work, we focus on the *embedding* problem. Given a document network, our objective is to "embed" (or reduce) the documents' high-dimensional representations (both in terms of text as well as network connnectivity) in a low-dimensional space that would still preserve as much of the "properties" of the original data as possible. Embedding is a well-recognized problem in machine learning (see Section 4.2). However, existing methods have not been designed with a document network in mind. We identify two issues that affect the fittingness of these methods for embedding a document network. The first issue is the *lack of connection between text and network*. Most methods have been designed either for embedding text documents, or for embedding a network. Obtaining either one embedding alone may offer a potentially distorted or incomplete view of the data. Obtaining both embeddings separately may produce two different representations that are not easily reconciled.

The second issue is the relative *lack of semantic interpretability*. Previous embedding methods produce low-dimensional representations that are not easily interpretable (other than as axes of the scatterplot visualization). In this respect, we are inspired by topic modeling [14], which obtains low-rank representations (i.e., topics) that are semantically interpretable (through high-probability words of each topic). However, topic modeling is not a solution to the embedding problem. For instance, to produce two-dimensional (2D) visualization, we can represent docu-

ments' topic distributions on a 2D simplex space, but this is only possible for three topics, which would be severely limiting as most applications of topic modeling require tens, if not hundreds, of topics [14].

4.1.1 Overview

To address the above issues, we propose a holistic and integrated approach based on two key principles. The first principle is to *embed both text and network representations of a document into a single unified low-rank representation*. This is grounded in the intuition that text content and network connectivity can inform each other. On one hand, text content can help to resolve ambiguities in the network. For instance, unobserved edges in a network may indicate either a genuine absence or a missing presence. If two documents are different in text content, the former is more likely than the latter. On the other hand, network connectivity can help to resolve the ambiguities in text through observed edges among documents that use different words for the same concept (synonymy), or missing edges among documents that use common words to refer to different concepts (polysemy).

The second principle is to *incorporate both a topic model and an embedding model within a single joint model*. To make our discussion more concrete, without loss of generality, we assume that the low-rank embedding takes the form of 2D visualization coordinates. This joint modeling is mutually beneficial to both topic modeling and visualization. By incorporating a topic model, we can infuse the visualization with semantic interpretability. Each point on a 2D scatterplot can be associated with the most likely topics or words [62]. By incorporating an embedding model, the mapping between topics and visualization may eventually offer a natural interface for user interaction to tune the underlying topic model [28].

We are thus motivated to tie together the four representations of each document in a document network, namely: the two high-dimensional representations in terms of word occurrences and network connectivity respectively, the intermediate representation in terms of a topic distribution as in topic modeling, as well as the low-rank

representation in terms of visualization coordinates as in embedding. One framework to join these disparate representations is *generative modeling*, a probabilistic model for the generation of observable data through modeling random variables (that encode the representations mentioned above). Generative modeling has been the bedrock for much of the topic modeling works that build on [14], though it has not been as widely applied to embedding.

4.1.2 Contributions

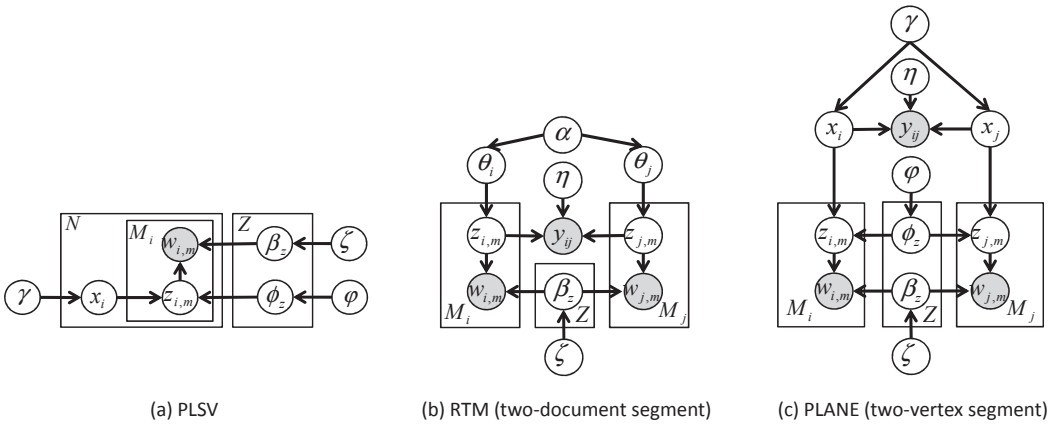
First, our novelty arises from the holistic approach to topic-based embedding of document networks. In comparison, previous works, reviewed in Section 4.2, have attempted this as separate segments, namely: embedding of documents, embedding of networks, or topic modeling, but have not recognized the embedding of a document network as a distinct problem to be addressed in its own entirety.

Second, to address this problem, we develop a generative modeling approach, and propose a model called PLANE, which stands for *P*robabilistic *LA*tent Document *N*etwork *E*mbedding. In Section 4.3, we describe the process of generation of observable data (text and network) from latent representations (topics and visualization coordinates). In Section 4.4, we outline an MAP inference algorithm to estimate the hidden parameters of this model through EM.

Third, to validate this model, we conduct comprehensive experiments (Section 4.5) on four real-life document networks derived from a benchmark collection of academic publications. We compare our model, quantitatively as well as qualitatively, against comparable baselines on both aspects (embedding and topic modeling) on a number of objective evaluation metrics.

4.2 Related Work

In terms of embedding. While we focus on embedding a document network, there are previous efforts on embedding documents, or embedding a network, which we



(a) PLSV (b) RTM (two-document segment) (c) PLANE (two-vertex segment)

Figure 4.1: Graphical Models of PLSV (a), RTM (b) and PLANE (c)

review below.

To embed documents, we can employ embedding techniques, which take as input N high-dimensional vectors $\{v_i\}_{i=1}^N$ and generate as output N low-rank vectors $\{x_i\}_{i=1}^N$. For instance, the v_i's may be the bag-of-words representations of documents, and the x_i's may be visualization coordinates. Good embedding produces x_i's that represent the v_i's "faithfully". In traditional embedding [70, 104, 115], this criterion is frequently formulated as preserving the distances among v_i's in the distances among x_i's. More recent approaches [61, 117] formulate this in terms of probabilities.

Recent works advocate having an intermediate representation, which is the topic space. The closest one to ours is PLSV [62], which pioneers the integration of topic modeling and visualization in a joint model. Figure 4.1(a) shows the graphical model of PLSV. Its generative process is as follows. For each topic z, we draw its word distribution β_z from a Dirichlet with parameter ζ, as well as its coordinate ϕ_z from a Normal distribution with mean 0 and variance φ^{-1}. For each document v_i, we draw its coordinate x_i from Normal with mean 0 and variance γ^{-1}. To generate each of the M_i words in v_i, we draw a topic $z_{i,m}$ based on the relative distance between x_i and topic coordinates, and draw a word from the selected topic's word distribution $\beta_{z_{i,m}}$. Since PLSV models only documents, our model builds on it by integrating a network model.

To embed a network, we can employ graph embedding techniques, of which there are broadly two main categories of approaches. The first category is *spectral embedding*, where the focus is on dimensionality reduction. For instance, the adjacency matrix representing the graph can be used as input to SVD [48] or PCA [65], whose objective is compressibility (preserving the variance in the data). To produce a low-dimensional embedding, the first few principal eigenvectors (with the largest eigenvalues) can be used as the coordinates $\{x_i\}_{i=1}^{N}$. This approach has been widely used for various large-scale graphs [113]. Building on this, SPE [108] attempts to preserve the neighboring structure as well, but since it is formulated as semidefinite programming, it is computationally very expensive for large-scale graphs [113].

The second category is *spring embedding*, also known as force-directed graph drawing. One example is the Fruchterman and Reingold layout [45] (FR-layout), which simulates a force system where spring-like attractive forces on links pull connected nodes together. The simulation is repeated iteratively till a mechanical equilibrium state is reached (energy minimization). Another approach Kamada and Kawai layout [66] (KK-layout) is also based on the idea of a balanced spring system and energy minimization, but achieves faster convergence due to the use of derivatives. These layouts are commonly found in graph visualization programs [6, 42].

In terms of topic modeling. Topic modeling is originally designed for documents [14], where each document is associated with a topic distribution, and each topic is associated with a word distribution. There also exists similar statistical modeling of networks as surveyed in [47]. For instance, in mixed membership stochastic blockmodel [1], each user is associated with a distribution over "communities", which explain the generation of links among users.

Recognizing the wide availability and applicability of document networks as a distinct data type, subsequent works seek to combine text and networks. One example is through a regularization framework [86], which however is not a joint model, and therefore does not model the generation of links. Yet others [24, 80, 90, 111] focus on modeling the generation of both text documents and network links

jointly.

Our work builds on the Relational Topic Model (RTM) [24], which we review briefly below. Its graphical model is shown in Figure 4.1(b). Each document v_i is associated with a topic distribution θ_i. To generate the m^{th} word in v_i, we first pick a topic $z_{i,m}$ from θ_i, then pick a word $w_{i,m}$ from $z_{i,m}$'s topic multinomial $\beta_{z_{i,m}}$. θ_i and β_z have Dirichlet priors of α and ζ respectively. In turn, each link y_{ij} between a pair of documents v_i and v_j is generated from a link probability function based on the topics that occur in v_i and v_j. The more they share common topics, the more likely there to be a link between them. There are a number of key differences between RTM and PLANE. Most importantly, we need to consider the low-rank embedding objective. We also model link generation based on coordinates instead of topic distributions. In our model likelihood, we also incorporate "virtual" negative links, not just observed positive links (see Section 4.3).

There are also some works on visualizing topic models [23, 29, 50, 123], where the focus is on visualizing which topics are important in a corpus, or which words are important in a topic. While they convey some information visually, they are orthogonal to our objective. They are not low-rank embedding techniques, and do not produce a low-rank representation for each document, which can also be used in non-visualization applications such as dimensionality reduction or compression.

4.3 Generative Model

Here, we describe the framework and the generative process of our proposed model PLANE, whose graphical representation in terms of a plate diagram is shown in Figure 4.1(c).

Framework. We consider as input a document network, represented as a graph $G = (V, E)$. V is a set of N vertices. Each vertex $v_i \in V$ refers to a document, and is associated with a bag of words. We denote $w_{i,m}$ to be the m^{th} word token in v_i, and M_i to be the total number of word tokens in v_i. Each token has a symbol

drawn from the vocabulary of words W. In turn, E is a set of edges in G, where each edge $e_{ij} \in E$ connects two vertices v_i and v_j. In this work, we would model an undirected graph, i.e., $e_{ij} = e_{ji}$, as our emphasis is on connectivity, rather than on directionality. The model could still apply to directed graphs by dropping the edge directions. In this paper, we use the term "edge" and "link" interchangeably.

As output, we aim for dual objectives as follows.

- *Embedding*: For each vertex v_i, we seek to learn its low-rank representation x_i, expressed as coordinates on a D-dimensional space. In this paper, which is framed in terms of embedding in a visualization space, we assume $D = 2$, without loss of generality.
- *Topic Modeling*: For each vertex v_i, we also seek to learn its representation in the topic space, expressed as a probability distribution $\{\mathrm{P}(z|v_i)\}_{z=1}^{Z}$ over a specified number of Z topics, where $D \ll Z \ll |W|$ is expected in most cases. Correspondingly, each topic z is associated with β_z, a probability distribution over words $\{\mathrm{P}(w|\beta_z)\}_{w \in W}$, where words with high probabilities provide semantic meaning to the topic.

To unify the dual objectives above, we need to concretely define how the two objectives are correlated with each other. This can be achieved by a mapping function from the visualization space to the topic space. Towards realizing this mapping, we associate each topic z with a visualization coordinate ϕ_z in the same D-dimensional space. If we model each ϕ_z to be the mean of a unit-variance Gaussian, and x_i to have been drawn from a mixture of Gaussians centered at ϕ_z's (with uniform mixture weights), we can express $\mathrm{P}(z|v_i)$ as the *responsibility* of the z's component of the Gaussian mixtures [13], as shown in Equation 4.1, which has also been used in [61, 62]. Here, $||\cdot||$ is the Euclidean norm defined on the visualization space, and $\Phi = \{\phi_z\}_{z=1}^{Z}$ refers to the collection of all topic coordinates.

$$\mathrm{P}(z|v_i) = \mathrm{P}(z|x_i, \Phi) = \frac{\exp(-\frac{1}{2}||x_i - \phi_z||^2)}{\sum_{z'=1}^{Z} \exp(-\frac{1}{2}||x_i - \phi_{z'}||^2)} \tag{4.1}$$

This mapping has an intuitive meaning. The closer is x_i to ϕ_z in the visualiza-

tion space, the greater is the probability of topic z in vertex v_i. It follows that if two vertices are close in the visualization space, they will also share similar topic distributions, thus encoding the above-mentioned embedding objective of finding similar low-rank representations for documents with similar high-dimensional representations.

Generative Process. We now describe the full generative process of our proposed model PLANE below.

1. For each topic $z = 1, \ldots, Z$:
   (a) Draw z's word distribution: $\beta_z \sim \text{Dirichlet}(\zeta)$
   (b) Draw z's coordinate: $\phi_z \sim \text{Normal}(0, \varphi^{-1} I)$
2. For each vertex v_i, where $i = 1, \ldots, N$:
   (a) Draw v_i's coordinate: $x_i \sim \text{Normal}(0, \gamma^{-1} I)$
   (b) For each word $w_{i,m}$, where $m = 1, \ldots, M_i$:
      i. Draw a topic: $z_{i,m} \sim \text{Categorical}(\{\text{P}(z|x_i, \Phi)\}_{z=1}^{Z})$
      ii. Draw a word: $w_{i,m} \sim \text{Categorical}(\beta_{z_{i,m}})$
3. For each pair of vertices v_i and v_j:
   (a) Draw e_{ij}'s binary indicator: $y_{ij} \sim \text{Bernoulli}\left(\text{P}(y_{ij} = 1|x_i, x_j, \eta)\right)$

Step 1 shows the generation of the parameters for each topic z. Like classical topic models [14], its word distribution β_z has a Dirichlet prior (with hyper parameter ζ). Its visualization coordinate ϕ_z has a Normal prior (centered at 0 with precision φ). The mean at 0 determines the locality of the visualization.

Step 2a shows the generation of parameter for each vertex v_i, which is its visualization coordinate x_i, from a Normal distribution with mean 0 and precision γ. Following Equation 4.1, this coordinate is mapped to v_i's representation in the topic space, which is a probability distribution over the Z topics, i.e., $\{\text{P}(z|x_i, \Phi)\}_{z=1}^{Z}$.

Step 2b encodes the *document embedding* step, where the "embedded" low-dimensional representation x_i generates the high-dimensional text representation

(bag of words). Based on x_i's topic space representation, we repeatedly draw a topic $z_{i,n}$ from $\{\mathrm{P}(z|x_i, \Phi)\}_{z=1}^{Z}$, followed by drawing a word $w_{i,m}$ from the topic's word distribution $\beta_{z_{i,m}}$.

Step 3 encodes the *network embedding* step, where the "embedded" low-dimensional representation x_i generates the high-dimensional network representation (i.e., which other vertices v_i is connected to). We associate each edge e_{ij} with a binary random variable denoted by y_{ij}, with a value of 1 if the edge is present ($e_{ij} \in E$), and 0 otherwise ($e_{ij} \notin E$). This random variable is drawn from a Bernoulli distribution. The Bernoulli parameter is denoted by $\mathrm{P}(y_{ij} = 1|x_i, x_j, \eta) \in [0, 1]$, which determines the probability that an edge exists between two vertices based on the vertices' latent coordinates x_i and x_j, and a parameter η (to be defined shortly).

Naturally, for network embedding, we desire that connected vertices would share similar embedded parameters. In that sense, the more similar are x_i and x_j, the higher is the $\mathrm{P}(y_{ij} = 1|x_i, x_j, \eta)$. Since x_i and x_j are coordinates, their "similarity" can be measured in terms of Euclidean distance $||x_i - x_j||$. To transform this distance into a probability value, we adopt the exponential probability function [13], as shown in Equation 4.2, where η is a parameter to be learned. In this work, we seek to study the connectivity hypothesis itself. While there could be other ways to realize the edge probability function, we keep the exploration in that direction to future work.

$$\mathrm{P}(y_{ij} = 1|x_i, x_j, \eta) = \exp(-\eta \cdot ||x_i - x_j||^2) \tag{4.2}$$

Our modeling of edge probability function based on distance ties together all the representations (document, networks, visualization coordinates, topics). This sets us apart from others that model only subsets of these representations (e.g., documents and networks but not visualization [24], documents and visualization but not networks [62]).

Model Likelihood. PLANE's graphical model in Figure 4.1(c) shows how the various representations are related to one another. Importantly, the observed (shaded) variables are only the words $\{w_{i,m}\}$ in vertex v_i, as well as the edges' indicators

$\{y_{ij}\}$. Equation 4.3 shows the log-likelihood function for generating these observed variables in the input graph $G = (V, E)$ based on the hidden parameters, such as embedding coordinates $\{x_i\}$ and topic multinomials $\{\beta_z\}$. The first component corresponds to the text associated with vertices in V. The second component corresponds to the edges.

$$\mathcal{L}(G) = \sum_{i=1}^{N} \sum_{m=1}^{M_i} \log \sum_{z=1}^{Z} \mathrm{P}(w_{i,m}|\beta_z)\mathrm{P}(z|x_i, \Phi) + \sum_{ij} \log \mathrm{P}(y_{ij}|x_i, x_j, \eta) \tag{4.3}$$

We need to decide how to model observed and unobserved edges. One way is to set $y_{ij} = 1$ when an edge is observed between vertices v_i and v_j, and $y_{ij} = 0$ otherwise. As stated in [24], this approach may be inappropriate when the absence of an edge cannot be used as evidence for $y_{ij} = 0$. To resolve this, they decided to model only observed edges (i.e, y_{ij} is either 1 or unobserved) [24]. While doing so can speed up computation, it falls short of the full discriminating power because the hidden structure of the corpora cannot be described fully only based on the positive observations ($y_{ij} = 1$). The negative observations ($y_{ij} = 0$) should also be considered.

Due to the reason above, we decide to model both observed and unobserved edges. We treat observed edges as positive observations ($y_{ij} = 1$). For unobserved edges, we assume that only a subset of them would be negative ($y_{ij} = 0$). It is not necessary to specify which particular edges are negative. Let ρ be the expected number of these "virtual" negative observations (to be learned from the data), and $U = \frac{N \times (N-1)}{2} - |E|$ be the total number of unobserved edges. The expected log likelihood of these negative observations is as follows.

$$\frac{\rho}{U} \sum_{e_{ij} \notin E} \log \mathrm{P}(y_{ij} = 0|x_i, x_j, \eta) \tag{4.4}$$

Therefore, the final log-likelihood of our model will be computed as follows.

$$\mathcal{L}(G) = \sum_{i=1}^{N}\sum_{n=1}^{M_i}\log\sum_{z=1}^{Z}\mathrm{P}(w_{i,m}|\beta_z)\mathrm{P}(z|x_i,\Phi)+ \sum_{e_{ij}\in E}\log\mathrm{P}(y_{ij}=1|x_i,x_j,\eta)+ \frac{\rho}{U}\sum_{e_{ij}\notin E}\log\mathrm{P}(y_{ij}=0|x_i,x_j,\eta) \tag{4.5}$$

4.4 Parameter Estimation

We estimate the parameters based on maximum a posteriori (MAP) estimation using EM algorithm [39]. The parameters that need to be estimated are the word probabilities $\{\beta_z\}_{z=1}^{Z}$, the topic coordinates Φ, the vertex coordinates $\{x_i\}_{i=1}^{N}$. η and ρ will also be learned from data. Since η and ρ are positive, let $\eta = \eta_{sqr}^2$ and $\rho = \rho_{sqr}^2$. Instead of directly learning η and ρ, we will learn η_{sqr} and ρ_{sqr} to avoid imposing the positivity constraints when optimizing the likelihood. We denote the collection of the unknown parameters as Ψ.

The conditional expectation of the complete-data log likelihood in MAP estimation with priors is:

$$\begin{aligned}\mathcal{Q}(\Psi|\hat{\Psi}) = &\sum_{i=1}^{N}\sum_{n=1}^{M_i}\sum_{z=1}^{Z}\mathrm{P}(z|i,m,\hat{\Psi})\log\left[\mathrm{P}(z|x_i,\Phi)\mathrm{P}(w_{i,m}|\beta_z)\right] \\ &+\sum_{i=1}^{N}\log(\mathrm{P}(x_i))+\sum_{z=1}^{Z}\log(\mathrm{P}(\phi_z))+\sum_{z=1}^{Z}\log(\mathrm{P}(\beta_z)) \\ &+\sum_{e_{ij}\in E}\log\mathrm{P}(y_{ij}=1|x_i,x_j,\eta)+ \\ &+\frac{\rho}{U}\sum_{e_{ij}\notin E}\log\mathrm{P}(y_{ij}=0|x_i,x_j,\eta)\end{aligned}$$

$\hat{\Psi}$ is the current estimate. $\mathrm{P}(z|i,m,\hat{\Psi})$ is the class posterior probability of the i^{th} document and the m^{th} word in the current estimate. $\mathrm{P}(\beta_z)$ is a symmetric Dirichlet prior with parameter ζ for word probability β_z. $\mathrm{P}(x_i)$ and $\mathrm{P}(\phi_z)$ are Gaussian priors with a zero mean and a spherical covariance for the document coordinates x_i and topic coordinates ϕ_z. We set the hyper-parameters to $\zeta = 0.01$, $\varphi = 0.1N$ and

$\gamma = 0.1Z$ as in [62].

In the E-step, $\mathrm{P}(z|i, m, \hat{\Psi})$ is updated as follows.

$$\mathrm{P}(z|i, m, \hat{\Psi}) = \frac{\mathrm{P}(z|\hat{x}_i, \hat{\Phi})\mathrm{P}(w_{i,m}|\hat{\beta}_z)}{\sum_{z'=1}^{Z} \mathrm{P}(z'|\hat{x}_i, \hat{\Phi})\mathrm{P}(w_{i,m}|\hat{\beta}_{z'})}$$

In the M-step, by maximizing $\mathcal{Q}(\Psi|\hat{\Psi})$ w.r.t β_{zw}, the next estimate of word probability β_{zw} is as follows.

$$\beta_{zw} = \frac{\sum_{i=1}^{N}\sum_{m=1}^{M_i} \mathrm{I}(w_{i,m} = w)\mathrm{P}(z|i, m, \hat{\Psi}) + \zeta}{\sum_{w'=1}^{W}\sum_{i=1}^{N}\sum_{n=1}^{M_i} \mathrm{I}(w_{i,m} = w')\mathrm{P}(z|i, m, \hat{\Psi}) + \zeta W}$$

$\mathrm{I}(.)$ is the indicator function. ϕ_z and x_i cannot be solved in a closed form, and are estimated by maximizing $\mathcal{Q}(\Psi|\hat{\Psi})$ using quasi-Newton [79].

We compute the gradients of $\mathcal{Q}(\Psi|\hat{\Psi})$ w.r.t ϕ_z, x_i, ρ_{sqr}, η_{sqr} respectively as follows.

$$\begin{aligned}
\frac{\partial Q}{\partial \phi_z} &= \sum_{i=1}^{N}\sum_{m=1}^{M_i} \big(\mathrm{P}(z|x_i, \Phi) - \mathrm{P}(z|i, m, \hat{\Psi})\big)(\phi_z - x_i) - \varphi\phi_z \\
\frac{\partial Q}{\partial x_i} &= \sum_{m=1}^{M_i}\sum_{z=1}^{Z} \big(\mathrm{P}(z|x_i, \Phi) - \mathrm{P}(z|i, m, \hat{\Psi})\big)(x_i - \phi_z) - \gamma x_i \\
&\quad - \sum_{e_{ij}\in E} 4\eta_{sqr}^2(x_i - x_j) \\
&\quad + \frac{4\rho_{sqr}^2\eta_{sqr}^2}{U} \sum_{e_{ij}\notin E} (x_i - x_j)\frac{\exp(-\eta_{sqr}^2 \cdot ||x_i - x_j||^2)}{(1 - \exp(-\eta_{sqr}^2 \cdot ||x_i - x_j||^2))} \\
\frac{\partial Q}{\partial \rho_{sqr}} &= \frac{2\rho_{sqr}}{U} \sum_{e_{ij}\notin E} \log(1 - \exp(-\eta_{sqr}^2 \cdot ||x_i - x_j||^2)) \\
\frac{\partial Q}{\partial \eta_{sqr}} &= -2\eta_{sqr} \sum_{e_{ij}\in E} ||x_i - x_j||^2 \\
&\quad + \frac{2\rho_{sqr}^2\eta_{sqr}}{U} \sum_{e_{ij}\notin E} ||x_i - x_j||^2 \frac{\exp(-\eta_{sqr}^2 \cdot ||x_i - x_j||^2)}{(1 - \exp(-\eta_{sqr}^2 \cdot ||x_i - x_j||^2))}
\end{aligned}$$

4.5 Experiments

The objective of experiments is to validate the effectiveness of our topic-based embedding method PLANE. First, we describe the setup, in terms of the datasets (4.5.1)

as well as the comparable baselines (4.5.2). Thereafter, we conduct the primary comparison in terms of the goodness of embedding coordinates (4.5.3). This is done both quantitatively by using the coordinates as features in a classification task, as well as qualitatively by inspecting some example visualizations. Finally, we compare the effectiveness of PLANE as a topic model for document network (4.5.4).

