A SCHEMA CONVERSION APPROACH FOR CONSTRUCTING HETEROGENEOUS INFORMATION NETWORKS FROM DOCUMENTS

BY

HYUNG SUL KIM

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2017

Urbana, Illinois

Doctoral Committee:

Professor Jiawei Han, Chair
Associate Professor Julia Hockenmaier
Professor ChengXiang Zhai
Doctor Pavel Dmitriev, Microsoft

ABSTRACT

Information networks with multi-typed nodes and edges with different semantics are called heterogenous information networks. Since heterogeneous information networks embed more complex information than homogeneous information networks due to their multi-typed nodes and edges, mining such networks has produced richer knowledge and insights.

To extend the application of heterogeneous information network analysis to document analysis, it is necessary to build information networks from a collection of documents while preserving important information in the documents.

This thesis describes a schema conversion approach to apply data mining techniques on the outcomes of natural language processing (NLP) tools to construct heterogeneous information networks.

First, we utilize named entity recognition (NER) tools to explore networks over entities, topics, and words to demonstrate how a probabilistic model can convert the data schema of the NER tools. Second, we address a pattern mining method to construct a network with authors, documents, and writing styles by extracting discriminative writing styles from parse trees and converting them into nodes in a network. Third, we introduce a clustering method to merge redundant nodes in an information network with documents, claims, subjective, objective, and verbs. We use a semantic role labeling (SRL) tool to get initial network structures from news articles, and merge duplicated nodes using a similarity measure SynRank. Finally, we present a novel event mining framework for extracting high-quality structured event knowledge from large, redundant, and noisy news data. The proposed framework ProxiModel utilizes named entity recognition, time expression extraction, and phrase mining tools to get event information from documents.

To my family, for their love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Information networks are one of very expressive forms to represent data. Most of the data we have can be converted or naturally represented in information networks. For example, web pages with hyperlinks, Facebook friendships, research co-authorship, gene regulation networks, and interactions of proteins are represented in information networks.

Among such networks, those with multi-typed nodes and edges with different semantics are called *heterogenous information networks*. Since heterogeneous information networks embed more complex information than homogeneous information networks due to their multi-typed nodes and edges, mining such networks has produced richer knowledge and insights.

Several heterogeneous information network anlayses have outperformed previous homogenous information network analysis in different tasks such as clustering [1, 2], classification [3], trustworthiness analysis [4], relationship prediction [5, 6], and similarity search [7, 8].

However, those studies are limited to structured data like movie, book, and DBLP databases because it is easier to use existing networks than to construct networks from unstructured data like documents.

Thank to the advance of natural language processing (NLP), documents can be enriched with structural information like parsing trees, semantic role labels, entities, and semantic meanings in vector space. Such NLP methods have opened up a new application domain of heterogenous information network analysis.

My past and current research focus on utilizing NLP methods to construct information networks from documents.

As the first step, I use news articles as the main data source because 1) they are well-written in terms of syntax and wording where NLP tools perform well, 2) they convey information related to real stories or events that are interests of most people, 3) there are many redundant information in a

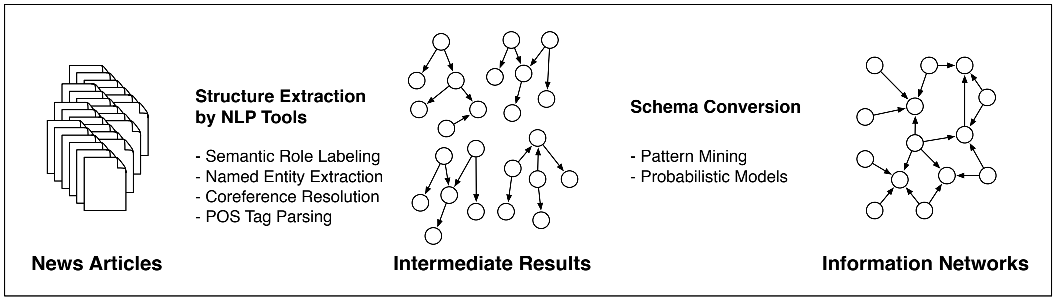


Figure 1.1: Schema Conversion

collection of news articles which lead to build stable and dense networks, and 4) they are easily accessible with other useful attributes like author names, timestamps, topics, main entities, and publishers.

Moreover, in the age of information overload, we can easily collect or access copora that cover the same topic such as multiple news reports on the same or similar events from different news agencies, and reviews about the same or similar products or services. Such a collection of documents is called a monolingual comparable corpus. A monolingual comparable corpus is defined as a collection of documents in the same language (*e.g.*, English) that overlap in the information they convey. Such a corpus is an important source to construct an information networks because documents in the corpus complement to each other, and redundancies of information in the corpus can measure the popularity and confidence of the information.

Network construction methods differ, depending on a target information network schema. When the output schema of a NLP method matches with a target information network schema, it is straightforward to construct information networks by the NLP method. When an information network schema is different from the output schema of a NLP method, we need to convert or extract nodes and relationships of the network schema from the output. This process can be done by pattern mining or probabilistic models. Figure 1.1 summarizes the schema conversion process.

In this dissertation, we investigate the schema conversion problems in information network construction with different NLP tools: tree parsing, entity annotation, semantic role labeling, and phrase mining. The main challenges are 1) how to encode the information from documents in multi-typed nodes and edges, and 2) how to remove redundancies and erroneous data.

In the first part of the dissertation, I propose a probabilistic graphical model to construct a network with documents, words, topics, and entities in order to demonstrate how a probabilistic model can convert the data schema. We annotate news articles with associated entities using an entity extraction tool, and build a network using a topic model. Experiments on real datasets demonstrate the effectiveness of our approach over several state-of-the-art baselines.

In the second part of the dissertation, I address a pattern mining method to construct a network with authors, documents, and writing styles. Writing styles are abstract nodes which are not explicitly shown in sentences. Using a pattern mining algorithm, we extract discriminative writing styles from sentences and convert them into nodes in a network. We show that this approach reduces the computational burden of using complex syntactic structures. Comprehensive experiments on real-world datasets demonstrate that our approach is reliable than previous studies.

In the third part of the dissertation, I introduce a clustering method to merge redundant nodes in an information network with documents, claims, subjective, objective, and verbs. We use a semantic role labeling (SRL) tool to get initial network structures from news articles, and merge duplicated nodes using a similarity measure SynRank.

In the fourth part of the dissertation, I present a novel event mining framework for extracting high-quality structured event knowledge from large, redundant, and noisy news data. The proposed framework jointly derives connections between different events by modeling the event correlation within each individual document as well as across the corpus. A proximity network-based approach to event mining, ProxiModel, constructs proximity networks as a data model to capture the corpus-level co-occurrence statistics for candidate event descriptors, attributes, as well as their connections.

The conclusion for the dissertation is discussed in the last section.

CHAPTER 2

RELATED WORK

My thesis explore three different approaches to construct information networks: pattern-based feature generation, topic models, and similarity measures in heterogenous information networks. A brief overview of these related methods is discussed in this section.

2.1 Information Network Construction

Information network construction has been studied in various domains.

First, several endeavors have been to build knowledge bases from web sources. Unlike unstructured data like documents, web pages are semi-structured data, where formatted layouts and hyperlinks serve as the cues of targeted information. DBpedia [9] and Freebase [10] are community efforts to extract structured information from Wikipedia. Mostly, they utilize the infobox in each Wikipedia webpage to get attributes and relationships of the corresponding entity.

EntityCube [11], KnowItAll [12], and NELL [13] treat the web pages as unstructured data and use NLP methods to extract entity relationships and general knowledge represented by networks. Since their methods are web-scale, they focus on the efficiency of their methods using simple NLP methods and probabilistic models.

WINACS (Web-based Information Network Analysis for Computer Science) [14], is a project that constructs a web-based compuster science information network using hyperlinks in the webpages of the computer science departments and professors. They use list finding, entity discovery, and record linkage with databases to harvest necessary information from webpages to build an information network.

Event extraction can be viewed as a process of constructing information networks where events and their attributes are nodes and linked by edges. For example, there are event extraction methods for a collection of documents [15] and tweets [16]. Similarly, in biology, there are biomedical event extraction [17] and protein interaction network construction [18].

2.2 Topic Models for Network Construction

Starting with the great success of Probabilistic Latent Semantic Analysis (PLSA) [19] and Latent Dirichlet Allocation (LDA) [20], there have been numerous proposals for topic models that identify patterns of word occurrences in large collections of documents which reflect the underlying topics represented in the collection, and can then be used to organize, search, index and browse large collection of documents [21].

While traditional topic models treat each document as a bag of words, documents are in fact associated with richer attributes: for example, news articles are associated with people, organizations or locations, many tweets are associated with geo-locations and timestamps, research articles are associated with authors, and webpages are associated with link information. This has opened up interesting opportunities and challenges for document analysis.

To deal with the different types of attributes associated with documents, different topic models have been proposed: (1) Topic Over Time [22] and Dynamic Topic Models [23] are designed for documents with timestamps, (2) GeoFolk [24] and Latent Geographical Topic Analysis [25] are proposed for documents with GPS information, (3) Author Models [26] and Autor Topic Models [27] deal with documents with author lists, and (4) Link-LDA [28] and Block-LDA [29] are designed for dealing with documents with hyperlinks, citations, and other forms of link information.

As shown in NetClus [1], iTopic [30], and TMBP [30], documents can be represented in information networks without any complicated conversion. Figure 2.1 shows the schema of input documents and the schema of the output of three topic models as examples: LDA, AM, and Link-LDA.

Latent Dirichlet Allocation (LDA) [20] is one of the most well-known topic models. It assumes that a document is generated via a mixture of topics. In its generative process, for each document d, a multinomial distribution θ_d

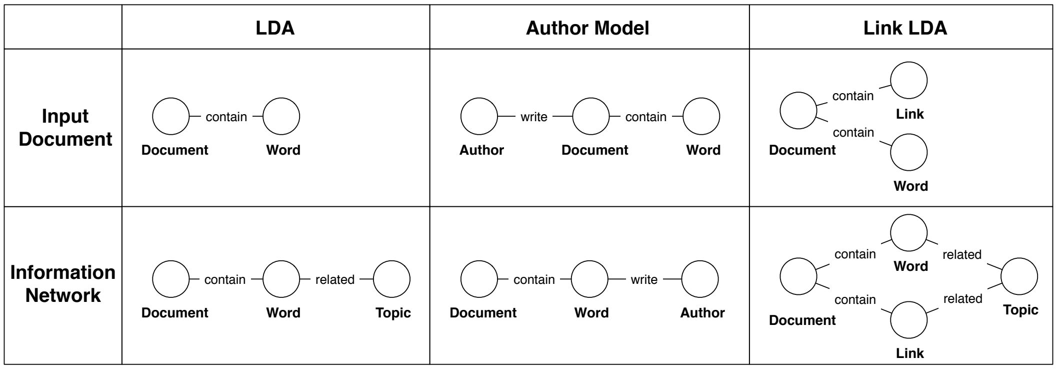


Figure 2.1: Topic Models and Related Information Networks

over topics is drawn from a Dirichlet prior with α. Then for each word, a topic $z_{d,i}$ is drawn from θ_d, and a word $w_{d,i}$ is generated by randomly sampling from a topic-specific multinomial distribution $\phi_{z_{d,i}}$.

The Author Model (AM) [26] is originally proposed for multi-labeled documents, where each label could represent a class or an entity. In other words, for each document d, the set of associated labels, $\boldsymbol{E}_d$, is given. For each word, a label $e_{d,i}$ is uniformly chosen from $\boldsymbol{E}_d$, and $w_{d,i}$ is generated by randomly sampling from a label-specific multinomial distribution $\varphi_{e_{d,i}}$. However, the AM only captures term distributions for each entity without investigating further the hidden patterns (topics) in documents.

Link-LDA [28] is proposed for scientific publication with citations. In this model, documents consist of a bag of words and a bag of citations. For each document d, a multinomial distribution θ_d over topics is drawn from a Dirichlet prior with α. Then, for each word, a topic $z_{d,i}$ is drawn from θ_d, and a word $w_{d,i}$ is generated from randomly choosing from a topic specific multinomial distribution $\phi_{z_{d,i}}$ over words. For each citation, a topic $z_{d,j}$ is drawn from θ_d, and a citation $e_{d,i}$ is generated from randomly sampling from a topic specific multinomial distribution $\varphi_{z_{d,j}}$ over citations.

2.3 Similarity Measures in Information Networks for Identifying Duplicates

Duplicates or redundancies in information networks may degrade the performance of information network analysis. Such redundant nodes in information networks can be detected using similarity measures.

There have been proposed several structural similarity measures including Random walk with restart [31], Simrank [32], P-Rank [8], and DISTINCT [33].

Random walk with restart [31] is an extension of the famous ranking algorithm, PageRank, to measure similarity between nodes by assigning a small probability to restart a current random walk at a pivot node. Simrank [32] is recursively defined on a information network to capture the intuition that two nodes are similar if they are referenced by similar entities. P-Rank [8] further extends Simrank to differentiate in-link and out-link relationships in information networks. DISTINCT [33] combines content similarity and structural similarity of two nodes in information networks.

2.4 Event Extraction from Documents

Many attempts have been made to extract events from text corpora. These approaches can be categorized into NLP-based contextual analysis approaches and data mining approaches.

In the NLP literature, many approaches employ rich features to model event extraction as a parsing problem. McClosky *et al.* perform event extraction by creating a tree of event-argument relations and using this as a representation for reranking of the dependency parser [34]. NLP event extraction techniques have even been applied to extracting biomedical events from text literature such as binding, regulation, and gene-protein interactions; these techniques rely on a rich feature-set for classification [35]. Other methods employ tagging and matching specified event patterns to perform large-scale event extraction; redundancy is reduced by automatically generating rulesets for event merging [36]. While these NLP-based methods often obtain high-quality results, their dependency on parsing, user-defined patterns, and annotated data reduces effectiveness across multiple sources. While these methods may show acceptable performance in a closed-domain such as when the types of events are known before-hand, they suffer in an open-domain scenario.

In the data mining literature, a variety of methods have been introduced for extracting underlying events from news corpora. Using a probabilistic model that incorporates both content, time, and location information, Li *et al.* develop a unified generative model where, for each article, a single latent event generates observable event descriptors such as location, people, keywords and timestamps [37]. This HISCOVERY framework first applies NLP entity recognition tools to extract persons, locations, and dates/times, then uses this data in its generative model. However, it makes the strong assumption that each news article references a single event, a requirement we relax in our probabilistic model.

Other approaches in the data mining literature apply clustering and document relevancy measures to organize documents into coherent events. These methods often employ heuristic clustering approaches based on intra-cluster similarity to agglomeratively form event clusters. Naughton *et al.* annotate sentences with event labels then aggregate these sentences into a structured form and create coherent event summaries [38]. They also apply machine learning to extract event-containing sentences and propose two metrics for event sentence clustering to identify, integrate, and summarize news events from multiple sources [39]. Further clustering approaches agglomeratively merge and prune event clusters to identify discriminative events [40]. Lam *et al.* cluster documents into events and detect new events by first extracting discriminative "concept terms", named entities, and other identifying information and using these features, cluster documents into existing and new events [41]. These clustering approaches are document-level event analysis, defining an event as a collection of topically related article. These works are not suitable for fine-grained event analysis.

CHAPTER 3

TOPIC MODELS FOR NETWORK CONSTRUCTION

3.1 Overview

In this chapter, we are interested in topic analysis for collections of documents associated with sets of entities. The ability to capture the association of documents with real-world entities or concepts holds great promise over traditional keyword-based approaches (cf. Google's "knowledge graph", which enhances search results by linking documents to entities[1]). In a similar vein, we argue that it is also highly desirable to build topic models that can capture the complex patterns involving the entities associated with documents. Almost any document is associated with some set of real-worlds entities. For instance, news articles may mention people, organizations or locations, research papers are associated with authors, medical records are associated with patients, doctors, diseases and so on. Many documents are explicitly associated with entities such as authors or publications via metadata. But since we are now quite successful at wide-coverage named-entity extraction from raw text [42, 43], we can also capture the implicit associations of documents to the entities mentioned in them.

In addition to the term distributions for each topic, we may therefore also wish to know the term distributions for each entity, or topic-entity pair. Namely, letting z, e, and w denote a topic, an entity, and a word, respectively, we want to design a topic model that can answer the following queries: $P(w|z)$, $P(w|e)$, and $P(w|e, z)$.

For example, in a collection of computer science research articles, we may want to find a topic called data mining, and understand it by browsing its word distribution $P(w|\textsf{Data Mining})$. If we want to identify the topics that a specific researcher, e.g. Judea Pearl, the 2011 winner of the A.M. Turing Award,

[1] http://www.google.com/insidesearch/features/search/knowledge.html

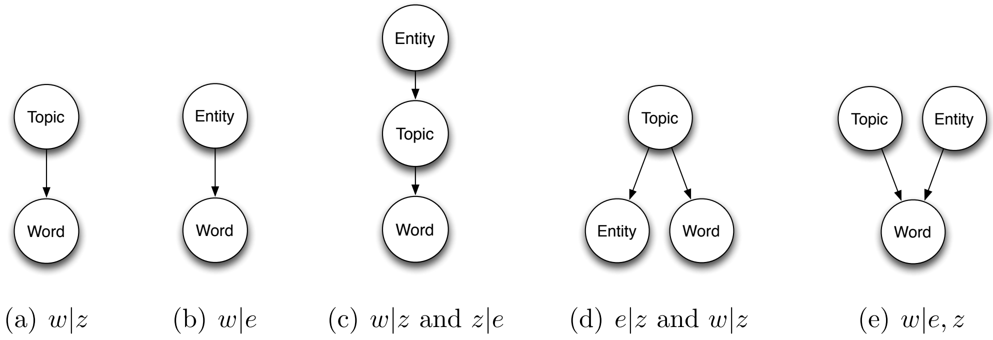
has worked on, we may want to browse the word distribution $P(w|\textsf{Judea Pearl})$. We can also have better understanding of his contribution to specific areas such as data mining or artificial intelligence through $P(w|\textsf{Judea Pearl}, \textsf{Data Mining})$ or $P(w|\textsf{Judea Pearl}, \textsf{A.I.})$, or the difference of focus of his data mining related works from data mining in general by comparing $P(w|\textsf{Judea Pearl}, \textsf{Data Mining})$ with $P(w|\textsf{Data Mining})$. We may also wish to compare his artificial intelligence related works with another leading researcher in that field, e.g. Michael Jordan, by comparing $P(w|\textsf{Judea Pearl}, \textsf{A.I.})$ with $P(w|\textsf{Michael Jordan}, \textsf{A.I.})$.

As another example, in a collection of news articles about Japan's Tsunami in 2011, we can find frequently mentioned words related to relief efforts by $P(w|\textsf{Relief Efforts})$, related to the United States by $P(w|\textsf{United States})$, and the term distribution related to the relief efforts of the United States by $P(w|\textsf{United States}, \textsf{Relief Efforts})$. Also, we can learn about Naoto Kan who was Japan's Prime Minister at the time by $P(w|\textsf{Naoto Kan})$, his actions on the tsunami disaster by $P(w|\textsf{Naoto Kan}, \textsf{Tsunami})$, and his actions on the economic damages by $P(w|\textsf{Naoto Kan}, \textsf{Economic Damages})$.

To the best of our knowledge, there are no previous studies that have modeled $P(w|e, z)$ directly: they assume either $P(w|e, z) = P(w|z)$ or $P(w|e, z) = P(w|e)$ by introducing different types of conditional dependency relations among topics, entities, and words. In Figure 3.1, we summarize the dependency structures among these variables in several well-known topic models. In LDA (Figure 3.1(a)), words are drawn for a given topic, and entities are not modeled. In the Author Model, words are drawn for a given author, and topics are not modeled (Figure 3.1(b)). In the Author Topic Model, topics are drawn for a given author, and words are drawn for a given topic (Figure 3.1(c)). In Link-LDA, entities are drawn for a given topic, and words are drawn for a given topic (Figure 3.1(d)). In Figure 3.1(a), Figure 3.1(c), and Figure 3.1(d), $P(w|e, z) = P(w|z)$ is assumed whereas in Figure 3.1(b), $P(w|e, z) = P(w|e)$ is assumed.

However, in many documents collections, these independence assumptions are not valid. For example, Judea Pearl published many papers in several different domains, including artificial intelligence and data mining. On the one hand, with the assumption of $P(w|e, z) = P(w|e)$, $P(w|\textsf{Judea Pearl}, \textsf{A.I.}) = P(w|\textsf{Judea Pearl}, \textsf{Data Mining})$[2], but obviously papers from different topics may not

[2] This problem cannot be solved by simple counting, as we are not sure who has contributed to a particular term when a paper is written by multiple authors.



(a) $w|z$ (b) $w|e$ (c) $w|z$ and $z|e$ (d) $e|z$ and $w|z$ (e) $w|e,z$

Figure 3.1: Different dependencies among topic, entity, and word

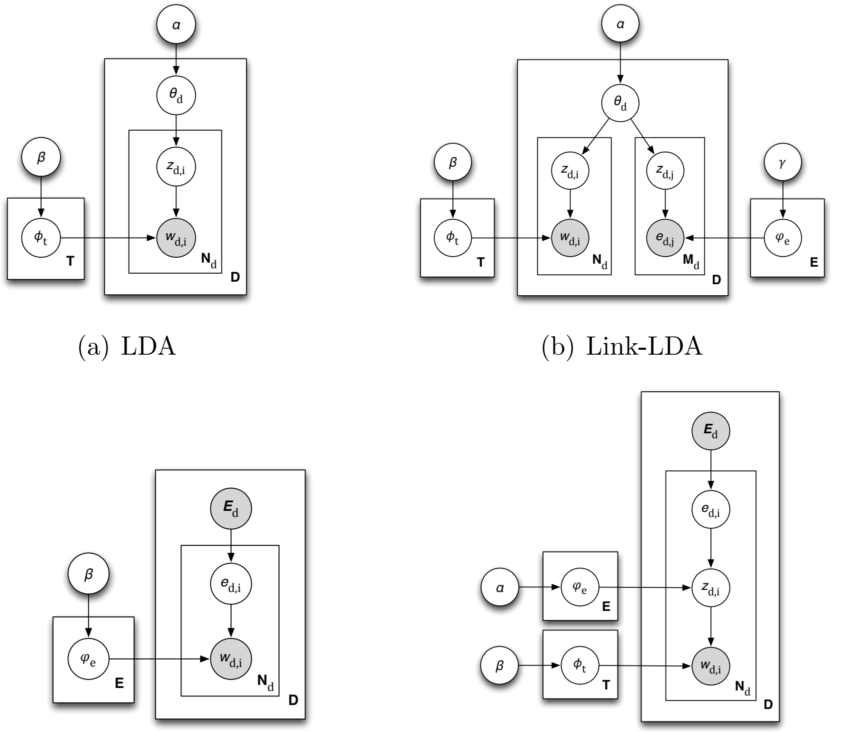
use the same terms. On the other hand, with the assumption of $P(w|e,z) = P(w|z)$, $P(w|\text{Judea Pearl, A.I.}) = P(w|\text{Michael Jordan, A.I.})$, but different authors usually use different terms even in the same research area. Therefore there is a necessity for us to model the correlation of words between a pair of an entity and a topic by directly modeling $P(w|e,z)$ as shown in Figure 3.1(e). In order to solve this problem, we propose a novel topic model named Entity Topic Model (ETM) for analyzing a given collection of documents with given entities. ETM not only models the generative process of a term given its topic and entity information, but also models the correlation of entity-term and topic-term distributions. We show that LDA and the Author Model are special cases of our model with different parameter settings. A Gibbs sampling-based algorithm is proposed to learn the model. Experiments on real datesets demonstrate the effectiveness of our approach over several state-of-the-art baselines.

The major contributions of this chapter are summarized in the following.

1. We identify a general type of task for topic modeling, i.e. designing topic models for documents with entity information.

2. We propose a novel Entity Topic Model (ETM) which solve this task by explicitly modeling the term correlation between entities and topics. We also define a Gibbs sampling-based algorithm to learn the model.

3. We demonstrate the power of our new model over several state-of-the-art baselines by using two real-world datasets.



(a) LDA

(b) Link-LDA

(c) Author Model

(d) Author Topic Model

Figure 3.2: Four related models with different dependencies among topics(z), entities(e), and words(w)

3.2 Problem Statement

The input to the ETM model is a collection of documents in which each document has a set of associated entities. A document d is associated with a term vector, $\boldsymbol{w}_d$, where each $w_{d,i}$ is chosen from the vocabulary of W, and an entity vector $\boldsymbol{E}_d$, chosen from a set of entities of size E. A collection of D documents is defined by $\mathcal{D} = \{\langle \boldsymbol{w}_1, \boldsymbol{E}_1 \rangle, \ldots, \langle \boldsymbol{w}_D, \boldsymbol{E}_D \rangle\}$. (The notation used in this chapter is summarized in Table 3.1). The goal is to discover word patterns for each pair of an entity and a topic. In other words, we want to discover the hidden topics in the documents, as well as the word distributions for a given entity e and a topic z, $P(w|e, z)$, which follows a multinomial distribution with parameter $\psi_{e,z}$.

The biggest challenge is that there are too many parameters to be estimated when modeling $P(w|e, z)$ directly. With E entities, T topics, and W words in a given collection, we need to estimate $O(ETW)$ parameters, which will most likely cause overfitting. In order to solve this problem, we propose

Table 3.1: Notation used in this chapter

Symbol	Description
D	number of documents
T	number of topics
W	number of words
E	number of entities
N_d	number of word tokens in document d
θ_d	multinomial distribution of topics specific to document d
ϑ_d	multinomial distribution of entities specific to document d
$\boldsymbol{w}_d$	bag of words associated with document d
$\boldsymbol{E}_d$	list of entities associated with document d
$z_{d,i}$	topic associated with the ith token in document d
$e_{d,i}$	entity associated with the ith token in document d
$w_{d,i}$	ith token in document d
ϕ_z	asymmetric Dirichlet prior for topic t
φ_e	asymmetric Dirichlet prior for entity e
$\psi_{e,z}$	multinomial distribution of words specific to entity e and topic t
$\mathcal{D}$	set of all documents
$\mathcal{Z}$	set of all topic assignments $\{e_{d,i}\}$
$\mathcal{E}$	set of all entity assignments $\{e_{d,i}\}$
Φ	set of all parameters in the model

a novel parameter smoothing method by designing hierarchical Dirichlet priors for the multinomial distribution of $P(w|e,z)$, where intuitively $P(w|e,z)$ is determined by the term distribution for the entity $P(w|e)$ and the term distribution for the topic $P(w|z)$. In particular, we use a weighted linear combination of ϕ_z and φ_e as the Dirichlet prior for $\psi_{e,z}$, where ϕ_z is an asymmetric Dirichlet parameter vector for each topic z, and φ_e is an asymmetric Dirichlet parameter vector for each entity e.

In the ETM model, we design a process for generating all the terms in a document that is associated with a given set of entities. Note that the entities that a document is associated with are not generated, but are assumed to be given. This assumption is also used in the author model and author topic model. However, in contrast to these models, we no longer assume entities are generated uniformly, but follow a multinomial distribution ϑ_d.

3.3 Entity Topic Model

In this section, we formally define our problem, introduce our topic model, and finally provide a Gibbs sampling-based learning algorithm. The graphical representation for ETM is shown in Figure 3.3, and the detailed explanations are given in the following.

3.3.1 Generative Process

The hypothesis at the heart of our model is that different entities are described with different word patterns or word distributions, and that the words used to describe an entity can change with the topic. In other words, $P(w|e_i, z) \neq P(w|e_j, z)$ if $e_i \neq e_j$ and $P(w|e, z_i) \neq P(w|e, z_j)$ if $z_i \neq z_j$.

As shown in Algorithm 1, for each document d, a multinomial distribution θ_d over topics is drawn from a Dirichlet prior with α_0, and another multinomial distribution ϑ_d over the associated entity set $\boldsymbol{E}_d$ is drawn from a Dirichlet prior with α_1. Note that instead of selecting an entity uniformly from $\boldsymbol{E}_d$ as in the author model and author topic model [26, 27], we draw it from a document-specific multinomial distribution ϑ_d over $\boldsymbol{E}_d$. This is due to the assumption that each entity in $\boldsymbol{E}_d$ has a different weight in generating a document d. For example, when writing a research article, different authors make different contributions. Then, to generate each word, a topic $z_{d,i}$ is drawn from θ_d, an entity $e_{d,i}$ is drawn from ϑ_d, and word $w_{d,i}$ is generated by randomly sampling from an entity and topic specific multinomial distribution $\psi_{e_{d,i}, z_{d,i}}$. That is, each term is associated with a entity-topic pair, and the generation of term is dependent on both factors.

3.3.2 Shared Asymmetric Dirichlet Priors

In this section, we will explain how to model the word distributions $P(w|e, z)$ for each entity-topic pair (e, z). As addressed in Section 3.3.1, our model uses two contexts, entity e and topic z, to generate word w. One of the important issues for statistical language models and topic models is data sparsity, which is the phenomenon of not observing enough data in a corpus to learn accurate model parameters. Effective smoothing techniques [44] are required

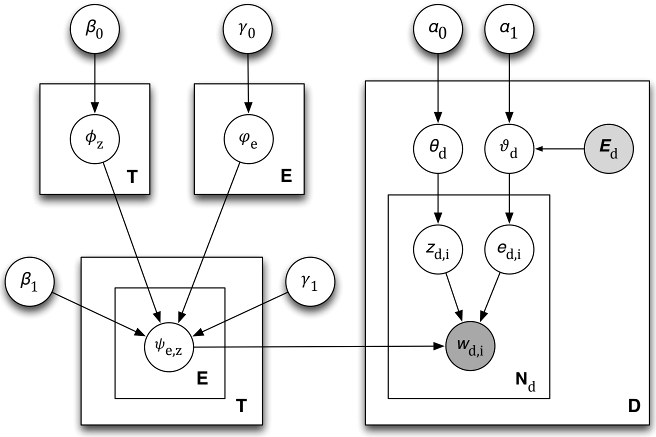


Figure 3.3: A graphical representation of ETM

to alleviate this problem. A well-known smoothing technique is to use symmetric Dirichlet priors with fixed, uniform concentration parameters. This allows any topic to generate any word with non-zero probability. However, recent studies [45] have shown that the quality of topic models can be significantly enhanced by considering asymmetric Dirichlet priors, an idea we adopt in ETM. We use the intuition that the word distribution for (e, z) pair should be dependent on word distributions for both entity e and topic z, and share some similarity with both of them. For example, the word distribution for *Judea Pearl* in *Data Mining* should be similar to the word distribution for *Judea Pearl* and the word distribution for *Data Mining* separately. Therefore, the prior for $\psi_{e,z}$ could be designed as some function of word distributions for e and z. More specifically, suppose that we have some common word patterns φ_e for an entity e across topics, and ϕ_z for a topic z across entities. We use a linear combination of φ_e and ϕ_z as Dirichlet prior of $\psi_{e,z}$:

$$\psi_{e,z} \sim Dir(\beta_1\phi_z + \gamma_1\varphi_e)$$

Since such common word patterns are not necessary symmetric, their linear combination is asymmetric. By sharing common word patterns as priors, we can get better word smoothing, and with a much smaller parameter space, i.e., EW for φ_e and TW for ϕ_z.

Algorithm 1 Entity Topic Models

```
for each topic $z$ do
    Draw $\phi_z \sim Dir(\beta_0)$
end for
for each entity $e$ do
    Draw $\varphi_e \sim Dir(\gamma_0)$
end for
for each $(e, z)$ do
    Draw $\psi_{e,z} \sim Dir(\beta_1 \phi_z + \gamma_1 \varphi_e)$
end for
for each document $d$ do
    Draw $\theta_d \sim Dir(\alpha_0)$
    Draw $\vartheta_d \sim Dir(\alpha_1; \boldsymbol{E}_d)$
    for each $i \in 1, \ldots, N_d$ do
        Draw $z_{d,i} \sim Multi(\theta_d)$
        Draw $e_{d,i} \sim Multi(\vartheta_d)$
        Draw $w_{d,i} \sim Multi(\psi_{e_{d,i}, z_{d,i}})$
    end for
end for
```

3.3.3 Model Learning

We use Gibbs sampling to learn the model. Specifically, we repeatedly sample the entity-topic pair for each word in the document collection, given the entity-pair of assignment to all the rest words $(\mathcal{Z}, \mathcal{E})$ as well as the priors (Φ). This conditional posterior of assignment $(e_{d,i}, z_{d,i})$ to the ith word $w_{d,i}$ in document d is:

$$\begin{aligned} & P(z_{d,i}, e_{d,i} | w_{d,i}, \mathcal{Z}_{\setminus d,i}, \mathcal{E}_{\setminus d,i}, \Phi) \\ \propto \quad & P(w_{d,i} | z_{d,i}, e_{d,i}, \mathcal{Z}_{\setminus d,i}, \mathcal{E}_{\setminus d,i}, \Phi) \\ & P(z_{d,i} | \mathcal{Z}_{\setminus d,i}, \Phi) \\ & P(e_{d,i} | \mathcal{E}_{\setminus d,i}, \Phi) \end{aligned} \tag{3.1}$$

where sub- or super-script "$\setminus d, i$" denotes a quantity excluding data from position i in document d.

The second and third terms on the right-hand side are straightforward:

$$P(z_{d,i} | \mathcal{Z}_{\setminus d,i}, \Phi) \propto \frac{N_{z_{d,i}|d}^{\setminus d,i} + \frac{\alpha_0}{T}}{N_d - 1 + \alpha_0} \tag{3.2}$$

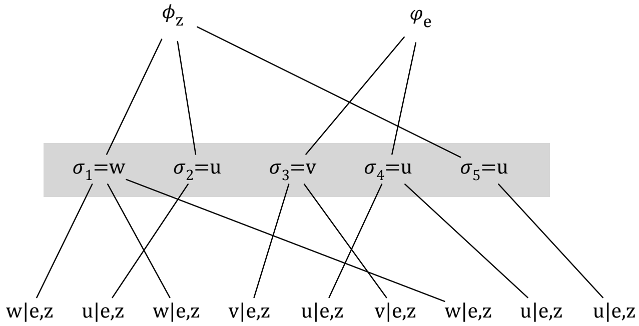


Figure 3.4: The generative process of nine words from $\psi_{e,z}$ that has $\beta_1\phi_z + \gamma_1\varphi_e$ as its prior

$$P(e_{d,i}|\mathcal{E}, \Phi) \propto \frac{N^{\setminus d,i}_{e_{d,i}|d} + \frac{\alpha_1}{|\boldsymbol{E}_d|}}{N_d - 1 + \alpha_1} \tag{3.3}$$

where $N^{\setminus d,i}_{z_{d,i}|d}$ is the number of word tokens assigned with topic $z_{d,i}$ except ith token in document d, and $N^{\setminus d,i}_{e_{d,i}|d}$ is the number of word tokens assigned with entity $e_{d,i}$ except ith token in document d.

In order to better understand the first term on the right-hand side, we describe its generative process[3]. Figure 3.4 depicts the process of drawing nine words from the Dirichlet-multinomial $\psi_{e,z}$ that has $\beta_1\phi_z + \gamma_1\varphi_e$ as its prior. This process introduces a set of internal draws $\{\sigma_1, \sigma_2, \dots\}$. Those internal draws are chosen when a word is generated from $\psi_{e,z}$. When drawing the first word, there are no previous internal draws, and σ_1 is drawn from either ϕ_z with probability $\frac{\beta_1}{\beta_1+\gamma_1}$ or φ_e with probability $\frac{\gamma_1}{\beta_1+\gamma_1}$. In the example of Figure 3.4, σ_1 is drawn from ϕ_z. The second word is drawn by selecting σ_1 with probability proportional to the number of previous words that are from σ_1, a new draw from ϕ_z with probability proportional to β_1, or a new draw from φ_e with probability proportional to γ_1. In the case of Figure 3.4, the second word is drawn by the new draw σ_2 from ϕ_z. The next words are drawn with the same procedure.

Let $N_{w|e,z}$ denote the number of word-w tokens assigned with the pair (e, z), $\hat{N}_{w|z}$ denote the number of internal draws in $\{\sigma_1, \sigma_2, \dots\}$ whose values are w drawn from ϕ_z, and $\hat{N}_{w|e}$ denote the number of internal draws in $\{\sigma_1, \sigma_2, \dots\}$ whose values are w drawn from φ_e. Also, let $N_{\cdot|e,z} = \sum_{w \in W} N_{w|e,z}$, $\hat{N}_{\cdot|z} =$

[3] Our word generative process is an extension of the generative process described in [45], where they have only one base measure while ours has two base measures.

$\sum_{w \in W} \hat{N}_{w|z}$, and $\hat{N}_{\cdot|e} = \sum_{w \in W} \hat{N}_{w|e}$. Then, the predictive probability of word w in given z, e, $\mathcal{Z}$, $\mathcal{E}$, and Φ is:

$$\begin{aligned} & P(w|z, e, \mathcal{Z}, \mathcal{E}, \Phi) \\ & = \frac{N_{w|e,z} + \beta_1 \frac{\hat{N}_{w|z} + \frac{\beta_0}{W}}{\hat{N}_{\cdot|z} + \beta_0} + \gamma_1 \frac{\hat{N}_{w|e} + \frac{\gamma_0}{W}}{\hat{N}_{\cdot|e} + \gamma_0}}{N_{\cdot|e,z} + \beta_1 + \gamma_1} \end{aligned} \tag{3.4}$$

By combining Equation 2, 3, and 4, we can compute Equation 1.

Once we obtain entity-topic pair assignments for each word, we can estimate the parameters in the model accordingly.

3.3.4 Discussions on Special Cases

Another advantage of our model is that it has connections to previous topic models, and it turns out that LDA and the Author Model are both special (limiting) cases of our model.

If the concentration parameter β_1 is large and γ_1 is small relative to $N_{e,z}$, then counts $N_{e,z}$ are effectively ignored, and lead to have $P(w|z, e, \mathcal{Z}, \mathcal{E}, \Phi) \approx P(w|z, \mathcal{Z}, \mathcal{E}, \Phi)$. As $\beta_1 \to \infty$ and $\gamma_1 \to 0$, the role of entities in the model becomes ignored, and our model approaches to LDA.

By contrast, if concentration parameter γ_1 is large and β_1 is small relative to $N_{e,z}$, our model will have $P(w|z, e, \mathcal{Z}, \mathcal{E}, \Phi) \approx P(w|e, \mathcal{Z}, \mathcal{E}, \Phi)$. As $\beta_1 \to \infty$ and $\gamma_1 \to 0$, the role of topics in the model becomes ignored, and our model approaches to Author Model.

3.4 Experiments

We have two types of datasets to evaluate our model: a news article dataset and a DBLP dataset. In the news article datasets, we collected articles about Japan's Tsunami(2011) and London's Riot(2011) from NewsBank[4].

In Japan's tsunami disaster, a massive 8.9-magnitude earthquake shook Japan on March 11, 2011, causing a devastating tsunami to the coast of Japan. Due to the tsunami, the nuclear power plants in Fukushima were damaged, and one of the reactors in the Fukushima No. 1 nuclear plant

[4] `http://www.newsbank.com/`

partially melted down in the following day. As a result, the nuclear accident caused the exposure of nuclear radiation near the plant. We searched articles with "Japan Tsunami" keywords, and collected 2,000 articles published from Mar. 11, 2011 to Apr. 11, 2011.

In London's riot, Mark Duggan, a 29-year-old Tottenham resident, was shot and killed by a police. His death was followed by a protest against police and disturbances began on August 6, 2011. The violence spread across several cities, including Birmingham, Bristol Liverpool, and Manchester. More than 3,000 people were arrested and £200 million worth of property damage was incurred. We searched articles with "London Riot" keywords, and collected 2,000 articles published from Aug. 6, 2011 to Sep. 6, 2011.

Since news articles do not contain associated entity sets explicitly, we extracted entities mentioned in the articles. We used Zemanta[5], a high-performance online entity extraction and disambiguation service that links extracted entities to Wikipedia entries. Despite of many other available entity annotation tools, Zemanta was chosen because it has very high throughput and high precision [46]. After extracting entities, we discarded infrequent entities that appear in less than 5 documents. We also removed stop words and infrequent words that appear less than 5 documents.

The Digital Bibliography and Library Project (DBLP)[6] is a collection of bibliographic information on major computer science journals and proceedings. Each paper is represented by a bag of words that appear in the abstract and title of the paper. Also, its associated entity set is defined as the set of authors. In this experiment, we use a subset of the DBLP records that belongs to four areas: databases, data mining, information retrieval and artificial intelligence. We discard authors with less than 5 publications in our corpus. We again removed stop words and infrequent words that appear less than 5 documents. The three datasets are summarized in Table 3.2.

Our main claim is that word distributions should depend on associated entities as well as topics. For each dataset, as a case study, we show how word distributions change over topics with a fixed entity, and over entities with a fixed topic. In addition, we show rankings of entities for each topic as by-products of our model.

Finally, we compare our model with several baselines in terms of perplexity,

[5] http://zemanta.com

[6] http://www.informatik.uni-trier.de/~ley/db/

Table 3.2: Three Datasets with Statistics

Dataset Name	D	E	W	$avg(\|\boldsymbol{E}_d\|)$	$avg(N_d)$
Japan Tsunami	2,000	596	10,104	11.49	243.30
London Riot	2,000	585	11,016	11.57	233.68
DBLP	20,860	3,251	11,609	1.79	96.51

and investigate the parameters of our model.

For Japan's Tsunami and London's Riot datasets, we used $T = 20$, $\beta_1 = 100$, $\gamma_1 = 10$, and set other hyperparameters to 0.1. For the DBLP dataset, we used $T = 50$, $\beta_1 = 1000$, $\gamma_1 = 1$, and set other hyperparameters to 0.1. We used a relatively small number of topics when visually investigating word distributions from ETM. The evaluation of our model for different number of topics will be addressed in Section 3.4.1. The hyperparameters will be addressed in Section 3.4.2.