4.5.1 Datasets

For repeatability, we rely on a publicly-available benchmark data source, which is a representative example of document networks. Cora[1] is a collection of academic publications and their citation networks from various categories [85]. Each document is an abstract. Two documents are connected by an undirected edge if one document cites the other. Documents in Cora are divided into general categories. Following [130], we use the following categories as four separate datasets: *Data Structure* (DS), *Hardware and Architecture* (HA), *Machine Learning* (ML), and *Programming Language* (PL).

For each dataset, each document is further classified into one of several sub-fields. For DS, the nine sub-fields are: *Computational Complexity*, *Computational Geometry*, *Formal Languages*, *Hashing*, *Logic*, *Parallel*, *Quantum Computing*, *Randomized*, and *Sorting*. The other three datasets each have their own respective sub-fields as well. We treat these sub-fields as class labels, which are not used as input, but rather for evaluation in Section 4.5.3. We also remove documents that are not connected to any document within the same dataset.

Table 4.1 lists the sizes of these datasets in terms of the number of classes, documents, edges, and the vocabulary sizes.

4.5.2 Comparative Methods

In Table 4.2, we list the methods that we will be comparing, and highlight the properties of each method.

[1] `http://people.cs.umass.edu/~mccallum/data/cora-classify.tar.gz`

Table 4.1: Datasets of Cora

	#classes	#documents	#edges	vocabulary
Data Structure (DS)	9	570	1336	3085
Hardware and Architecture (HA)	6	223	515	2073
Machine Learning (MA)	7	1980	5638	4431
Programming Language (PL)	9	1553	4851	4105

Table 4.2: Comparative Methods

	Document embedding	Network embedding	Topic model	Joint model
PLANE	✓	✓	✓	✓
RTM+PE	✓	✓	✓	
PLSV	✓		✓	✓
KK		✓		
SVD		✓		

Proposed approach. As a topic-based embedding model, **PLANE** is our method that models both document and network embeddings, as well as topic model in a joint manner.

Pipelined approach. Since there is no other existing model with all the properties, the most direct baseline is a composite approach that pipelines two methods. First, a document network is reduced into a set of topic distributions (one for each document) by the relational topic model RTM [24]. As recommended in [24], α is set such that the total mass of the Dirichlet hyperparameter is 5. ζ is set to 0.01 (same as PLANE and PLSV) following [62]. Then, these topic distributions are embedded in a 2D visualization space using PE [61], an embedding approach designed for probability distributions. This composite, called **RTM+PE**, is our primary baseline that allows us to validate the utility of modeling both topics and embedding jointly, as opposed to modeling them separately.

Document embedding. While document embedding is not a direct baseline, because it does not model the network aspect, a comparison to it allows us to evaluate the contribution of network embedding to our model. As a representative of document embedding, the closest one to ours is **PLSV** [62], which models topic-based document embedding.

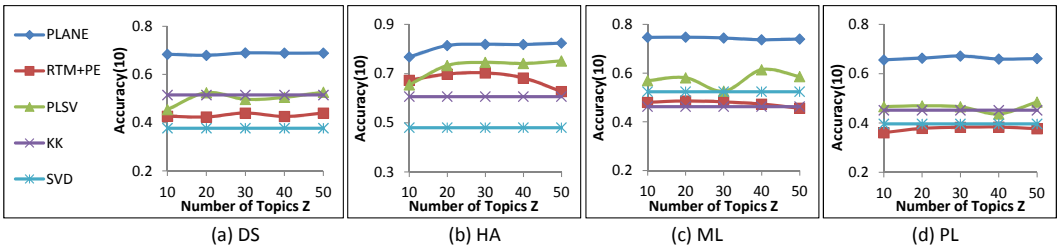


Figure 4.2: Accuracy at $k = 10$ nearest neighbors for varying number of topics Z

Network embedding. Network embedding is not a direct baseline either, because it models neither documents nor topics. For completeness, we include a comparison to two categories of network embedding. As a representative of spring embedding, we use **KK** layout [66]. As a representative of spectral embedding, we use **SVD** [48]. These are among the most popular methods in their respective categories.

For the probabilistic methods (i.e., PLANE, RTM+PE, PLSV), we average the performance numbers across ten independent runs. For each run, the parameter estimation is based on 100 learning iterations. We set the number of iterations for each Gibbs sampling E-step of RTM to 1000. As much as possible, we have used public implementations. For RTM, we use its original authors' implementation[2]. For KK, we use the implementation in the JUNG library[3]. For SVD, we use the implementation in R software[4]. We implement our own method PLANE, as well as the baselines PE and PLSV[5].

4.5.3 Embedding

As our primary objective is to embed a document network in a low-dimensional space, we first evaluate the quality of the resulting embedding coordinates against all the baselines.

Metric. Since embedding seeks to "preserve" the original data as much as possible in the reduced dimensions, one well-accepted means for embedding evaluation

[2] `http://cran.r-project.org/web/packages/lda/`
[3] `http://jung.sourceforge.net/`
[4] `http://stat.ethz.ch/R-manual/R-devel/library/base/html/svd.html`
[5] We could not find a public or an original implementation by their authors.

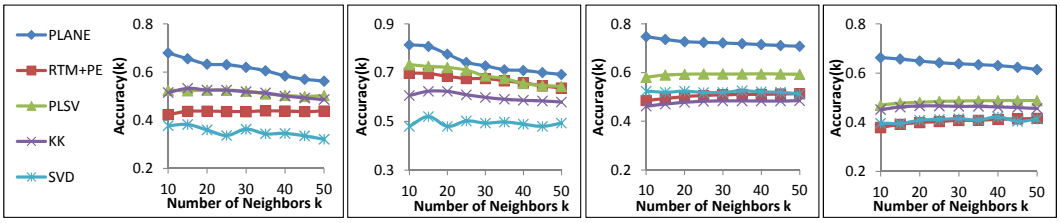


Figure 4.3: Accuracy at varying k nearest neighbors for $Z = 20$ topics

is to use the low-dimensional coordinates as features in a learning task [62, 108]. Since class labels are available (but not used as input), we conduct evaluation based on classification. The more the features help to predict the classes, the more the low-dimensional coordinates (features) have preserved the properties of the data (embedding objective). Because what is evaluated are the features, we use a simple k-nearest neighbor classification. For each document, we hide its true label, and predict its label as the majority label among its k-nearest neighbors (based on Euclidean distance in the embedding coordinates). The metric $accuracy(k)$ is the fraction of documents for which the predicted label matches the hidden true label.

Vary number of topics. First, we investigate the effect of the number of topics Z on accuracy. Figure 4.2 shows the $accuracy(10)$ values for the four datasets. Similar observations regarding the relative standings of various methods can be made for other k values as well.

The accuracy values are relatively stable across different numbers of topics. Figure 4.2(b) for HA shows a small increase from $Z = 10$ to $Z = 20$, after which accuracies remain flat. For subsequent experiments, we will use $Z = 20$ by default.

In absolute terms, PLANE achieves high accuracies of around 0.8 for HA, and 0.7 for DS, ML, and PL. This is notable as PLANE only uses 2-dimensional features for the k-NN classification. This helps to validate the quality of the embedding in preserving the high-dimensional representations.

In relative terms, PLANE has higher accuracies than all the baselines. This outperformance is statistically significant in all cases. It outperforms RTM+PE, which helps to validate the utility of having a joint modeling of embedding and topics.

It also outperforms document embedding (PLSV) and network embedding (KK, SVD), which justifies embedding documents and network with a unified low-rank representation.

Among the baselines themselves, there is no consistent ordering across datasets in terms of which is better. For the network embedding KK and SVD, accuracies are flat across different Z's because they are not topic-based approaches.

Vary neighborhood size. We now investigate how the accuracy is affected by different neighborhood sizes for the k-NN classification. Figure 4.3 shows the $accuracy(k)$ values for the four datasets when $Z = 20$. As shown by the earlier consistency among different Z's, similar observations can be made for other number of topics as well. For all the methods, there is a general tendency that accuracy decreases at larger k's. This is reasonable, because as k increases, we use a greater number of neighbors to arrive at the classification, which dilutes the quality of classification. Importantly, in relative terms, the outperformance by PLANE still stands across different k's, for the reasons explained above.

Visualization. To gain a sense of the visualization quality obtained by embedding the documents in a two-dimensional scatterplot, we show several examples for the various datasets.

We begin with the Data Structure (DS) dataset. Figure 4.4(a) shows the visualization generated by PLANE. Each document is a dot placed in the scatterplot according to their 2D embedding coordinates. Each dot is painted with a color that represents its sub-field or class. The legend specifies the colors assigned to each class. Edges are lines between two connected documents. There are two key observations. First, note how the different classes are quite well-separated from one another (the class information itself was never used for learning). The red *Parallel* documents are at the lower right, while the grey *Sorting* documents are at the center. Second, note how the edges are hardly visible, which is a good sign because it means connected documents are placed as close neighbors in the visualization space. Otherwise, we would have witnessed criss-crossing lines all over. These

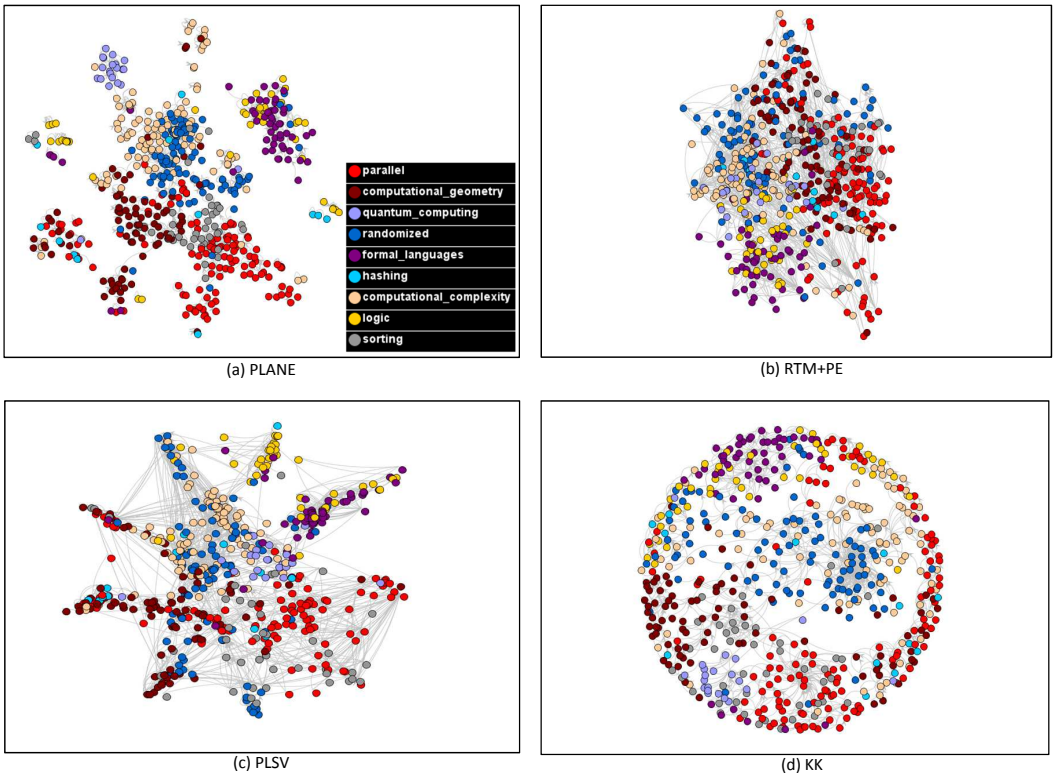


Figure 4.4: Visualizations of Data Structure (DS) dataset for $Z = 20$ (best seen in color)

observations support the hypothesis that having a joint model for embedding documents and network results in better embedding overall.

Still for the DS dataset, Figure 4.4(b) for RTM+PE does not show a good separation between classes, and has many criss-crossing edge lines. This is because while the network is used to influence the topic distributions, because of the disjoint embedding through PE, the network effect does not get enforced in the embedding process. PLSV in Figure 4.4(c) looks more coherent than RTM+PE, but not as clean as PLANE. For one thing, the grey *sorting* documents are spread apart, while in PLANE they are clustered together. For another thing, there are still criss-crossing edges due to separation of connected documents as PLSV models text content only. In contrast, KK in Figure 4.4(d) models only network embedding. Thus connected edges are tightly clustered together. However, because it does not model content, documents of the same class without connection to each other are spread far apart (e.g., the red *parallel*). Due to space constraint, here we do not show SVD (which

has the lowest accuracy for DS in Figure 4.3(a)).

To show that the observations for PLANE apply to other datasets as well, in Figure 4.5, we show PLANE's visualization for HA, ML, and PL datasets. Evidently, PLANE can group together documents of the same class well, and place connected documents as neighbors in the visualization space.

4.5.4 Topic Modeling

While our main objective is to improve the embedding of document networks, it is important to ensure that the gains in embedding and visualization quality have not come at the expense of the topic model. Since ours is a topic model for a document network, the appropriate comparison is to a baseline that also models the generation of both words and links, namely RTM [24]. In the following, we compare PLANE and RTM, in terms of the topic words, as well as the links.

Topic Interpretability

As modeling topics with embedding is to improve the interpretability of embedding, we evaluate the topics on how interpretable the topic words are.

Metric. Pointwise Mutual Information (PMI) is an established measure for how coherent the top words in a topic are [92]. PMI for two words w_i and w_j is defined in Equation 4.6.

$$\text{PMI}(w_i, w_j) = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)} \tag{4.6}$$

PMI uses an external corpus to estimate $p(w_i, w_j)$ and $p(w_i)$. As in [92], we use *Google Web 1T 5-gram Version 1* [15], a corpus of n-grams generated from 1 trillion word tokens. $p(w_i)$ is estimated from the frequencies of 1-grams. $p(w_i, w_j)$ is estimated from the frequencies of 5-grams. For each topic, we average the pairwise PMI's among the topic's top 10 words. For each model, we average the topic-level PMI's. Higher PMI indicates that the words in a topic are correlated, and the topic is more coherent and interpretable.

Table 4.3: PMI Scores for Topic Interpretability ($Z = 20$)

	DS	HA	ML	PL	*Average*
PLANE	0.59	0.53	0.43	0.51	*0.51*
RTM	0.54	0.48	0.51	0.50	*0.50*

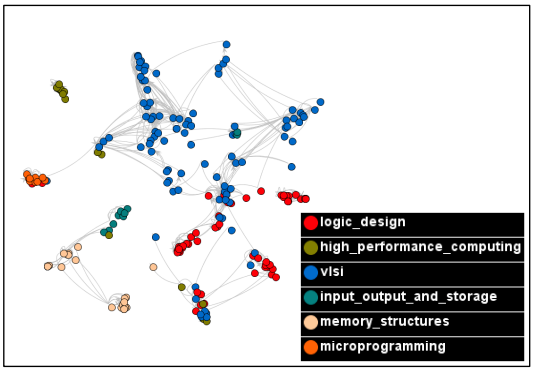
PMI Scores. Table 4.3 shows the PMI scores for the four datasets for $Z = 20$ topics. The figures for other numbers of topics are consistent as well. Averaging across the four datasets, PLANE and RTM have very similar PMI's of around 0.5. This suggests that PLANE is at least not inferior to RTM, even with the constraint of modeling embedding coordinates. This shows a great promise by PLANE in enriching the visualization with coherent semantic interpretability.

Link Generation Probability

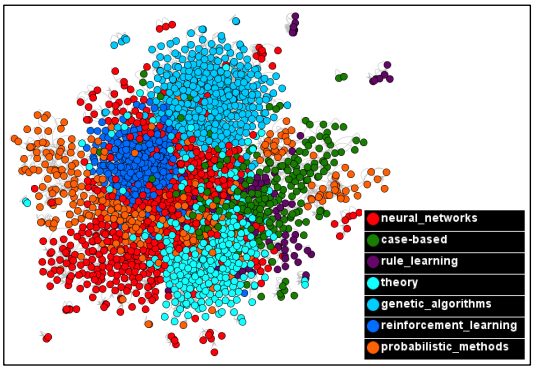
In addition to words, both PLANE and RTM also model edges or links. In order to evaluate their effectiveness in modeling link generation, we compare the two methods in terms of link prediction. Note that this is confined to an evaluation task, and our goal is not to propose or compare to state-of-the-art link prediction methods.

For each document (with at least three links), we randomly hide one link. In total, we have around 13%-14% of all links hidden. The task is thus to predict these hidden links based on the observations on the texts and the remaining links. To estimate these hidden links, for PLANE, we use the document coordinates to compute the probability of a hidden link according to Equation 4.2. For RTM, we compute the probability of a hidden link according to [24], which shares a comparable exponential link probability function but based on topic distributions (instead of latent coordinates).

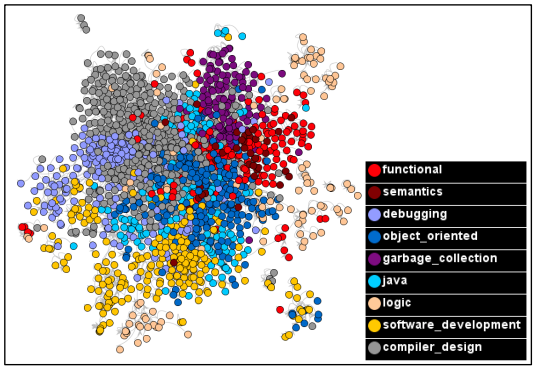
Metric. One possibility is to compute the likelihood of generating these hidden links. However, this may not be an appropriate measure, because we will be computing only the likelihood of some links being present (but not of links being absent), thus favoring a model that simply produces higher probability values across the board for all possible links. For instance, consider how in Equation 4.2, one can



(a) Hardware and Architecture (HA)



(b) Machine Learning (ML)



(c) Programming Language (PL)

Figure 4.5: PLANE's Visualizations for $Z = 20$ (best seen in color)

Table 4.4: MRR Scores for Link Prediction ($Z = 20$)

	DS	HA	ML	PL	*Average*
PLANE	0.328	0.207	0.194	0.219	*0.237*
RTM	0.005	0.009	0.0001	0.002	*0.004*

produce a higher likelihood simply with a lower η, even while keeping all x_i and x_j's the same, which is inappropriate because the model complexity is in deriving the coordinates to determine which documents should (or should not) be neighbors.

Therefore, a more appropriate metric is to evaluate whether the model assigns a higher probability to the hidden link (which is factually present, though not used for learning) than to other unobserved links. For each document with a hidden link, we rank all the unobserved links of this document in terms of their generation probabilities. The highest rank is 1. Intuitively, the hidden link is expected to have a rank as close to 1 as possible, because it is indeed a factual link that was simply hidden from the model. We borrow a metric from information retrieval, called mean reciprocal rank [34] or MRR, which is defined in Equation 4.7, where E' is the set of hidden links, and $\mathrm{rank}(\mathrm{e_{ij}})$ is the ranking of the hidden link e_{ij} among the unobserved links of the document from which it is hidden. The higher is the MRR of a method, the better is the method at placing the hidden links in the high ranks.

$$\mathrm{MRR} = \frac{1}{|E'|} \sum_{e_{ij} \in E'} \frac{1}{\mathrm{rank}(\mathrm{e_{ij}})} \tag{4.7}$$

MRR Scores. Table 4.4 shows the MRR scores for the four datasets for $Z = 20$ topics. The figures for other numbers of topics are consistent as well. We see that PLANE produces significantly higher MRR scores than RTM across all the datasets. Averaging across the datasets, PLANE has a score of 0.237, which implies that it generally places the hidden links in the top 5 in terms of link generation probability. In contrast, RTM's score of 0.005 implies that the hidden links tend to be placed around the two hundredths' rank positions.

We attribute PLANE's higher performance in this task to the way we infer the parameters of the model. As discussed in Section 4.3, by modeling some amount of "virtual" negative links we force the model to discriminate between close neighbors

(more likely to be positive links) and distant documents (more likely to be negative links). In contrast, by modeling only positive links, RTM is not as able to sharply discriminate genuine neighbors from unrelated documents. The trade-off is that PLANE requires more run time than RTM, because the former models both positive as well as "virtual" negative links, whereas RTM models positive links only (of which there are relatively few in a sparse network).

4.6 Conclusion

We address the problem of embedding a document network's high-dimensional representations in terms of text and network connectivity in a low-dimensional space. We formulate this as a generative model tying together the various representations of a document (words, links, topics, and coordinates), which we call PLANE. Through comprehensive experiments on four real-life datasets extracted from the Cora collection, we show that it outperforms existing baselines in topic modeling, document embedding, and network embedding, especially in terms of the quality of embedding coordinates (as features in classification and scatterplot visualization). For future work, we plan to consider extensions such as generalizing to directed graph, and pursuing computational optimizations such as hyper-threading or parallel processing.

Part II

Modeling Document Representation

Chapter 5

Modeling Spherical Representation

In this chapter, we address the semantic visualization problem. Given a corpus of documents, the objective is to simultaneously learn the topic distributions as well as the visualization coordinates of documents. We propose to develop a semantic visualization model that approximates L^2-normalized data directly. The key is to associate each document with three representations: a coordinate in the visualization space, a multinomial distribution in the topic space, and a directional vector in a high-dimensional unit hypersphere in the word space. We join these representations in a unified generative model, and describe its parameter estimation through variational inference. Comprehensive experiments on real-life text datasets show that the proposed method outperforms the existing baselines on objective evaluation metrics for visualization quality and topic interpretability.

5.1 Introduction

In this chapter, we propose a semantic visualization model for data with *spherical representation*. This refers to data whose instances can each be represented as a vector of unit length in a high-dimensional hypersphere [4], with dimensionality commensurate with the number of features. In other words, we are dealing with L^2-normalized feature vectors as input. One important category of such data that we focus on in this work is text document. A document can be naturally repre-

sented as a normalized term vector, as done in the classical vector space model [105]. Stated more formally, the input to the problem is a corpus of documents $\mathcal{D} = \{d_1, d_2, \ldots, d_N\}$, where every d_n is represented by an L^2-normalized term vector ν_n. We seek to learn, for each d_n, a probability distribution θ_n over Z topics (*semantic*), and a coordinate x_n on a low-dimensional space (*visualization*). While we frame the discussion here in terms of documents and words, our technique is applicable to other data types for which both visualization and semantic interpretability are important, as long as they can be expressed in terms of spherical representation.

Previous Approach. Jointly modeling topics and visualization coordinates is pioneered by PLSV [62] (reviewed briefly in Section 2.1.1). It is aimed at *dyadic data*, whereby every observation involves a couple (d, w) of word w's occurrence in document d. The observations for a document can be summarized as an integer vector of word counts in $\mathbb{N}^{|W|}$, where W is the vocabulary. Like its topic modeling predecessors [14, 57], PLSV uses the word count vectors to maximize the likelihood of generating individual words based on the learned latent multinomial distribution over words $\{\mathrm{P}(w|d_n)\}_{w \in W}$. Here, $\mathrm{P}(w|d_n)$ is obtained from topics' word distribution $\mathrm{P}(w|z)$ and document's topic distribution $\mathrm{P}(z|d_n)$, i.e., $\mathrm{P}(w|d_n) = \sum_{z=1}^{Z} \mathrm{P}(w|z)\mathrm{P}(z|d_n)$.

The stated aim of most visualization approaches is to recover a low-dimensional manifold embedded within the high-dimensional space of the original data [55, 70, 104, 117]. Key to manifold learning is the capacity for approximating the similarities and differences among data instances [9]. In this respect, multinomial modeling of dyadic data has a couple of downsides [101]. For one thing, it primarily models word presences, but does not directly model word absences. The likelihood of a document is defined over only words present in the document. For another thing, it is also sensitive to document lengths. If one document were to contain two copies of each word in another document, the two documents would have different likelihoods, even though the word distributions in the two documents are effectively identical.