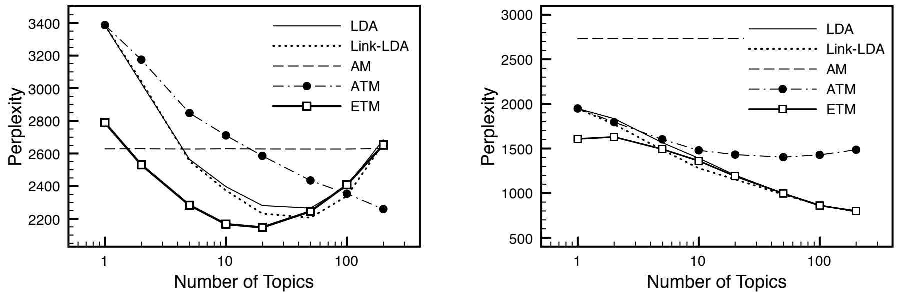
3.4.1 Perplexity Analysis

We compare our model with several baselines: LDA [20], Link-LDA [28], AM [26], and ATM [27]. Their hyperparameters are set to 0.1 except ATM, where the author suggested its hyperparameter settings: $\alpha = \frac{50}{T}$ and $\beta = 0.01$ in Figure 3.2(d). Perplexity is a standard measure for estimating the performance of a probabilistic model. We evaluate our model and compare with the baselines by estimating the perplexity of unseen held-out documents given some training documents. A better model will have a lower perplexity of held-out documents, on average. Perplexity is defined as $exp(-\frac{\log P(\mathcal{D}^{\text{test}}|\mathcal{D}^{\text{train}})}{\sum_{d\in\mathcal{D}^{\text{test}}} N_d})$. Let Φ denote the set of all parameters in a topic model. Then,

$$P(\mathcal{D}^{\text{test}}|\mathcal{D}^{\text{train}}) = \int P(\mathcal{D}^{\text{test}}|\Phi)P(\Phi|D^{\text{train}})\mathrm{d}\Phi$$

This integral can be approximated by averaging $P(\mathcal{D}^{\text{test}}|\Phi)$ under samples from $P(\Phi|D^{\text{train}})$. We used a Gibbs sampling to get 20 samples of Φ and *left-to-right* evaluation algorithm [47] to approximate $P(\mathcal{D}^{\text{test}}|\Phi)$. Note that AM, ATM, and ETM have generative processes of words for a given set of entities. Thus, $P(\mathcal{D}^{\text{test}}|\Phi)$ is defined as follows:

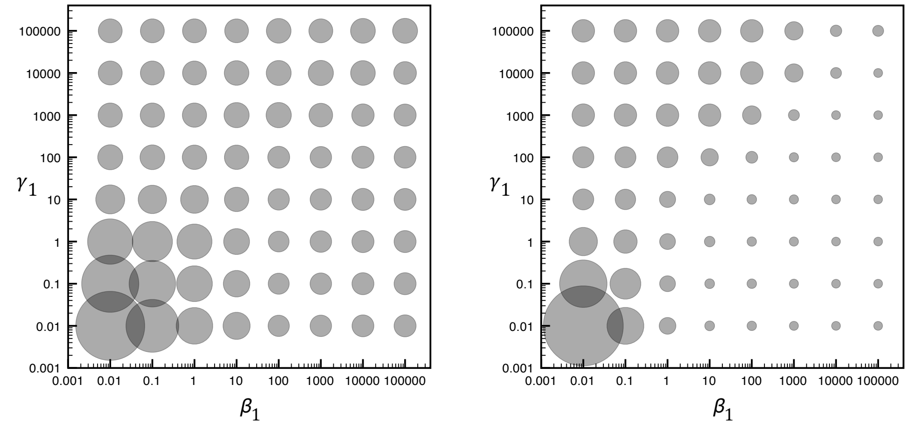
$$P(\mathcal{D}^{\text{test}}|\Phi) = \prod_{d\in\mathcal{D}^{\text{test}}} P(\boldsymbol{w}_d|\boldsymbol{E}_d, \Phi)$$



(a) Japan's Tsunami

(b) DBLP

Figure 3.5: Perplexity values for different number of topics

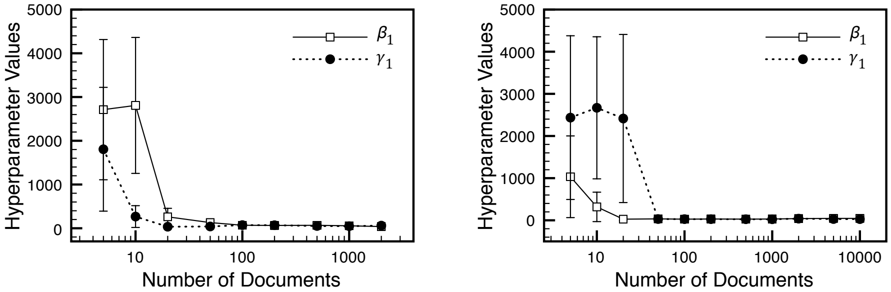


(a) Japan's Tsunami

(b) DBLP

Figure 3.6: Perplexity values for different β_1 and γ_1. The size of circle at each data point is proportional to its perplexity value.

We randomly sample 80% of the data as $\mathcal{D}^{\text{training}}$ and use the remaining 20% as $\mathcal{D}^{\text{test}}$. Figure 3.5 shows the perplexity values of our model and the baselines for different number of topics. Note that because AM does not have topics in its model, it has the same value regardless of the number of topics. Also, because LDA does not have entities in its model, LDA cannot take advantage of given associated entity sets. Generally, Link-LDA is slightly better than LDA because it uses the given associated entity sets as extra information to learn topic distributions in documents. Since ATM models a document generative process for a given set of entities, it is expected to have lower perplexity values than LDA. However, their experiments [27] with the corpus of NIPS papers showed that ATM has higher perplexity values than LDA because ATM model has large number of parameters to be estimated,



(a) For different number of documents (b) For different number of documents

Figure 3.7: The changes of the sampled parameters β_1 and γ_1 over the number of documents

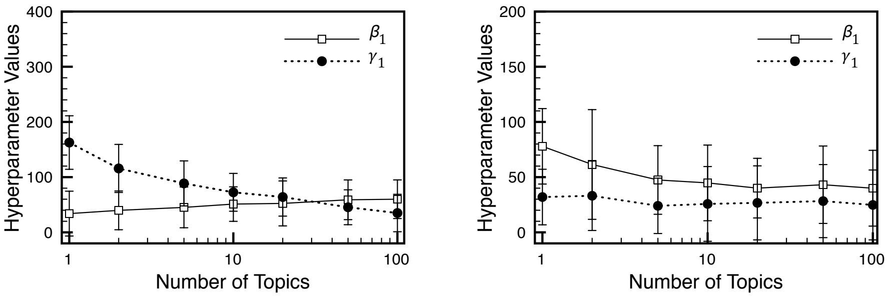
limiting its generalization performance. For DBLP dataset, ATM also has higher perplexity values than LDA as shown in Figure 3.5(b).

In Japan's Tsuanmi dataset, the perplexity values of LDA and Link-LDA decrease until they reach the lowest values at $T = 50$, and then begin to increase. When $T > 50$, LDA and Link-LDA have too many parameters in their models, causing an overfitting problem. The perplexity value of ETM decrease until it reaches to the lowest value at $T = 20$, and then begin to increase due to the overfitting problem like LDA and Link-LDA. Until $T = 20$, ETM outperforms the four baselines, and its lowest perplexity value is lower than the lowest perplexity values of the other models.

In the DBLP dataset, ETM has similar perplexity values as LDA and Link-LDA. The main reason is that most of the words in the research articles are related to research topics, and entity-specific topic-independent words are relatively rare in the corpus. For example, some coined words by an author can be entity-specific and topic-independent words, but such words are relatively rare unless the terms become popular in their related research communities. ETM, however, is still the best among all the models when the number of topics is small, and comparative to LDA and Link-LDA when number of topics is increasing, and much better than ATM for all the settings.

3.4.2 Parameter Studies

Among the six hyperparameters in our model, β_1 and γ_1 play the most important role. Depending on their values, our model slides between LDA and



(a) For different number of topics (b) For different number of topics

Figure 3.8: The changes of the sampled parameters β_1 and γ_1 over the number of documents and the number of topics in DBLP dataset

AM. For a given collection of documents, these parameters can be tuned by the perplexity analysis. Figure 3.6 shows the perplexity values for different values of β_1 and γ_1. For each pair of β_1 and γ_1, the size of circle is proportional to its perplexity value (smaller is better). For Japan's Tsunami dataset, ETM has the lowest perplexity value at $\beta_1 = 100$ and $\gamma_1 = 10$. For DBLP dataset, ETM has the lowest perplexity value at $\beta_1 = 1000$ and $\gamma_1 = 1$. With Figure 3.6, we can find appropriate parameter values for β_1, and γ_1. In addition, we can understand the characteristics of the corpus: topic-related words are dominant in DBLP dataset, while topic-related words and entity-related words are relatively balanced in Japan's Tsunami dataset.

Instead of enumerating parameter values and evaluating to find appropriate values, we can estimate them directly from a given corpus by sampling. As many studies suggested [45, 48], concentration parameters like β_1 and γ_1 can be given broad Gamma priors and inferred using slice sampling [49].

For Japan's Tsunami dataset, we sampled β_1 and γ_1. First, we sample them for different number of documents. In Figure 3.7, when training documents are very few, the sampled hyperparameters β_1 and γ_1 become large, leading to reduce its parameter space by weighting more on priors. Next, we sampled β_1 and γ_1 for different number of topics. As shown in Figure 3.8(a), β_1 increases as the number of topics increases. This is due to the quality of topics. When the model has better quality of topics, the word distributions $P(w|e, z)$ depend more on the topic z than the entity e. However, in Figure 3.8(b), β_1 and γ_1 are steady because in DBLP dataset words depend on topics more than authors when enough documents are given to model topics,

Table 3.3: Naoto Kan's Entity Prior (φ_e) and Word Distributions ($\psi_{e,z}$) of his Related Topics

Naoto Kan	Relief Efforts	Nuclear Accident	Economic Effects
kan	bodies	kan	prime
minister	search	minister	rule
prime	kan	prime	bill
naoto	people	naoto	kan
government	troops	nuclear	powerful
tokyo	car	radiation	business
crisis	crisis	plant	minister
troops	prime	evacuated	naoto
friday	confirmed	yukio	mind
party	business	reactors	term
assistance	told	urged	starting
democratic	minister	time	past
asked	lost	televised	march
kans	concrete	complex	loans
house	coastal	situation	financing
situation	centers	cabinet	economic
mr	center	fears	disaster
efforts	soldiers	crippled	april
conference	naoto	indoors	kans
spokeswoman	leaks	statement	yen

and such dependencies do not change even when we increase the number of topics.

3.4.3 Case Study 1: Japan's Tsunami

Since T is set to 20, we get 20 topics, including Tsunami, Nuclear Accident, Nuclear Radiation, Economic Effects, Industrial Effects, Relief Efforts, Tsunami Rescue, and so on[7].

Naoto Kan, who was the prime minister of Japan during the incident, was frequently mentioned in the corpus. He was involved in many topics like Relief Efforts, Nuclear Accident, and Economic Effects.

First, the top 20 words in the entity prior φ_e of Naoto Kan are shown in the first column in Table 3.3[8]. The entity prior can be interpreted as entity-

[7] The topics are manually named based on their word distributions.

[8] For simplicity, we omitted parameter values, and listed the top words

related and topic-independent word distribution for Naoto Kan. Combining with topic priors, the entity prior helps to shape the word distributions ($\psi_{e,z}$) of Naoto Kan in different contexts.

To support our main claim, we compare the word distributions ($\psi_{e,z}$) for Naoto Kan across different topics. Here, we show Naoto Kan in three different topics – Relief Efforts, Nuclear Accident, and Economic Effects. The top words are listed in the rest of columns in Table 3.3 based on their $\psi_{e,z}$ values. Note that there are "troops", "soldiers", "bodies", and "search" in Relief Efforts since the Japanese government had sent 50,000 troops for the rescue and recovery efforts, and "yukio" in Nuclear Accident refers to Yukio Edano who was the chief secretary of Japan's cabinet, leading the government to combat the aftermath of Nuclear Accident. As shown in Table 3.3, the word distributions ($\psi_{e,z}$) related to Naoto Kan vary significantly across the topics.

In the first column in Table 3.4, the top 20 words of the topic prior ϕ_z of Relief Efforts are listed. The topic prior can be interpreted as topic-related and entity-independent word distribution for Relief Efforts. The topic prior help to learn the word distributions ($\psi_{e,z}$) related to the entities involved in Relief Efforts.

We compare the word distributions ($\psi_{e,z}$) of three entities in the context of Relief Efforts – American Red Cross, Korea, and Tokyo. Even though American Red Cross and Korea are entities that had supported the Japanese people, their word distributions ($\psi_{e,z}$) are different: Korea has "sympathy" and "personal" in the top 20 words, and American Red Cross has "efforts" and "raise". Tokyo has the words "family", "friend", "home", and "email" because many articles mentioned that many people contacted with their family or friends in Tokyo via phone and e-mail. As shown in Table 3.4, the word distributions ($\psi_{e,z}$) related to Relief Efforts also change over the related entities and fit more to the entities.

As by-products, we can rank entities for each topic and rank topics for each entity. In contrast to ATM [27], our model does not model the relationship between entities and topics directly. Our model, however, can get their relationship indirectly for a given assignments $\mathcal{E}$ and $\mathcal{Z}$. Let $N_{\cdot|e,z}$ denote the number of words that are assigned with (e, z). Also, let $N_{\cdot|\cdot,z} = \sum_e N_{\cdot|e,z}$ and $N_{\cdot|e,\cdot} = \sum_z N_{\cdot|e,z}$. Then, $P(e|z, \mathcal{E}, \mathcal{Z}, \Phi) = \frac{N_{\cdot|e,z}}{N_{\cdot|\cdot,z}}$, and $P(z|e, \mathcal{E}, \mathcal{Z}, \Phi) = \frac{N_{\cdot|e,z}}{N_{\cdot|e,\cdot}}$. Based on $P(e|z, \mathcal{E}, \mathcal{Z}, \Phi)$ and $P(z|e, \mathcal{E}, \mathcal{Z}, \Phi)$, we can rank entities for each topic, and rank topics for each entity.

Table 3.4: Relief Effort's Topic Prior (ϕ_z) and Word Distributions ($\psi_{e,z}$) of Its Related Entities

Relief Efforts	American Red Cross	Korea	Tokyo
japan	cross	japan	people
japanese	red	japanese	japan
people	japan	korea	friends
tsunami	american	korean	japanese
earthquake	relief	donations	tokyo
disaster	support	koreans	tsunami
world	donations	sympathy	back
relief	donation	march	earthquake
money	disaster	earthquake	home
time	raise	helping	email
country	march	hard	devastating
damage	efforts	victims	family
friends	affected	support	earthquakes
information	tsunami	collected	student
aid	victims	quake	miles
week	earthquake	personal	concerned
affected	money	people	watch
nation	thursday	news	live
march	people	money	concern
devastation	located	important	close

Table 3.5 shows two topics and their entity rankings. Nuclear Accident and Nuclear Radiation have three entities in common in the top entities: Tokyo Electric Power Company, Fukushima Nuclear Power Plant, and Potassium iodide. Tokyo Electric Power Company is the operating company of Fukushima Nuclear Power Plant, and one of the nuclear reactors in Fukushima Nuclear Power Plant had been damaged and started to melt down. Potassium iodide is an inorganic compound that is used as drugs to prevent Thyroid cancer caused by radioactive chemicals. However, the rest of entities are very different. Note that there are Nuclear Regulatory Commission, U.S. Environmental Protection Agency, and Seawater in the top entities of Nuclear Accident: Nuclear Regulatory Commission oversees nuclear reactor safety, U.S. Environmental Protection Agency protects human health and the environment by enforcing related regulations, and Seawater was used to cool down the nuclear reactor. On the other hand, there are Iodine-131, Caesium, Thyroid, and Tap Water in the top entities of Nuclear Radiation: Iodine-131 and Caesium are the emitters of strong gamma radiation

Table 3.5: Entity Rankings for Different Topics in Japan Tsunami Dataset

Nuclear Accident	Nuclear Radiation
Nuclear Regulatory Commission	Tokyo Electric Power Company
Nuclear power plant	Fukushima Nuclear Power Plant
Chernobyl disaster	Electrical grid
Japan	Tap water
Tokyo Electric Power Company	Caesium
Libya	Iodine-131
Potassium iodide	Thyroid
U.S. Environmental Protection Agency	Radiation
Seawater	Yukio Edano
Fukushima Nuclear Power Plant	Raw Milk
Barack Obama	Potassium iodide
Automotive industry	Thyroid cancer

that causes cancers and even death. Those radioactive chemicals can dissolve in water, and people may get exposed to the radioactive chemicals by drinking Tap Water.

Similarly, it is possible to analyze topic rankings for each entity.

3.4.4 Case Study 2: DBLP – Research Articles

In this section, we performed a similar analysis with the DBLP corpus. As introduced in Section 3.1, Judea Pearl is the 2011 winner of the A.M. Turing Award for *"for fundamental contributions to artificial intelligence through the development of a calculus for probabilistic and causal reasoning."*[9] He is credited for inventing Bayesian networks, and several inference methods in the models. He later developed a theory of causal and counterfactual inference based on structural models.

First, the top 20 words in the entity prior φ_e of Judea Pearl are shown in the first column in Table 3.6. The entity prior can be interpreted as the word distribution of his general methodologies, approaches, or research interests for Judea Pearl. There are "casual" and "counterfactual" in his entity prior φ_e, indicating his research interests are casual and counterfactual inference across his research topics. Combining with topic priors, his entity prior helps to shape the word distributions $\psi_{e,z}$ of Judea Pearl in different research topics.

[9] http://amturing.acm.org/award_winners/pearl_2658896.cfm

Table 3.6: Judea Pearl's Entity Prior (φ_e) and Word Distributions ($\psi_{e,z}$) of His Related Research Topics

Judea Pearl	Knowledge Representation	Reasoning	Bayesian Network
causal	reasoning	logic	causal
revisited	default	dependencies	distributions
optimality	causal	probabilistic	models
markovian	formal	graphs	markovian
counterfactual	systems	directed	semi
explanations	computational	representing	identification
symbolic	specificity	dags	characterization
independence	causality	programs	recursive
path	diagnostic	reasoning	joint
specificity	inheritance	conditional	variables
scout	representation	bases	data
independencies	model	based	effects
dependence	knowledge	efficient	algorithm
proven	system	networks	based
embracing	inference	programming	clustering
dags	common	probability	network
tolerating	rule	undirected	arbitrary
economy	embracing	causal	graph
states	coherence	inference	bayesian
counterfactuals	belief	belief	networks

Based on $P(z|\mathsf{Judea\ Pearl}, \mathcal{E}, \mathcal{Z})$, we selected his top 3 research topics: Knowledge Represetation (KR), Reasoning, and Bayesian Network[10]. With his general approaches "casual" and "counterfactual", he has involved in these research topics. The top words of the word distributions ($\psi_{e,z}$) are listed in the rest of the columns in Table 3.6, indicating how his approach is applied in different research topics.

For KR, we select the top three authors based on $P(e|\mathsf{KR}, \mathcal{E}, \mathcal{Z})$. Even though they have published papers on KR, their approaches are very different from each other. Pedro Domingos has focused on learning Markov logic networks, Benjamin Kuipers has developed qualitative models to express states of incomplete knowledge about continuous mechanisms and QSIM algorithm for qualitative simulation. Marzena Kryszkiewicz has taken very different approaches that use frequent patterns to generate rules as knowledge. Even

[10] These topics are manually named based on their topic priors.

Table 3.7: Knowledege Represetntation's Topic Prior (ϕ_z) and Word Distributions ($\psi_{e,z}$) of Its Related Entities

Knowledge Representation	Pedro Domingos	Benjamin Kuipers	Marzena Kryszkiewicz
knowledge	logic	qualitative	frequent
reasoning	markov	simulation	patterns
system	networks	reasoning	representation
representation	learning	knowledge	free
based	world	quantitative	disjunction
design	mlns	systems	generalized
systems	knowledge	incomplete	concise
qualitative	order	mechanism	based
logic	structure	abstraction	generators
theory	logical	envisionment	representations
learning	real	physical	negations
domain	models	behavior	oriented
planning	unifying	causal	support
problem	ilp	system	knowledge
agent	systems	description	sets
expert	purely	expert	condensed
model	representation	process	reasoning
approach	reasoning	behaviors	survey
models	viewing	logic	system
support	mln	model	borders

though the three authors have very different approaches for the same research topic, and thus have very different word distributions, our model aligns them under KR topic to make them comparable. This comparison is not possible without modeling $P(w|e, z)$.

As we did for Japan's Tsunami dataset, we can rank entities for each topic, and topics for each entity, as shown in other studies [27, 1]. Due to the space limit, we will not show them in this study.