Proposed Approach. Spherical representation could address the above-mentioned issues, leading towards better approximation of similarities among documents, and thus towards better manifold learning and visualization. In the spherical space, relationships between documents are measured as cosine similarity $\in [0, 1]$, which is the angular distance between two directional unit vectors. Firstly, two documents would have higher cosine similarity, not only if some words in common are present, but also if some other words in common are absent. Secondly, the normalization of all documents to unit vectors effectively neutralizes the impact of document lengths. Moreover, there is indicative evidence from the literature that a spherical approach will be promising in terms of dimensionality reduction. For instance, the spherical topic model SAM [101] performs significantly better than the multinomial topic model LDA [14], when used as a dimensionality reduction technique.

There are further advantages to spherical representation. For one thing, there is a greater degree of *flexibility* in admitting different L^2-normalized representations, e.g., term frequency *tf* or *tf-idf* or other feature vectors. For another thing, there is a greater degree of *expressiveness*, as an L^2-normalized vector can have both positive and negative elements, representing the degrees of word presences and absences respectively. Inspired by [101], this expressiveness engenders a change in the topic definition, from multinomial word distribution to a unit term vector. Given a topic, we no longer associate a word with a probability value, but rather with a real value that expresses the word's presence or absence (the sign) and relative importance (the weight).

Contributions. Our problem formulation is novel because to the best of our knowledge, we are the first to address semantic visualization for spherical representation (*first contribution*). We propose a generative model called SSE, which stands for *S*pherical *S*emantic *E*mbedding. In Section 5.2.1, we develop the full generative process of SSE (*second contribution*). To learn its parameters, we describe an estimation based on variational inference in Section 5.2.2 (*third contribution*). In Section 5.3, we validate SSE through experiments on publicly available real-life

datasets, showing significant gains in visualization quality and topic interpretability (*fourth contribution*). We conclude in Section 5.4.

5.2 Spherical Semantic Embedding

5.2.1 Generative Model

Document Representations. We associate each document with representations in three different spaces.

- We model the *visualization space* as a Cartesian plane, where relationships can be visualized spatially in terms of Euclidean distances. This space is low-dimensional, and without loss of generality, we assume it has two dimensions (2D). Each document d_n is associated with 2D coordinates x_n. This is consistent with visualization techniques oriented towards dimensionality reduction [55, 117].
- We model the *topic space* as a $(Z-1)$-simplex, where Z is the number of topics. This is consistent with the practice in most topic models [14, 57, 101]. Each document d_n occupies a point θ_n in the simplex, which codes for a multinomial distribution over the topics $\{\mathrm{P}(z|d_n)\}_{z=1}^{Z}$.
- We model the *word space* as a $(|W|-1)$-dimensional unit sphere in $\mathbb{R}^{|W|}$, where W is the vocabulary. Each document d_n is associated with a directional, unit-length vector ν_n. For instance, ν_n could be a *tf-idf* vector, or other L^2-normalized vector. This is consistent with the vector space model [105], and spherical models [4, 101].

Of the three representations of d_n, only ν_n is observed, while x_n and θ_n are latent. A key step towards integrating visualization and topic modeling is to define a mapping between the spaces to ensure a consistency among the representations. In defining the mapping, we associate each topic z with representations in both the visualization space ϕ_z and the word space τ_z. The coordinate ϕ_z reveals where

a topics is in the visualization space, allowing users to observe the relationships between documents and topics. The word vector τ_z reveals the topic semantics in terms of the relative importance of various words within τ_z.

Visualization Space to Topic Space. As both documents and topics have coordinates in the visualization space, their relationship can be expressed in terms of distances $||x_n - \phi_z||$. Intuitively, the closer is x_n to a topic's ϕ_z, the higher is $\theta_{n,z}$ or the probability of topic z for document d_n. One framework to relate variables based on distances is Radial Basis Function or RBF [16], which defines a function $\lambda(||x_n - \phi_z||)$ in terms of how far a data point (e.g., x_n) is from a center (e.g., ϕ_z). The function λ may take on various forms, e.g., Gaussian, multi-quadric, polyharmonic spline.

RBF network [13] is frequently used to build a function approximation. We use an RBF network as a "kernel" for the mapping between coordinates and topic distributions. To express θ_n as a function of x_n, we consider the normalized architecture of RBF network, with three layers. The input layer consists of one input node (x_n). The hidden layer consists of Z number of normalized RBF activation functions. Each is centered at ϕ_z and computes $\frac{\lambda(||x_n-\phi_z||)}{\sum_{z'=1}^{Z}\lambda(||x_n-\phi_{z'}||)}$. The linear output layer consists of Z output nodes. Each output node $y_z(x_n)$ corresponds to $\theta_{n,z}$, which is a linear combination of the RBF functions, as shown in Equation 5.1. Here, $w_{z,z'}$ is the weight of influence of the RBF function of z' on the $\theta_{n,z}$, with the constraint $\sum_{z'=1}^{Z} w_{z,z'} = 1$.

$$\theta_{n,z} = y_z(x_n) = \frac{\sum_{z'=1}^{Z} w_{z,z'} \cdot \lambda(||x_n - \phi_{z'}||)}{\sum_{z'=1}^{Z} \lambda(||x_n - \phi_{z'}||)} \tag{5.1}$$

While Equation 5.1 is the general form, to instantiate a specific mapping function, we need to determine both the assignment of $w_{z,z'}$ and the form of the function λ. In this work, we will experiment with a special case (λ is Gaussian and $w_{z,z'} = 1$ when $z = z'$ and 0 otherwise), which yields the function in Equation 5.2, where Φ refers to the collective set of ϕ_z's. This specific function has appeared previously in the baseline [62] that we will compare to, and this design decision helps to estab-

lish parity for comparative purposes. In future work, we will explore other function instantiations.

$$\theta_{n,z} = \mathrm{P}(z|x_n, \Phi) = \frac{\exp(-\frac{1}{2}||x_n - \phi_z||^2)}{\sum_{z'=1}^{Z} \exp(-\frac{1}{2}||x_n - \phi_{z'}||^2)} \tag{5.2}$$

Topic Space to Word Space. For d_n, we also need to bridge θ_n to its word space representation ν_n. As introduced previously, each topic z also has a word space representation τ_z. Because θ_n is essentially a topic distribution, we adopt a similar practice as in conventional topic model, which represents a document's word distribution as a weighted average (based on topic distribution) of the topics' word distributions. In our context, it means taking a weighted average of the topics' spherical unit vectors τ_z's, weighted by $\theta_{n,z}$, followed by L^2-normalization to return the mean vector to unit length, i.e., $\tau_n = \frac{\sum_{z=1}^{Z} \theta_{n,z} \cdot \tau_z}{||\sum_{z=1}^{Z} \theta_{n,z} \cdot \tau_z||}$.

To avoid overfitting, instead of equating ν_n to τ_n, we assume a probabilistic process where ν_n is drawn from a distribution centered at τ_n. Because ν_n and τ_n are both directional vectors, we turn to directional statistics [83]. In particular, von Mises-Fisher (vMF) distribution [84] was previously used to model documents [4, 101]. Equation 5.3 specifies the probability density function (p.d.f.) for a random unit vector ν, given mean directional vector μ, and concentration parameter κ. Note how the p.d.f. is parameterized by the cosine similarity $\mu^T \nu$ between the mean direction μ and ν, which is effectively the angular distance between the two unit vectors. The higher the κ, the more concentrated the distribution is around μ. The distribution is unimodal for $\kappa > 0$, and is uniform for $\kappa = 0$. C_D is the normalization constant, defined in Equation 5.4, where I_r denotes the modified Bessel function of the first kind and order r.

$$\mathrm{vMF}(\nu; \mu, \kappa) = C_D(\kappa) \exp(\kappa \mu^T \nu) \tag{5.3}$$

$$C_D(\kappa) = \frac{\kappa^{D/2-1}}{(2\pi)^{D/2} I_{D/2-1}(\kappa)} \tag{5.4}$$

We can then express ν_n as a draw from a vMF distribution with mean direction τ_n, i.e., $\nu_n \sim \mathrm{vMF}(\tau_n, \kappa)$.

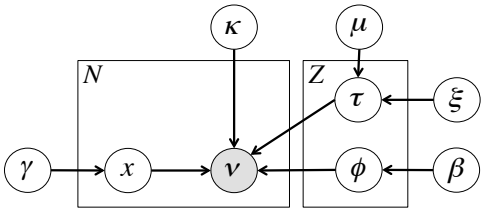


Figure 5.1: Graphical Model of SSE

Generative Process. We join the three representations into a generative model, with graphical representation as in Figure 5.1. The generative process of SSE is as follows:

1. Draw the corpus mean direction: $\mu \sim \mathrm{vMF}(m, \kappa_0)$
2. For each topic $z = 1, \ldots, Z$:
   - Draw z's coordinate: $\phi_z \sim \mathrm{Normal}(0, \beta^{-1} I)$
   - Draw z's spherical direction: $\tau_z \sim \mathrm{vMF}(\mu, \xi)$
3. For each document d_n, where $n = 1, \ldots, N$:
   - Draw d_n's coordinate: $x_n \sim \mathrm{Normal}(0, \gamma^{-1} I)$
   - Derive d_n's topic distribution: $\theta_{n,z} = \mathrm{P}(z|x_n, \Phi) = \frac{\exp(-\frac{1}{2}||x_n - \phi_z||^2)}{\sum_{z'=1}^{Z} \exp(-\frac{1}{2}||x_n - \phi_{z'}||^2)}$
   - Derive d_n's spherical average: $\tau_n = \frac{\sum_{z=1}^{Z} \theta_{n,z} \cdot \tau_z}{||\sum_{z=1}^{Z} \theta_{n,z} \cdot \tau_z||}$
   - Draw d_n's spherical direction: $\nu_n \sim \mathrm{vMF}(\tau_n, \kappa)$

In Step 1, we draw the corpus mean direction μ. In Step 2, we draw, for each topic, a visualization coordinate ϕ_z and a spherical direction τ_z. In Step 3, we draw, for each document, a visualization coordinate x_n, which we use to compute topic distribution θ_n as a function of document and topics' coordinates. θ_n together with different topics' τ_z's are used to compute the weighted average of topics' directions, denoted τ_n. After normalizing τ_n to a unit-length vector, we draw ν_n from a vMF with mean τ_n. Though the observed ν_n is usually positive (e.g., *tf-idf*), the latent τ_n may contain negative elements, which reflect unlikely words.

5.2.2 Parameter Estimation

To estimate the parameters in SSE, we employ variational EM with maximum a posteriori (MAP) estimation. The unknown parameters are the coordinates for documents (collectively $\chi = \{x_n\}$) and for topics (collectively $\Phi = \{\phi_z\}$), the directional vectors for topics (collectively $\mathcal{T} = \{\tau_z\}$) and the hyperparameters ξ, m. Given a corpus $\mathcal{D}$, which are represented as L^2-normalized term vectors $\mathcal{V} = \{\nu_n\}_{n=1}^N$, we infer the posterior distribution $\mathrm{P}(\mathcal{T}, \mu|\mathcal{V}, \chi, \Phi, \beta, \gamma, \xi, m, \kappa_0, \kappa)$ of the directional vectors for topics (collectively $\mathcal{T} = \{\tau_z\}$) and the corpus mean direction μ.

We approximate the posterior using the following variational distribution:

$$q(\mathcal{T}, \mu|\tilde{\mu}, \xi) = q(\mathcal{T}|\tilde{\mu}, \xi)q(\mu|\tilde{m}, \kappa_0)$$

where $q(\tau_z) = \mathrm{vMF}(\tau_z|\tilde{\mu}, \xi)$, $q(\mu_z) = \mathrm{vMF}(\mu_z|\tilde{m}_z, \kappa_0)$ and the variational parameters are $\tilde{\mu}, \tilde{m}$. Given this variational distribution q, we have a lower bound $\mathcal{L}(\tilde{\mu}, \tilde{m})$ on the log likelihood with priors over the document and topic visualization coordinate x_n, ϕ_z, as follows:

$$\begin{aligned}
\mathcal{L}(\tilde{\mu}, \tilde{m}) &= \mathbb{E}_q\big[\log p(\mathcal{V}, \mathcal{T}, \mu)\big] - \mathbb{E}_q\big[\log q(\mathcal{T}, \mu|\tilde{\mu}, \xi)\big] \\
&\quad + \sum_{n=1}^{N} \log p(x_n) + \sum_{z=1}^{Z} \log p(\phi_z) \\
&= \mathbb{E}_q\big[\log p(\mathcal{V}|\mathcal{T}, \chi, \Phi)\big] + \mathbb{E}_q\big[\log p(\mathcal{T}|\mu, \xi)\big] \\
&\quad + \mathbb{E}_q\big[\log p(\mu)\big] - \mathbb{E}_q\big[\log p(\mathcal{T}|\tilde{\mu}, \xi)\big] - \mathbb{E}_q\big[\log p(\mu|\tilde{m}, \kappa_0)\big] \\
&\quad + \sum_{n=1}^{N} \log p(x_n) + \sum_{z=1}^{Z} \log p(\phi_z)
\end{aligned}$$

In the E-step, we optimize the lower bound $\mathcal{L}(\tilde{\mu}, \tilde{m})$ with respect to the variational parameters $\tilde{\mu}, \tilde{m}$. In the M-step, the lower bound is optimized with respect to the parameters χ, Φ, ξ, m. We alternate E and M-steps until some appropriate convergence criterion is reached. We use gradient-based numerical optimization method such as the quasi-Newton method to update $\tilde{\mu}, \chi, \Phi, \xi$.

E-step. Let $\rho_n = \mathbb{E}\big[\tau_n\big]^T \nu_n$ where $n \in \{1...N\}$ ranges over the documents.

Taking the gradients of $\mathcal{L}(\tilde{\mu}, \tilde{m})$ w.r.t $\tilde{\mu}$, we have:

$$\bigtriangledown_{\tilde{\mu}_z}\mathcal{L} = A_W(\xi)A_W(\kappa_0)\xi\tilde{m}_z + \kappa\sum_{n=1}^{N}\bigtriangledown_{\tilde{\mu}_z}\rho_z$$

where $A_p(c)$ denotes the mean resultant length of a vMF distribution of dimension p with concentration c. Since $\mathbb{E}\big[\tau_n\big]$ does not have a closed form, following [101] we approximate it as:

$$\mathbb{E}\big[\tau_n\big] \approx \mathbb{E}\big[\sum_{z=1}^{Z}\theta_{n,z}\cdot\tau_z\big]\mathbb{E}\big[||\sum_{z=1}^{Z}\theta_{n,z}\cdot\tau_z||^2\big]^{-1/2}$$

We refer to $\mathbb{E}\big[||\sum_{z=1}^{Z}\theta_{n,z}\cdot\tau_z||^2\big]$ as S_n. ρ_n will be approximated as:

$$\rho_n \approx A_W(\xi)S_n^{-1/2}(\tilde{\mu}\theta_n)^T\nu_n$$

where

$$S_n = (1 - A_W(\xi)^2)\sum_z\theta_{nz}^2 + A_W(\xi)^2||\tilde{\mu}\theta_n||^2$$

Taking the gradients of ρ_n w.r.t $\tilde{\mu}_j$, yields:

$$\bigtriangledown_{\tilde{\mu}_j}\rho_n = A_W(\xi)\big(\frac{\theta_{n,j}\nu_n}{\sqrt{S_n}} - \frac{(\tilde{\mu}\theta_n)^T\nu_n}{2S_n^{3/2}}.\bigtriangledown_{\tilde{\mu}_j}S_n\big)$$

where

$$\bigtriangledown_{\tilde{\mu}_j}S_n = 2A_W(\xi)^2\theta_{n,j}\tilde{\mu}\theta_n$$

The variational corpus mean $\tilde{m}$ has a closed form update rule:

$$\tilde{m} \propto \kappa_0 m + A_W(\xi)\xi\sum_{z=1}^{Z}\tilde{\mu}_z$$

M-step. In the M-step, taking gradients of $\mathcal{L}(\tilde{\mu}, \tilde{m})$ w.r.t ξ, we have:

$$\bigtriangledown_{\xi}\mathcal{L} = (\bigtriangledown_{\xi}A_W(\xi)\xi + A_W(\xi))(A_W(\kappa_0)\tilde{m}^T\sum_z\tilde{\mu}_z - Z) + \kappa\sum_{n=1}^{N}\bigtriangledown_{\xi}\rho_n$$

where

$$\bigtriangledown_\xi \rho_n = \big(\bigtriangledown_\xi A_W(\xi) S_n^{-1/2} - \frac{1}{2} A_W(\xi) S_n^{-3/2} \bigtriangledown_\xi S_n \big) (\tilde{\mu}\theta_n)^T \nu_n$$

and

$$\bigtriangledown_\xi S_n = 2A_W(\xi) \bigtriangledown_\xi A_W(\xi)(||\tilde{\mu}\theta_n||^2 - \sum_z \theta_{nz}^2)$$

The corpus mean m has a closed form update rule as follows:

$$m \propto \sum_z \tilde{\mu}_z$$

Taking the gradients of $\mathcal{L}(\tilde{\mu}, \tilde{m})$ w.r.t x_n, we have:

$$\bigtriangledown_{x_n} \mathcal{L} = \kappa A_W(\xi) \big(- \frac{\bigtriangledown_{x_n} S_n}{2S_n^{3/2}} \tilde{\mu}\theta_n + \frac{\tilde{\mu} \bigtriangledown_{x_n} \theta_n}{\sqrt{S_n}} \big)^T \nu_n - \gamma x_n$$

where

$$\bigtriangledown_{x_n} S_n = 2(1 - A_W(\xi)^2) \sum_z \bigtriangledown_{x_n} \theta_{nz} \theta_{nz} + 2A_W(\xi)^2 \theta_n^T \tilde{\mu}^T \tilde{\mu} \bigtriangledown_{x_n} \theta_n$$

Taking the gradients of $\mathcal{L}(\tilde{\mu}, \tilde{m})$ w.r.t ϕ_z, we have:

$$\bigtriangledown_{\phi_z} \mathcal{L} = \kappa A_W(\xi) \big(- \frac{\bigtriangledown_{\phi_z} S_n}{2S_n^{3/2}} \tilde{\mu}\theta_n + \frac{\tilde{\mu} \bigtriangledown_{\phi_z} \theta_n}{\sqrt{S_n}} \big)^T \nu_n - \beta \phi_z$$

where

$$\bigtriangledown_{\phi_z} S_n = 2(1 - A_W(\xi)^2) \sum_{z'} \bigtriangledown_{\phi_z} \theta_{nz'} \theta_{nz'} + 2A_W(\xi)^2 \theta_n^T \tilde{\mu}^T \tilde{\mu} \bigtriangledown_{\phi_z} \theta_n$$

5.3 Experiments

We conduct comprehensive experiments to evaluate the effectiveness of SSE, in terms of the quality of its outputs (primarily visualization, but also topic model).

5.3.1 Experimental Setup

Datasets. We rely on three publicly-available[1], real-life datasets which are $20News$, $Reuters8$ and $Cade12$ [20]. The description of the three datasets as well as how we process the data are given in Section 3.5.1.

L^2-normalized Representation. SSE admits different options for the L^2 representation of a document. The option that is most well-recognized in the information retrieval literature is *tf-idf*. We experimented with several alternatives, such as word count or term frequency (*tf*), and found *tf-idf* to give the best results. This echoes the finding in [101], which concluded that *tf-idf* was a better document representation than *tf*. Thus, we will use *tf-idf* in the experiments.

5.3.2 Comparative Methods

The comparative methods, and their attributes, are summarized in Table 5.1. **SSE** is our proposed method. A proper comparison is to another approach that jointly models visualization and topics, i.e., PLSV [62], which we use as the primary baseline. For completeness, we include other baselines in visualization (PE). While not direct competitors, they allow us to highlight certain aspects of our model.

PLSV [62] is a semantic visualization method based on multinomial modeling for dyadic data. Therefore, it is the proper baseline to SSE, allowing us to investigate the effects of SSE's modeling of *spherical representation*. For PLSV, we use the same settings as in the original paper [62] ($\beta = 0.1N$ and $\gamma = 0.1Z$, which we apply to SSE as well). We implement PLSV on our own (its authors have not made their implementation available), and verify that the results are similar to those reported in the original paper [62].

PE [61] stands for Parameteric Embedding. It is also one of the state-of-the-art approaches in visualization, but is aimed at visualizing discrete probability distributions (e.g., class or topic distributions). PE cannot stand alone, as it needs to be coupled with a method that produces topic distributions. Including PE allows us to

[1] `http://web.ist.utl.pt/acardoso/datasets/`

	Visualization	Topic model	Joint model	Spherical Representation
SSE	✓	✓	✓	✓
PLSV	✓	✓	✓	
PE (SAM)	✓	✓		✓
PE (LDA)	✓	✓		

Table 5.1: Comparative Methods

investigate the effects of modeling visualization and topic model *jointly*, as opposed to obtaining topic model separately before feeding it into PE. To produce the topic distributions, we experiment with two other topic models, as follows. **PE (LDA)** couples PE with LDA [14], which operates in the simplex word space. For LDA, we use the implementation[2] by its first author D. Blei. **PE (SAM)** couples PE with SAM [101], which operates in the spherical word space. For SAM, we use the implementation[3] by an author A. Waters with default settings ($\kappa_0 = 10, \kappa = 5000$, which we apply to SSE as well). [62] showed that PE with PLSA [57] is inferior to PLSV.

For visualization, we will be comparing SSE against PLSV, PE (SAM) and PE (LDA). We also investigate the topic models, comparing SSE against PLSV, and the two topic models used with PE, i.e., SAM and LDA. As input, for models with spherical representation (see Table 5.1), we use *tf-idf* vector (as explained in Section 5.3.1). For the multinomial models, we use their regular inputs (word counts).

5.3.3 Visualization Quality

Metric. The utility of a scatterplot visualization is in allowing the user to perceive similarities between documents through their distances in the visualization space. Our emphasis is on the strength of the dimensionality reduction, rather than on the user interface aspect. There exists established metrics to measure dimensionality reduction objectively.

One such approach is to rely on the available class labels as ground truth. Intuitively, documents of the same class are more likely to be similar than documents

[2]http://www.cs.princeton.edu/ blei/lda-c

[3]`https://github.com/austinwaters/py-sam`

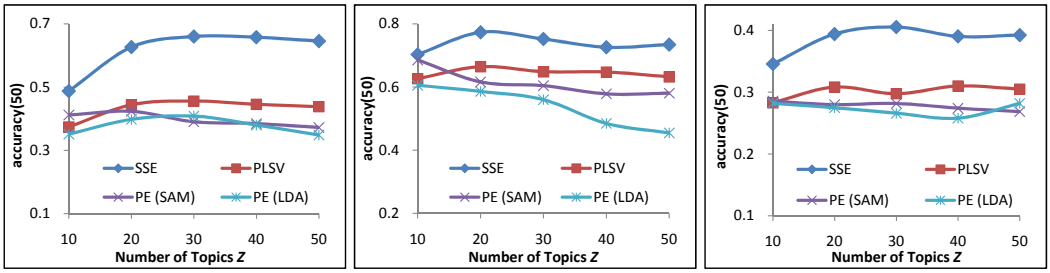


Figure 5.2: Visualization Quality: Vary Number of Topics Z

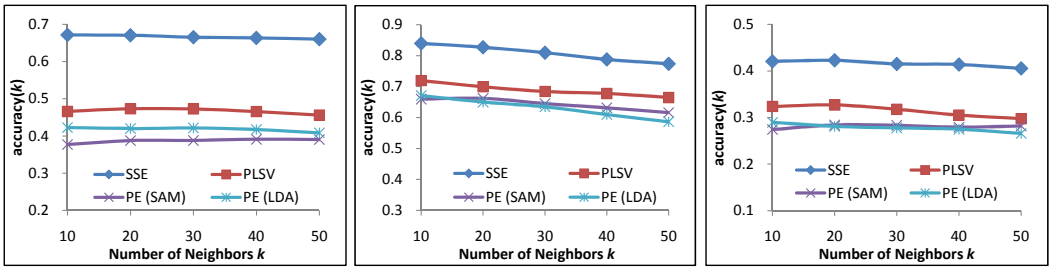


Figure 5.3: Visualization Quality: Vary Number of Neighbors k

from different classes. A good visualization will "encode" this intuition, by placing documents of the same class nearby, and documents of different classes apart in the visualization space. Since dimensionality reduction means that the lower-dimensional representation still preserves the "properties" of the data, we can measure how well a visualization output reflects this intuition, by employing each document's visualization coordinates as a reduced "feature vector" in a classification task.