3.4.5 Case Study 3: London's Riot

In this section, we show the power of ETM in document retrieval. Since we have topics and entities for organizing documents, we can use a pair of an entity e and a topic z as a query to retrieve relevant documents. For a given query $\langle e, z \rangle$, we rank the documents by a score function as follows:

Table 3.8: Entity Rankings for Different Topics in DBLP Dataset

Text Classification	Web Search
Andrew McCallum	Barry Smyth
Haym Hirsh	Bing Liu
Rong Jin	Ryen W. White
Doina Precup	Marius Pasca
Wenyuan Dai	Wei-Ying Ma
Aidong Zhang	Hongkun Zhao
Rayid Ghani	Marc Najork
Qiang Yang	Yannis Papakonstantinou
Kotagiri Ramamohanarao	Krithi Ramamritham
Alexandru Niculescu-Mizil	Qiang Yang
Vikas Sindhwani	Ji-Rong Wen
Massih-Reza Amini	Hua-Jun Zeng

$$Score(\langle e, z\rangle, d) = \frac{\sum_i I(e_{d,i} = e \wedge z_{d,i} = z)}{N_d}$$

We first ranked and listed top 5 documents using three different queries that have the same topic but different entities. The three queries and the corresponding top 5 documents are shown in Table 3.9. We fixed topics as Violence and used the three entities Mark Duggan, Olympic Game, and Water Cannon to see the difference between the top ranked documents with the queries.

With the query ⟨Mark Duggan, Violence⟩, we found news articles that explain how the death of Mark Duggan caused a protest and spread the violence. Using the query ⟨Olympic Game, Violence⟩, we searched news articles that explain how the violence affected the preparation of London Olympics that was less than a year away. In addition, using the query ⟨Water Cannon, Violence⟩, we found news articles about the riot control tactics by police including using water cannons.

Likewise, we fixed entities as Police and used different topics. We used three different topics Riot Control, Society, and Sporting Fixtures as queries. Table 3.10 shows the top ranked documents.

As shown in this section, we can use ETM to organize and retrieve documents in two different dimensions or aspects – topics and entities – for a given corpus.

Table 3.9: Queries of $\langle *, \mathsf{Violence} \rangle$ and Titles of Relevant News Articles

Query: $\langle \mathsf{Mark\ Duggan}, \mathsf{Violence} \rangle$
1. As trouble spreads, did police fire the bullet that sparked riots ? - As trouble spreads, did police fire first shots in gun drama? 2. Police apologise to Duggan family for failing to keep them informed 3. The truf war that fuelled riots 4. A dead man and a crucial question: should police have shot Mark Duggan? - News Cahal Milmo and Rob Hastings reconstruct the fateful events of Thursday evening that sparked three days of rioting 5. How fatal shooting of mini-cab driver sparked protests
Query: $\langle \mathsf{Olympic\ Game}, \mathsf{Violence} \rangle$
1. Burning rings of fire - Olympic organisers feel the strain as riots sweep across London 2. Lawless London a worry for Games organisers, says Shirvington 3. Riots in London a concern with Olympics looming 4. London deploys extra officers to quiet riots 5. Three die as riots flare up again
Query: $\langle \mathsf{Water\ Cannon}, \mathsf{Violence} \rangle$
1. Plastic bullets authorised for use on British mainland: 16,000 officers flood capital as authorities change tactics: Trouble flares up again in Manchester and Birmingham 2. The fightback is under way: PM's pledge to battered cities 3. Sporadic nature of violence complicates challenge for police 4. Residents demand tougher policing after third night of burning, looting 5. Cameron allows water cannon to crush riots

Table 3.10: Queries of $\langle \mathsf{Police}, * \rangle$ and Titles of Relevant News Articles

Query: $\langle \mathsf{Police}, \mathsf{Riot\ Control} \rangle$

1. Police strength crucial for a strong community
2. Western Morning News: Police officers cover for striking control room staff
3. Keep your water cannon: What police most need to quell riots isn't fancy new weapons but unequivocal support from the public
4. PM defends bid for police "figurehead"
5. More police, not less - your letters

Query: $\langle \mathsf{Police}, \mathsf{Society} \rangle$

1. Many flaws in policing plan
2. The Daily Telegraph: We have the chance to recover Britain's streets for civilisation
3. Met police are being spread too thinly
4. Another perspective on London riots - London's raging riots spread north, Aug. 9
5. Britain's August riots - Civil disorder and looting hits Britain

Query: $\langle \mathsf{Police}, \mathsf{Sporting\ Fixtures} \rangle$

1. Cheltenham's game called off because of violence
2. Broadcasters forced to hand over riot footage to the police
3. Riots force Tottenham postponement
4. Riots were "disgusting", says assistant chief constable
5. U.K. police don't take aim, but critics open fire

CHAPTER 4

PATTERN MINING FOR FEATURE NODE GENERATION

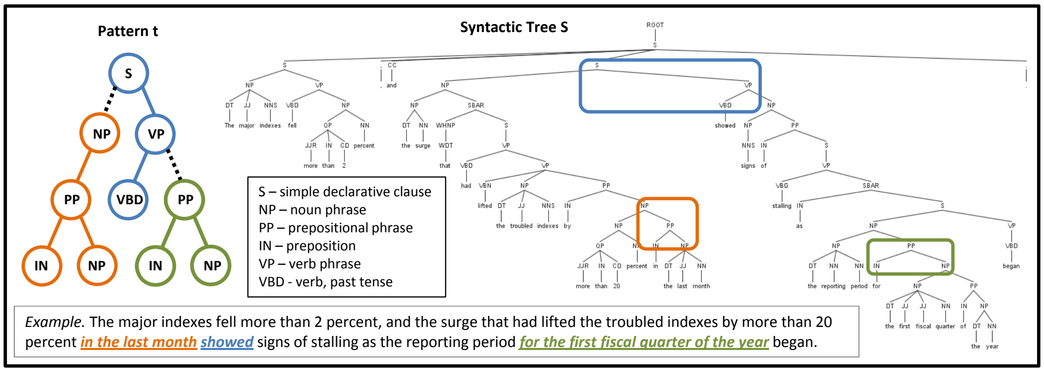
4.1 Overview

In computational linguistics and text mining domains, there are three classical classification problems: topic classification, genre classification, and authorship classification. Among these three problems, arguably the most difficult is the classification of documents in terms of their authorship (known as *authorship classification*, authorship attribution and/or authorship discrimination). This problem can be thought of as classifying documents based on the writing styles of the authors. This is a nontrivial problem even for humans: while a human can easily identify the topic and genre of a given document, identifying its authorship is harder. If the documents are in the same topic and genre, the task becomes much harder.

In the era of excessive electronic texts, authorship classification has become more important than ever before with a wide variety of applications. Besides the early works of analyzing the disputed plays of *Shakespeare*(1887) [50] or anonymous documents of *The Federalist Papers*(1964) [51], it could also be used to identify authors of short 'for sale' messages in a newsgroup [52] and even for forensic investigations by identifying authorship of e-mail messages [53]. Detecting plagiarism or copyright infringement of unauthorized reuse of source code by establishing a profile of an author's style is another important application of authorship classification [54].

Existing approaches to authorship classification use various methods to extract effective features, the most common of which include style markers such as function words [55, 56, 57, 58] and grammatical elements such as part of speech (*POS*) tags [59, 60, 61]. Function words are common words (*e.g.* articles, prepositions, pronouns) that have little semantic content of their own but usually indicate a grammatical relationship or generic property.

Recently, there have been several papers that claimed function words are more effective than other types of style markers [56, 61, 62].



S – simple declarative clause
NP – noun phrase
PP – prepositional phrase
IN – preposition
VP – verb phrase
VBD - verb, past tense

Example. The major indexes fell more than 2 percent, and the surge that had lifted the troubled indexes by more than 20 percent ***in the last month*** ***showed*** signs of stalling as the reporting period ***for the first fiscal quarter of the year*** began.

Figure 4.1: A 2-*ee* subtree t is mined from two *The New York Times* journalists Jack Healy and Eric Dash who worked in the same business department. On average, 21.2% of Jack's sentences contained t while only 7.2% of Eric's sentences contained t.

Unfortunately, research on more complex syntactic structures has not been practical because of the lack of a reliable, automatic tool which retrieves syntactic structures, and because of the high computational cost associated with syntactic structure-based algorithms. Instead, several variations of *POS* tags [55, 60, 63] and rather simple syntactic structures like rewrite rules [59, 60, 63] have been proposed. Among them, bigram *POS* tags and rewrite rules showed reliable performance in various dataset configurations.

Recently, several advanced techniques have been developed which greatly improved the performance of *Natural Language Processing*(*NLP*) tools[1] enabling reliable, highly accurate sentence parsing into a syntactic tree of *POS* tags. A *syntactic tree* is a rooted and ordered tree that is labeled with *POS* tags that represent the syntactic structure of a sentence. Based on the syntactic trees parsed by these tools, we propose a novel syntactic feature set of tree fragments allowing at most k-embedded edges (in short, a k-*ee* subtree). We say there is an embedded edge between two nodes if and only if they are in an ancestor-descendant relationship but not in a parent-child relationship. Compared with previous feature sets that consist of parts of distinct connected subtree components, our new feature set captures the relationship

[1] We used Stanford Parser (`http://nlp.stanford.edu/software/lex-parser.shtml`), but there are more tools available like **N**atural **L**anguage **T**ool**K**it (NLTK) package (`http://www.nltk.org`).

between k+1 connected subtree components of a syntactic tree, which leads to a better representation of datasets consisting of long and complex sentences. Figure 4.1 gives an example of a *k-ee* subtree t for $k = 2$. Pattern t is composed of three smaller subtrees, which are connected by two embedded edges (S,NP) and (VP,PP). The differences in pattern distributions between two authors suggest that a set of *k-ee* subtrees can be utilized as a good feature set for authorship classification.

To reduce the number of features, we only mine a set of frequent and discriminative *k-ee* subtrees, which results in higher accuracy by avoiding overfitting to the training data and by not generating non-discriminative features that often degrade the performance. This task is commonly referred to as pattern-based classification. The original pattern-based classification technique employed a two-step procedure called *generate-and-test* which generates all frequent and closed candidate patterns and then selects the discriminative patterns among them [64]. Unfortunately, it is still intractable to use this *generate-and-test* methodology to get discriminative patterns because there are simply too many candidate patterns.

For this reason, there have been quite a few works which *directly* mine discriminative patterns without generating all candidates [65, 66, 67]. Yet, these existing works cannot be directly applied to our problem setting because they require the feature values to be binary. Instead, we require numeric feature values because a (syntactic) feature can occur multiple times in a document and usually the number of occurrences implies its importance. Existing works are all based on binary-valued features and their theorems and proofs are not easily extendable to numeric-valued features. A recent work ([68]) showed that it has more gain to use numeric values than to discretize them into binary values. It also proposed a new way to directly mining discriminative numeric features by solving a linear programming optimization problem. But all these previous works mine top-1 pattern iteratively until the mined patterns cover the entire data. To cope with this issue, we derive an upper bound of a discriminative score of numeric-valued features, and develop an efficient algorithm that mines in one iteration a set of discriminative patterns to be used for classification purpose.

To validate the utility of our new feature set compared to others, for fair comparisons, we apply the same *SVM* classification algorithm using various feature sets on several real data collections. Because of its high and reliable performance, *SVM* has commonly been used to compare the effectiveness of feature sets [60, 61, 63]. Experimental results demonstrate the effectiveness of the proposed *k-ee* subtree features in comparison to the well-known existing feature sets of function words, *POS* tags, and rewrite rules. We demonstrate that by using *k-ee* subtrees as the feature set we outperform the existing feature sets by 8.23% on average and show that it is significantly better from other approaches by t-test with 95% confidence level.

In summary, the contributions of this chapter are as follows:

- We propose a new feature set of *k-ee* subtrees for authorship classification.

- We develop an efficient algorithm to directly mine discriminative *k-ee* subtrees, which are not binary but numeric valued features, in one iteration.

- Through comprehensive experiments on various datasets, we demonstrate the utility of our proposed framework to provide an effective solution for the authorship classification problem.

The rest of the chapter is organized as follows. In Section 4.2, we introduce various preliminary concepts and define our new feature set of *k-ee* subtrees. Section 4.3 explains a *branch-and-bound* framework of discriminative *k-ee* subtree mining. We report experimental results in Section 4.4.

4.2 k-Embedded-Edge Subtree

Previous authorship attribution approaches adopted function words, *POS* tags, and rewrite rules as a feature set to build a classification model. Even though they achieved good accuracy, there still exists room for a more meaningful feature set to improve the performance. In this section, we describe rewrite rules which are somewhat complex syntactic structures that hold more syntactic information than the other two feature sets. Also, we define our new feature set of *k-ee* subtree patterns.

4.2.1 Rewrite Rule

In [59], rewrite rules were considered to be building blocks of a syntactic tree, just as words are building blocks of a sentence. Here, a *syntactic tree* is a rooted and ordered tree which is labeled with *POS* tags that represents the syntactic structure of a sentence. Its interior nodes are labeled by non-terminals of the grammar, and the leaf nodes are labeled by terminals.

Compared to previous approaches that utilized function words and *POS* tags, rewrite rules can hold functional structure information of the sentence. In linguistics, a rewrite rule is in the form of "$X \rightarrow Y$" where X is a syntactic category label and Y is a sequence of such labels such that X can be replaced by Y in generating the constituent structure of a sentence. For example, "*NP* → *DT*+*JJ*+*JJ*+*NN*" means that a noun phrase (*NP*) consists of a determiner (*DT*) followed by two adjectives (JJ) and a noun (*NN*).

There is a limit when using rewrite rules as features of a classification model. First, because of the restriction that the entire rule cannot be broken into smaller parts, no similarity between rules are considered. A large number of slightly different rules are all counted as independent features. For instance, a rewrite rule "*NP* → *DT*+*JJ*+*NN*", missing one JJ from the above example, becomes a separate rewrite rule. Second, the expressibility of rewrite rules is limited because they must adhere to a very strict two-level tree structure, which does not allow the entire rule to be broken into smaller parts. For example, the relationships between rewrite rules are missing, which can hold more refined syntactic information. For these reasons, we developed a new feature set of *k-ee* tree patterns that are flexible and complex enough to represent the syntactic structure information of a sentence.

4.2.2 k-Embedded-Edge Subtree

To overcome the drawbacks of simple syntactic feature sets used in previous approaches, we explore more complex syntactic features. Induced subtrees of a syntactic tree are one of the candidate feature sets whose features are multi-level tree fragments used to model the complex syntactic structure of a sentence. Here, we define a tree t to be an *induced subtree* of a tree s if there exists an identity mapping from t to s preserving all parent-child relationships

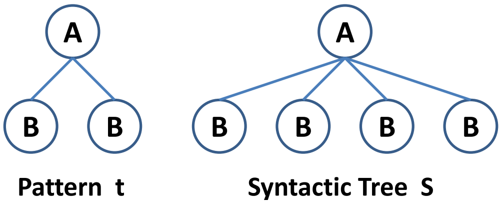


Figure 4.2: Example of overcounting overlapped *k-ee* subtree occurrences

between the nodes of t. Our pilot experiments showed that a small number of combinations of those induced subtrees could achieve even higher accuracy, which motivated us to define *k-ee* subtrees for our new feature set. Based on this motivation, we designed a new tree pattern that can capture this phenomenon.

Definition 1. *We define an* **embedded** *edge e of a tree s to be a pair of two nodes with an ancestor-descendant relationship. We define a* **k-embedded-edge subtree** (shortly, **k-ee subtree**) *t of a tree s to be a set of induced subtrees of s that can be connected by at most k embedded edges (not with parent-child relationships) for a user specified value k.*

The number of *k-ee* subtrees would be exponential on the number of trees and their sizes. We define a minimum support θ to ensure we only mine general common patterns that will be applicable to test data thus avoiding overfitting. We define the *support* of a feature t (denoted by $sup(t)$) to be the total number of sentences in training data that contains t. We say t is *frequent* if and only if $sup(t) \geq \theta$ for a user-specified minimum support threshold θ.

4.2.3 Document Representation based on Discriminative k-ee Patterns

The frequency of a pattern in a document (or a set of syntactic trees) is quite important in the sense that it can be a good measure to discriminate the writing styles of different authors. Well-known features like function words, and the *POS* tag-adapted *bag-of-words* approach use the number of occurrences in a document as their frequency measure. However, unlike function words and *POS* tags, *k-ee* subtrees cannot simply adapt the same frequency measure because it generates overlapped occurrences, which would lead to an exaggerated frequency value. Figure 4.2 is an illustration of this overcounting problem. The syntactic tree S has only one A and four Bs, but the number of occurrences of pattern t becomes 6. More generally, if A has n Bs as its children in S, then the occurrence count of pattern t becomes $O(n^2)$. Since we allow k embedded edges for a *k-ee* subtree, this overcounting problem will be even more amplified.

Our observation that a document is parsed into a set of syntactic trees (of sentences) gave us an insight to define the frequency measure of a *k-ee* subtree in a more natural way by counting the number of syntactic trees of a document that contain the pattern.

Definition 2. *We define the* **frequency** *of a k-*ee *subtree t in a document d (denoted by $freq(t, d)$) to be the number of syntactic trees (*i.e.*, parsed sentences) in d that contain t over the total number of sentences in d.*

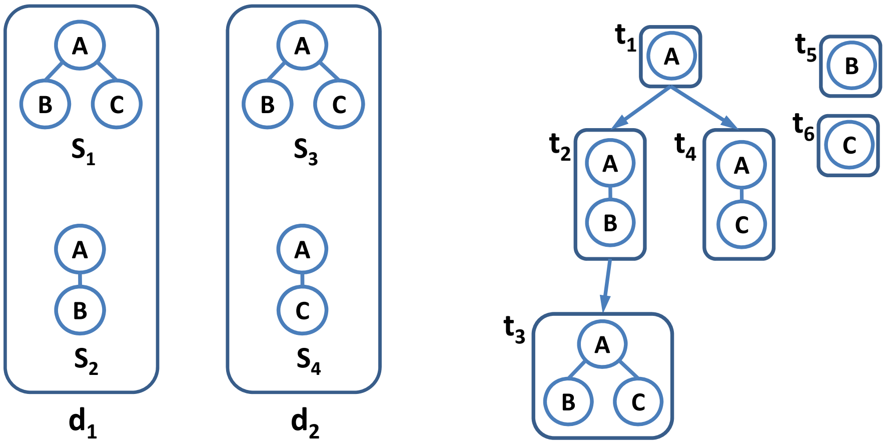
We will discuss how to mine discriminative *k-ee* subtree patterns in the following section (Section 4.3). For here, suppose we already have them in a set $P = \{t_1, \cdots, t_n\}$. Then, we can express a document d as a vector of their frequencies as $d = (freq(t_1, d), \cdots, freq(t_n, d))$.

4.3 Discriminative k-ee Subtree Mining

In the previous section, we introduced *k-ee* subtrees as a new feature set for authorship classification. These patterns hold more expressive syntactic information than other features and are flexible enough to consider partial matchings of syntactic trees, but the number of *k-ee* subtrees is above our control. Therefore, we need to directly mine a small number of discriminative patterns not only to reduce the number of features but also to mine significant patterns which has been shown to improve classification accuracy [69]. In this section, we present a branch-and-bound framework to solve this problem.

4.3.1 Mining Frequent k-ee Subtrees: Pattern-Growth Approach

We do not generate candidate *k-ee* subtrees and check for frequent attributes. Instead, we find a frequent *k-ee* subtree and extend it by adding a node that



(a) A toy database $\mathcal{D}$ with 2 documents. Each document has 2 syntactic trees.

(b) Pattern growth of k-*ee* subtrees with $\theta = 0.5$ and k=0

Figure 4.3: Database $\mathcal{D}$ and its frequent k-*ee* subtrees

is guaranteed to be frequent in a depth-first manner, which enables several pruning techniques for frequent and discriminative pattern mining. We first introduce how to efficiently mine frequent patterns based on pattern-growth approach by using projected database [70, 71], and then explain pruning techniques to mine discriminative patterns.

We illustrate the procedure for pattern-growth approach as follows. First, find a size-1 frequent k-*ee* subtree t in the training dataset $\mathcal{D}$. Second, project the postfix of each occurrence of t in the syntactic trees of $\mathcal{D}$ into a new database $\mathcal{D}_t$. A *postfix* of an occurrence of t in a syntactic tree s is a forest of the nodes of s appearing after the occurrence of t in a pre-order scan of s. Third, find a frequent node v in $\mathcal{D}_t$ that can be attached to the rightmost path of t that forms a k-*ee* subtree. Once v is frequent in $\mathcal{D}_t$, it ensures that the extended pattern is also frequent, so we do not need to scan the whole database $\mathcal{D}$ again. Note that, in this study, we consider a node v attached to t by an (induced) edge different from the one attached by an embedded edge. Fourth, recursively go back to the second step with the extended pattern for every frequent node we find. Note that the projected database of a pattern t keeps shrinking as the mining process moves on and t becomes a bigger superpattern.

Example 1. *Figure 4.3 shows an example of the pattern-growth approach to mine 0-*ee *subtrees from a database $\mathcal{D}$ of four syntactic trees when minimum* *support threshold is 0.5. Each pattern is indexed in pattern-generation order. We first search for size-1 frequent patterns, which are t_1, t_5 and t_6. We choose t_1 as a starting point, and find frequent nodes that can be attached to t_1 from its projected database. We find that nodes B and C are frequent, and we extend t_1 to t_2 by adding a node B. Similar procedures are recursively performed until we mine all frequent patterns.*

4.3.2 Binned Information Gain Score

In previous subsections, we presented a *pattern-growth* method to mine frequent patterns, but the resulted patterns may still be too many. Based on the study that the patterns with high discriminative score can improve the classification performance [69], we first evaluate the discriminative power of a *k-ee* subtree. Note that most of the well-known discriminative scores (*e.g.* information gain, fisher score) have upper bound on binary feature values not on numeric feature values [69, 65, 67, 72]. In this subsection, we define a new discriminativeness score, *binned information gain*, and derive its upper bound on the numeric feature values to enable a *branch-and-bound* framework to mine discriminative patterns on numeric feature values.

Definition 3. *For a user specified number* n*, we divide range* $[0, 1]$ *of the relative sentence frequency per document of* t *into a partition* p *of equi-width* n *bins:* $p_1 = [0, \frac{1}{n})$, $p_2 = [\frac{1}{n}, \frac{2}{n})$, $\cdots$, $p_{n-1} = [\frac{n-2}{n}, \frac{n-1}{n})$, $p_n = [\frac{n-1}{n}, 1]$*. For a given partition* p *and* m *classes* $C_1, \cdots, C_m$*, we define the* binned conditional entropy *of* t *by*

$$H(C|X) = -\sum_{i=1}^{n} P(X \in p_i) \sum_{k=1}^{m} P(C_k|X \in p_i) \log p(C_k|X \in p_i)$$

and binned information Gain *of* t *by* $IG(C|X) = H(C) - H(C|X)$ *where* $H(C) = -\sum_{k=1}^{m} p(C_k) \log p(C_k)$*.*

A pattern t will have a large binned information gain score if the frequency distribution imbalance between the classes becomes bigger for each bin, which means t is significant to discriminate classes.

Figure 4.4 presents binned information gain score distributions of various feature sets such as function words (FW), *POS* tags (POS), bigram *POS* tags (BPOS), rewrite rules (RR), and *k-ee* subtrees for k=0, 1, and 2 (0-*ee*,

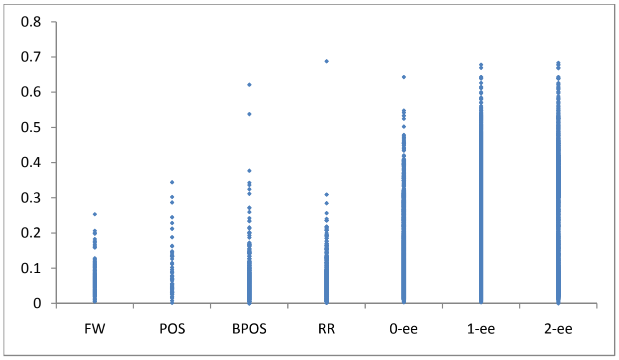


Figure 4.4: Binned information gain score distribution of various feature sets

1-*ee*, and 2-*ee*, respectively). We can easily see that the highest scores are mostly from *k*-*ee* subtrees, which implies that they can be more meaningful than other features – an assertion we later test in the experiments section.

For a tree pattern t, we denote binned information gain of t by $IG(t)$ and information gain upper bound of t and its superpatterns by $IG_{ub}(t)$. Given a k-*ee* subtree t and a partition p, we define (A, B, p) to be a *frequency distribution* of t where $A = (A_1, \ldots, A_n)$ and $B = (B_1, \ldots, B_n)$ with A_i and B_i being the number of documents in class C_1 and C_2 respectively for each bin p_i of a partition p. Denote (A', B', p) as a frequency distribution of a super pattern t' of t. The following two lemmas describe the properties of (A, B, p) and (A', B', p) that will be used to prove the main theorem to derive the upper bound of binned information gain.

Lemma 1. *For any $k = 2, \ldots, n$, the following four inequalities hold for a k-*ee *subtree t and its superpattern t': $\sum_{i=k}^{n} A'_i \le \sum_{i=k}^{n} A_i$, $\sum_{i=1}^{k-1} A'_i \ge \sum_{i=1}^{k-1} A_i$, $\sum_{i=k}^{n} B'_i \le \sum_{i=k}^{n} B_i$, and $\sum_{i=1}^{k-1} B'_i \ge \sum_{i=1}^{k-1} B_i$.*

Proof. Since t' is a superpattern of t, $\sum_{i=k}^{n} A'_i \le \sum_{i=k}^{n} A_i$ for $k \ge 2$. Therefore, $\sum_{i=1}^{k-1} A_i = |C_1| - \sum_{i=k}^{n} A_i \le |C_1| - \sum_{i=k}^{n} A'_i = \sum_{i=1}^{k-1} A'_i$ where $|C_i|$ is the number of documents in class C_i. Similar proof for B_i. □

The following lemma shows the condition to get the upper bound of binned information gain for a special case when only the first two bins of frequency distribution are different.

Lemma 2. *For a given frequency distribution (A, B, p), let (A', B', p) be a frequency distribution with $A'_1 = A_1 + x$, $A'_2 = A_2 - x$ $(0 \le x \le A_2)$ and* *the rest unchanged. If $\frac{A_1}{A_1+B_1} \geq \frac{A_2}{A_2+B_2}$, then (A', B', p) achieves its minimum conditional entropy when $x = A_2$. Otherwise, it achieves its minimum conditional entropy when $x = 0$.*

Proof. Let $f(x)$ be the conditional entropy of $(A'$,B',$p)$ and N be the total number of documents. Then,

$$\begin{aligned}
\mathcal{T}f(x) &= \frac{A_1+B_1+x}{N}\Big(-\frac{A_1+x}{A_1+B_1+x}\log\frac{A_1+x}{A_1+B_1+x} \\
&- \frac{B_1}{A_1+B_1+x}\log\frac{B_1}{A_1+B_1+x}\Big) \\
&+ \frac{A_2+B_2-x}{N}\Big(-\frac{A_2-x}{A_2+B_2-x}\log\frac{A_2-x}{A_2+B_2-x} \\
&- \frac{B_2}{A_2+B_2-x}\log\frac{B_2}{A_2+B_2-x}\Big) \\
&+ \sum_{i=3}^{n} P(X \in p_i)\sum_{k=1}^{2} P(C_k|X \in p_i)\log p(C_k|X \in p_i) \\
f'(x) &= \frac{1}{N}\log\left(\frac{A_1+B_1+x}{A_1+x}\cdot\frac{A_2-x}{A_2+B_2-x}\right)
\end{aligned}$$

If $\frac{A_1}{A_1+B_1} \geq \frac{A_2}{A_2+B_2}$, $f'(x) \leq 0$. Otherwise, $f'(x) > 0$. □

The following theorem describes that the binned information gain upper bound exists and is determined by the frequency distribution of the first two bins.

Theorem 1. *Given a tree pattern t, its super patterns including itself have a conditional entropy lower bound in the frequency distribution (A', B', p) of one of the following two forms: (1) $A'_1 = A_1 + A_2$, $B'_2 = \sum_{i=2}^{n} B_i$, $B'_1 = B_1$, $B'_i = 0$ $(i = 2, \ldots, n)$ and $A'_i = A_i$ $(i = 3, \ldots, n)$ (2) $B'_1 = B_1 + B_2$, $A'_2 = \sum_{i=2}^{n} A_i$, $A'_1 = A_1$, $A'_i = 0$ $(i = 2, \ldots, n)$ and $B'_i = B_i$ $(i = 3, \ldots, n)$.*

Proof. Suppose $(\bar{A}, \bar{B}, p)$ is a frequency distribution of a superpattern $\bar{t}$ of t with minimum conditional entropy whose form is in neither cases. Denote $P_i = \frac{\bar{A}_i}{\bar{A}_i+\bar{B}_i}$ and $Q_i = \frac{\bar{B}_i}{\bar{A}_i+\bar{B}_i}$ $(i = 1, \ldots, n)$. By generalizing Lemma 2, either $P_i < P_{i+1}$ or $P_{i+1} = 0$ $(i = 1, \ldots, n-1)$. Symmetrically, either $Q_i < Q_{i+1}$ or $Q_{i+1} = 0$ $(i = 1, \ldots, n-1)$. Then, for all $i = 2, \ldots, n$, either $P_i = 0$ or $Q_i = 0$. ($\because$ Assume $P_i \neq 0$ and $Q_i \neq 0$ for some i. Then, $P_{i-1} < P_i$ and $Q_{i-1} < Q_i$. But, $1 - P_{i-1} = Q_{i-1} < Q_i = 1 - P_i$ which is a contradiction.) Therefore, either $P_2 = 0$ or $Q_2 = 0$. Without loss of generality, say $P_2 = 0$. Then, we can get another distribution $(\bar{A}'$,$\bar{B}'$,$p)$ where $\bar{B}'_2 = \sum_{i=2}^{n} \bar{B}_i$, $\bar{B}'_i = 0$ for $(i = 3, \ldots, n)$, and the rest unchanged from $(\bar{A}$,$\bar{B}$,$p)$. Since its conditional entropy at each bin p_i ($i = 2, \ldots, n$) becomes 0, it has smaller or the same conditional entropy with ($\bar{A}$,$\bar{B}$,p). By the assumption that ($\bar{A}$,$\bar{B}$,p) has the minimum conditional entropy, their conditional entropy are the same. By Lemma 1, $\bar{A}'_1 \geq A_1 + A_2$ and $\bar{B}'_1 \geq B_1$ ($\because \bar{A}'_2 = 0$ since $P_2 = 0$). If either $\bar{A}'_1 > A_1 + A_2$ or $\bar{B}'_1 > B_1$, then the conditional entropy of ($\bar{A}'$,$\bar{B}'$,p) becomes higher than the conditional entropy of (A',B',p) in the first form of the theorem which is a contradiction to our assumption that the conditional entropy of ($\bar{A}$,$\bar{B}$,p) is minimum. Similar contradiction can be derived when $Q_2 = 0$. □

4.3.3 Modified Sequential Coverage Method

The binned information gain measure and its upper bound described in Section 4.3.2 enables a *branch-and-bound* framework, and we can simply perform the feature selection procedure in a traditional *sequential coverage* way as follows ([65, 67]). First, we mine the most discriminative *k-ee* subtree and add it to the feature set. Second, we remove trees that contain the extracted pattern and compute binned information gain scores of the remaining patterns on the updated database. In this way, redundant patterns will have a small chance to be selected. Third, we go back to the first step until either the dataset becomes empty or no more patterns are mined. Once the feature selection procedure is complete, we get a small number of discriminative *k-ee* subtrees. Based on the feature set F of these patterns, we use the document representation described in Section 4.2.3 to train a classification model.

But this procedure is inefficient when many discriminative patterns need to be mined because the *sequential coverage* method described above is based on iteratively mining one discriminative pattern for each iteration. We observe that the object of iterative approach is to find non-repetitive discriminative patterns. For this purpose, previous works simply applied the decision tree scheme of feature selection either (1) to a sequential coverage method to be used for SVM classification model [65, 67] or (2) to a decision tree classification model directly [73]. The difference between them is that the former recursively mines the dataset that does not contain the pattern, and the latter recursively mines both datasets containing and not containing the pattern. But both approaches need to recompute discriminativeness scores of the patterns on the updated database paying an expensive computational cost, which does not really involve removing repetitive patterns. We propose to use a modified sequential coverage method which does not recompute the binned information gain scores at step 2 of the traditional sequential coverage method described above.

4.3.4 Direct Discriminative k-ee Subtree Mining

In this section, we design a novel algorithm to efficiently mine discriminative patterns in a single iteration. We compute the binned information gain score only once, and apply the sequential coverage method without recomputing the binned information gain scores. Moreover, we propose an efficient way of mining the discriminative patterns in one iteration.

Here, we define some terms and symbols that will be used for the rest of the section. We denote $\mathbf{t} \models \mathbf{s}$ when a *k-ee* subtree t is contained in a tree s. We define $\mathbf{S_t} = \{s \in \mathcal{D} | t \models s\}$ to be a set of trees in a tree dataset $\mathcal{D}$ that contain t. Also, we define $\mathbf{A_t} = \{p : \text{k-ee subtree} | \exists s \in S_t, p = argmax_{p \models s} IG(p)\}$ to be a set of patterns that achieve the highest discriminative score among all patterns in some trees that contain t, and $\mathbf{B_t}$ to be a set of arbitrary patterns from each tree of S_t. We denote $\mathbf{F}$ to be a set of discriminative *k-ee* subtrees in $\mathcal{D}$ mined by the modified sequential coverage method.

The following lemma characterizes discriminative patterns mined by sequential coverage.

Lemma 3. *For a given tree dataset $\mathcal{D}$,*

$$F = \{t | \exists s \in \mathcal{D} \text{ such that } t = argmax_{p \models s} IG(p)\}.$$

Proof. By the definition of the modified sequential coverage method mentioned in Section 4.3.3. □

Lemma 3 explains that the discriminative patterns mined by the modified sequential coverage method are indeed the most discriminative patterns for some trees of $\mathcal{D}$. Based on this observation, we derive a pruning method by *branch-and-bound* approach in the following proposition.

Proposition 1. (Branch-and-Bound (BB) Pruning)
If $IG_{ub}(t) < \min_{p \in A_t} IG(p)$, then no superpattern t' of t is in F.

Proof. Since $S_t \supseteq S_{t'}$, $IG_{ub}(t) < \min_{p \in A_t} IG(p) \leq \min_{p \in A_{t'}} IG(p)$. That is, t' cannot be the most discriminative pattern for any tree in $S_{t'}$. □

Corollary 1. *If $IG_{ub}(t) < \min_{p \in B_t} IG(p)$, then no superpattern t' of t is in F.*

Proof. By definition of A_t, $IG_{ub}(t) < \min_{p \in B_t} IG(p) \leq \min_{p \in A_t} IG(p)$. □

In case $IG_{ub}(t) = \min_{p \in B_t} IG(p)$, we also skip mining $\mathcal{D}_t$ since any tree containing a superpattern t' of t will also contain another pattern that has higher or the same discriminative score.

Once we know an upper bound of the discriminative score of t's superpatterns, we can use the *BB* pruning method described in Proposition 1. Unfortunately, as alluded to earlier, this is a nontrivial task because the feature values are numeric instead of binary. In Section 4.3.2, we partitioned the numeric range $[0, 1]$ into a finite number of bins and derived the upper bound of binned information gain score by checking a constant number of cases (at most 2 cases) regardless to the number of bins.

In the mining process, since we do not know A_t, we set B_t to be the set of current best patterns of S_t and apply Corollary 1 as a *BB* pruning condition. For that reason, we maintain current best patterns for each tree.

Example 2. *Consider the example from Figure 4.3. Suppose class c_1 has a document d_1 and class c_2 has a document d_2 from a database $\mathcal{D}$. Let the number of bins for binned information gain be 3 (i.e. $n = 3$). We first mine t_1, compute its discriminative score ($IG(t_1) = 0$) and update current B_{t_1} ($B_{t_1} = \emptyset$) by checking t_1. Now, $B_{t_1} = \{t_1\}$. Since $IG_{ub}(t_1) = 1 > \min_{p \in B_{t_1}} IG(p) = 0$, we move on to next pattern t_2 without pruning. We compute t_2's discriminative score ($IG(t_2) = 1$), and update $B_{t_2} = \{t_1\}$ to be $B_{t_2} = \{t_2\}$. Since $IG_{ub}(t_2) = 1 = \min_{p \in B_t} IG(p)$, we can skip generating t_3.*

Following the original sequential coverage methodology mentioned in Section 4.3.3, when a *k-ee* subtree t is generated the trees containing t are removed. But in real classification tasks, we may want to generate multiple patterns to represent a tree to improve accuracy. To address this issue, we use a minimum feature coverage threshold δ introduced in [65], *i.e.*, a tree is removed when it is covered by at least δ discriminative patterns. Lemma

3 and Proposition 1 can easily be adapted with the feature coverage parameter δ by maintaining top-δ patterns for each tree and using δ-th highest discriminative score as a cut-off threshold for each tree.

In summary, we proposed a *branch-and-bound* framework of authorship classification. During the process, the algorithm retains and updates the most discriminative patterns $Opt(s)$ of each tree input, and at the end they become F. The basic framework is to expand the patterns from small to large sizes in pattern-growth approach. Before we expand current pattern t into a larger one, we compute the upper bound of the binned information gain of all superpatterns of t. Based on BB pruning described in Corollary 1, if the upper bound value is not greater than the current minimum $Opt(s)$ from all trees (s) containing t, then we can safely skip exploring superpatterns of t.

4.4 Experiments

In this section, we present an empirical evaluation in order to validate the performance of our *k-ee* subtree based authorship classification. We also analyze the effect of the parameters of *k-ee* subtree patterns presented in this chapter. The experiments are designed to test the usefulness of *k-ee* subtrees, as a new feature set, for authorship classification.

We first show accuracy comparison on various feature sets and then analyze the effect of the parameters of k-ee subtree approach. For the accuracy comparison with other feature sets, we conducted binary authorship classification as well as multiple authorship classification tasks. By default, we used the number of embedded edge $k = 1$, minimum support threshold $\theta = 0$, the number of bins $n = 10$, and minimum feature threshold $\delta = 3$ for discriminative k-ee subtree mining. In Tables 4.3 and 4.4, boldface denotes the best result for each dataset.

4.4.1 Datasets

For the following experiments, we used public data collections extracted from the *TREC* corpus [74] and *The New York Times*[2].

[2] http://www.nytimes.com

Table 4.1: Characteristics of data collections

Data	# Authors	Doc	Doc/Author	Sentence	Word
NTNews	4	400	100	19,161	381,450
Movie	4	2,177	415 – 598	51,086	1,299,682
TREC	7	6,336	804 – 1,003	169,767	3,964,865

From *The New York Times* we collected two different types of datasets: news articles and movie reviews. For the news articles, we randomly selected two journalists from the business department, and two other journalists from the health department who were the main contributors in their departments.[3] We collected datasets assuming that the journalists in the same department are likely to write articles on the same topic and genre using similar words.

For the movie reviews, we used four movie critics from the *The New York Times*. It has three main critics whom we used. We added another randomly selected critic who is one of the major contributors.[4] We collected this data because most of the movies reviewed by the critics overlapped. We assumed movie reviews of the same movie will be on the same topic and genre using similar words.

We also used news articles from the Associated Press (AP) subcollection of the public TREC corpus. The AP collection has over 200,000 documents by more than 2,380 distinct authors. We followed the same experimental configurations as previous works [56, 61] did by using the same datasets from the same seven authors[5] they used. The statistics of each data collection are described in Table 4.1. Note that the class distributions (or the number of documents per author) are mostly balanced, and in this way we do not have to consider the effect of skewed data.

4.4.2 Evaluation Methodology

To evaluate the performance, we performed multiclass classification on each data collection using *SVM* with linear kernel. Specifically, we decomposed

[3] Eric Dash and Jack Healy from the business department, and Denise Grady and Gina Kolata from the health department.

[4] The three main critics of *The New York Times* are A. O. Scott, Manohla Dargis, and Stephen Holden. The other critic we used is Jeannette Catsoulis.

[5] The authors are Barry Schweid, Chet Currier, Dave Skidmore, David Dishneau, Don Kendall, Martin Crutsinger, and Rita Beamish.

Table 4.2: Number of features for *FW*, *POS*, *RR* and *k-ee* feature sets

Data	FW	POS	BPOS	RR	0-ee	1-ee	2-ee
NTNews	308	74	1088	3929	119.2	257.8	453.1
Movie	308	74	1088	9029.2	306.2	575.1	1015.6
TREC	308	74	1088	8278	254.4	570.5	1107

the multiclass problem into binary problems via one-versus-one method, and paired the authors of each data collection and conducted binary classification on these pairwise datasets. For each dataset, we conducted 5-fold cross validation, and averaged the accuracy as a measure of the performance. For each fold, training data was used to mine the syntactic features and to get a classification model while test data was only used for evaluation purposes. For each training data, we used another 5-fold cross validation to determine appropriate parameter values for the classification model (linear SVM). In this way, our evaluation ensured that there is no information leak from the test data for the classification task.

We used the number of occurrences of each feature as a feature value for the syntactic features except *k-ee* subtrees which used a new frequency measure defined in Definition 2. For the fair comparison, we used the same classifier. In [55, 61], it is shown that *SVM* achieves reliable performance with high accuracy for authorship classification and the choice of the *SVM* kernel has little or no effect on the performance.

4.4.3 Comparison Feature Sets

To show how effectively our new feature set of *k-ee* subtrees works, we compared the authorship classification performance with other syntactic features such as function words (FW), unigram *POS* tags (POS), bigram *POS* tags (BPOS), and rewrite rules (RR). As for function words, we took the list of 308 function words from [75]. We used 74 *POS* tags from from the stanford parser. 1,088 Bigram *POS* tags were identified from the leaves of syntactic trees. Rewrite rules and *k*-ee subtrees were generated by mining parsed sentences of syntactic *POS*-tagged trees.