The choice of the classification method is not germane, because it is the feature vector that is being evaluated. In favor of simplicity, we employ kNN classification. For each document, we hide its class label, and predict a label by majority voting among its k-nearest neighbors as determined by Euclidean distance on the visualization space. The accuracy at k or $accuracy(k)$ is the fraction of documents whose predicted label based on kNN matches the true label. The higher the accuracy, the better is a visualization at encoding the class information. 1 is the highest possible accuracy, and 0 the lowest. The same metric was also used in [62].

For relative comparison, we set $k = 50$, i.e., measuring $accuracy(50)$, which is appropriate, as the datasets contain 50 documents from each class. Setting $k << 50$ may not sufficiently penalize a visualization that splits documents of the same class into multiple small clusters in different localities.

Vary Number of Topics Z. We now compare the performance of various methods. In Figure 5.2, we plot the $accuracy(50)$ as we vary the number of topics Z from 10 to 50. The three sub-plots (a), (b), and (c) correspond to the three datasets $20News$, $Reuters8$, and $Cade12$ respectively.

In terms of SSE's performance as the number of topics varies:

(#1) As the number of topics Z increases, initially there is an improvement in accuracy, most notably between $Z = 10$ and $Z = 30$. Thereafter, accuracies either remain flat or drop slightly as Z increases further. The best performance by SSE is 0.66 on $20News$ (at $Z = 30$), 0.77 on $Reuters8$ (at $Z = 20$), and 0.41 on $Cade12$ (at $Z = 30$).

(#2) SSE achieves a drastic reduction in dimensionality from thousands (vocabulary size) to two (visualization), while preserving the relationship between data points. The above accuracies as measured in the reduced dimensionality (visualization) approach closely the accuracies of kNN when using the full dimensionality (i.e., *tf-idf* input vectors), which are 0.73 on $20News$, 0.85 on $Reuters8$, and 0.52 on $Cade12$. This shows that SSE's low-dimensional representation has high approximation ratios of 90% for $20News$ and $Reuters8$ and 78% for $Cade12$ in kNN accuracies, underlining the quality of dimensionality reduction achieved.

(#3) The varying accuracies across datasets indicate their relative difficulties, with $20News$ in between $Reuters8$ (the least difficult) and $Cade12$ (the most difficult).

In terms of SSE's comparison to baselines:

(#1) SSE has significantly higher accuracies than PLSV (the main baseline). In relative terms, SSE improves upon PLSV's accuracy by 30–48% on $20News$, by 12–16% on $Reuters8$, and by 22–36% on $Cade12$. This indicates that *spherical*

representation of word space helps to improve the visualization.

(#2) SSE also outperforms PE (SAM) by a large margin. Since SSE and SAM share a spherical representation of topics in the word space, this outperformance by SSE can be attributed to *jointly* modeling topics and visualization. This is further suported by how PLSV (which also jointly models topics and visualization) outperforms PE (LDA), even as they share multinomial modeling of topic words.

Vary Number of Neighbors k. In Figure 5.3 we investigate the effects of different neighborhood size k's at specific settings of topics ($Z = 30$ for $20News$, $Z = 20$ for $Reuters8$, and $Z = 30$ for $Cade12$). These are Z settings where SSE performs best, but similar observations can be drawn for other Z settings. The focus here is on the number of neighbors, rather than on the relative comparison against the baselines again, so we apply the same Z for all methods.

(#1) As k increases from 10 to 50, the $accuracy(k)$ tends to decrease. This is expected because a small k is very conservative, where we are only concerned with the immediate neighbors, which tend to be very similar. As k increases, the neighborhood considered in the kNN is larger, with a higher chance of having neighbors of a different class.

(#2) The gradients of the decrease vary among methods. Most methods, such as SSE, are relatively stable. This stability across different k's is a good sign, indicating that documents of the same class are placed in the same general locality.

In summary, the experiments show that SSE overall produces a significant gain in visualization quality over the baselines, as measured in terms of its accuracy in kNN classification with coordinates as features.

5.3.4 Topic Interpretability

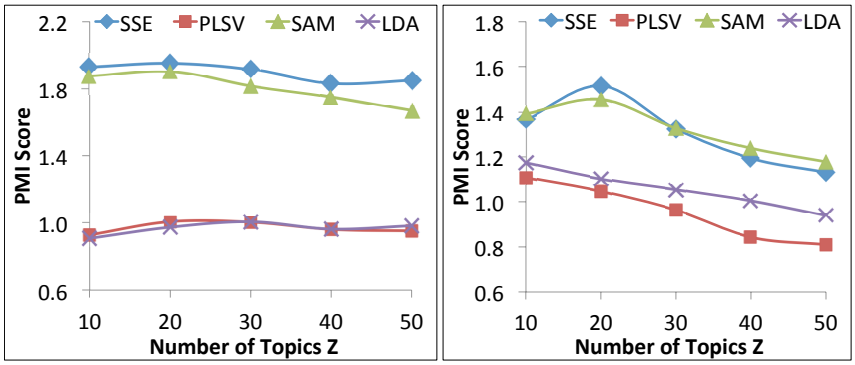
We also investigate whether the gain in visualization comes at the expense of the topic model. We compare SSE with baselines PLSV, LDA, and SAM in terms of topic model.

Metric. There are several evaluation methods for topic models proposed in the literature. One is perplexity [14], which measures the log-likelihood on unseen test data. Perplexity is *intrinsic*, i.e., dependent on the specific probability model, and may be inappropriate when comparing models with drastically different probability models, e.g., PLSV or LDA that uses multinomial models, versus SSE or SAM that uses vMF distributions. We thus need an *extrinsic* evaluation that compares these models using external validation.

In our setting, interpretability is important, because the topic model serves to provide semantics to the visualization of the data at hand. To human subjects, interpretability is closely related to coherence [91], i.e., how much the top keywords in each topic are "associated" with each other. After an extensive study of evaluation methods for coherence, [91] identifies Pointwise Mutual Information (PMI) as the best measure, in terms of having the greatest correlation with human judgments. We therefore adopt PMI as a metric. PMI is based on term cooccurrences. For a pair of words w_i and w_j, PMI is defined as $\log \frac{p(w_i,w_j)}{p(w_i)p(w_j)}$. For a topic, we average the pairwise PMI's among the top 10 words of that topic. For a topic model, we average PMI across the topics. Intuitively, PMI is higher (better), if each topic features words that are highly correlated with one another.

Key to PMI is the use of an external corpus to estimate $p(w_i, w_j)$ and $p(w_i)$. Following [92], we use *Google Web 1T 5-gram Version 1* [15], a corpus of n-grams generated from 1 trillion word tokens. $p(w_i)$ is estimated from the frequencies of 1-grams. $p(w_i, w_j)$ is estimated from the frequencies of 5-grams, as recommended in [92]. We show the PMI for the English-based $20News$ in Figure 5.4(a) and $Reuters8$ in Figure 5.4(b). $Cade12$ is not included because we do not possess a large-scale n-gram corpus for Brazilian Portuguese.

Vary Number of Topics Z. From Figure 5.4, we draw the following observations on topic interpretability. **(#1)** SSE outperforms PLSV, and SAM outperforms LDA, in terms of PMI scores, across various topic settings, on $20News$ and $Reuters8$. It indicates that spherical models (SSE and SAM) produce topics that are



a. *20News* b. *Reuters8*

Figure 5.4: Topic Interpretability (PMI Score)

more coherent and interpretable than multinomial models (PLSV and LDA). This is consistent with the conclusion reached in [101], which conducts an evaluation of coherence using human judges. This concurrence helps to show that our automatic evaluation on an external corpus is consistent with human judgments.

(#2) SSE performs similarly to SAM, with slightly higher PMI scores on $20News$, but comparable scores on $Reuters8$. This can be explained by their common modeling of topics in the spherical space. Since SSE also needs to deal with visualization constraints, it is notable that the gains in visualization quality have not hurt, and have even sometimes helped the topic model.

(#3) PLSV performs similarly to LDA on $20News$, but slightly worse on $Reuters8$, which is not surprising since they both share a similar multinomial modeling of topics but PLSV also faces constraints to fit the visualization task.

In summary, the experiments show that by incorporating spherical representation, SSE's significant gain in visualization does not come at the expense of the topic model.

5.3.5 Qualitative Comparison

To gain a sense of the visualization quality, we show example visualization outputs for $20News$ and $Reuters8$. $Cade12$ is not shown here due to space constraint.

20News. The visualizations for $20News$ are shown in Figure 5.5 for $Z = 30$. Each document has a coordinate in the scatterplot. To aid identification, documents

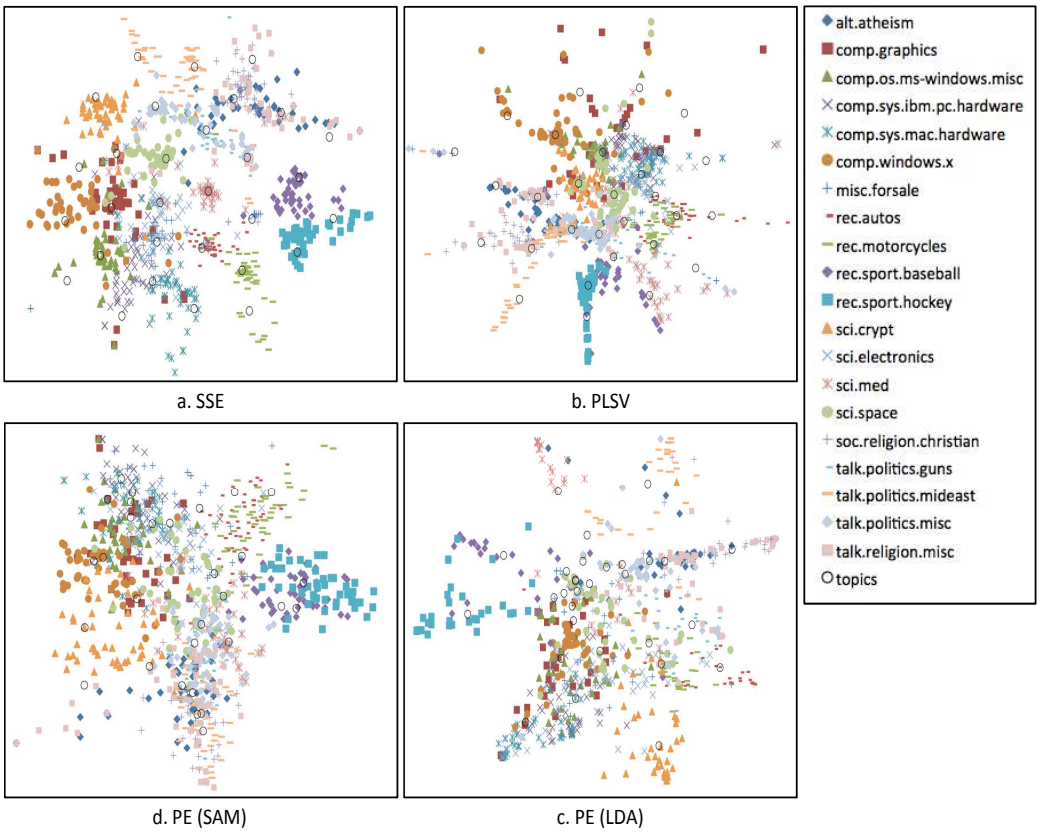


Figure 5.5: Visualization of $20News$ for $Z = 30$ topics (best viewed in color)

are drawn with a colored marker based on their class (see legend). Topics are drawn as black, hollow circles.

SSE's visualization in Figure 5.5(a) shows better separation of different classes. For instance, there are distinct blue cluster and purple cluster on the right for *rec.sport.hockey* and *rec.sport.baseball* classes respectively, green and red clusters on the lower right for *rec.motorcycles* and *rec.autos*, etc. Interestingly, not only are documents of the same class placed nearby, but related classes are also neighboring one another, with recreational classes *rec.** on the lower right, computer classes *comp.** on the lower left, science clases *sci.** at the center and upper left, while classes related to politics and religion are on the upper right. Comparatively, PLSV in Figure 5.5(b) suffers from greater crowding at the mid-section. PE (LDA) in Figure 5.5(d) and PE (SAM) in in Figure 5.5(e) are weaker. The relative ranking in visualization quality largely mirrors the earlier finding on quantitative accuracy, with SSE being the best, followed by PLSV, and then the two PE approaches.

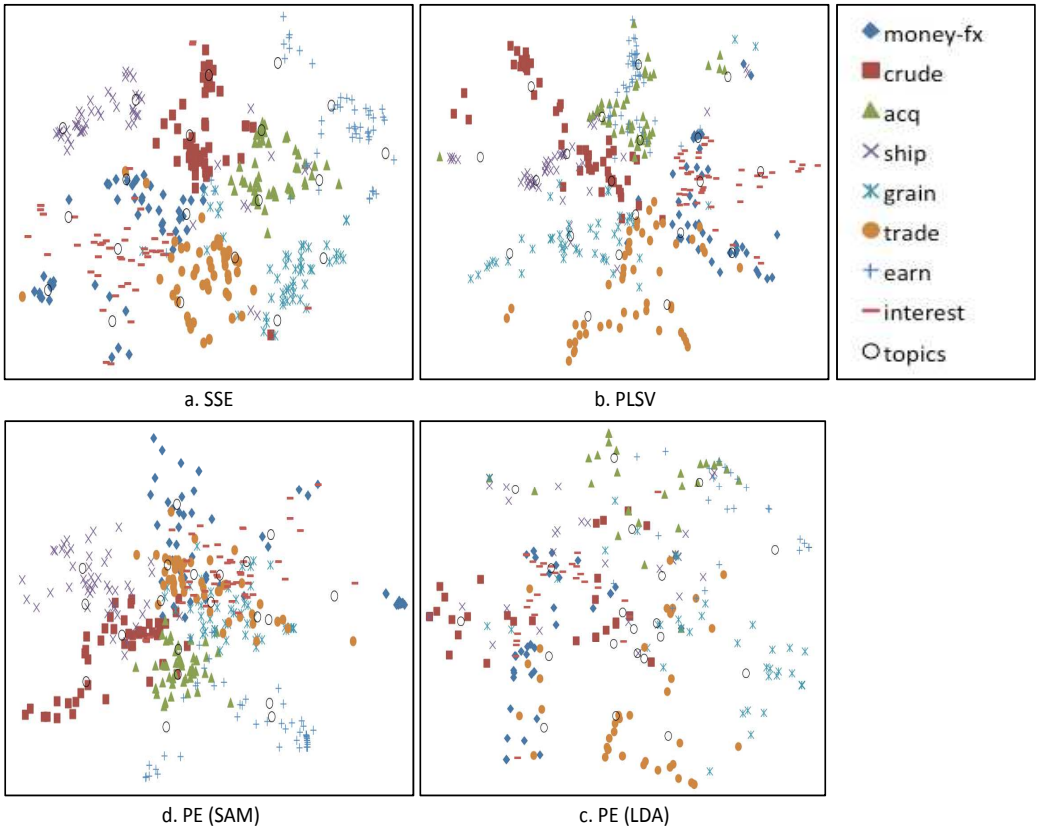


Figure 5.6: Visualization of $Reuters8$ for $Z = 20$ topics (best viewed in color)

Topic 0	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9
5 Most Positive Weights									
hockey	bike	car	window	apple	jesus	god	israel	doctor	space
team	dod	engine	software	sale	christ	religion	israeli	patient	launch
cup	motorcycle	mile	product	monitor	christian	truth	arab	treatment	moon
playoff	ride	ford	price	computer	god	belief	jew	medicine	flight
nhl	rider	mustang	user	price	sin	existence	jewish	symptom	nasa
5 Most Negative Weights									
pitch	ford	circuit	video	scsi-2	scholar	reporter	encryption	objective	algorithm
pitcher	detector	amp	bus	scsi-1	addition	government	armenian	religion	file
inning	oort	board	slot	burst	wingate	livesey	algorithm	jew	driver
bullpen	sensor	lady	wiretap	scsi	sea	corruption	science	key	nice
giant	firearm	1983	ide	16-bit	livesey	theological	armenia	god	motorcycle

Table 5.2: Positive and Negative Words in Each Topic for $20News$ by SSE for $Z = 30$ (a selection of 10)

To show that the SSE's visualization is backed by a good topic model, we show some topic words in Table 5.2. One property of spherical representation is that each topic may have both positive and negative words. We show the five most positive words, and the five most negative words. Only 10 topics out of 30 are shown due to space constraint. Looking at the positive words, we see that the topics cover some classes very well, such as hockey, motorcycle, car, windows, apple,

christianity, religion, middle eastern politics, medicine, and space. Looking at the negative words, we see that the topics also define what classes they are not. Topic 0 is about hockey, and not baseball. Topic 1 is about motorcycles, and not cars. Topic 3 is about software, and not hardware.

Reuters8. The visualizations for $Reuters8$ are shown in Figure 5.6 for $Z = 20$. Generally, it is an easier dataset, and most methods perform better than for $20News$. Comparatively, SSE still produces the clearest separation between classes, and similar observations apply as before.

5.4 Conclusion

In this work, we address the problem of semantic visualization that jointly models visualization and topics. Our model, Spherical Semantic Embedding or SSE is designed for data with spherical representation, i.e., L^2-normalized term vectors. Its generative model associates each document with a triplet of representations, namely: a coordinate in the Euclidean visualization space, a multinomial topic distribution in the topic space, as well as a normalized term vector in the spherical word space. Comprehensive experiments on benchmark datasets show that SSE shows significantly improved performance when compared to existing state-of-the-art baselines in terms of visualization quality, as well as topic interpretability.

Chapter 6

Modeling Bag of Word Vectors

In this chapter, we address the problem of semantic visualization for short texts. Such documents are increasingly common, including tweets, search snippets, news headlines, or status updates. Due to their short lengths, it is difficult to model semantics as the word co-occurrences in such a corpus are very sparse. Our approach is to incorporate auxiliary information, such as word embeddings from a larger corpus, to supplement the lack of co-occurrences. This requires the development of a novel semantic visualization model that seamlessly integrates visualization coordinates, topic distributions, and word vectors. We propose a model called GaussianSV, which outperforms pipelined baselines that derive topic models and visualization coordinates as disjoint steps, as well as semantic visualization baselines that do not consider word embeddings.

6.1 Introduction

As formulated in Section 1.1, *semantic visualization* refers to jointly modeling topics and visualization. Given a corpus of documents, we seek to learn for each document, its coordinate in a 2D Euclidean space for visualization, as well as its topic distribution. Of primary concern in this chapter is semantic visualization for *short texts*, which make up an increasing fraction of texts generated today, owing to the proliferation of mobile devices and prevalence of social media. For instance, tweets

are limited to 140 characters. Other types of short text, e.g., search snippets, news headlines, or status updates are not much longer. Short text's limitation in modeling semantics is well-documented in various contexts [87, 109, 112].

Existing semantic visualization models are not designed for short texts. For example, PLSV [62] represents documents as bags of words, and topic distributions are inferred from word co-occurrences in documents. This assumes sufficiency in word co-occurrences to discover meaningful topics. This may be valid for regular-length documents, but not for short texts, due to the extreme sparsity of words in such documents. Methods based on tf-idf vectors, such as SSE [74] would also suffer, because tf-idf vectors are not efficient for short text analysis [126]. Many words appear only once in a short document, and may appear in only a few documents. Consequently tf and idf are not very distinguishable in short texts.

Approach. There are several possible directions to deal with short text. Not all are suitable for semantic visualization. For instance, it is possible to combine a few short texts into a longer "pseudo-document", e.g., grouping tweets of one user. However, this would not allow us visualize individual short texts, in order to view their relationships, as they are now aggregated into one pseudo-document displayed as a single element. For another instance, we could constrain the topic model to assign one topic to all words within a short text to enforce word co-occurrences. However, this still would not fully resolve the issue of the sparsity of word co-occurrences.

The direction taken in this chapter is to attack the main issue of sparsity, by supplementing short texts with auxiliary information from a larger external corpus. Outside of semantic visualization, this was explored in the context of topic modeling (without visualization), by incorporating topics learned from Wikipedia [100] or jointly learning two sets of topics on short and auxiliary long texts [64].

Specifically, we seek to leverage word embeddings, which have gained increasing attention for their ability to express the conceptual similarity of words. Models such as Word2Vec [88] and GloVe [99] learn a continuous vector in an embedding

space for each word. They are trained on a large corpora (e.g., Wikipedia, Google news). We postulate that word vectors would be a useful form of auxiliary information in the context of semantic visualization for short texts, as the conceptual similarities learned from the huge corpus and encoded in word vectors can supplement lack of word co-occurrences in short-texts.

There are two potential approaches to using word vectors. The first is what we term a *pipelined* approach, by employing topic models that work with word vectors [37, 58] to produce the topic distributions of short texts, which are then mapped to visualization coordinates using an appropriate dimensionality reduction technique. The second is what we term a *joint* approach, by designing a single model that incorporates visualization coordinates, topic distributions, and word vectors within an integrated generative process. Inspired by the precedence established by previous semantic visualization works on bag of words [62] showing the advantage of a joint approach, we surmise that joint modeling is a promising approach for semantic visualization using word embeddings.

Contributions. We make the following contributions. *Firstly*, as far as we are aware, we are the first to propose semantic visualization for short texts. *Secondly*, we design a novel semantic visualization model that leverages word embeddings. Our model, called *Gaussian Semantic Visualization* or GaussianSV, assumes that each topic is characterized by a Gaussian distribution on the word embedding space. Section 6.2 presents the model in detail including its generative process as well as how to learn its parameters based on MAP estimation. *Thirdly*, we evaluate our model on two public real-life short text datasets in Section 6.3. To validate our joint modeling, one class of baselines consist of pipelined approaches that apply dimensionality reduction to the outputs of topic models with word embeddings. To validate our modeling of word embeddings, the other class of baselines consist of semantic visualization models not using word vectors.

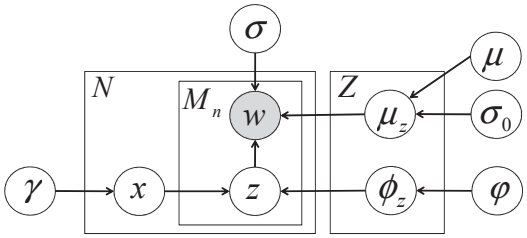


Figure 6.1: Graphical Model of GaussianSV

6.2 Gaussian Semantic Visualization

In this section, we describe our proposed model GaussianSV, whose graphical model is shown in Figure 6.1.

Our input is a corpus of documents $\mathcal{D} = \{d_1, \ldots, d_N\}$. Each document d_n is a bag of words. Denote w_{nm} to be the m^{th} word in document d_n, and M_n to be the number of words in d_n. Each word w in the vocabulary W is represented as a p-dimensional continuous vector, which has been learned from an external corpus using some word embedding model. For popular word embeddings [88, 99], p is usually in the hundreds.

Our objective is two-fold. First, we seek to derive as output the visualization coordinate x_n for each document d_n. Without loss of generality, in the following, we assume x_n is 2-dimensional for visualization. Second, we also seek to derive each document's topic distribution over Z topics $\{\mathrm{P}(z|d_n)\}_{z=1}^{Z}$. Each topic z is associated with a probability distribution $\{\mathrm{P}(w|z)\}_{w \in W}$ over words in the vocabulary W. The words with the highest probabilities given a topic usually help to provide some interpretable meaning to a topic.

6.2.1 Generative Process

In a conventional topic model, such as LDA [14] or PLSA [57], a topic is represented by a multinomial distribution over words. Some previous works on semantic visualization [62, 75] are also based on such topic representation.

The key difference is that in our context a word is not just a discrete outcome of a multinomial process, but rather a continuous vector in the embedding space. We need another way to characterize a topic, as well as to model the generation of words

due to that topic. Inspired by [37], we associate each topic z with a continuous vector μ_z resident in the same p-dimensional word embedding space. This allows us to model the word generation due to a topic as a Gaussian distribution, centered at the μ_z vector, with spherical covariance. In other words, a word w_{nm} belonging to topic z will be drawn according to the following probability:

$$\mathrm{P}(w_{nm}|\mu_z, \sigma) = \big(\frac{\sigma}{2\pi}\big)^{\frac{p}{2}} \exp(-\frac{\sigma}{2} \parallel w_{nm} - \mu_z \parallel^2), \tag{6.1}$$

where σ is a hyper-parameter.