In the table 4.2, we show the average sizes of feature sets for each data collection. To get the number of features of rewrite rules and *k*-ee subtrees, we computed the average value of the number of distinct features of 5-fold

Table 4.3: Accuracy Comparison on Different Number of Authors and Various Data Collections

Data	# Authors	FW	POS	BPOS	RR	*k-ee*
NTNews	2	92.25	86.67	90.42	89.75	**94.25**
	3	87.08	78.17	83.97	82.17	**90.83**
	4	82.75	71.25	79.45	75.25	**87.75**
Movie	2	93.18	88.99	84.17	92.88	**95.62**
	3	88.03	81.77	82.17	88.45	**92.89**
	4	84.00	76.23	80.25	85.11	**91.30**
TREC	2	93.33	92.43	93.95	95.07	**96.04**
	3	88.63	87.12	89.64	91.49	**93.43**
	4	85.10	83.03	86.30	88.67	**91.50**
	5	82.24	79.71	83.51	86.31	**89.95**
	6	79.80	76.87	81.10	84.26	**88.56**
	7	77.62	74.53	78.92	82.46	**87.37**
Average		86.14	81.40	84.45	86.87	**91.62**

training data for each feature set and dataset. As expected, rewrite rules generated much larger number of features than all the other feature sets. It is noticeable that the number of *k-ee* subtrees are far less than the number of bigram *POS* tags and rewrite rules, and sometimes even less than the number of function words. For the rest of the section, we will show that our small sized new feature set of *k-ee* subtrees outperforms all the other feature sets.

4.4.4 Overall Effectiveness

Based on the accuracy results in Table 4.3, our new feature set of *k-ee* subtrees achieved the highest performance of the comparison feature sets. Overall, most feature sets showed high accuracy on binary authorship classification tasks. But when the number of authors was increased, the performance gaps between *k-ee* subtree feature set and all the others became larger.

It is true that bigram *POS* tags and rewrite rules catch deeper insights of an author's writing style since they are more complex and have much larger number of features than *POS* tags. But we conclude that a feature set of *k-ee* subtrees can characterize an author's writing style even better since (1) it allows even more complex syntactic structures than rewrite rules as features, (2) its size is much smaller than the feature set of bigram *POS*

Table 4.4: Accuracy Comparison on binary authorship classification of *The New York Times* news articles. Two journalists Dash and Healy from the business department are denoted by B_1 and B_2, and two journalists Grady and Kolata from the health department are denoted by H_1 and H_2

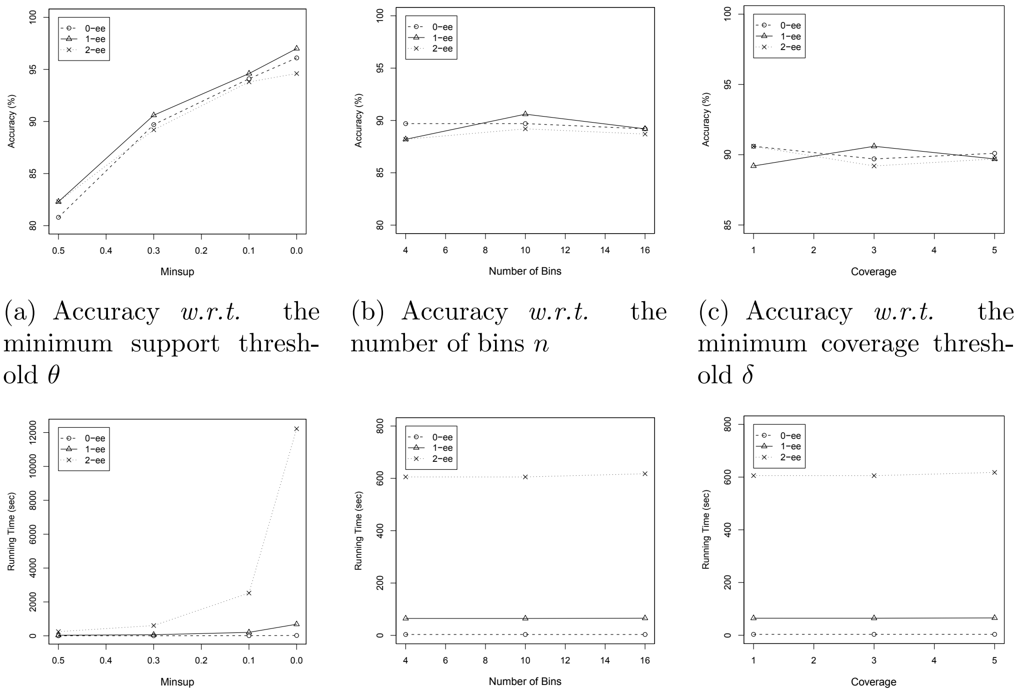
Author Pair	FW	POS	BPOS	RR	k-ee
(B_1,B_2)	91.5	87	**95**	94	94
(B_1,H_1)	94	85	92	91	**95**
(B_1,H_2)	95.5	92.5	95	**96**	94
(B_2,H_1)	95	92.5	94.5	92.5	**97.5**
(B_2,H_2)	97	95.5	96.5	97.5	**98**
(H_1,H_2)	80.5	67.5	69.5	67.5	**87**
Average	92.25	86.67	90.42	89.75	**94.25**

tags and rewrite rules, and (3) it achieved better accuracies. Note that the feature set of function words reliably showed reasonable accuracies as previous works mentioned [56, 61, 62]. It achieved better than *POS* tags and sometimes even better than bigram *POS* tags and rewrite rules. This is because function words have two different aspects together (syntactic and lexical) while *POS* tags only have a syntactic aspect. But complex syntactic structures can complement the lack of lexical aspect of the features, since the feature sets of rewrite rules and *k-ee* subtrees showed higher accuracies than function words.

On average, the feature set of *k-ee* subtrees improved performance over the other feature sets about 8.23% (overall), 6.36% (function word), 12.56% (*POS*), 8.49% (bigram *POS*) and 5.50% (rewrite rule).

We also performed a significance test on the feature sets over *k-ee* subtrees. We used two-tailed t-test on the accuracy results in Table 4.3, and all their t values (FW:3.18, POS:5.02, BPOS: 4.49, RR: 2.69) indicated that the performance of *k-ee* subtree patterns are significantly different from (or, better than) all the others (95% confidence interval, threshold:2.07).

Note that we could mine *k-ee* subtrees even for minimum support $\theta = 0$, a task rarely done in previous works because too many patterns were generated from the mining process.



(a) Accuracy *w.r.t.* the minimum support threshold θ

(b) Accuracy *w.r.t.* the number of bins n

(c) Accuracy *w.r.t.* the minimum coverage threshold δ

(d) Running time *w.r.t.* the minimum support threshold θ

(e) Running time *w.r.t.* the number of bins n

(f) Running time *w.r.t.* the minimum coverage threshold δ

Figure 4.5: Performance Comparisons on Different Parameter Settings

4.4.5 Problem Difficulty Analysis

As we explained in Section 4.4.1, the datasets of *The New York Times* news articles were collected to identify the difficulty of classification problem. We assumed that the journalists from the same departments will be hard to classify because they might use similar terms on the same topic and genre. As expected, classification results in Table 4.4 show that classifying journalists from different departments was easier than journalists from same departments.

Note that the last row of Table 4.4 shows extremely worse performance than other cases. We manually analyzed the news articles of H_1 and H_2, and found that their writing styles were quite informal using several quotations which made it the hardest dataset. Even for this hard task, our approach got the highest accuracy with a big gap.

4.4.6 Parameter Analysis

In Figure 4.5, we analyze the role of each parameter used to mine discriminative *k-ee* subtrees. All experiments were conducted for binary classification of two movie critics Stephen Holden and Jeannette Catsoulis. Similar trends could be found from other datasets. For default values, we used $\theta = 0.3$, $n = 10$, and $\delta = 3$. Overall, we found that 1-*ee* subtree feature set showed the best performance. It could be mined with almost in a constant time even with no minimum support threshold. But, when the number of embedded edges increased (*e.g.* $k = 2$), *k-ee* feature set showed worse accuracies because it tended to overfit to the training data. Moreover, it took exponential time to run when minimum support threshold gets smaller. It is good to know that we do not need too complicated syntactic structures (with a high k), because the computation would be too expensive to make our proposed feature set useful.

There are two parameters, n and δ, which are related to our binned information gain score. Based on Figure 4.5, they did not significantly affect the running time, but somehow affected the accuracy. However, since they achieved the peak within a small range, it was not difficult to optimize their values in our experiments.

CHAPTER 5

CLUSTERING REDUNDANT NODES

5.1 Overview

Document representation is a fundamental problem for user comprehension and understanding [76, 77, 78], and is also critical to various text processing tasks like text categorization [79] and retrieval [80]. Because of its simplicity and effectiveness, the bag-of-words representation is widely adopted in most of document processing tasks, especially in text categorization [81]. However, there are several areas that other representations outperform the bag-of-words where it is needed to capture complex semantics of text, including phrasal, syntactic and more sophisticated linguistic structures [82, 83, 84].

Analyzing monolingual comparable corpora is one of the areas where the bag-of-words representation has limitations. Monolingual comparable corpora is defined as a collection of documents in the same language (*e.g.*, English) that overlap in the information they convey. In the age of information overload, we can easily collect or access such corpora that cover the same topic such as multiple news reports on the same or similar events from different news agencies, and reviews about the same or similar products or services.

Beyond several studies on monolingual comparable corpora, which study sentence alignments [85] and paraphrasing rules [86], analyzing monolingual comparable corpora has many potential applications. First, the analysis can give a comprehensive summary about one event, fact, or entity because documents in a comparable corpus cover different perspectives of the topic. Second, the analysis can derive a set of consistent information across documents, which helps remove some trivial or misleading information. This application is close related to trustworthiness analysis, where many studies on structured data like movie databases [4] and sensor data [87] have been

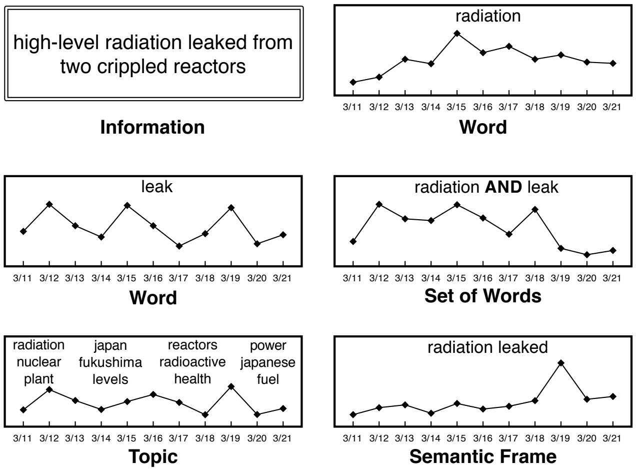


Figure 5.1: 5 different representations for information and their trend plots

done, but not in unstructured data like documents. Third, analyzing monolingual comparable corpora can track the trend of information when each document has timestamp.

As the first step toward the analysis of monolingual comparable corpora, we propose the use of *frame*, a high-level semantic feature derived by semantic role labeling (SRL) [88], as the basic unit for document represent in comparable corpora. In Figure 5.1, we demonstrate the power of semantic frame. Specifically, a collection of news articles about Japan's 2011 Tsunami (which caused radiation leaked from two crippled nuclear reactors in March 19th) is used as a comparable corpora. We use 5 different kinds of representations for this particular information, and measure the popularity using the occurrences of the representations within the corpus, and draw the trends in Figure 5.1. As shown in the figure, the semantic frame is the only one that isolates the information and detect the peak in March 19th. We will further discuss on this aspect in Section 5.4.

Semantic frame has proved its superiority in various applications including information extraction [89] and question answering [90]. Each frame is a verb-argument structure from a sentence, and is arranged as a subject-verb-object *triplet* where each part is associated with a set of words. By extracting triplets we can find the most important semantic information from a set of documents, and can serve as a better representation for other tasks like event tracking.

However, a higher level document representation usually results in a higher

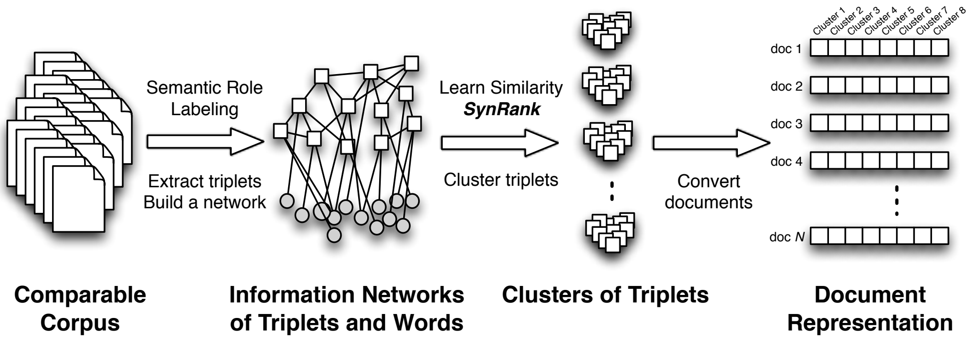


Figure 5.2: Overview of Our Proposed Framework

complexity feature space, which leads to sparser document model due to the variational forms. For example, "radiation leaked" in one news article can appear as "the level of radiation increased" in another article. In this chapter, we try to resolve the sparsity challenge when dealing with frame-based document representation, by grouping semantically similar frames together.

An information network-based approach is developed to define similarity between frames, by which similar frames can be better grouped together due to the propagation of similarity along different types of network links. We first construct a syntactic structure between each frame-derived triplet and its words. Then, a bi-typed *information network* is built for a corpus by extracting all the nodes and links from different documents, where nodes represent words and triplets, and links exist between them if they are connected in their original syntactic structure. We further propose a link-based similarity measure, called *SynRank*, to calculate the similarity between triplets in an iterative way, where we design different iterative formulas for different types of objects by considering their semantic meanings. Then we can cluster similar frames together according to the obtained similarity. One representative triplet will be selected for one cluster, and documents are represented by the corresponding frames (see Figure 5.2).

Finally, we validate the effectiveness of our similarity measure comparing with other baselines on several real-world datasets. The results show that the frame-based document representation is more interpretable and comprehensive than baseline methods.

We summarize our contributions of this work as follows:

1. We propose a novel frame-based document representation method which

can capture the document semantics and represent comparable corpora in a comprehensive and concise way.

2. We propose to construct an information network from the corpus, and develop a link-based similarity measure called SynRank to capture the similarity between frames and similarity between words jointly, in an iterative and global way.

3. Experiments on real-world datasets show the power of the new document representation method, compared with several baseline approaches.

5.2 Problem Statement

In this section, we introduce preliminary knowledge about semantic frame and provide an overview of our proposed frame-based document representation method.

5.2.1 Raw Semantic Frame Extraction

Different from bag-of-words representation, which misses the semantic relationships among words, semantic frames aim at capturing the most important elements such as entities and their relationships from a sentence, defined as follows.

Definition 4. A **semantic frame** $f \in F$ is a verb-argument structure in a sentence that describes a type of event, relation, or entity and the participants in it [91].

This definition is based on the semantic role formalism of PropBank [92]. As seen from Figure 5.3, extracted frames contain richer information than word and less information (usually single fact, statement, or proposition) than sentences. Notice that, there could be several frames derived from one sentence, and the number of semantic frames in a sentence equals to the number of verbs in the sentence. In this work, we use SRL tool SENNA parser [93] for raw frame extraction, which is reported to have about 74% F_1 measure on CoNLL 2005 benchmark dataset.

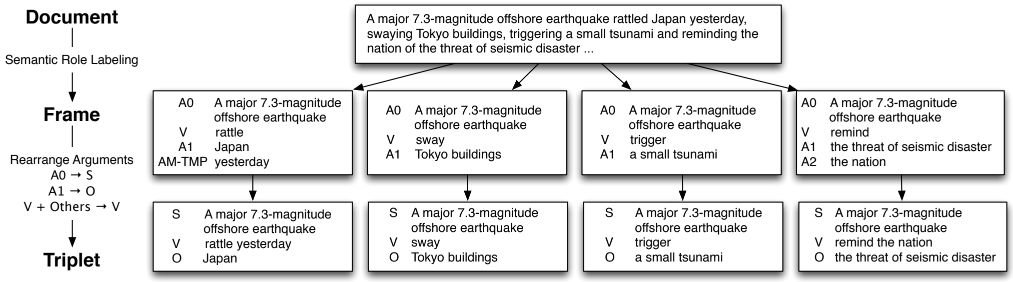

Document
Semantic Role Labeling
Frame
Rearrange Arguments
A0 → S
A1 → O
V + Others → V
Triplet
A major 7.3-magnitude offshore earthquake rattled Japan yesterday, swaying Tokyo buildings, triggering a small tsunami and reminding the nation of the threat of seismic disaster ...
A0 A major 7.3-magnitude offshore earthquake
V rattle
A1 Japan
AM-TMP yesterday
A0 A major 7.3-magnitude offshore earthquake
V sway
A1 Tokyo buildings
A0 A major 7.3-magnitude offshore earthquake
V trigger
A1 a small tsunami
A0 A major 7.3-magnitude offshore earthquake
V remind
A1 the threat of seismic disaster
A2 the nation
S A major 7.3-magnitude offshore earthquake
V rattle yesterday
O Japan
S A major 7.3-magnitude offshore earthquake
V sway
O Tokyo buildings
S A major 7.3-magnitude offshore earthquake
V trigger
O a small tsunami
S A major 7.3-magnitude offshore earthquake
V remind the nation
O the threat of seismic disaster

Figure 5.3: An Illustrative Example for Process of Extracting Triplets from Document.

We further formulate each semantic frame into a **triplet** of subjective, verb and objective (see Figure 5.3), to preserve semantic roles and content in an effective and concise way.

Definition 5. We denote the **semantic triplet** as $t = (s, v, o)$, where s is *subjective word set* consisting of words with A_0 SRL tags in frame f, o is *objective word set* consisting of words with A_1 SRL tags in frame f, and v is *verb word set* containing verb and all the other arguments such as A_2, AM-TMP and AM-LOC, where A_0 represents the subjective, A_1 represents the objective, A_2 represents indirect objective, AM-TMP represents temporal modifier, and AM-LOC represents the location modifier.

By re-structuring frames into triplets, we have a much clearer structure of each frame. However, these raw triplets cannot be directly used as features to represent documents because there still exists many semantically similar variations (e.g., “earthquake hit Japan” and “quake struck Japan”), leading to a high-complexity feature space and thus sparse document representation. To resolve this, we first construct a semantic text information network among words and triplets, and then propose a link-based similarity function to measure their similarity. Similar triplets are grouped into clusters based on the similarity and the frame corresponding to representative triplet in each cluster will be selected as the final representation feature for documents.

The overall framework of the process can be summarized into the following three steps (see also Figure 5.2).

1. **Raw semantic frame extraction.** In this step, raw semantic frames and corresponding semantic triplets are first extracted from sentences

in documents through semantic role labeling tool (see Figure 5.3).

2. **Semantic text information network construction.** We construct a semantic information network for words and triplets extracted from corpus (see Figure 5.4 and 5.5), which provides a novel view that different text objects are connected by semantic links.

3. **Link-based triplet clustering for document representation.** Finally, we propose a link-based similarity measure, and cluster triplets into different groups based on it. We select the most representative one in each cluster for final representation of the documents.

The first step is easily done by semantic role labeling tool, we now introduce Step 2 and 3 in following sections.

5.2.2 Semantic Text Information Network Construction

In order to merge similar triplets, we need a way to measure similarities between them, which is a problem related to the paraphrase detection task. One of the paraphrase detection methods is leveraging synonyms from a knowledge base such as WordNet [94] to improve the detection performance [95]. However, this kind of approaches are limited for the synonyms in the general usages. For example, the word "threat" is frequently used to refer the word "radiation" in the Japan's tsunami corpus[1], but their similarity in Wordnet is 1.743[2], which is lower than the similarity score of 1.897 between "buildings" and "cars." Thus, it is important to derive a *corpus-based similarity measure* for words in order to measure the similarities of triplet.

To meet this need, we propose to cluster similar triplets using an information network approach, where various text objects and their connections are captured by a *semantic text information network*. As we will show in Section 5.4, it is much more effective to compute the word similarity and frame similarity jointly and globally in a unified framework instead of computing them separately by utilizing links in this text information network.

Definition 6. A **semantic text information network** is a bi-typed undirected graph containing two types of object sets T (triplets) and W (words).

[1] There was a nuclear accident and radiation leaks following the tsunami.

[2] This similarity is computed using Leacock & Chodorow [96].

For each triplet $t \in T$, it has links to a set of words in W, as well as links to its neighbor triplets as its *context.* The link types are defined by their relations: links from triplet to its contextual triplets belong to *triplet-triplet (TT)* relation; links between triplets and its words belong to *triplet-word* TW relation.

The network schema of the information network is shown in Figure 5.4. Notice that words are distinguished by different semantic roles (S,V,O) such as "S: earthquake" and "O: tsunami" in Figure 5.4.

For a triplet node t in the network, the neighbors of t are denoted by $N_R(t)$, where $R \in \{TT, TW\}$ represents the link type. We denote the context of triplet t as $N_{TT_\sigma}(t)$, where σ is the size of the context window, i.e., the number of nearby triplets that are considered as its context in a document. For simplicity, we denote it by $N_{TT}(t)$.

Based on the semantic text information network, we derive a semantic similarity measure for triplets by analyzing *triplet-triplet* and *triplet-word* links. The intuition behind this measure is that similar text objects share *similar context* around them and *similar content* within them. The details will be introduced in Section 5.3.

5.3 SynRank: A Link-Based Semantic Similarity Measure

In this section, we explain in details how link information in the semantic text information network can be leveraged to cluster triplets, where different types of relations, i.e., *triplet-triplet* relation and *triplet-word* relation, are considered simultaneously. We first introduce a novel similarity measure, called SynRank, then show how to compute SynRank, and finally the clustering algorithm for triplets based on this similarity measure.

Similar to SimRank [32], which measures the similarity between objects in a network based on the assumption that "two objects are similar if they share similar neighbors," we propose our link-based similarity measure, following the intuition that "similar triplets share *similar context* around them and *similar content* within them." In particular, a triplet is most similar to itself, with maximum score 1.

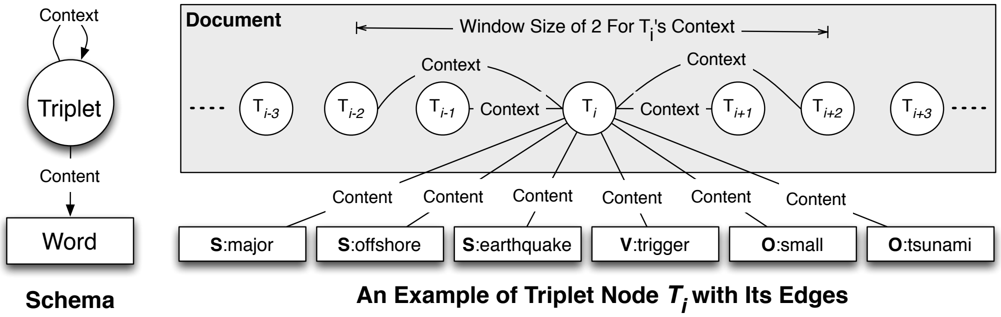


Figure 5.4: Meta-Schema of Semantic Text Information Network

SynRank deals with different types of relations (i.e., triplet-triplet context relation and triplet-word content relation) simultaneously with different updating mechanisms, which distinguishes itself from other link-based measures such as SimRank [32] and P-Rank [8]. Iteratively computing SynRank function can propagate similarities between object pairs in a global manner, *i.e.*, word similarity and triplet similarity are mutually adjusted according to the whole corpus (see Figure 5.9).

We formulate above intuition into a link-based similarity measure function, called **SynRank**, which takes the recursive form as follows. For two triplets nodes t_i and t_j, at the k-th iteration of SynRank, if $t_i = t_j$, then $s_T^{(k)}(t_i, t_j)$ is set to be 1; otherwise,

$$s_T^{(k)}(t_i, t_j) = C \cdot \Big[(1-\lambda) \cdot s_{TW}^{(k)}(t_i, t_j) + \lambda \cdot s_{TT}^{(k)}(t_i, t_j)\Big], \tag{5.1}$$

where $s_{TW}^{(k)}(t_i, t_j)$ and $s_{TT}^{(k)}(t_i, t_j)$ denote content similarity based on *triplet-word* (TW) relation and contextual similarity based on *triplet-triplet* (TT) relation at k-th iteration, respectively. λ is a trade-off parameter, and constant $C \in [0, 1]$ is a damping factor similar as the one in SimRank [32].

Note that for a semantic text information network with $|T|$ triplets, a set of $|T|^2$ SynRank equations needs to be computed. We use $\mathbf{S}_T \in \mathbb{R}^{|T| \times |T|}$ to denote the triplet similarity matrix, where $\mathbf{S}_T(i, j) = s_T(t_i, t_j)$.

Other essential updating formula, including content-based triplet similarity $s_{TW}(t_i, t_j)$, context-based triplet similarity $s_{TT}(t_i, t_j)$, and word similarity S_W, are further introduced as follows.

5.3.1 Content-based Triplet Similarity

Given a pair of triplets t_i and t_j, their content-based similarity, $s_{TW}(t_i, t_j)$, is defined according to the similarity between their content neighbors $N_{TW}(t_i)$ and $N_{TW}(t_j)$.

Example 1 (Similar triplets with similar content)**.**

$$t_1 = (\textit{S:}\{\textit{An earthquake}\}, \textit{V:}\{\textit{unleashed}\}, \textit{O:}\{\textit{7.3m waves}\});$$
$$t_2 = (\textit{S:}\{\textit{A 8.9 quake}\}, \textit{V:}\{\textit{unleashed}\}, \textit{O:}\{\textit{a tsunami wave}\})$$

Just like above example, triplets are thought to be similar if they have same/synonymous terms in subjectives, verbs, and objectives, respectively.

Assumption 1. In semantic text information network, two triplet nodes t_i and t_j are said to be *content-based similar* if many of their linked words $a \in N_{TW}(t_i)$ and $b \in N_{TW}(t_j)$ are similar.

Following the assumption, a recursive equation for updating $s_{TW}(t_i, t_j)$ can be derived. If $t_i = t_j$, then $s^{(k)}_{TW}(t_i, t_j) = 1$; otherwise,

$$s^{(k)}_{TW}(t_i, t_j) = \sum_{a \in N_{TW}(t_i)} \sum_{b \in N_{TW}(t_j)} \frac{f_{t_i,a} \cdot f_{t_j,b} \cdot s^{(k-1)}_{W}(a, b)}{F_{TW}(t_i) F_{TW}(t_j)}, \tag{5.2}$$

where $f_{t_i,a}$ denotes the occurrence frequency of word a in triplet t_i, and $F_{TW}(t_i)$ denotes total word occurrence in t_i, *i.e.*, $F_{TW}(t_i) = \sum_{a \in N_{TW}(t_i)} f_{t_i,a}$. Here, $s_W(\cdot, \cdot)$ is the similarity between words, which will be introduced in Section 5.3.3. We rewrite Equation (5.2) into matrix form

$$\mathbf{S}^{(k)}_{TW} = \mathbf{D} \cdot \mathbf{S}^{(k-1)}_{W} \cdot \mathbf{D}^T, \tag{5.3}$$

where we define matrices $\mathbf{D} \in \mathbb{R}^{|T| \times |W|}$, $\mathbf{S}_{TW} \in \mathbb{R}^{|T| \times |T|}$, and $\mathbf{S}_W \in \mathbb{R}^{|W| \times |W|}$ as $\mathbf{D}(i, j) = f_{t_i,w_j} / F_{TW}(t_i)$, $\mathbf{S}_{TW}(i, j) = s_{TW}(t_i, t_j)$, and $\mathbf{S}_W(i, j) = s_W(w_i, w_j)$, respectively. $|W|$ denotes number of unique words in the corpus. The computational complexity of Equation (5.3) is $\mathcal{O}(|T|^2 L^2)$, where L is the maximum number of words in a triplet.

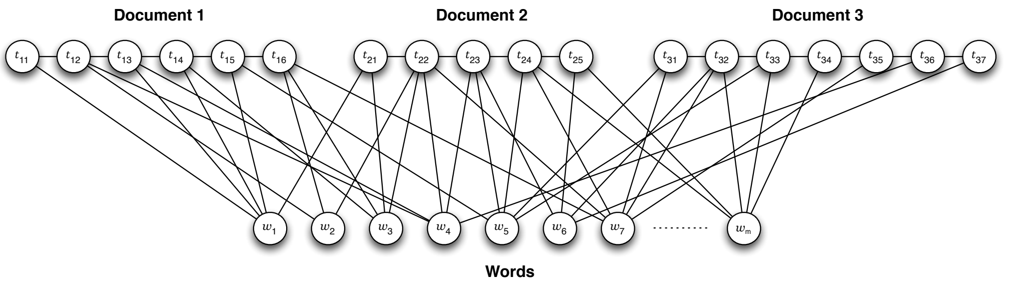


Figure 5.5: An Example of Semantic Text Information Network on Three Documents and with Context Window Size 1 ($\sigma = 1$).

5.3.2 Context-based Triplet Similarity

It is not sufficient to fully measure semantic similarity between two triplets by only their contents. In some cases, there could be only a few words inside the two triplets that are same/synonymous. We then propose to evaluate contextual similarity between two triplets, $s_{TT}(t_i, t_j)$, based on their contextual neighbors $N_{TT}(t_i)$ and $N_{TT}(t_j)$.

Example 2 (Similar triplets with similar context)**.**

$$\begin{aligned} t_1 &= (S{:}\{\textit{The first wave}\}, V{:}\{\textit{hit}\}, O{:}\{\textit{coasts in Japan}\}); \\ t_2 &= (S{:}\{\textit{A wave over 5 feet}\}, V{:}\{\textit{struck}\}, O{:}\{\textit{there}\}) \end{aligned}$$

Many articles in our Japan Tsunami news dataset reported not only the tsunamis in Japan, but also the Hawaii's tsunamis. Thus, by merely looking at t_2, we have no idea about where the wave struck. Intuitively, we can seek context of t_1 and t_2 as complementary reference. More specifically, context of a triplet is defined by neighbor triplets within a size σ window in its document (see Figure 5.5). For example, if the contexts of t_1 and t_2 are both about "Japan coasts," t_1 and t_2 become similar to each other.

Assumption 2. In semantic text information network, two triplet nodes t_i and t_j are said to be *context-based similar* if their linked triplets in the context windows $a \in N_{TT}(t_i)$ and $b \in N_{TT}(t_j)$ are similar.

Remind that for content-based measure of Equation (5.2), each word in triplet t_i will be compared with each word in t_j. However, in context-based measure, it may be meaningless to compare t_i's neighbor that talks about current fact with t_j's neighbor which can be a quotation. Our method in

Equation (5.4) is to compare each of t_i's neighbor a only with the neighbor of t_j that is *most similar* to a. With above intuition, we derive a recursive equation for $s_{TT}(t_i, t_j)$ by iterating over neighbors of t_i and t_j. At k-th iteration,

$$\begin{aligned} s_{TT}^{(k)}(t_i, t_j) &= \eta(t_i, t_j) \cdot \left(\sum_{a \in N_{TT}(t_i)} \max_{b \in N_{TT}(t_j)} s_T^{(k-1)}(a, b) \right. \\ &\quad \left. + \sum_{a \in N_{TT}(t_j)} \max_{b \in N_{TT}(t_i)} s_T^{(k-1)}(a, b) \right), \qquad (5.4) \end{aligned}$$

where $\eta(t_i, t_j) = \frac{1}{|N_{TT}(t_i)| + |N_{TT}(t_j)|}$ denotes the number of triplet pairs in summation, which will scale the final similarity score into [0, 1].

The computational complexity for calculating Equation (5.4) for all triplets is $\mathcal{O}(|T|^2\sigma^2)$. Note that by using some pruning strategy, we actually do no have to compute pairwise similarity for triplets. Due to space limit, we do not discuss the pruning issue in details here.

5.3.3 Corpus-based Word Similarity

Recall that in Equation (5.2), content-based triplet similarity $\mathbf{S}_{TW}$ is measured based on word similarity $\mathbf{S}_W$. In this section, we will address the problem of how to define a good word similarity $\mathbf{S}_W$.

The most straightforward way to calculate S_W is simply using the identity matrix, which only leverages the fact that a word is only similar to itself. A better strategy might be using some predefined thesaurus such as WordNet [94] to capture more sophisticated similarity structure between words. However, these methods are not able to capture the corpus-specific information. For example, "Japan" and "Tsunami" should be treated more similar in a news corpus about Japan Tsunami than in a corpus about the study of Tsunami's nature. Also, words in semantic text information network are distinguished by different semantic roles (S,V,O) denoting different semantic information, which cannot be well distinguished by knowledge-based approaches.

To address this problem, we propose to adaptively and iteratively update word similarity so that $\mathbf{S}_W$ and $\mathbf{S}_T$ can mutually enhance each other. Intuitively, a good word similarity should generate content-based triplet similarity $\mathbf{S}_{TW} = \mathbf{D}\mathbf{S}_W\mathbf{D}^T$ consistent with triplet similarity $\mathbf{S}_T$.

Assumption 3. In semantic text information network, corpus-specific information (*i.e.*, context of triplet) is well embedded into word similarity $\mathbf{S}_W$ if content-based triplet similarity $\mathbf{S}_{TW}$ is consistent with triplet similarity $\mathbf{S}_T$.

Suppose at the k-th iteration of SynRank, the triplet similarity $\mathbf{S}_T^{(k)}$ is derived by Equation (5.1), based on above assumption, we update $\mathbf{S}_W$ by approximately solving the optimization problem as follows:

$$\mathbf{S}_W^{(k)} = \text{argmin}_{\mathbf{S}_W}\mathcal{L}(\mathbf{S}_W) = \|\mathbf{S}_T^{(k)} - \mathbf{D}\mathbf{S}_W\mathbf{D}^T\|_F^2, \tag{5.5}$$

where $\|\mathbf{X}\|_F = (\sum_{i,j} X_{ij}^2)^{\frac{1}{2}}$ is matrix Frobenius norm for measuring how consistent the two matrices are. Objective function $\mathcal{L}(\mathbf{S}_W)$ in Equation (5.5) measures the difference between content-based triplet similarity $\mathbf{D}\mathbf{S}_W\mathbf{D}^T$ and current triplet similarity $\mathbf{S}_T^{(k)}$. By minimizing it, we have the optimal solution as follows:

$$\hat{\mathbf{S}}_W^{(k)} = (\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T\mathbf{S}_T^{(k)}\mathbf{D}(\mathbf{D}^T\mathbf{D})^{-1}, \tag{5.6}$$

whose computational complexity is $\mathcal{O}(|W||T|^2)$.

In order to enforce word similarity to fall in the range of $[0, 1]$, post-processing $\hat{S_W}^{(k)}$ is further performed by

$$s_W^{(k)}(t_i, t_j) = \begin{cases} 1, & \text{if } t_i = t_j. \\ \\ \max(\frac{\hat{\mathbf{S}}_W^{(k+1)}(t_i,t_j)}{\|\hat{\mathbf{S}}_W^{(k)}\|_F^2}, \quad 0), & \text{otherwise;} \end{cases} \tag{5.7}$$

If $\mathbf{D}^T\mathbf{D}$ is not invertible, $\mathbf{S}_W$ can be updated approximately based on gradient descent method

$$\mathbf{S}_W^{(k)} = \mathbf{S}_W^{(k-1)} - \alpha \cdot \Big\{(\mathbf{D}^T\mathbf{D})\mathbf{S}_W^{(k-1)}(\mathbf{D}^T\mathbf{D}) - \mathbf{D}^T\mathbf{S}_T^{(k)}\mathbf{D}\Big\} \tag{5.8}$$

where α is the step size. In our experimental setting, we have $|T| \gg |W|$, and thus $\mathbf{D}^T\mathbf{D}$ is in practice of full rank and invertible.

Algorithm 2 SynRank

```
Input: tuning parameters $C$ and $\lambda$, frequency matrix $\mathbf{D}$.
Initialize $\mathbf{S}_W^{(0)} = \mathbf{I}_{|W|}$ and $\mathbf{S}_T^{(0)} = \mathbf{I}_{|T|}$.
for $k = 1 \to maxIter$ do
    Compute content-based similarity matrix $\mathbf{S}_{TW}^{(k)}$ based on $\mathbf{S}_W^{(k-1)}$ by Equation (5.2);
    Compute context-based similarity matrix $\mathbf{S}_{TT}^{(k)}$ based on $\mathbf{S}_T^{(k-1)}$ by Equation (5.4);
    Calculate triplet similarity matrix $\mathbf{S}_T^{(k)}$ based on $\mathbf{S}_{TW}^{(k)}$ and $\mathbf{S}_{TT}^{(k)}$ using Equation (2);
    Update $\mathbf{S}_W^{(k)}$ based on $\mathbf{S}_T^{(k)}$ by Equation (5.6) and post-process it following Equation (5.7).
end for
Output: Converged matrices $\mathbf{S}_T^{(\infty)}$ and $\mathbf{S}_W^{(\infty)}$.
```

5.3.4 Algorithm for SynRank

Similar to SimRank, solution to the SynRank equations can be derived by iterations leading to a fixed-point. Starting with $\mathbf{S}_W^{(0)} = \mathbf{I}_{|W|}$ and $\mathbf{S}_T^{(0)} = \mathbf{I}_{|T|}$ as lower bounds of the actual SynRank scores, we successively and alternatively compute $\mathbf{S}_T^{(k)}$ based on $\mathbf{S}_{TW}^{(k-1)}$ and $\mathbf{S}_{TT}^{(k-1)}$ by Equation (5.1), and $\mathbf{S}_W^{(k)}$ based on $\mathbf{S}_T^{(k)}$ by Equation (5.6), respectively.

Algorithm 2 summarizes the iterative procedure for computing SynRank. Based on each of the similarity computation procedures in previous sections, computational cost for SynRank is $\mathcal{O}(K \cdot |T|^2|W|)$, where typically $|T| \gg |W|$, and K is the number of iterations needed for SynRank.

5.3.5 Triplet Clustering for Representative Frame

Once we compute triplet similarity $\mathbf{S}_T$ by SynRank, various off-the-shelf clustering algorithms (*e.g.* DBSCAN [97] and Affinity Propagation [98]) can then be applied to group these triplets together into clusters. From each cluster, we select one triplet which best summarizes the cluster and use its corresponding frame as the representative frame. Finally, each document is described by the corresponding representative frames derived from all triplet clusters.

More precisely, given triplet similarity matrix $\mathbf{S}_T(i, j)$, and suppose there is totally K frame clusters $\mathcal{C} = \{\mathcal{C}_1, ..., \mathcal{C}_K\}$ derived from the triplet clustering algorithm, we calculate the K representative frames $\{\hat{f}_1, ..., \hat{f}_K\}$ corresponding to the K clusters as follows:

$$\hat{f}_k = \underset{f_i \in C_k}{\operatorname{argmin}} \sum_{f_j \in C_k} \left(\mathbf{S}_T(i,j)\right)^2, \quad k = 1, ..., K. \tag{5.9}$$

Each document d is then summarized by a bag of representative frames $d = \{\hat{f}_1, ..., \hat{f}_{K_d}\}$, where K_d is the total number of clusters involved by frames of d.

We choose to use DBSCAN as our triplet clustering algorithm because it has the notion of noise objects, and does not require the number of clusters as an input. Like many other cluster algorithms, DBSCAN have tuning parameters for a given dataset. The two parameters $MinPts$ and Eps [97] are tuned in our experiments so that each news article has at most 100 different frames, and at most 3 same frames. The assumption is that each news article has at most 100 different statements or facts, and should not repeat to mention the same information more than 3 times because they are well-written articles. These constraints can be relaxed for different types of documents like blog posts.

5.4 Experiments

In this section, we first explain how we obtain three real-world monolingual comparable corpora.

As we addressed in Section 5.1, it is important to make the document representation space dense by clustering redundant features. We evaluate our information network-based similarity computation algorithm, SynRank, on labeled datasets. Since better similarity measures lead to better clustering, we demonstrate the effectiveness of SynRank by evaluating semantic similarities using the precision at K measure.

Then, we demonstrate the effectiveness of the frame-based document representation by the event tracking analysis of monolingual comparable corpora.