To derive the visualization, in addition to the coordinate x_n associated with each document d_n, we also assign each topic z a latent coordinate ϕ_z in the same visualization space. With documents and topics residing in the same Euclidean space, spatial distances between documents and topics can represent their relationship. Intuitively, documents close to each other would tend to talk about the same topics (that are also located near those documents). We thus express a document d_n's distribution over topics, in terms of the Euclidean distances between x_n and topic coordinate ϕ_z, as follows:

$$\mathrm{P}(z|x_n, \Phi) = \frac{\exp(-\frac{1}{2}||x_n - \phi_z||^2)}{\sum_{z'=1}^{Z} \exp(-\frac{1}{2}||x_n - \phi_{z'}||^2)} \tag{6.2}$$

where $\mathrm{P}(z|x_n, \Phi)$ is the probability of topic z in document d_n and $\Phi = \{\phi_z\}_{z=1}^{Z}$ is the set of topic coordinates.

Our objective is to derive the coordinates of documents and topics in the visualization space, as well as the distribution over Z topics $\{\mathrm{P}(z|d_n)\}_{z=1}^{Z}$ for each document d_n. We also derive the mean μ_z for each topic z. Note that we do not derive word vectors, but consider them as input to our model.

The generative process is now described as follows:

1. For each topic $z = 1, \ldots, Z$:
   (a) Draw z's mean: $\mu_z \sim \mathrm{Normal}(\boldsymbol{\mu}, \sigma_0^{-1}\mathrm{I})$
   (b) Draw z's coordinate: $\phi_z \sim \mathrm{Normal}(0, \varphi^{-1}I)$

2. For each document d_n, where $n = 1, \ldots, N$:

   (a) Draw d_n's coordinate: $x_n \sim \mathrm{Normal}(0, \gamma^{-1}I)$

   (b) For each word $w_{nm} \in d_n$:

      i. Draw a topic: $z \sim \mathrm{Multi}(\{\mathrm{P}(z|x_n, \Phi)\}_{z=1}^{Z})$

      ii. Draw a word: $w_{nm} \sim \mathrm{Normal}(\mu_z, \sigma^{-1}\mathrm{I})$

The first step concerns the generation of topics' mean vectors and visualization coordinates. The second step concerns the generation of documents' coordinates, and words (represented as word vectors) within each document.

Notably, by representing documents and topics in the same visualization space, as well as words and topics in the same word embedding space, the topics play a crucial role as conduits between the two spaces. Therefore, documents that contain similar words are more likely to share similar topics. Here, "similar" words could be the same words, frequently co-occurring words, and owing to the use of word embeddings: also different words that are close in the word embedding space. For short texts in particular, the latter is expected to be especially significant, because of lower word frequencies and weaker role of word co-occcurrences.

6.2.2 Parameter Estimation

The parameters are estimated based on maximum a posteriori estimation (MAP) using EM algorithm [39]. The unknown parameters that need to be estimated include document coordinates $\chi = \{x_n\}_{n=1}^{N}$, topic coordinates $\Phi = \{\phi_z\}_{z=1}^{Z}$, and topic mean vectors $\Pi = \{\mu_z\}_{z=1}^{Z}$, collectively denoted as $\Psi = \{\chi, \Phi, \Pi\}$.

Given the generative process described earlier, the log likelihood can be expressed as follows:

$$\mathcal{L}(\Psi|\mathcal{D}) = \sum_{n=1}^{N}\sum_{m=1}^{M_n}\log\sum_{z=1}^{Z}\mathrm{P}(z|x_n, \Phi)\mathrm{P}(w_{nm}|\mu_z, \sigma) \tag{6.3}$$

The conditional expectation of the complete-data log likelihood with priors is as

follows:

$$\mathcal{Q}(\Psi|\hat{\Psi}) = \sum_{n=1}^{N}\sum_{m=1}^{M_n}\sum_{z=1}^{Z} \mathrm{P}(z|n,m,\hat{\Psi}) \log\left[\mathrm{P}(z|x_n,\Phi)\mathrm{P}(w_{nm}|\mu_z,\sigma)\right] + \sum_{n=1}^{N}\log(\mathrm{P}(x_n)) + \sum_{z=1}^{Z}\log(\mathrm{P}(\phi_z)) + \sum_{z=1}^{Z}\log(\mathrm{P}(\mu_z)),$$

where $\hat{\Psi}$ is the current estimate. $\mathrm{P}(z|n,m,\hat{\Psi})$ is the class posterior probability of the n^{th} document and the m^{th} word in the current estimate. $\mathrm{P}(x_n)$ and $\mathrm{P}(\phi_z)$ are Gaussian priors with a zero mean and a spherical covariance for the document coordinates x_n and topic coordinates ϕ_z:

$$\mathrm{P}(x_n) = \Big(\frac{\gamma}{2\pi}\Big)^{\frac{D}{2}} \exp\Big(-\frac{\gamma}{2} \parallel x_n \parallel^2\Big), \tag{6.4}$$

$$\mathrm{P}(\phi_z) = \Big(\frac{\varphi}{2\pi}\Big)^{\frac{D}{2}} \exp\Big(-\frac{\varphi}{2} \parallel \phi_z \parallel^2\Big), \tag{6.5}$$

where we set the hyper-parameters to $\gamma = 0.1Z$ and $\varphi = 0.1N$ following PLSV [62].

We put a Gaussian prior over μ_z with hyper-parameter σ_0 and mean $\boldsymbol{\mu}$ which is set to the average of all word vectors in the vocabulary.

$$\mathrm{P}(\mu_z) = \big(\frac{\sigma_0}{2\pi}\big)^{\frac{p}{2}} \exp(-\frac{\sigma_0}{2} \parallel \mu_z - \boldsymbol{\mu} \parallel^2) \tag{6.6}$$

We use EM algorithm to estimate the parameters. In the E-step, we compute $\mathrm{P}(z|n,m,\hat{\Psi})$ as in Equation 6.7. We then update $\Psi = \{\chi, \Phi, \Pi\}$ in the M-step. μ_z is updated using Equation 6.8. To update ϕ_z and x_n, we use gradient-based numerical optimization method such as the quasi-Newton method [79] because the gradients cannot be solved in a closed form. We alternate the E- and M-steps until some appropriate convergence criterion is reached.

E-step:

$$\mathrm{P}(z|n,m,\hat{\Psi}) = \frac{\mathrm{P}(z|\hat{x}_n,\hat{\Phi})\mathrm{P}(w_{nm}|\hat{\mu}_z,\hat{\Sigma}_z)}{\sum_{z'=1}^{Z}\mathrm{P}(z'|\hat{x}_n,\hat{\Phi})\mathrm{P}(w_{nm}|\hat{\mu}_{z'},\hat{\Sigma}_{z'}))} \tag{6.7}$$

M-step:

$$\frac{\partial \mathcal{Q}(\Psi|\hat{\Psi})}{\partial \phi_z} = \sum_{n=1}^{N}\sum_{m=1}^{M_n}\big(\mathrm{P}(z|x_n,\Phi) - \mathrm{P}(z|n,m,\hat{\Psi})\big)(\phi_z - x_n) - \beta\phi_z$$
$$\frac{\partial \mathcal{Q}(\Psi|\hat{\Psi})}{\partial x_n} = \sum_{m=1}^{M_n}\sum_{z=1}^{Z}\big(\mathrm{P}(z|x_n,\Phi) - \mathrm{P}(z|n,m,\hat{\Psi})\big)(x_n - \phi_z) - \gamma x_n$$
$$\mu_z = \frac{\sum_{n=1}^{N}\sum_{m=1}^{M_n}\big(\mathrm{P}(z|n,m,\hat{\Psi})\sigma w_{nm}\big) + \sigma_0\boldsymbol{\mu}}{\sum_{n=1}^{N}\sum_{m=1}^{M_n}\mathrm{P}(z|n,m,\hat{\Psi})\sigma + \sigma_0} \tag{6.8}$$

6.3 Experiments

The objective is to evaluate the effectiveness of GaussianSV for visualizing short texts and the quality of its topic model.

6.3.1 Experimental Setup

Datasets. We use short texts from two real-life public datasets. The first is BBC[1] [49], which consists of 2,225 BBC news articles from 2004-2005, divided into 5 classes. The second is $SearchSnippet$[2] [100], which consists of 12,340 Web search snippets categorized into 8 classes. For each BBC article, we only use its title and headline, which is comparable in length to $SearchSnippet$. For word embedding, we use the pre-trained 300-dimensional word vectors from *Word2Vec* trained on Google News[3]. For each dataset, we remove stopwords, perform stemming, and remove words that do not have pre-trained word vectors. The average document length is 14.1 words for BBC and 14.9 words for $SearchSnippet$.

Although the datasets have classes, the class information is not used for learning, as semantic visualization is an unsupervised task. The classes are used later for validation, with the hypothesis that a good visualization would tend cluster documents of the same class. Following the practice in previous semantic visualization works [62, 75], for each dataset, we sample 50 documents per class in order to

[1] http://mlg.ucd.ie/datasets/bbc.html
[2] http://jwebpro.sourceforge.net/data-web-snippets.tar.gz
[3] https://code.google.com/archive/p/word2vec/

create a well-balanced dataset. Therefore, each sample of $SearchSnippet$ has 400 documents, and that of BBC has 250 documents respectively. As the methods are probabilistic, we create 5 samples for each dataset, and run each sample 5 times. The reported performance numbers are averaged across 25 runs.

Comparative Methods. We compare our GaussianSV[4] model to two classes of baselines that generate both topic model and visualization coordinates, as listed in Table 6.1. GLDA [37] modeled a topic as a distribution over word vectors. LCTM [58] modeled a topic as a distribution of concepts, where each concept defined another distribution of word vectors. GPUDMM [78] and LFDMM [93] extended DMM [94] that assigned all words in a short text to only one topic. While these topic models were not meant for visualization, their output topic distributions could be mapped to a 2D space using the dimensionality reduction meant for probability distributions, i.e., Parametric Embedding or PE [61]. In Section 6.3, we will compare to such pipelined baselines involving GLDA, LCTM, and GPUDMM (which had been shown to outperform LF-DMM in [78]).

	Visualization	Topic model	Joint model	Word vectors
GaussianSV	✓	✓	✓	✓
PLSV	✓	✓	✓	
SEMAFORE	✓	✓	✓	
SSE	✓	✓	✓	
GLDA/PE	✓	✓		✓
LCTM/PE	✓	✓		✓
GPUDMM/PE	✓	✓		✓

Table 6.1: Comparative Methods

The first class of baselines are semantic visualization techniques that do not rely on word vectors. These include PLSV[5], SEMAFORE[6], and SSE[7]. Comparison to these models help to validate the contributions of word vectors.

The second class of baselines are not semantic visualization models per se.

[4]We choose appropriate values for ρ_0 and ρ. $\rho_0 = 10000$ and $\rho = 100$ work well for most of the cases in our experiments.

[5]We use the implementation by `https://github.com/tuanlvm/SEMAFORE`.

[6]We use the author implementation in `https://github.com/tuanlvm/SEMAFORE`.

[7]We use the implementation obtained from the authors.

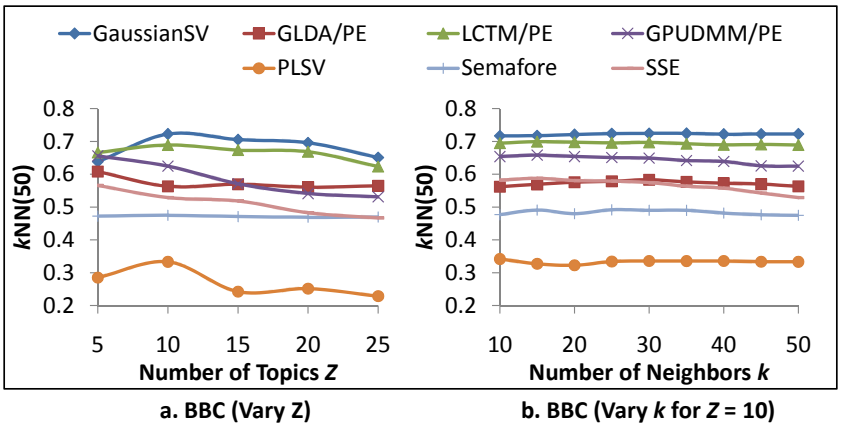


Figure 6.2: kNN Accuracy Comparison on BBC

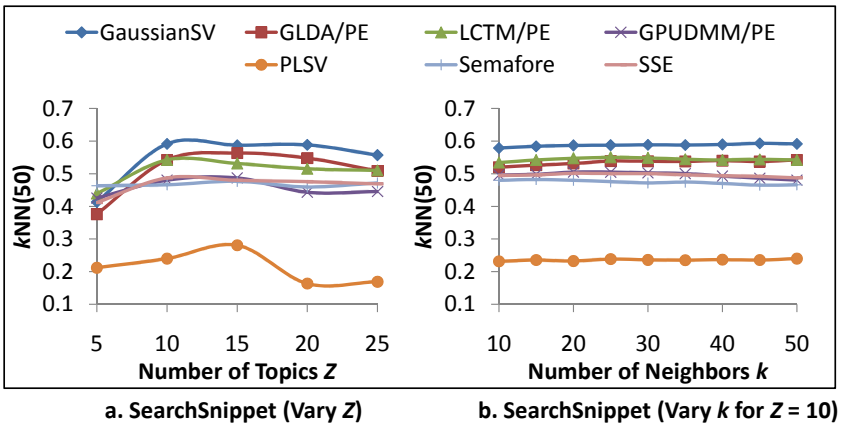


Figure 6.3: kNN Accuracy Comparison on $SearchSnippet$

Rather they are a pipeline of topic models that incorporate word vectors, i.e., GLDA[8], LCTM[9], and GPUDMM[10], followed by PE [61] for mapping topic distributions into visualization space. Comparison to these help to validate the contributions of joint modeling.

6.3.2 Visualization Quality

Metric. A good visualization is expected to keep similar documents close, and keep different documents far in the visualization space. We rely on k nearest neighbors

[8]We use the author implementation at `https://github.com/rajarshd/Gaussian_LDA`, set degree of freedom $\nu = 1000p$, and use default values for other parameters.

[9]We use the author implementation at `https://github.com/weihua916/LCTM`. The number of concepts is 500, and the noise of each concept is 0.001. Other parameters are set to default.

[10]We use the author implementation at `https://github.com/NobodyWHU/GPUDMM` with default parameters.

(kNN) classification accuracy to measure the visualization quality. This is an established metric for semantic visualization [62, 74, 75] for objectivity and repeatability. For each document, we hide its true class and assign it to the majority class determined by its k nearest neighbors in the visualization space. The accuracy is the fraction of documents that are assigned correctly to its true class.

Results. We report kNN accuracy on BBC in Figure 6.2 and on $SearchSnippet$ in Figure 6.3. At first, we set k = 50 as the datasets contain 50 documents from each class. Later, we also show kNN accuracy at different k.

In Figure 6.2a and Figure 6.3a, we vary the number of topics Z. The results show that methods with word vectors (i.e., GaussianSV, GLDA/PE, LCTM/PE and GPUDMM/PE) deal with short texts better than conventional semantic visualization techniques (i.e., PLSV, Semafore and SSE). The latter suffer due to the sparsity of word co-occurrences.

Among those leveraging word vectors, our method GaussianSV performs significantly better than the others. For BBC, comparing to LCTM/PE that has the closest performance, we gain 4-5% improvement for 10 to 25 topics. Paired samples t-test indicate that the improvement is significant at 0.05 level or lower in all cases, except for $Z = 25$. At 5 topics, LCTM/PE is slightly better, but it is not significant even at 0.1 level. For $SearchSnippet$, except for $Z = 5$, we beat the two closest baselines GLDA/PE and LCTM/PE by 4-14% with statistical significance at 0.05 level or lower. These improvements show that joint modeling to leverage word embeddings is better for semantic visualization of short texts.

In Figures 6.2b and 6.3b, we vary k while fixing $Z = 10$. The performances are not affected much by k. Similar observations regarding the comparisons can be drawn as before.

Example Visualizations. Figure 6.5 shows the visualization of each method on BBC. Documents are represented as coloured points placed according to their coordinates. Topic coordinates are represented as hollow circles. GaussianSV separates the 5 classes well. PLSV tends to mix the classes together. Semafore is

better than PLSV, as it produces some clusters, although it cannot differentiate documents belonging to $business$ and $tech$. SSE differentiates those two classes better, but the $business$ documents are spread all over instead of being grouped together like in GaussianSV's visualization. SSE also mixes some documents belonging to $entertainment$, $politics$ and $sport$ at the bottom, which is not the case in GaussianSV's visualization. The classes are not separated well in GLDA/PE's visualization, especially for those documents at the center. GPUDMM/PE separates $business$ and $tech$ well, but it divides $politics$ into two sub-clusters which could reduce the kNN accuracy. In addition, it also mixes some documents of $entertainment$ and $sport$, while GaussianSV can differentiate them. LCTM/PE provides a good visualization, however it still mixes some documents of $business$ and $politics$ together near the center. GaussianSV is better than LCTM/PE at separating them.

The visualization produced by each method on $SearchSnippet$ in Figure 6.6 shows similar trends. PLSV does not visualize well the dataset. Semafore and SSE are better than PLSV but still mix some documents belonging to different classes together. GPUDMM/PE, by leveraging word embeddings, provides better clusters in the visualization but cannot differentiate $culture - arts - entertainment$ and $sports$ on the top. This is not case in GaussianSV's visualization. Similar to GaussianSV, GLDA/PE and LCTM/PE can separate well $engineering$ and $health$. However, GLDA/PE does not separate well $culture - arts - entertainment$ by letting some documents overlap with other documents from other classes at the center. LCTM/PE has the same problem. It mixes $culture - arts - entertainment$ with some documents from other classes such as $computers$.

6.3.3 Topic Coherence

We investigate whether while providing better visualization, our method still maintains the quality of the topic model.

Metric. One measure for topic model quality that has some agreement with human

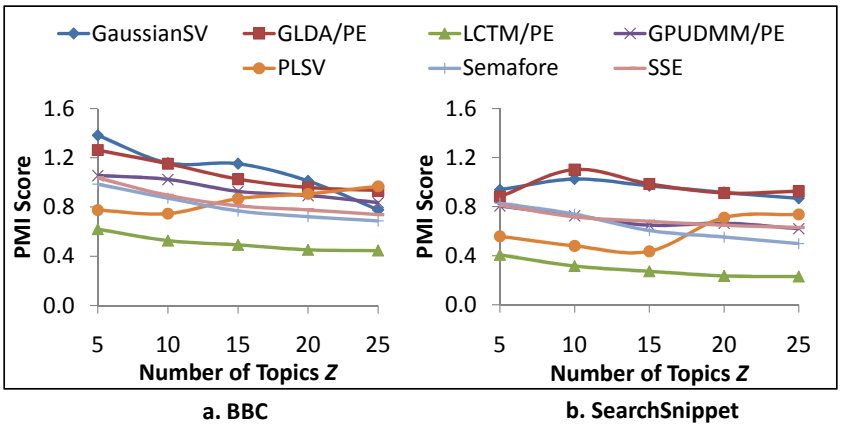


Figure 6.4: Topic Coherence (PMI Score)

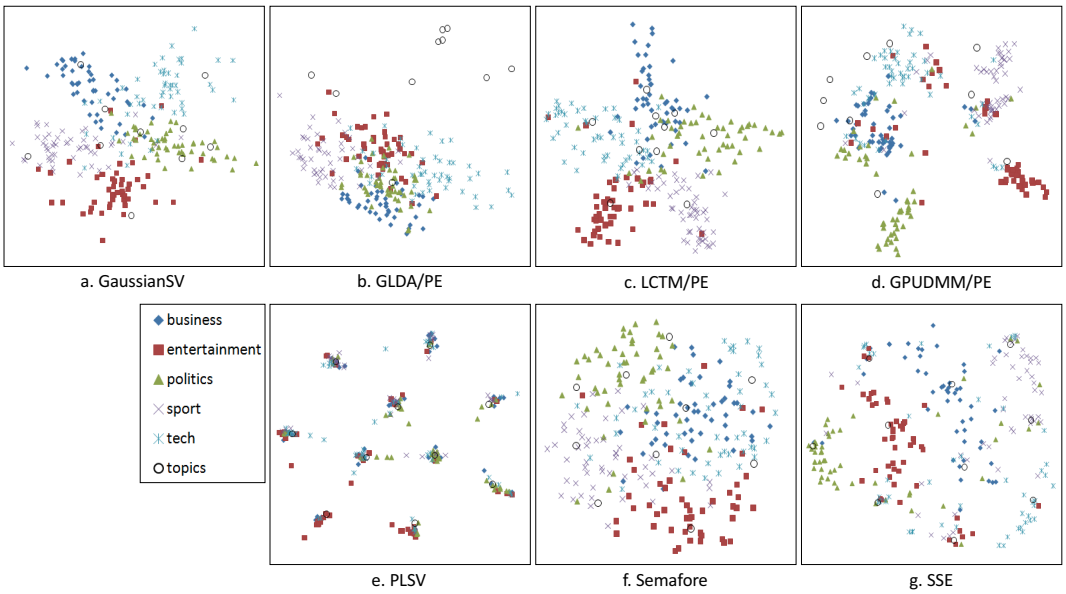


Figure 6.5: Visualization of BBC for $Z = 10$ (*best seen in colour*)

judgment is topic coherence [91], which looks at how the top keywords in each topic are related to each other in terms of semantic meaning. As suggested by [91], we rely on Pointwise Mutual Information (PMI) to evaluate topic coherence. PMI is described in detail in Section 5.3.4.

Results. Figure 6.4 shows the PMI scores for various number of topics Z. Evidently, GaussianSV has comparable PMI score to GLDA/PE, and performs better than the other methods across different Z, which shows that GaussianSV produces at least a comparable topic model, while having better visualization. As examples, Table 6.2 shows the top 5 words of each topic for $Z = 10$ for BBC and $SearchSnippet$.

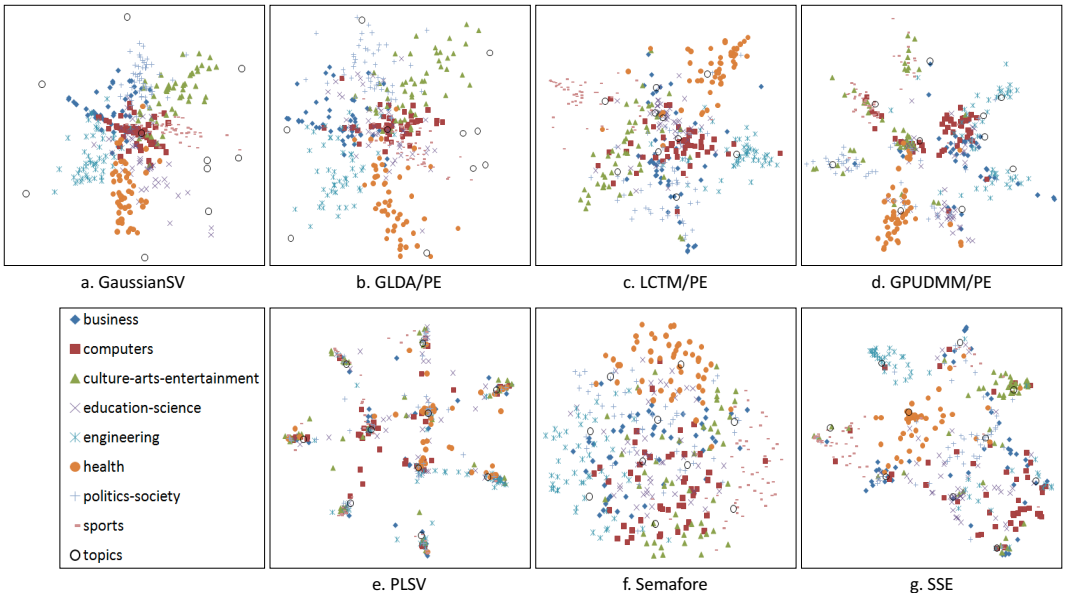


Figure 6.6: Visualization of $SearchSnippet$ for $Z = 10$ (*best seen in colour*)

BBC		$SearchSnippet$	
ID	Top 5 words	ID	Top 5 words
0	government, election, vote, proposal, referendum	0	software, technology, database, computer, system
1	player, star, boss, manager, director	1	game, sport, football, tournament, basketball
2	film, music, movie, musical, musician	2	democratic, political, democracy, government, politics
3	market, company, economy, price, economic	3	engine, cylinder, piston, turbine, compressor
4	internet, mobile, computer, digital, browser	4	health, medical, cancer, diagnosis, doctor
5	win, season, victory, championship, game	5	market, business, export, industry, manufacturing
6	gordon, thompson, alex, bryan, bennett	6	science, university, mathematics, academic, faculty
7	bring, leave, push, accept, seek	7	kind, type, aspect, work, approach
8	big, good, real, great, major	8	usa, carl, bryant, donnie, eric
9	man, woman, girl, boy, teenager	9	news, web, website, blog, online

Table 6.2: Top Words in Each Topic by GaussianSV for $Z = 10$

6.4 Conclusion

We propose GaussianSV model, a semantic visualization model for short text, which leverages word vectors obtained from a larger external corpus to supplement the sparsity of short texts. The model performs well on real-life short text datasets against semantic visualization baselines, as well as against pipelined baselines, validating both the value of incorporating word embeddings and that of joint modeling.

Part III

Applications of Semantic Visualization

Chapter 7

SemVis: Semantic Visualization for Interactive Topical Analysis

Exploratory analysis of a text corpus is an important task that can be aided by informative visualization. One spatially-oriented form of document visualization is a scatterplot, whereby every document is associated with a coordinate, and relationships among documents can be perceived through their spatial distances. Semantic visualization further infuses the visualization space with latent semantics, by incorporating a topic model that has a representation in the visualization space, allowing users to also perceive relationships between documents and topics spatially. In this chapter, we illustrate how a semantic visualization system called SemVis could be used to navigate a text corpus interactively and topically via browsing and searching.

7.1 Introduction

With digitization of content, there are increasingly more tasks that involve exploration of a text corpus for the purpose of building a general understanding of the corpus, as well as extracting specific information. For instance, a scientist conducts literature review, a financial analyst digests economic reports, a patent officer examines prior art, a legal researcher looks for precedence, etc. These scenarios

involve various information needs, e.g., what the corpus is about in general, what the predominant concepts or topics are, which documents are relevant to a particular search intent, which other documents are related to the current document. Such information needs are indeed the subjects of study in various areas of IR. What is of particular interest to us is the use of visualization.

There are various visualization paradigms. We focus on dimensionality reduction. The original representation of a document is often a bag of words. It is a high-dimensional representation, with dimensionality equal to the size of the vocabulary. One way to visualize documents and the relationships among them is to reduce their high-dimensional representation into a low-dimensional one that preserves their similarities [70, 117]. Each document is associated with a coordinate in a 2D or 3D scatterplot. Similarities among documents can be perceived spatially via their close distances.

However, such a visualization, on its own, is not designed for revealing the main "themes" of a corpus. Understanding the main themes or latent semantics in a corpus is the objective of topic modeling [14, 57]. A topic model associates each document with a probability distribution over topics, where the semantics of each topic can be interpreted by the topic's word distribution. It is common to model tens to hundreds of topics in a corpus. In other words, the topical dimensionality is still too high to be visualized directly.

Our objective is to infuse the visualization with latent semantics. Recent developments in *semantic visualization* present methods [62, 74, 75] that jointly model topics and visualization coordinates. In this paradigm, documents and topics are respectively associated with latent visualization coordinates (to be learned). A document's topic distribution is a function of the relative distance between the document's coordinate and each topic's coordinate. As a result, we can visualize the relationship not only between a pair of documents, but also between a document and a topic. Moreover, the visualization space is now also a continuous semantic space, as every coordinate (even an empty spot) codes for a distribution over topics,

and by extension also a distribution over words, lending semantic interpretation to any point in the visualization space.

Contributions We showcase SemVis, a semantic visualization system for interactive topical analysis. This is a demonstrable system that is built on, and is generically compatible with PLSV [62], SEMAFORE (Chapter 3), and SSE (Chapter 5). We illustrate the interactive topical analysis features and the capabilities of SemVis in browsing and searching scenarios in Section 7.2, and briefly outline the implementation architecture in Section 7.3.

7.2 Interactive Topical Analysis

We describe the features of the visualization system SemVis, assuming that the coordinates and the topic distributions have been learned from the corpus as in the previous section. For the running example, we use a corpus based on $20News$[1] and learn 30 topics.

Browsing. Figure 7.1 shows the main screen of SemVis. Item (1) is the black canvas space for displaying the visualization. In this canvas, we display a 2D scatterplot of documents and topics based on their respective coordinates. Each document is a circle, and each topic is a square. As a visual cue, each topic is associated with a color. Item (2) is a legend of topics, listing the top words with the highest probabilities for each topic to aid topic intrepretation. While a document's coordinate codes for a probability distribution over all topics, for ease of identification, a document is colored the same as the topic with the largest probability in that document.

The layout as well as the coloring of documents and topics in the canvas reveal an overview of the corpus, in terms of the various topics that are relevant to the corpus, as well as the relationship among documents. We can perceive when documents are similar, both through their close distances as well as similar colors. Each cluster also tends to be "anchored" by a topic. Intuitively, documents in between

[1] `http://ana.cachopo.org/datasets-for-single-label-text-categorization`

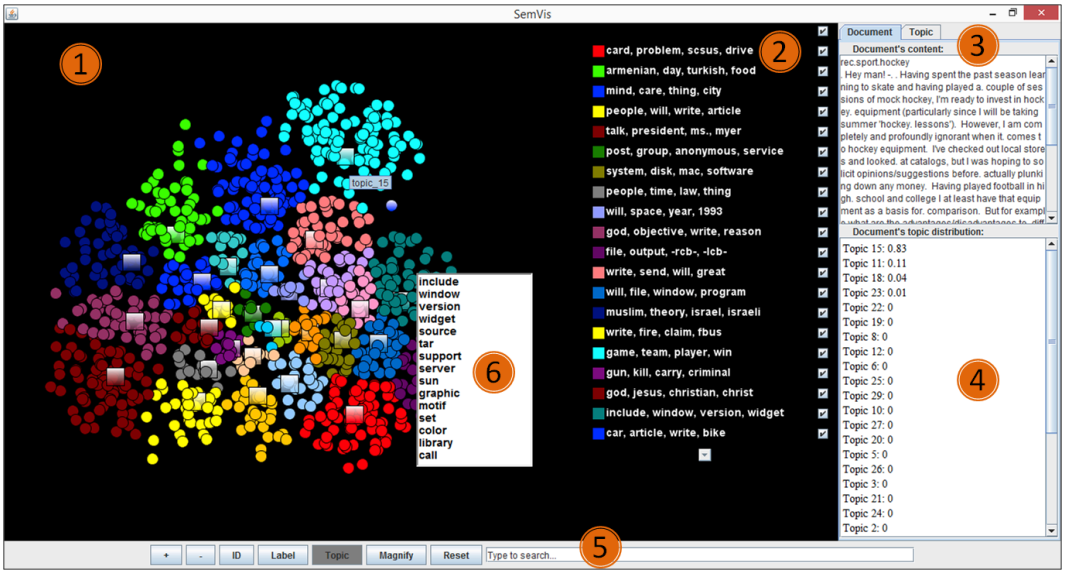


Figure 7.1: Browsing Interface

two topics have significant probabilities for both.

For a detailed view, users can click on a circle (document) to see its content displayed on item (3), and its topic distribution on item (4). In addition to that, a list of interactive functions are provided on item (5). This includes magnifying or zooming in and out of the canvas to focus on a specific region of the space.

Every point x in the visualization space is associated with a topic distribution $P(z|x, \Phi)$ (see Equation 2.3). Taking into account each topic z's distribution over words $P(w|\theta_z)$, the point's word distribution can be obtained via $P(w|x) = \sum_z P(w|\theta_z)P(z|x, \Phi)$. At any point in this visualization space, the user can right-click to see the distribution of words corresponding to that point in space. For an example, item (6) in Figure 7.1 shows the list of top words associated with the coordinate on the top left corner of the list.

Searching. Other than browsing a corpus for a general understanding, users may also need to focus on subsets of documents that are relevant to a specific search intent. This role is traditionally served by a search engine that returns a ranked list of the most relevant documents. While this is a very familiar interface to most search users today, there are some aspects for which a visualization could be beneficial. For one, a query may be ambiguous, with a few different senses, e.g., "apple" the

company vs. “apple” the fruit. A ranked list frequently interleaves results of different senses. For another, results within the ranked list may have a natural clustering structure, e.g., news about the same event. The ranking by relevance alone may not capture this, requiring additional processing.

Figure 7.2 shows our search interface. Currently, we support two query types. The first type is *textual query*. User can type in a query, and the most relevant results[2] are returned and displayed on a 2D visualization space. Here, we indicate the degree of relevance by the size of the circle, i.e., a more relevant document is drawn as a larger circle. The left panel of Figure 7.2 shows an example query *“fast drive”*. We can see clearly three clusters of results: a red cluster on the bottom right, a green cluster at the centre, and a blue cluster on the top left. This reveals that the query is indeed ambiguous, and it can be associated with several topics or senses. The red topic is about *card, problem, scsus, drive*, suggesting that the query is probably interpreted as about a fast driver software for some computer component. The green topic is about *system, disk, mac, software*, pointing to a fast hard disk drive. The blue topic is about *car, article, write, bike*, implying a fast driving car or bike.

The user may wish to refine the query to find more documents of a particular sense or topic. This is where the second query type, *spatial query*, may be useful. Because every document is associated with a coordinate in a Euclidean space, the user can specify any coordinate on the visualization space, and we can return the “most relevant” or the closest documents within a radius. Continuing the previous example, if the user decides to focus on any one of the three clusters, she can execute a spatial query by double-clicking a specific coordinate. On the right-hand side of Figure 7.2, we show three small panels, illustrating the hypothetical scenarios in which the user is interested in one of the three localities. Each panel corresponds to a spatial query, centered at the coordinate marked with an ‘x’. This is akin to a visual interface for query reformulation.

[2]We return up to 50 most relevant results, which is a configurable number.

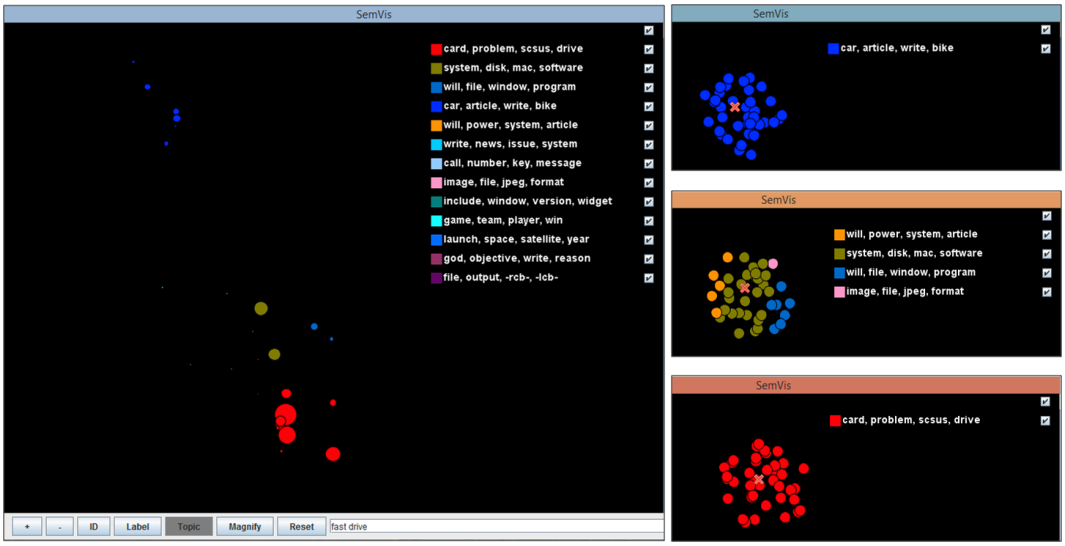


Figure 7.2: Search Interface: text query (left) and spatial queries (right). Topics of retrieved documents are shown in the legend.

7.3 Implementation

We briefly discuss the implementation details. Figure 7.3 shows the framework of SemVis. It has three main modules.

Given a document corpus, the first module, *Semantic Visualization*, helps to build a topic model and visualization of the corpus. We use SEMAFORE [75], for which the implementation is publicly available[3]. Nevertheless, the SemVis framework outlined in Figure 7.3 is still compatible with other algorithms such as PLSV [62] or SSE [74], or even pipelines of a topic model, e.g., LDA [14], followed by embedding, e.g., PE [61].

The second module, *Content and Spatial Indexing*, provides functions for indexing the corpus. We use Apache Lucene 6.4.1[4] implemented in Java for indexing the corpus. We index two kinds of information. The first is the document text content. The second is the visualization coordinates of documents.

The third module, *User Interface*, provides environment for users to interact and explore the corpus. It has two main functions which are browsing and searching. Users are provided with controls for performing these two tasks easily, such as drag-

[3] https://github.com/tuanlvm/SEMAFORE
[4] http://lucene.apache.org/core/6_4_1/

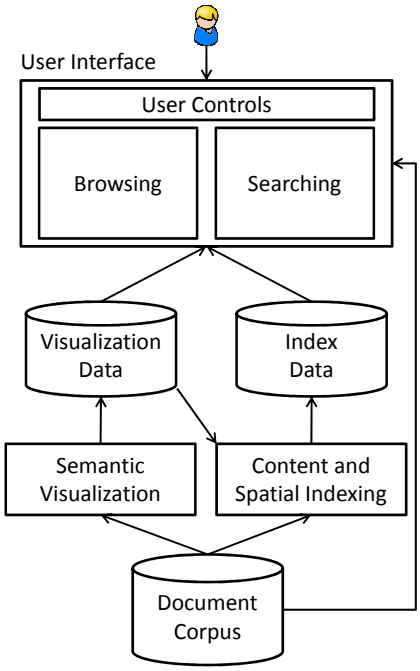


Figure 7.3: Framework of SemVis

ging, selecting, zooming and magnification, as well as a search box. To implement this module, we rely on Jung Library 2.0.1[5] written in Java.

Data. We rely on several English corpora. One is $20News$[1], which is also used to provide the illustrations in this paper. The documents are newsgroup articles from 20 classes. For $20News$, we sample a subset of 1000 documents which contains 50 documents per class. The class labels are not used for generating the topics and coordinates, but can still be visualized if desired..

We also rely on several text corpora obtained from $Cora$[6], which is a collection of abstracts of academic publications from various categories. From $Cora$, we carve out four smaller text corpora based on categories, namely *Data Structure* with 570 documents, *Hardware and Architecture* with 223 documents, *Machine Learning* with 1980 documents, and *Programming Language* with 1553 documents.

7.4 Conclusion

SemVis is a demonstrable system for interactive topical analysis via spatial visual-

[5] `http://jung.sourceforge.net/`
[6] http://people.cs.umass.edu/mccallum/data/cora-classify.tar.gz

ization, supported by the rigor of the underlying semantic visualization algorithms in deriving topics and coordinates. We hope to spark a continuing conversation on the applicability of semantic visualization for text analysis tasks.

Chapter 8

Word Clouds for Visual Comparison of Documents

Word cloud is a visualization form for text that is recognized for its aesthetic, social, and analytical values. Here, we are concerned with deepening its analytical value for visual comparison of documents. To aid comparative analysis of two or more documents, users need to be able to perceive similarities and differences among documents through their word clouds. However, as we are dealing with text, approaches that treat words independently may impede accurate discernment of similarities among word clouds containing different words of related meanings. We therefore motivate the principle of displaying related words in a coherent manner, and propose to realize it through modeling the latent aspects of words. Our WORD FLOCK solution brings together latent variable analysis for embedding and aspect modeling, and calibrated layout algorithm within a synchronized word cloud generation framework. We present the quantitative and qualitative results on real-life text corpora, showcasing how the word clouds are useful in preserving the information content of documents so as to allow more accurate visual comparison of documents.

8.1 Introduction

The abundance of text motivates the development of text analysis tools. One such need is to aid users in comparing several documents. For instance, a user may go through Web search results to determine how they differ from one another. A researcher needs to get an overview of various papers within a proceeding or a journal issue. Similar needs are faced by librarians or analysts. In such scenarios, users need to quickly gain a sense of whether several documents are similar.

Visualization may help in document comparison, by providing visual representations that allow users to perceive similarities and differences tangibly. There are various visualization forms. One is a scatterplot, showing documents as coordinates in a 2 or 3-dimensional space [70]. While it allows easy determination of whether two documents are similar (based on their distance in the scatterplot), it is not effective in conveying contents, which are important in providing meaning or justification to similarities.

Therefore, we focus on another visual representation, i.e., a *word cloud* displaying a subset of words within a document, by assigning greater visual prominence to more important words. Because a word cloud still displays the actual words, it is better at conveying the content of the corresponding document than a scatterplot. In addition, word cloud as a visualization form is extremely popular [121]. For instance, Wordle[1] has generated more than 1.4 million publicly posted word clouds [110].

Problem. We seek effective visual comparison of documents via word clouds. Ideally, documents with similar contents have word clouds of similar appearances. Traditional approaches fall short of this ideal, as word clouds of different documents are generated independently using a layout algorithm [106, 121]. Two documents may feature similar words that are placed in different colors and positions within their respective word clouds, placing a burden on the viewer in corroborating their similarities.

[1] `http://www.wordle.net/`

The state-of-the-art approach, *Word Storms* [22], employs a synchronized generation of the word clouds of all documents within a corpus. A word is expected to have the same color and position across word clouds. This aims to reduce the cognitive effort needed for comparing word clouds. However, it has two shortcomings. First, it only seeks to synchronize the appearance of each distinct word. This is problematic, as text frequently uses different words to refer to the same concept. Second, its synchronization of all word clouds imposes sizeable runtime requirement that prevents real-time generation of word clouds.

These issues arise because word clouds are still high-dimensional representations, with dimensionality the size of the vocabulary. Our insight is that a word cloud can encode information at *several dimensionalities* simultaneously. In addition to the actual words, the *position* of a word in the two-dimensional canvas space can reflect some two-dimensional word representation that captures relatedness among words, such as embedding that assigns nearby coordinates to "related" words, e.g., [70]. We can also have the word *color* reflect some k-dimensional word representation that captures k latent "aspects" of words. Each aspect may capture words of similar meaning or words often used together to describe a certain concept.

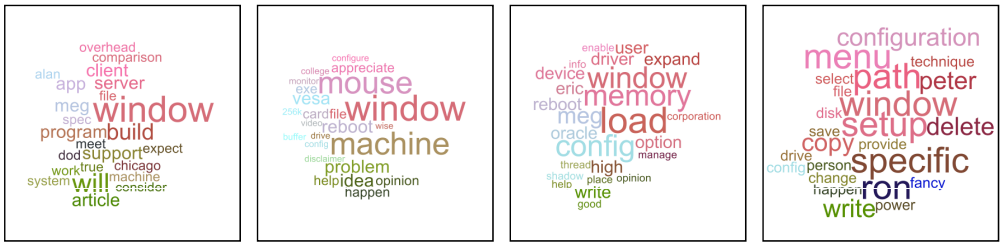
Approach. To realize the vision of multiple dimensionalities within a word cloud, we propose a technique called WORD FLOCK. The name is inspired by the idiom "birds of a feather flock together". In our case, *words* of a "feather" (similar aspects/colors) flock together (similar positions).

To illustrate how word clouds could provide effective visual comparison of documents, Figure 8.1 and Figure 8.2 show example word clouds generated by WORD FLOCK for documents in the $20News$ dataset[2]. The four word clouds in Figure 8.1 are for documents from the *comp.os.ms-windows.misc* category, pertaining to Windows computing. Words such as "window", "file", and "memory" have similar colors (reddish hue) and positions (top right) across the four word clouds. The four word clouds in Figure 8.2 are for documents from the *rec.sport.baseball* category,

[2] `http://web.ist.utl.pt/acardoso/datasets/`

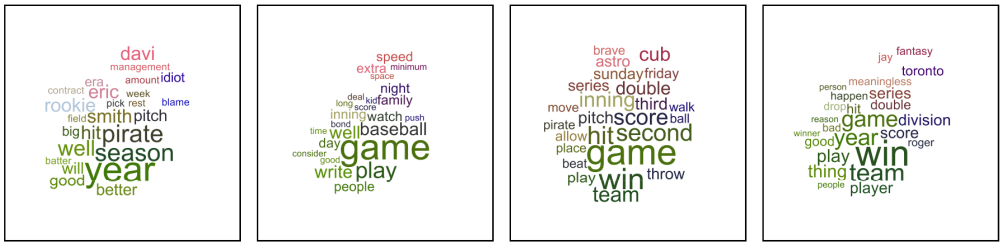
with words such as “hit”, “play”, “game”, and “team” sharing similar greenish hues and bottom left positions. Quick perusal of them is sufficient to convey which documents are similar (same category). Note that category labels had not been used in generating word clouds.

WORD FLOCK is underpinned by a novel approach of employing latent variable analysis for word cloud generation. Given a vocabulary of words, we seek to learn their latent representations in two forms. The first is coordinate representation in a two-dimensional space, which is derived from a latent embedding model. The second is a probability distribution over k latent aspects. This representation learning phase can be done offline once for a given vocabulary. Thereafter, we generate a word cloud for a document online, incorporating these representations in a calibrated layout algorithm.



a. Doc101 by Word Flock b. Doc111 by Word Flock c. Doc132 by Word Flock d. Doc136 by Word Flock

Figure 8.1: Word clouds by WORD FLOCK for 4 documents from *comp.os.ms-windows.misc* of $20News$ (best seen in color)



a. Doc472 by Word Flock b. Doc473 by Word Flock c. Doc495 by Word Flock d. Doc499 by Word Flock

Figure 8.2: Word clouds by WORD FLOCK for 4 documents from *rec.sport.baseball* of $20News$ (best seen in color)

Contributions. We make the following contributions:

- WORD FLOCK is the first to integrate *two levels* of “synchronization” principles for word clouds: *similar documents* share similar word clouds, and *related words* of the same latent aspects are displayed similarly.

- WORD FLOCK is novel in employing latent variable analysis through *joint* usage of *embedding* (synchronized positioning) and *latent aspect modeling* (coloring) among words of similar concepts.
- Comprehensive experiments on real-life document corpora showcase the effectiveness of WORD FLOCK via an empirical comparison to the baseline *Word Storms* [22], on objective quantitative metrics, as well as a user study.
- The *two-phase* approach attains the synchronization of word representations *offline*, so as to obviate the need to generate all word clouds together. This allows an *online* generation of individual word clouds at near-instant speed, which eludes the baseline *Word Storms*.

8.2 Overview of WORD FLOCK

Problem Statement. We assume that the scope is defined by a vocabulary $\mathcal{W}$, the set of words that could appear in any word cloud in a corpus. As input, we are given a corpus of documents $\mathcal{D} = \{d_1, d_2, \ldots, d_N\}$. Every document d_n consists of words drawn from the vocabulary $\mathcal{W}$. The objective is to generate a set of word clouds $\{C_1, C_2, \ldots, C_N\}$, one for each document in $\mathcal{D}$, so as to aid visual comparison of documents through their respective word clouds.

We display each word in a cloud according to a number of visual attributes. There are various visual variables within a word cloud, and in general different visual features may be good for different types of information [7]. Here, we make use of three visual attributes: font size, color, and position. Each word w in C_n for d_n is associated with a tuple $\langle s_w, p_w, l_w \rangle$, where s_w is the font size, p_w is the word position in terms of 2D coordinates, and l_w is the color. Fixing the orientation to horizontal prevents the cognitive overload of reading randomly oriented words.

Solution Framework. We now discuss the principles for the design of our word cloud algorithm WORD FLOCK.

Principle #1: Display related words similarly. Words in a document are not

independent. Some words may capture a particular concept or aspect. It is far easier to understand a word cloud in terms of a small number of coherent concepts, rather than in terms of a large number of independent words. Among the visual attributes, we rely on *position* (p_w) and *color* (l_w). Through a dimensionality reduction task known as *embedding*, we seek to derive coordinates for each word in a latent two-dimensional space, such that two related words are nearby in this space. The continuous spectrum of color is also appropriate to convey the underlying aspects or concepts of words. We discover these aspects automatically through *latent aspect modeling*. We pursue these tasks jointly, computing them offline once for the corpus to support the synchronization of positions and colors across all word clouds.

Principle #2: Similar documents have similar word clouds. Aimed squarely at aiding visual comparison of documents, this principle motivates the coherent appearances of word clouds of similar documents. This is achieved by infusing and calibrating the layout algorithm with the coordinated positions and colors determined by the embedding and latent aspect. Consequently, the online layout of each new word cloud requires only a small marginal computational cost.

In the next two sections, we describe the two phases of the WORD FLOCK algorithm. Due to the limitation of canvas space, conventionally only the more important words are included [106]. Word prominence is indicated by the *font size* (s_w). There are various notions of "importance" of a word. Without loss of generality, here we use the well-accepted term frequency (*tf*), after removing stop words.

8.3 Embedding and Latent Aspect Modeling

The objective is to derive coordinates in a 2D space, as well as a k latent aspects of words. To do so, we need to associate words with informative feature space representation. By feature space representation, we refer to a feature vector $\mathbf{w}$ for each word w, capturing information on how words are associated with one another.

To express $\mathbf{w}$, each word w is considered as a "pseudo-document" containing all words that w co-occurs with. Each $\mathbf{w}$ is expressed in terms of word counts where each element of $\mathbf{w}$ corresponds to how frequently another word v co-occurs with w in some reference corpus. This corpus may be a large independent corpus (e.g., Wikipedia), or the specific corpus of interest. The co-occurence of two words can be determined by their appearance within a document or a window. This way of modeling is consistent with [131]. Intuitively, two different words with similar co-occurrence counts are likely to share a similar meaning.

The task is to reduce the high-dimensional $\{\mathbf{w}\}_{w\in\mathcal{W}}$ to lower-dimensional representations. In obtaining 2D coordinates $\{x_w\}_{w\in\mathcal{W}}$ for each word, the aim is similar to embedding, whereas in obtaining latent aspects of words, it is feasible to learn it from word cooccurrences [131]. While embedding and latent aspect modeling could be done independently, recent works [62, 72] show that it is beneficial to join the two tasks into a single joint model to ensure consistency in objectives. We therefore adapt the state-of-the-art model SEMAFORE (described in Chapter 3), originally designed for topics in documents, now to model latent word aspects. We apply the generative process of SEMAFORE to all "pseudo-documents" that represent for words. The parameters are learned from $\{\mathbf{w}\}_{w\in\mathcal{W}}$ based on maximum a posteriori estimation through EM [39]. The outputs are the coordinates x_w, as well as probability distribution over k latent aspects $\{\mathrm{P}(z|x_w,\Phi)\}_{z=1}^{k}$, for every word w in the vocabulary $\mathcal{W}$. These outputs underpin the online generation of word clouds described next.

8.4 Word Cloud Layout with Scale Calibration

We include only top M words in a document by weight (e.g., term frequency). The font size s_w is controlled by this weight.

The color of each word l_w is determined based on its aspect probabilities $\{\mathrm{P}(z|x_w,\Phi)\}_{z=1}^{k}$ from the first phase. l_w is expressed in terms of a color representation, such as RGB.

There are different schemes for transforming the aspect probabilities into word colors. For instance, we could assign each aspect a color, and for each word we take the weighted average of its aspects' colors, or that of the strongest aspect. However, these approaches would require associating a color to each topic, which itself is not a straightforward task.

A better approach to assign colors to words automatically, which we adopt here, is to have a color map based on the aspect probabilities. Since RGB colors lie in a three-dimensional (3D) space (i.e., R, G, and B axes), we employ PE [61] to find the embedding of the k aspect probabilities of all words into a 3D space. We then map these 3D coordinates to the RGB space using min-max normalization to find the word colors. This way, words with similar aspect probabilities would share similar colors.

The word position p_w should be similar, if not identical, to the word coordinate x_w from the first phase. There are two issues. First, the canvas space over which p_w is defined has a different scale from the embeding space of x_w. We therefore introduce a scaling factor Γ, i.e., $p_w = \Gamma \times x_w$.

Second, even if the former could be calibrated, some words may have similar x_w's, causing overcrowding. This is not unique to us. Classically, word clouds have had to deal with how to position words in a compact and non-overlapping way [106]. Similarly to *Word Storms* [22], we build on Wordle's algorithm. Our layout algorithm is shown in Algorithm 3. It works in a greedy and incremental manner. As indicated previously, our requirement is different in having to deal with the scale calibration issue.

The scaling factor Γ is calibrated so as to optimize an objective function that captures the aesthetic quality. First, it is desired that a word cloud is compact, expressed in terms of smaller distances of the final word positions $p_w{}'$ from the origin. Second, it is desired that similar words are placed close to one another. Suppose that η_w is the set of m closest neighboring words of w in C_n (based on their embedding coordinates x_w's). We would like w to have a final position $p_w{}'$

Algorithm 3 Spiral Algorithm with Calibration of Scaling

Require: M_n words for each document d_n, 2D coordinates $\{x_w\}_{w \in \mathcal{W}}$ obtained from embedding in offline phase, and a set of scaling factors Γ.

Ensure: Final scaling factor γ and for each document d_n, positions $\mathbf{p}_n$ of M_n words in the word cloud of d_n.

```
for each scaling factor γ ∈ Γ do
    for all document d_n, n ∈ {1, ..., N} do
        for all words w ∈ {w_1, ..., w_{M_n}} do
            Initialize p_w = γ × x_w
            while p_w intersects any previous words do
                Move p_w one step along a spiral path
            end while
        end for
    end for
    Compute the objective function value in Equation 8.1.
    Store the γ and all p_n with the best objective function value so far.
end for
```

that is as close as possible to its neighbors in η_w. To achieve this, we propose the objective function in Equation 8.1.

$$\sum_{n=1}^{|\mathcal{D}|} \sum_{w \in C_n} \left[||{p_w}'||^2 + \frac{1}{|\eta_w|} \sum_{v \in \eta_w} ||{p_w}' - {p_v}'||^2 \right] \tag{8.1}$$

During the calibration process, we investigate various scaling factors Γ to minimize Equation 8.1. We further advocate an offline calibration to arrive at a single Γ for any new document. There is usually a single scaling factor that works for most documents. This also saves time in the online generation of word cloud that only needs to run the layout algorithm.

8.5 Evaluation

Evaluating word clouds is challenging because of the various purposes that they could be aimed at, e.g., gisting, word recall [102]. We focus on the task of visual comparison of documents. This involves a multi-prong approach, including qualitative examples, quantitative metrics involving objective ground truth, as well as a user study.

8.5.1 Experimental Setup

First, we describe the experimental setup.

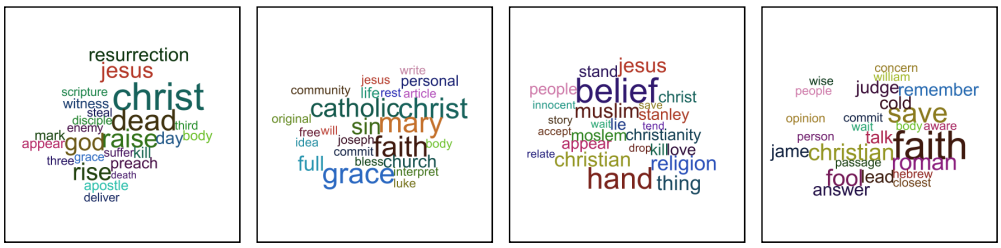
Datasets. We rely on two publicly available datasets of text documents, where each document has a known category label. These labels are not required for training. Rather, they are used in evaluation as an objective proxy for defining what constitute "similar" documents (i.e., same category). The two datasets[3] are: $20News$ containing newsgroup documents partitioned into 20 classes, and $Reuters$ containing newswire articles from 8 classes. To create a balanced dataset, we sample 50 documents from each class, resulting in 1000 and 400 documents respectively. After removing stopwords and infrequent words (< 5 occurrences), the vocabulary consists of 3744 words for $20News$, and 1933 words for $Reuters$.

Methods. WORD FLOCK incorporates synchronization principles for both documents and words for visual comparison of documents. The most appropriate baseline for this task is *Word Storm* [22], which applies synchronization of word clouds across documents, but does not address relatedness among words. We use the authors' implementation in GitHub[4].

As longer documents may result in word clouds that are too "busy", we show up to twenty five words based on weight. The same words are visualized by the comparative methods. For WORD FLOCK, we also need to specify the number of latent aspects of words k. We experiment with k in the range [5, 25]. For each k, we tune the scaling factor Γ to minimize the Equation 8.1. Γ is tuned with $|\eta_w| = 3$. The optimal Γ ranges from 20 to 25. Through experimentation, we discover that $k = 20$ works best for both $20News$ and $Reuters$. Note that the number of colors in a word cloud generated by Word Flock is not directly determined by k. We map the aspect distribution of words across the full RGB spectrum (see Section 8.4). If all words in a document were distinctly different, they would show up with different colors. However, words appearing within a document tend to be related. Usually

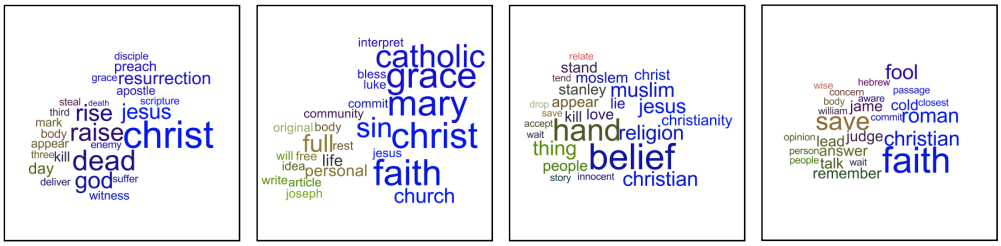
[3] `http://web.ist.utl.pt/acardoso/datasets/`
[4] `https://github.com/quimcastella/WordStorm`

only a small number of colors are seen in a word cloud by WORD FLOCK due to the relative coherence of words in a document.



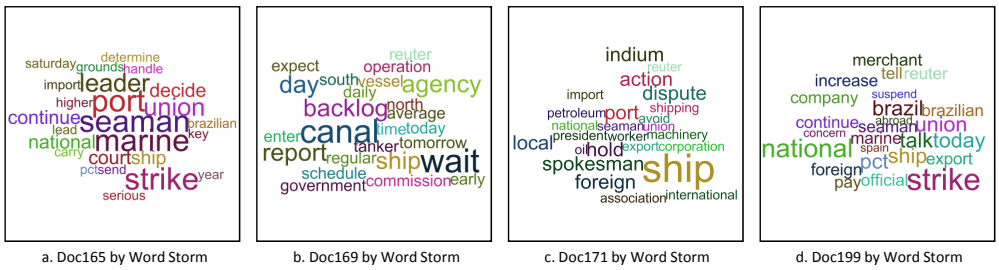
a. Doc762 by Word Storm b. Doc770 by Word Storm c. Doc788 by Word Storm d. Doc790 by Word Storm

Figure 8.3: Word clouds by *Word Storm* for 4 documents from *soc.religion.christian* of $20News$ (best seen in color)



a. Doc762 by Word Flock b. Doc770 by Word Flock c. Doc788 by Word Flock d. Doc790 by Word Flock

Figure 8.4: Word clouds by WORD FLOCK for 4 documents from *soc.religion.christian* of $20News$ (best seen in color)

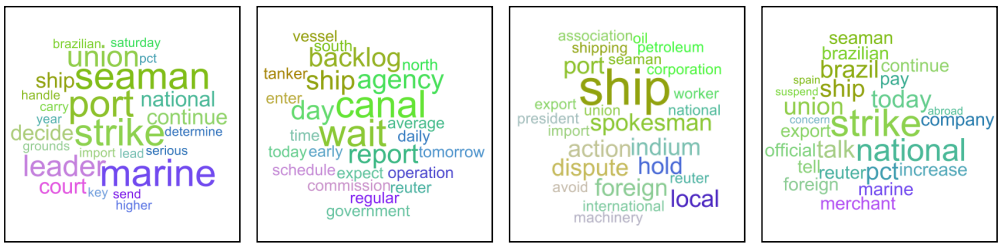


a. Doc165 by Word Storm b. Doc169 by Word Storm c. Doc171 by Word Storm d. Doc199 by Word Storm

Figure 8.5: Word clouds by *Word Storm* for 4 documents from *ship* of $Reuters$ (best seen in color)

8.5.2 Qualitative Analysis

We begin with an exploration of example word clouds from $20News$. Figure 8.3 shows *Word Storm*'s word clouds for four documents from the *soc.religion.christian* category. Figure 8.4 shows the corresponding word clouds by WORD FLOCK. There are several words that capture the semantic meaning of the documents, such as



a. Doc165 by Word Flock b. Doc169 by Word Flock c. Doc171 by Word Flock d. Doc199 by Word Flock

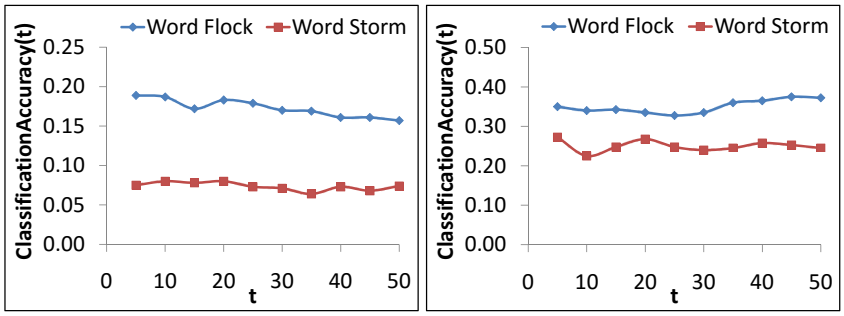
Figure 8.6: Word clouds by WORD FLOCK for 4 documents from *ship* of $Reuters$ (best seen in color)

"god", "christ", "jesus", "faith" and "christian". *Word Storm* disperses these words across each word cloud, because it does not address their relatedness and relies on having the exact same words for comparison, which fails when documents use different words. In Figure 8.3(a) and Figure 8.3(d), *Word Storm* uses the same location and color for "body", but other words are different across the two clouds. In contrast, WORD FLOCK groups related words in similar positions and colors, yielding four strikingly coherent word clouds. This is also evident from the previous examples of WORD FLOCK's word clouds for *comp.os.ms-windows.misc* category in Figure 8.1 and for *rec.sport.baseball* in Figure 8.2.

Examples from $Reuters$ also reveal the contrast between *Word Storm* and WORD FLOCK. Figures 8.5 and 8.6 shows the respective word clouds by *Word Storm* and WORD FLOCK for four documents from the *ship* category of $Reuters$. While related words such as "ship", "vessel", "canal", "port", "seaman", and "shipping" are grouped together by WORD FLOCK, these words are dispersed and have different colors in *Word Storm*'s word clouds.

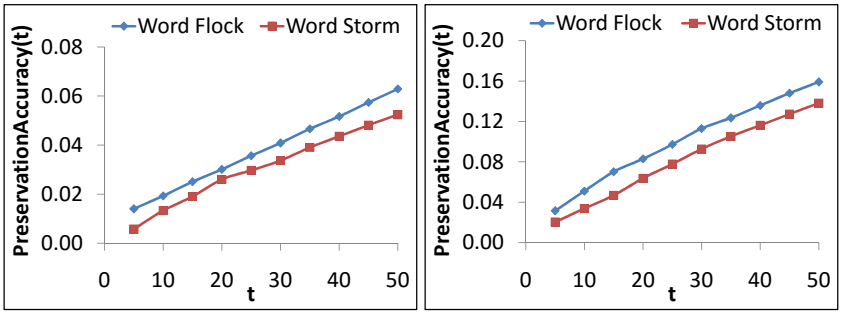
8.5.3 Classification

We seek further evidence through an automatic evaluation that offers a repeatable and objective validation. Each word cloud is represented as a vector of image pixels, where each pixel is represented by its RGB value. We validate how well the pixel representation of the word cloud images may be used as features in classification, with the simple nearest neighbors classifier. For every document, we hide



a. *20News* b. *Reuters*

Figure 8.7: $Classification Accuracy(t)$ for various t



a. *20News* b. *Reuters*

Figure 8.8: $Preservation Accuracy(t)$ for various t

its class label. We then identify its t-nearest neighbors based on cosine similarity over the pixel representations, and assign the document the majority class among its neighbors. $Classification Accuracy(t)$ is the fraction of documents for which the classification derives the correct labels. We average the accuracies across ten runs. This is merely for evaluation, and is not meant as a technique for document classification.

Figure 8.7(a) shows that WORD FLOCK has significantly higher accuracies than *Word Storm* on $20News$. A random classifier would have an accuracy of 0.05. *Word Storm* performs at around 0.08. WORD FLOCK attains more than 100% increase in accuracy over *Word Storm*. For $Reuters$ in Figure 8.7(b), WORD FLOCK is also better. The improvements over *Word Storm* are statistically significant at 0.01 level. The performance is closer because $Reuters$ is an "easier" dataset (a random classifier would attain 0.125 accuracy).

8.5.4 Neighborhood Preservation

Ideally, a word cloud should be a faithful representation of its original document. In the next evaluation task, given a query document, we seek to retrieve the t most similar documents. We consider the ground truth to be the most similar documents over the original text representation of documents (i.e., cosine similarity over the 25-word term frequency vectors). We define $PreservationAccuracy(t)$ as the fraction of t ground-truth documents that are "preserved" or identified among the t retrieved images (based on cosine similarity over the pixel representation). Figure 8.8(a) for $20News$ and Figure 8.8(b) for $Reuters$ show that WORD FLOCK has higher preservation accuracies than *Word Storm* over various t's. This indicates that the resulting word clouds by WORD FLOCK better preserve the similarities among the original documents. The difference between the two methods is statistically significant at 0.01 level in all cases.

8.5.5 User Study

We conduct a pilot user study on $20News$ to confirm our results in the quantitative analyses. The study involves two types of questions/tasks related to visual comparison of documents, which were similar to the study conducted in [22]. For the first type, each user views six clouds, and is asked to identify the most different one. Among the six, five come from the same category, and one (the ground truth) comes from a different category. For the second type, each user views one query cloud and six answer clouds, and is asked to identify which answer cloud is most similar to the query cloud. Among the six, only one (the ground truth) comes from the same category as the query.

For each type, a user has to complete 30 multiple-choice questions, with a time limit of 30 seconds per question. The clouds for each question are generated either by *Word Storm* or by WORD FLOCK and each user is randomly presented with one of the two versions. There are 6 users involved in the study. Therefore, each question is answered 3 times using *Word Storm* and 3 times using WORD FLOCK

Question	Accuracy (%)		Time (s)	
	Word Storm	WORD FLOCK	*Word Storm*	WORD FLOCK
Type 1: Select the most different cloud	71.1	**78.9**	15.7	**14.6**
Type 2: Select the cloud most similar to a given cloud	63.6	**70.0**	16.6	**16.1**

Table 8.1: Results of the user study (bold is better)

. The 6 clouds are sorted randomly and the users do not know how many methods there are, or which method is used for each question. We track accuracy and average time to answer each question.

Table 8.1 summarizes the results of the user study. For Type 1 questions, WORD FLOCK helps users to attain a higher accuracy, 78.9% as compared to 71.1% for *Word Storm*, and with less time too (the time spent to answer was reduced by about a second). For Type 2, WORD FLOCK also has a higher accuracy of 70.0% vs. 63.6% for *Word Storm*, again with slightly improved timing. The results are quite consistent among users, with 5 out of 6 users achieving higher accuracy with WORD FLOCK than with *Word Storm* for both types.

8.5.6 Brief Comment on Efficiency

We comment briefly on one efficiency advantage of WORD FLOCK over *Word Storm*, in our ability to generate individual word clouds in an online fashion. *Word Storm* must process all word clouds together. For the 1000 documents in $20News$, it requires 15 minutes on Intel Core i7 2.4Ghz machine with 8GB memory. Adding a new document requires looping over all the previously generated word clouds again to ensure consistency. In contrast, WORD FLOCK achieves synchronization offline, so as to enable online generation of each word cloud independently, which requires only between 100 to 200 millisecond for every new word cloud.

8.6 Conclusion

We are interested in producing effective word clouds for visual comparison of documents within a corpus. The key idea is to construct word clouds to show related words with similar appearances, to enhance cognition of aspects across multiple word clouds. WORD FLOCK achieves this via latent variable analysis, including offline embedding and latent aspect modeling, followed by online generation of word clouds. Through multi-faceted evaluation on two public datasets, we show evident outperformance by WORD FLOCK over the baseline.

There are several potential directions for future work. One direction is to further enrich the word clouds by encoding some useful information in other visual attributes such as word orientation. Another direction is to further investigate the use of word clouds in specific application scenarios such as document retrieval or document summarization.

Chapter 9

Conclusion and Future Work

9.1 Summary

This thesis considers the problem of visualizing document similarities on a scatterplot. Classical approaches to document visualization treat this as a dimensionality reduction problem where we want to directly reduce high-dimensional representation of documents (i.e., bags of words) into visualizable two or three dimensions. This thesis considers a new approach where documents have an intermediate representation in topic space, between original space and visualization space. This approach seeks to couple visualization with topic modeling by jointly modeling visualization and topics, which is referred to as the task of semantic visualization. This dissertation focuses on building probabilistic models for semantic visualization by modeling document relationship and document representation in addition to their texts. The objective is to improve the quality of the scatterplot visualization while maintaining topic quality.

In the first part for modeling document relationship, we propose two semantic visualization models. The first one, SEMAFORE is for modeling neighborhood structure (Chapter 3). The second one, PLANE is for modeling networked documents (Chapter 4). Experiments on real-life datasets show that SEMAFORE and PLANE significantly outperform the baselines in terms of visualization quality and

accuracy, while having a similar topic quality. This provides evidence that neighborhood structure and network structure, together with joint modeling of topics and visualization, are important for semantic visualization.

In the second part for modeling document representation, we consider different types of representation. In Chapter 5, we propose SSE, a semantic visualization model for spherical representation. Comprehensive experiments on benchmark datasets show that SSE shows significantly improved performance when compared to existing state-of-the-art baselines in terms of visualization quality, as well as topic interpretability. Another type of representation that we consider in Chapter 6 is bag of word vectors. Word vectors are known for its ability to deal with sparsity problem in short texts. Therefore, we propose a semantic visualization model called GaussianSV using word vectors for visualizing short texts. GaussianSV performs well on real-life short text datasets against semantic visualization baselines, as well as against pipelined baselines, validating both the value of modeling word vectors with semantic visualization. The good performance of these two models shows that by modeling different representations, we can improve scatterplot visualization quality for different types of dataset.

Finally, we attempt to find application of semantic visualization in various problems. In Chapter 7, we develop a system called SemVis for navigating a text corpus interactively and topically via browsing and searching. SemVis is supported by the rigor of the underlying semantic visualization algorithms in deriving topics and coordinates. Another application of semantic visualization is for single document visualization. In Chapter 8, we propose a new framework called WORD FLOCK for visual comparison of documents using word clouds. In this framework, a semantic visualization method is used to visualize words which are represented as pseudo-documents.

9.2 Future Work

For future work, we will explore different ways to combine proposed approaches, which simultaneously model neighborhood and network structures and different document representations. Another future work is to explore more use cases for SemVis. From this exploration, we may have new ideas to improve the usability of SemVis for supporting interactive topical exploration.

There are other interesting directions for future work. One direction is to focus on applications of semantic visualization. We could make use of semantic visualization for building an interactive topic model where users can provide feedback for learning a better topic model. The interface provided by semantic visualization, in which each document and topic have a coordinate, could be a promising way for users to provide feedback. By changing coordinates of documents and topics, users can tune the underlying topic model for achieving a more relevant output. Other potential applications of semantic visualization are to support a document organizer system or an augmented retrieval system. The visualization could potentially help in assigning categories to documents, by showing how closely related documents have been labeled. For augmented retrieval system, given a query, the results may include not just relevant documents, but also other similar documents (neighbors in the visualization).

Another interesting direction for future work is to extend semantic visualization for dealing with large scale dataset. To perform large scale semantic visualization, one approach we can take is to use faster algorithms for inference such as stochastic variational inference [56] to improve the training speed of the visualization algorithms. Other approaches such as online learning or creating an implementation to run on a cloud of computers to speed up the algorithm could be promising.

Another way to extend semantic visualization is focusing on its visualization form. Currently, we visualize a document collection on a single scatterplot. Given that the document collection is large, the curse of dimensionality can cause problems of representation and preservation where we cannot express and preserve faith-

fully all the relationships among documents by just using only one scatterplot. Therefore, one promising extension is using multiple scatterplots for semantic visualization. This extension may provide a way to solve the large scale problem as well. For example, we can visualize multiple scatterplots at the same time in a parallel fashion.

Bibliography

[1] Airoldi, E. M.; Blei, D. M.; Fienberg, S. E.; and Xing, E. P. 2009. Mixed membership stochastic blockmodels. In *NIPS*. 4.2

[2] Akkucuk, U., and Carroll, J. D. 2006. Paramap vs. isomap: a comparison of two nonlinear mapping algorithms. *Journal of Classification* 23(2):221–254. 3.5.1

[3] Bai, L.; Guo, J.; Lan, Y.; and Cheng, X. 2014. Local linear matrix factorization for document modeling. In *Advances in Information Retrieval*. Springer. 398–411. 3.3

[4] Banerjee, A.; Dhillon, I. S.; Ghosh, J.; and Sra, S. 2005. Clustering on the unit hypersphere using von Mises-Fisher distributions. In *JMLR*. 5.1, 5.2.1, 5.2.1

[5] Barth, L.; Kobourov, S. G.; and Pupyrev, S. 2014. Experimental comparison of semantic word clouds. In *Experimental Algorithms*. Springer. 247–258. 2.2.2

[6] Bastian, M.; Heymann, S.; Jacomy, M.; et al. 2009. Gephi: an open source software for exploring and manipulating networks. *ICWSM*. 4.2

[7] Bateman, S.; Gutwin, C.; and Nacenta, M. 2008. Seeing things in the clouds: The effect of visual features on tag cloud selections. In *HT*. 8.2

[8] Belkin, M., and Niyogi, P. 2001. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems (NIPS)*, volume 14, 585–591. 2

[9] Belkin, M., and Niyogi, P. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation* 15(6):1373–1396. 1.2.1, 3.1, 5.1

[10] Belkin, M.; Niyogi, P.; and Sindhwani, V. 2006. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research (JMLR)* 7:2399–2434. 1.2.1, 3.1, 3.1.1, 3.3.1, 3.3.1, 3.3.1

[11] Bernstein, M. S.; Suh, B.; Hong, L.; Chen, J.; Kairam, S.; and Chi, E. H. 2010. Eddi: Interactive topic-based browsing of social status streams. In *UIST*, 303–312. ACM. 2.2.2

[12] Bishop, C. M.; Svensén, M.; and Williams, C. K. 1998. GTM: The generative topographic mapping. *Neural Computation* 10(1):215–234. 2.1.1

[13] Bishop, C. M. 1995. *Neural Networks for Pattern Recognition*. Oxford University Press. 3.2.2, 4.3, 4.3, 5.2.1

[14] Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research (JMLR)* 3:993–1022. 1, 1.2.2, 2.1.1, 3.5.4, 4.1, 4.1.1, 4.2, 4.3, 5.1, 5.2.1, 5.3.2, 5.3.4, 6.2.1, 7.1, 7.3

[15] Brants, T., and Franz, A. 2006. *Web 1T 5-gram Version 1*. Linguistic Data Consortium, Philadelphia. 3.5.5, 4.5.4, 5.3.4

[16] Buhmann, M. D. 2000. Radial basis functions. *Acta Numerica 2000* 9. 3.2.2, 5.2.1

[17] Cai, D.; Mei, Q.; Han, J.; and Zhai, C. 2008. Modeling hidden topics on document manifold. In *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM)*. 1.2.1, 3.1, 3.2.1, 3.3, 3.3.1, 3.3.1

[18] Cai, D.; Wang, X.; and He, X. 2009. Probabilistic dyadic data analysis with local and global consistency. In *Proceedings of the International Conference on Machine Learning (ICML)*. 1.2.1, 3.1, 3.2.1, 3.3, 3.3.1, 3.3.1

[19] Cardoso-Cachopo, A. 2007a. Improving Methods for Single-label Text Categorization. PhD Thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa. 3.5.1

[20] Cardoso-Cachopo, A. 2007b. Improving Methods for Single-label Text Categorization. PdD Thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa. 5.3.1

[21] Carey, C., and Mahadevan, S. 2014. Manifold spanning graphs. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*. 3.3.2

[22] Castella, Q., and Sutton, C. 2014. Word storms: Multiples of word clouds for visual comparison of documents. In *WWW*. (document), 2.2.2, 2.10, 8.1, 8.1, 8.4, 8.5.1, 8.5.5

[23] Chaney, A. J.-B., and Blei, D. M. 2012. Visualizing topic models. In *Proceedings of the International AAAI Conference on Web and Social Media (ICWSM)*. 2.1.1, 2.1.2, 4.2

[24] Chang, J., and Blei, D. M. 2009. Relational topic models for document networks. In *AISTATS*. 4.2, 4.3, 4.3, 4.5.2, 4.5.4, 4.5.4

[25] Chang, J.; Gerrish, S.; Wang, C.; Boyd-Graber, J. L.; and Blei, D. M. 2009. Reading tea leaves: How humans interpret topic models. In *Advances in Neural Information Processing Systems*, 288–296. 3.5.5

[26] Chen, L., and Buja, A. 2009. Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis. *Journal of the American Statistical Association* 104(485):209–219. 3.5.1

[27] Chen, Y.-X.; Santamaría, R.; Butz, A.; and Therón, R. 2009. Tagclusters: Semantic aggregation of collaborative tags beyond tagclouds. In *Smart Graphics*, 56–67. Springer. 2.2.2

[28] Choo, J.; Lee, C.; Reddy, C. K.; and Park, H. 2013. Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE Transactions on Visualization and Computer Graphics* 19(12):1992–2001. 1, 2.1.1, 2.1.2, 4.1.1

[29] Chuang, J.; Manning, C. D.; and Heer, J. 2012. Termite: visualization techniques for assessing textual topic models. In *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI)*, 74–77. 2.1.1, 2.1.2, 4.2

[30] Coifman, R. R., and Lafon, S. 2006. Diffusion maps. *Applied and Computa-*

tional Harmonic Analysis 21(1):5 – 30. 1

[31] Collins, C.; Carpendale, S.; and Penn, G. 2009. Docuburst: Visualizing document content using language structure. In *Computer graphics forum*. Wiley Online Library. (document), 2.2.2, 2.2.2, 2.11

[32] Comon, P. 1994. Independent component analysis, a new concept? *Signal Processing* 36(3):287–314. 2.1.1

[33] Coppersmith, G., and Kelly, E. 2014. Dynamic wordclouds and vennclouds for exploratory data analysis. *Sponsor: Idibon* 22. 2.2.2

[34] Craswell, N. 2009. Mean reciprocal rank. In *Encyclopedia of Database Systems*. 4.5.4

[35] Cui, W.; Wu, Y.; Liu, S.; Wei, F.; Zhou, M. X.; and Qu, H. 2010. Context preserving dynamic word cloud visualization. In *PacificVis*. 2.2.2

[36] Cui, W.; Liu, S.; Tan, L.; Shi, C.; Song, Y.; Gao, Z.; Qu, H.; and Tong, X. 2011. Textflow: Towards better understanding of evolving topics in text. *IEEE transactions on visualization and computer graphics* 17(12):2412–2421. 2.1.2

[37] Das, R.; Zaheer, M.; and Dyer, C. 2015. Gaussian lda for topic models with word embeddings. In *Proceedings of the 53nd Annual Meeting of the Association for Computational Linguistics*. 1.2.2, 6.1, 6.2.1, 6.3.1

[38] DeCamp, P.; Frid-Jimenez, A.; Guiness, J.; and Roy, D. 2005. Gist icons: Seeing meaning in large bodies of literature. In *Proceedings of IEEE Symposium on Information Visualization*. 2.2.2, 2.2.2

[39] Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B* 39(1):1–38. 3.4, 4.4, 6.2.2, 8.3

[40] Dou, W.; Wang, X.; Chang, R.; and Ribarsky, W. 2011. ParallelTopics: A probabilistic approach to exploring document collections. In *VAST*. 2.1.1, 2.1.2

[41] Dumais, S.; Furnas, G.; Landauer, T.; Deerwester, S.; Deerwester, S.; et al. 1995. Latent semantic indexing. In *Proceedings of the Text Retrieval Conference*. 2.1.1

[42] Ellson, J.; Gansner, E.; Koutsofios, L.; North, S. C.; and Woodhull, G. 2002. Graphviz - open source graph drawing tools. In *Graph Drawing*. 4.2

[43] Fellbaum, C., ed. 1998. *WordNet: an electronic lexical database*. MIT Press. 2.2.2

[44] Fisher, R. A. 1936. The use of multiple measurements in taxonomic problems. *Annals of Eugenics* 7(2):179–188. 2.1.1

[45] Fruchterman, T. M., and Reingold, E. M. 1991. Graph drawing by force-directed placement. *Software: Practice and Experience*. 1.2.1, 4.2

[46] Fujimura, K.; Fujimura, S.; Matsubayashi, T.; Yamada, T.; and Okuda, H. 2008. Topigraphy: Visualization for large-scale tag clouds. In *WWW*, 1087–1088. ACM. 2.2.2

[47] Goldenberg, A.; Zheng, A. X.; Fienberg, S. E.; and Airoldi, E. M. 2010. A survey of statistical network models. *FTML*. 4.2

[48] Golub, G. H., and Van Loan, C. F. 2012. *Matrix Computations*, volume 3.

JHU Press. 2.1.1, 4.2, 4.5.2

[49] Greene, D., and Cunningham, P. 2006. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *ICML*, 377–384. 6.3.1

[50] Gretarsson, B.; O'donovan, J.; Bostandjiev, S.; Höllerer, T.; Asuncion, A.; Newman, D.; and Smyth, P. 2012. TopicNets: Visual analysis of large text corpora with topic modeling. *ACM Transactions on Intelligent Systems and Technology (TIST)* 3(2):23. (document), 2.1.1, 2.1.2, 2.5, 4.2

[51] Hassan-Montero, Y., and Herrero-Solana, V. 2006. Improving tag-clouds as visual information retrieval interfaces. In *InSciT*. 2.2.2

[52] Havre, S.; Hetzler, E.; Whitney, P.; and Nowell, L. 2002. Themeriver: Visualizing thematic changes in large document collections. *IEEE transactions on visualization and computer graphics* 8(1):9–20. (document), 2.4, 2.1.2

[53] Hearst, M. A. 1995. Tilebars: visualization of term distribution information in full text information access. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, 59–66. ACM Press/Addison-Wesley Publishing Co. 2.1.2

[54] Hein, M.; Audibert, J.-y.; and Luxburg, U. V. 2007. Graph laplacians and their convergence on random neighborhood graphs. In *Journal of Machine Learning Research*, 1325–1368. 1

[55] Hinton, G. E., and Roweis, S. T. 2002. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems (NIPS)*, 833–840. 2.1.1, 5.1, 5.2.1

[56] Hoffman, M. D.; Blei, D. M.; Wang, C.; and Paisley, J. W. 2013. Stochastic variational inference. *Journal of Machine Learning Research* 14(1):1303–1347. 9.2

[57] Hofmann, T. 1999. Probabilistic latent semantic indexing. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 50–57. 1, 1.1.2, 2.1.1, 2.1.1, 3.1, 5.1, 5.2.1, 5.3.2, 6.2.1, 7.1

[58] Hu, W., and Tsujii, J. 2016. A latent concept topic model for robust topic inference using word embeddings. In *The 54th Annual Meeting of the Association for Computational Linguistics*, 380. 1.2.2, 6.1, 6.3.1

[59] Hu, Y.; Boyd-Graber, J.; Satinoff, B.; and Smith, A. 2014. Interactive topic modeling. *Machine Learning* 95(3):423–469. 1, 2.1.1, 2.1.2

[60] Huh, S., and Fienberg, S. E. 2012. Discriminative topic modeling based on manifold learning. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 5(4):20. 1.2.1, 3.1, 3.2.1, 3.3, 3.3.1

[61] Iwata, T.; Saito, K.; Ueda, N.; Stromsten, S.; Griffiths, T. L.; and Tenenbaum, J. B. 2007. Parametric embedding for class visualization. *Neural Computation* 19(9):2536–2556. 2.1.1, 2.1.1, 3.5.4, 4.2, 4.3, 4.5.2, 5.3.2, 6.3.1, 6.3.1, 7.3, 8.4

[62] Iwata, T.; Yamada, T.; and Ueda, N. 2008. Probabilistic latent semantic visualization: topic model for visualizing documents. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 363–371. 1.1.2, 3a, 2.1.1, 3.1, 3.2, 3.2.2, 3.2.2, 3.4, 3.5.1, 3.5.4, 4.1.1,

4.2, 4.3, 4.3, 4.4, 4.5.2, 4.5.3, 5.1, 5.2.1, 5.3.2, 5.3.3, 6.1, 6.2.1, 6.2.2, 6.3.1, 6.3.2, 7.1, 7.3, 8.3

[63] Jebara, T.; Wang, J.; and Chang, S.-F. 2009. Graph construction and b-matching for semi-supervised learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 441–448. ACM. 2

[64] Jin, O.; Liu, N. N.; Zhao, K.; Yu, Y.; and Yang, Q. 2011. Transferring topical knowledge from auxiliary long texts for short text clustering. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, 775–784. ACM. 6.1

[65] Jolliffe, I. 2005. *Principal Component Analysis*. Wiley Online Library. 2.1.1, 4.2

[66] Kamada, T., and Kawai, S. 1989. An algorithm for drawing general undirected graphs. *Information Processing Letters*. 1.2.1, 4.2, 4.5.2

[67] Kim, M., and Torre, F. 2010. Local minima embedding. In *Proceedings of the International Conference on Machine Learning (ICML)*, 527–534. 2.1.1

[68] Knautz, K.; Soubusta, S.; and Stock, W. G. 2010. Tag clusters as information retrieval interfaces. In *HICSS*, 1–10. IEEE. 2.2.2

[69] Kohonen, T. 1990. The self-organizing map. *Proceedings of the IEEE* 78(9):1464–1480. 2.1.1

[70] Kruskal, J. B. 1964. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika* 29(1):1–27. 1, 2.1.1, 4.2, 5.1, 7.1, 8.1

[71] Lafferty, J. D., and Wasserman, L. 2007. Statistical analysis of semi-supervised regression. In *Advances in Neural Information Processing Systems (NIPS)*, 801–808. 1.2.1, 3.1

[72] Le, T. M. V., and Lauw, H. W. 2014a. Manifold learning for jointly modeling topic and visualization. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 1a, 3.3.1, 3.5.4, 8.3

[73] Le, T. M. V., and Lauw, H. W. 2014b. Probabilistic latent document network embedding. In *2014 IEEE International Conference on Data Mining*, 270–279. IEEE. 1b

[74] Le, T. M. V., and Lauw, H. W. 2014c. Semantic visualization for spherical representation. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1007–1016. ACM. 2a, 6.1, 6.3.2, 7.1, 7.3

[75] Le, T. M. V., and Lauw, H. W. 2016a. Semantic visualization with neighborhood graph regularization. *Journal of Artificial Intelligence Research* 55:1091–1133. 1a, 6.2.1, 6.3.1, 6.3.2, 7.1, 7.3

[76] Le, T. M. V., and Lauw, H. W. 2016b. Word clouds with latent variable analysis for visual comparison of documents. In *IJCAI*, 2536–2543. IJCAI/AAAI Press. 3b

[77] Le, T. M. V., and Lauw, H. W. 2017. Semantic visualization for short texts with word embeddings. In *IJCAI*. IJCAI/AAAI Press. 2b

[78] Li, C.; Wang, H.; Zhang, Z.; Sun, A.; and Ma, Z. 2016. Topic modeling for short texts with auxiliary word embeddings. In *Proceedings of the 39th Inter-*

national ACM SIGIR conference on Research and Development in Information Retrieval, 165–174. ACM. 6.3.1

[79] Liu, D. C., and Nocedal, J. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical Programming* 45:503–528. 3.4, 4.4, 6.2.2

[80] Liu, Y.; Niculescu-Mizil, A.; and Gryc, W. 2009. Topic-Link LDA: Joint models of topic and author community. In *ICML*. 4.2

[81] Lohmann, S.; Heimerl, F.; Bopp, F.; Burch, M.; and Ertl, T. 2015. Concentri cloud: Word cloud visualization for multiple text documents. In *InfoVis*, 114–120. IEEE. 2.2.2

[82] Manning, C. D.; Raghavan, P.; Schütze, H.; et al. 2008. *Introduction to Information Retrieval*, volume 1. Cambridge University Press Cambridge. 3.3.2

[83] Mardia, K. V., and Jupp, P. E. 2009. *Directional Statistics*, volume 494. Wiley.com. 5.2.1

[84] Mardia, K. V. 1975. Distribution theory for the von Mises-Fisher distribution and its application. In *A Modern Course on Statistical Distributions in Scientific Work*. Springer. 1.2.2, 5.2.1

[85] McCallum, A. K.; Nigam, K.; Rennie, J.; and Seymore, K. 2000. Automating the construction of internet portals with machine learning. *Information Retrieval*. 4.5.1

[86] Mei, Q.; Cai, D.; Zhang, D.; and Zhai, C. 2008. Topic modeling with network regularization. In *WWW*, 101–110. ACM. 4.2

[87] Metzler, D.; Dumais, S.; and Meek, C. 2007. Similarity measures for short segments of text. In *ECIR*, 16–27. 6.1

[88] Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*. 1.2.2, 6.1, 6.2

[89] Millar, J. R.; Peterson, G. L.; and Mendenhall, M. J. 2009. Document clustering and visualization with latent dirichlet allocation and self-organizing maps. In *FLAIRS Conference*, volume 21, 69–74. 2.1.1

[90] Nallapati, R., and Cohen, W. W. 2008. Link-PLSA-LDA: A new unsupervised model for topics and influence of blogs. In *ICWSM*. 4.2

[91] Newman, D.; Lau, J. H.; Grieser, K.; and Baldwin, T. 2010. Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 100–108. 3.5.5, 5.3.4, 6.3.3

[92] Newman, D.; Karimi, S.; and Cavedon, L. 2009. External evaluation of topic models. In *Australasian Document Computing Symposium (ADCS)*. 3.5.5, 4.5.4, 4.5.4, 5.3.4

[93] Nguyen, D. Q.; Billingsley, R.; Du, L.; and Johnson, M. 2015. Improving topic models with latent feature word representations. *Transactions of the Association for Computational Linguistics* 3:299–313. 1.2.2, 6.3.1

[94] Nigam, K.; McCallum, A. K.; Thrun, S.; and Mitchell, T. 2000. Text classification from labeled and unlabeled documents using em. *Machine Learning*

39(2-3):103–134. 6.3.1

[95] Oelke, D.; Strobelt, H.; Rohrdantz, C.; Gurevych, I.; and Deussen, O. 2014. Comparative exploration of document collections: a visual analytics approach. *Computer Graphics Forum* 33(3):201–210. 2.2.2

[96] Paley, W. B. 2002. Textarc: Showing word frequency and distribution in text. In *Poster presented at IEEE Symposium on Information Visualization*, volume 2002. 2.2.1

[97] Park, J., and Sandberg, I. W. 1991. Universal approximation using radial-basis-function networks. *Neural Computation* 3(2):246–257. 3.2.2

[98] Paulovich, F. V.; Toledo, F.; Telles, G. P.; Minghim, R.; and Nonato, L. G. 2012. Semantic wordification of document collections. *Computer Graphics Forum* 31(3pt3). 2.2.2

[99] Pennington, J.; Socher, R.; and Manning, C. D. 2014. GloVe: Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP)* 12. 1.2.2, 6.1, 6.2

[100] Phan, X.-H.; Nguyen, L.-M.; and Horiguchi, S. 2008. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th international conference on World Wide Web*, 91–100. ACM. 6.1, 6.3.1

[101] Reisinger, J.; Waters, A.; Silverthorn, B.; and Mooney, R. J. 2010. Spherical topic models. In *Proceedings of the International Conference on Machine Learning (ICML)*, 903–910. 2, 1.2.2, 5.1, 5.2.1, 5.2.1, 5.2.2, 5.3.1, 5.3.2, 5.3.4

[102] Rivadeneira, A. W.; Gruen, D. M.; Muller, M. J.; and Millen, D. R. 2007. Getting our head in the clouds: toward evaluation studies of tagclouds. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 995–998. ACM. 8.5

[103] Rodrigues, N. 2013. Analyzing textual data by multiple word clouds. Master's thesis, Universität Stuttgart, Holzgartenstr. 16, 70174 Stuttgart. 2.2.2

[104] Roweis, S. T., and Saul, L. K. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science* 290(5500):2323–2326. 2.1.1, 4.2, 5.1

[105] Salton, G.; Wong, A.; and Yang, C.-S. 1975. A vector space model for automatic indexing. *CACM* 18(11). 5.1, 5.2.1

[106] Seifert, C.; Kump, B.; Kienreich, W.; Granitzer, G.; and Granitzer, M. 2008. On the beauty and usability of tag clouds. In *InfoVis*. IEEE. 1.2.3, 8.1, 8.2, 8.4

[107] Shaw, B., and Jebara, T. 2007. Minimum volume embedding. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 460–467. 2.1.1

[108] Shaw, B., and Jebara, T. 2009. Structure preserving embedding. In *Proceedings of the International Conference on Machine Learning (ICML)*, 937–944. ACM. 1.2.1, 2.1.1, 4.2, 4.5.3

[109] Sriram, B.; Fuhry, D.; Demir, E.; Ferhatosmanoglu, H.; and Demirbas, M. 2010. Short text classification in twitter to improve information filtering. In *SIGIR*, 841–842. 6.1

[110] Steele, J., and Iliinsky, N. 2010. *Beautiful visualization: looking at data*

through the eyes of experts. " O'Reilly Media, Inc.". 2.2.1, 8.1

[111] Sun, Y.; Han, J.; Gao, J.; and Yu, Y. 2009. iTopicModel: Information network-integrated topic modeling. In *ICDM*. 4.2

[112] Sun, A. 2012. Short text classification using very few words. In *SIGIR*, 1145–1146. ACM. 6.1

[113] Talwalkar, A.; Kumar, S.; Mohri, M.; and Rowley, H. 2013. Large-scale SVD and manifold learning. *JMLR*. 4.2

[114] Tang, J.; Liu, J.; Zhang, M.; and Mei, Q. 2016. Visualizing large-scale and high-dimensional data. In *Proceedings of the 25th International Conference on World Wide Web*, 287–297. International World Wide Web Conferences Steering Committee. 3.5.1

[115] Tenenbaum, J. B.; De Silva, V.; and Langford, J. C. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science* 290(5500):2319–2323. 2.1.1, 4.2

[116] Ting, D.; Huang, L.; and Jordan, M. I. 2010. An analysis of the convergence of graph laplacians. In *Proceedings of the International Conference on Machine Learning (ICML)*. 1

[117] Van der Maaten, L., and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research (JMLR)* 9(2579-2605):85. 2.1.1, 3.2.2, 4.2, 5.1, 5.2.1, 7.1

[118] Van Der Maaten, L. 2014. Accelerating t-sne using tree-based algorithms. *Journal of machine learning research* 15(1):3221–3245. 3.5.1

[119] Van Ham, F.; Wattenberg, M.; and Viégas, F. B. 2009. Mapping text with phrase nets. *IEEE Trans. Vis. Comput. Graph.* 15(6):1169–1176. (document), 2.2.1, 2.9

[120] Viégas, F. B., and Wattenberg, M. 2008. Timelines tag clouds and the case for vernacular visualization. *Interactions* 15(4):49–52. 2.2.1

[121] Viegas, F. B.; Wattenberg, M.; and Feinberg, J. 2009. Participatory visualization with wordle. *TVCG* 15(6):1137–1144. 1.2.3, 2.2.1, 8.1

[122] Wattenberg, M., and Viégas, F. B. 2008. The word tree, an interactive visual concordance. *IEEE transactions on visualization and computer graphics* 14(6):1221–1228. (document), 2.2.1, 2.8

[123] Wei, F.; Liu, S.; Song, Y.; Pan, S.; Zhou, M. X.; Qian, W.; Shi, L.; Tan, L.; and Zhang, Q. 2010. Tiara: a visual exploratory text analytic system. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 153–162. (document), 2.1.1, 2.1.2, 2.6, 4.2

[124] Wu, Y.; Provan, T.; Wei, F.; Liu, S.; and Ma, K.-L. 2011. Semantic-preserving word clouds by seam carving. *Computer Graphics Forum* 30(3):741–750. 2.2.2

[125] Wu, H.; Bu, J.; Chen, C.; Zhu, J.; Zhang, L.; Liu, H.; Wang, C.; and Cai, D. 2012. Locally discriminative topic modeling. *Pattern Recognition* 45(1):617–625. 3.1

[126] Yan, X.; Guo, J.; Liu, S.; Cheng, X.-q.; and Wang, Y. 2012. Clustering short text using ncut-weighted non-negative matrix factorization. In *Proceedings of the*

21st ACM international conference on Information and knowledge management, 2259–2262. ACM. 6.1

[127] Zemel, R. S., and Carreira-Perpiñán, M. Á. 2004. Proximity graphs for clustering and manifold learning. In *Advances in Neural Information Processing Systems (NIPS)*, 225–232. 3.3.2, 2

[128] Zhou, D.; Bousquet, O.; Lal, T. N.; Weston, J.; and Schölkopf, B. 2004. Learning with local and global consistency. *Advances in Neural Information Processing Systems (NIPS)* 16(16). 1.2.1, 3.1, 3.1.1

[129] Zhu, X.; Ghahramani, Z.; Lafferty, J.; et al. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 3, 912–919. 1.2.1, 3.1, 3.1.1

[130] Zhu, S.; Yu, K.; Chi, Y.; and Gong, Y. 2007. Combining content and link for classification using matrix factorization. In *SIGIR*. 4.5.1

[131] Zuo, Y.; Zhao, J.; and Xu, K. 2015. Word network topic model: A simple but general solution for short and imbalanced texts. *Knowledge and Information Systems* 1–20. 8.3