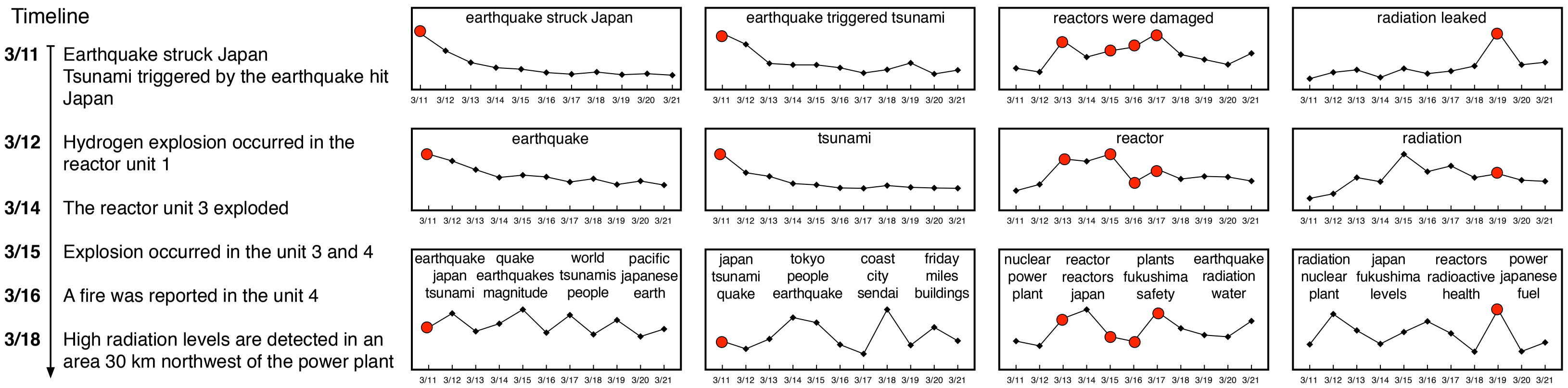


Figure 5.6: Event Tracking for Japan's Tsunami by Triplets (top), Words (middle), and Topics (bottom)

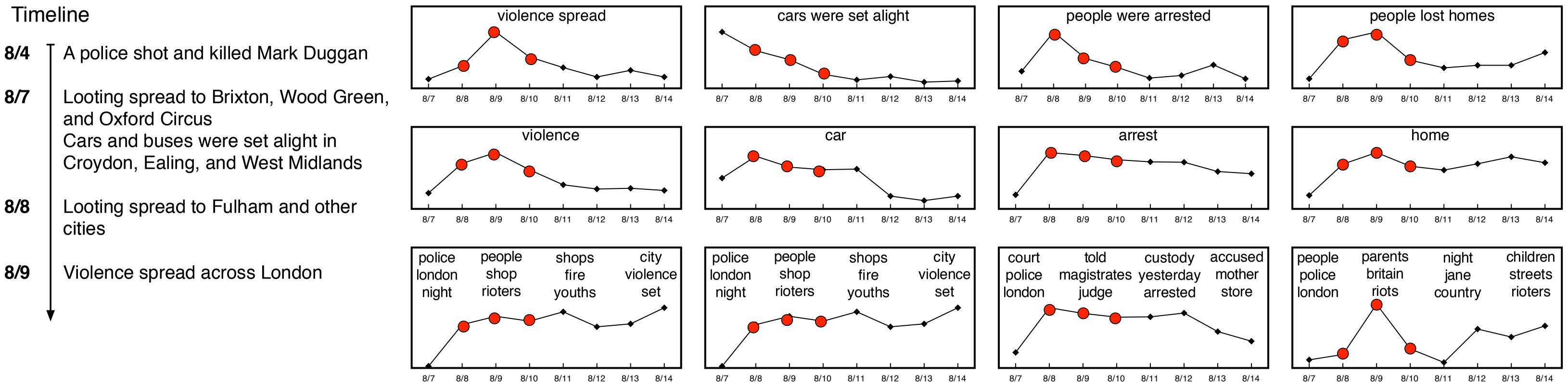


Figure 5.7: Event Tracking for London's Riot by Frames (top), Words (middle), and Topics (bottom)

Table 5.1: Description of Three Datasets in Our Experiments

Name	Docs	Sentences	Triplets	Words
Japan's Tsunami	22,108	608,723	402,601	13,114,356
London Riot	6,812	186,394	1,390,960	4,022,380
Egypt Revolution	1,759	70,211	140,348	1,493,745

5.4.1 Datasets

We use three different comparable corpora, collected from NewsBank[3], as datasets in the experiments. These corpora consist of news articles published by different news agencies about three news events: Japan's Tsunami (started from 3/11/2011), Egypt Revolution (started from 1/24/2011), and London Riot (started from 8/4/2011), respectively. Overview of the news events are provided as follows

- Japan's Tsunami[4]: A massive 8.9-magnitude earthquake shook Japan on March 11, 2011, causing a devastating tsunami to the coast of Japan. Due to the tsunami, the nuclear power plants in Fukushima were damaged, and one of the reactors in the Fukushima No. 1 nuclear plant partially melted down on the following day. As a result, the nuclear accident caused the exposure of nuclear radiation near the plant.

- Egypt Revolution[5]: Protests started on January 25, 2011, and thousands of people began taking to the streets to protest poverty, rampant unemployment, government corruption, and autocratic governance of President Hosni Mubarak, who has ruled the country for thirty years.

- London Riot[6]: Started from August 6, 2011, thousands of people took to the streets in several London boroughs as well as in cities and towns across England. Resulting chaos generated looting, arson, and mass deployment of police. The disturbances began after a protest in Tottenham, following a death of Mark Duggan, a local who was shot dead by police on August 4, 2011.

We searched news articles in NewsBank with keywords: "Japan Tsunami", "Egypt Revolution", and "London Riot", respectively, and collected articles

[3] http://www.newsbank.com

[4] http://en.wikipedia.org/wiki/2011_Tohoku_earthquake_and_tsunami

[5] http://en.wikipedia.org/wiki/2011_Egyptian_revolution

[6] http://en.wikipedia.org/wiki/2011_England_riots

for 11 days after the corresponding start date of each event. The statistics for the three datasets, and the statistics of semantic text information network constructed from them are shown in Table 5.1.

These datasets are available upon request.

5.4.2 Data Labeling

In order to quantitatively conduct empirical evaluations, we generated three labeled datasets (subsets of original ones in Table 5.1). Since getting pairwise labels for large datasets is very expensive, we sampled the datasets as follows:

We first chose one specific date from each dataset to increase the chance of having similar documents. Then, we randomly sampled 800 news articles published in the selected date. We performed a labeling procedure as follows: 1) randomly select a triplet t (called a query); 2) from each of our method and our baselines, generate the top 20 similar triplets to t; 3) combine the top 20 triplets of the all methods; 4) label the triplets. Repeating the steps 1-4, we can generate queries with its labeled pairs on which Precision at 20 (P@20) can be calculated.

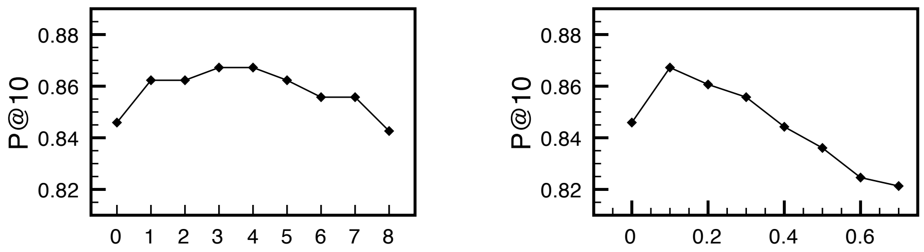
We asked two participants to label the pairs of triplets with two labels "same" and "different".

After eliminating pairs with different labels from two labelers (the inter-judge agreement rate was 86%), and rejecting queries with all positive and all negative cases, we have 650 queries for Japan corpus, 1,784 queries for London corpus, and 752 queries for Egypt corpus.

5.4.3 Quantitative Comparison with Baselines

In this section, we conduct a quantitative comparison between SynRank and other similarity measuring methods to demonstrate the effectiveness of our method on capturing semantic similarity between triplets. Methods based on unstructured text (non-link-based), and based on our semantic text information network are both considered as follows:

- Content (TF-IDF Based Cosine Similarity): This content-based baseline first indexes triplets into tf-idf vectors, and then computes their similarity by cosine similarity measure.



(a) Window Size σ

(b) Tradeoff Factor λ

Figure 5.8: Parameter Studies of σ and λ by P@10 on Japan's Tsunami Labeled Dataset: window size σ controls range of contexual information and λ controls the information trade-off between context and content of triplets.

- Corpus (Corpus-Based Distributional Similarity) [99]: This method computes distributional (corpus-based) similarity between words and compose them to get triplet similarity.
- WordNet (Knowledge-Based Similarity) [95]: It computes word similarity based on word synonym information from WordNet and compose them to calculate similarity of triplets.
- SimRank (Homogeneous Link-Based Similarity) [32]: Bipartite SimRank is applied on modified text information network where contextual links are removed since SimRank can only handle homogeneous links.
- P-Rank (Heterogeneous Link-Based Similarity) [8]: P-Rank is applied on our text information network by treating TT and TW relations as in-links and out-links in its framework.

We set all shared parameters between our method and those of baselines the same ($C = 0.8$, $\lambda = 0.1$), and the window size is set as $\sigma = 4$. We ran 20 iterations for SynRank, SimRank and P-Rank. The comparison of SynRank with the other five baseline methods in terms of P@5, P@10 and P@20 are shown in Table 5.3. It shows that SynRank outperforms other methods, demonstrating that leveraging both contextual and content information helps to measure similarities among triplets.

5.4.4 Parameter Study

Recall that the two parameters, σ and λ in SynRank formulas control their information gain between context and content. The window size σ controls

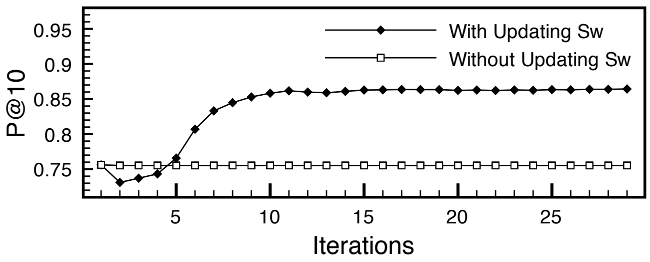


Figure 5.9: Performance Gain from Learning Corpus-based Word Similarity Jointly: P@10 over iterations is plotted, with or without updating the corpus-based word similarity matrix S_W, respectively, on Japan's Tsunami labeled dataset.

the range of contextual information, whereas λ in Equation (5.1) controls the information trade-off between the context and content of triplets. We now study the influence of parameters on SynRank's performance by measuring P@10 on Japan's Tsunami labeled dataset. Parameter study results on other data sets suggest similar trend. In Figure 5.8(a), SynRank gained best P@10 when $\sigma = 4$, and has relatively low P@10 when σ is small or large. As an extreme case, when $\sigma = 0$ it means no context of triplet is used in the calculation and only content is considered. Low P@10 at small σ indicates that context is useful to enhance similarity measure performance. Also, low P@10 at large σ demonstrates the fact that taking too large range of neighbor triplets as context may introduce too much unrelated and noisy information.

From Figure 5.8(b) we can examine the appropriate balance between content and contextual information in terms of similarity measuring performance. When $\lambda = 0$, we only make use of content information, which causes low performance gain. On the other hand, when λ goes close to 1, which means only context is leveraged, the performance gain also drops. We found the optimal value for λ is 0.1.

5.4.5 Corpus-based Word Similarity

In order to show the performance gain from corpus-based word similarity matrix updating, we plot the curve in Fig. 5.9 which shows the change of P@10 as SynRank iteration goes, i.e., S_W is updated iteratively. In the Figure, we show the P@10 with and without updating S_W (i.e., fix the word similarity matrix as $S_W = I_{|W|}$). Even though learning the word similarity from corpus leads to worse performance at first, it eventually enhances it and gets to a stable point, demonstrating that word similarity updating by the corpus bring usefulness.

5.4.6 Effectiveness of Frame-Based Document Representation

Many of the document representation studies [100] evaluate their proposed representation methods via specific applications like similarity search and document clustering. We choose the event tracking task because it is one of the key applications for monolingual comparable corpora analysis, and it is an interesting task for a collection of news articles.

We identified four important events from the Japan's Tsunami corpus and London Riot corpus. For each event, we searched for the best triplet clusters, keywords, and topics that describe the event, where topics are from LDA [77] with 20 topics. Then, we plot them by counting their occurrences in the corpus and normalizing by the number of documents in each date. Figure 5.6 and 5.7 show the trends in the order of triplets, words, and topics. The bottom row in Figure 5.6 and 5.7 show the event tracking by topics. The highest probability words from each topic are listed on each plot.

In Figure 5.6 and 5.7, we also indicated the timelines of the two corpora. The red dots in the trend plots indicate the consistent points with the timelines[7]. Thus, those red dots should be higher than other data points.

As shown in the two figures, in general, the frame-based event tracking performs better than the other two baselines. Quantitatively, we can take the average of the rankings of the red dots within the plots as an evaluation measure. For example, in the "reactor" plot, the four red dots ranked 1, 3, 4, and 9. The averages of the rankings of the 19 red dots for frames, words, and topics are respectively 2.33, 2.42, and 3.75. Since lower is better in this measure, the frame-based event tracking is better than the others.

The observation is that if an event cannot be described in a single keyword, it is hard to track events by the keyword. For example, "radiation leaked" cannot be described by a single word. Topic models are designed to model the theme of the words, which are more general concepts than events. It is hard to specify an event using topics. The topic trend plots have many

[7] Since the timestamps of the news articles are the publication dates, they are off by one from the timeline dates

Table 5.2: Topic Model Evaluation Survey

The Number of Topics	10	20	50
The Pairwise Agreement	33%	46%	33%

Table 5.3: Precision Evaluations of Different Compared Methods on Three Labeled Datasets

Method	Japan's Tsunami			London's Riot			Egypt Revolution		
	P@5	P@10	P@20	P@5	P@10	P@20	P@5	P@10	P@20
Content	0.767	0.698	0.653	0.853	0.787	0.719	0.848	0.773	0.689
Corpus	0.756	0.694	0.650	0.859	0.794	0.723	0.853	0.770	0.681
WordNet	0.770	0.711	0.664	0.854	0.791	0.722	0.850	0.767	0.683
SimRank	0.747	0.683	0.641	0.798	0.737	0.695	0.745	0.722	0.679
P-Rank	0.783	0.726	0.681	0.868	0.803	0.728	0.817	0.746	0.677
SynRank	**0.856**	**0.864**	**0.854**	**0.883**	**0.848**	**0.807**	**0.905**	**0.843**	**0.739**

peaks because one topic covers more than one event. These results show that topics are not suitable to specify an event. Increasing the number of topics does not help to specify events. The following survey experiment shows that increasing the number of topics does not make the topics more specific.

We make multiple choice questions. Each question has a one event description by a sentence and five choices of topics with top ranked words from the word distributions of the topics. Then, participants are asked to pick the most relevant topic for a given event description.

We first generated topics using LDA [77] for the London corpus, and for each of the four events, we selected five most relevant topics by looking at their word distributions and the rankings of several keywords. We repeated this survey for different number of topics (10, 20, and 50). We computed the pairwise agreements for the different number of topics as shown in Table 5.2. The pairwise agreement indirectly measures the specificity of topics for events. When the number of topics is 20, the pairwise agreement is lowest, which means the topics from LDA with 20 topics describe events better than those from LDA with 10 or 50 topics. Thus, increasing the number of topics does not improve the specificity of topics for events.

CHAPTER 6

MULTI-ATTRIBUTE PROXIMITY NETWORK MODELING

6.1 Overview

With the proliferation of digital media and newswires, massive online news data has become widely available. Subsequently, automated analysis of news events has become an important research issue since the sheer quantity of news events makes human analysis infeasible. An interesting common phenomenon among these large collections of news articles is that *these news corpora not only have high coverage of world-wide news events, but also contain a lot of partially overlapping information.* Partially overlapping information gives an opportunity to align articles and discover both what is important and what is correct within the collection.

More specifically, the statistical power available from information redundancy makes it possible to find and describe important events as well as their essential attributes such as time, location, as well as related organizations and persons. Moreover, it helps discover connections between events in news articles because news articles cover multiple related events together, contrasting to short documents like micro-blog posts which mostly cover a single event.

Discovering and visualizing events with their key descriptors, essential attributes, and their connections makes it possible to understand the big picture when bombarded with a huge amount of information in news articles. Effective event discovery can be used to summarize and navigate a news corpus and effectively retrieve nuggets of knowledge for a specific interest. It is thus desirable to build a system that, given a news corpus, discovers important events automatically, attributes key properties to them accurately, and connects them thematically.

There have been multiple approaches that summarize and visualize news

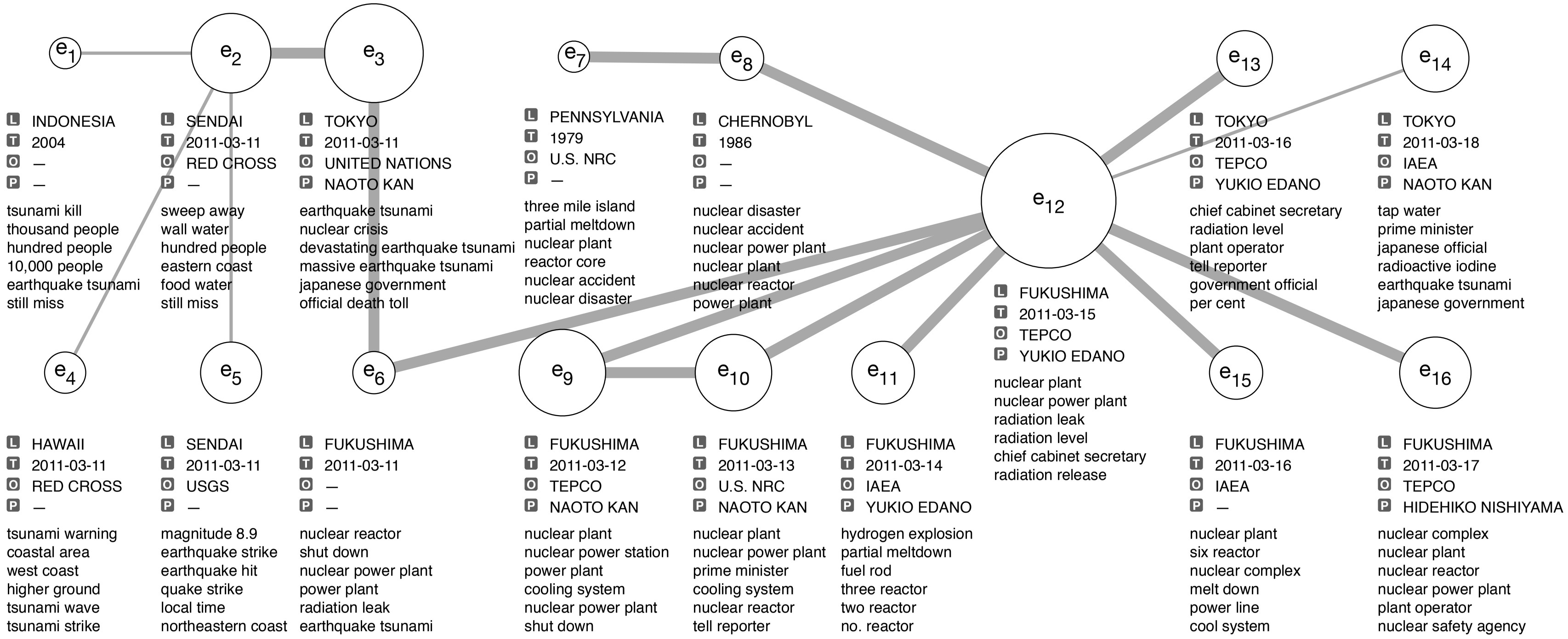


Figure 6.1: Events, their attributes and event connections for 2011 Japan tsunami and nuclear accident generated by ProxiModel.

events. However, they suffer from several limitations.

1. **Unigram-based event descriptors**: While some systems use *unigrams* (i.e., *single words*) as event descriptors [37, 101], it has been shown that *phrases* are more descriptive and interpretable than words [102, 103]. There are several studies that use phrases in information flow detection on the Web [104, 105] or in event detection with micro blogs [16, 106]. However, their phrases are not for describing events but for searching and linking multiple documents.

2. **Lack of key dimensions for event description**: A reader can better understand the context of an event if she knows several key dimension (or attribute) values of an event: *when* and *where* the event happened, and *who* or *which organizations* the event is related to [107]. Most of the studies do not have such attributes in their outputs, and some extract event attributes from meta data like publication dates and reporting locations, which can be misleading. Some key dimension values, such as persons or organizations, are often unavailable or inaccurate.

3. **Ignoring event connections within a single document**: Events naturally relate to each other. While these connections are often explicitly addressed within news articles, many event detection and tracking studies in micro blogs [108, 106, 16] and news articles [37, 101] make the strong assumption that each document describes a single event. While for short documents like micro blog posts, this assumption may hold, it often fails to hold for long documents like news articles which are more susceptible to event drift. Further, enforcing this assumption will lose event connection evidences found within a single document.

It is challenging but desirable to effectively mine and extract high-quality event knowledge from large, noisy text corpora consisting of partially repeated news articles. In this study, we develop a new approach, ProxiModel (**Proxi**mity network-based generative **model**), which leverages the notion of proximity: *If two instances co-occur in news articles closely and frequently, they have high proximity.* This notion of proximity is used to model events, descriptors, attributes, and connections.

Fig. 6.1 shows an example output of ProxiModel for the news collection about 2011 Japan tsunami and nuclear accident. There are 16 events shown, with automatically generated phrasal key descriptors and event attributes, where circle size represents the importance of events, and line width the strength of event connection.

By automatically identifying latent news events, their phrasal descriptors, attributes, and connections, ProxiModel provides an effective framework for organizing and exploring these huge amounts of data. Without understanding the meanings of sentences in news articles, our method models the events based on the notion of proximity. ProxiModel possesses several key qualities that differentiate it from other event detection methods and allow for high-quality event discovery and intuitive and interpretable organization of news: (1) it provides a big picture of events in news articles with rich information, which includes the importance of events, key phrasal descriptors, event attributes, and event connections, (2) it utilizes proximity information and regularizes sparsity in model parameters to find correct event attributes and connections from text, and (3) it uses a scalable data structure, called a proximity network, that stores necessary information from news articles.

6.2 Event Mining

In this section, we introduce a proximity network generated from a comparable news corpus and propose an event mining method on the constructed proximity network.

6.2.1 Event Definition

While bearing some similarities, event discovery has subtle differences from topic discovery or topic modeling. Traditionally, a topic is defined as a distribution of words [77]. An event, however, is associated with several key dimensions including location, time, person, organization, and a set of descriptive phrases as theme.

We first examine several key dimensions and primitives of events.

1. Time: Temporal expressions are extracted from documents and normalized to the form of the TIMEX3, which is a part of the TimeML annotation language [109]. Relative temporal expressions like "last night" and "yesterday" are also normalized by taking the report time or publication time of

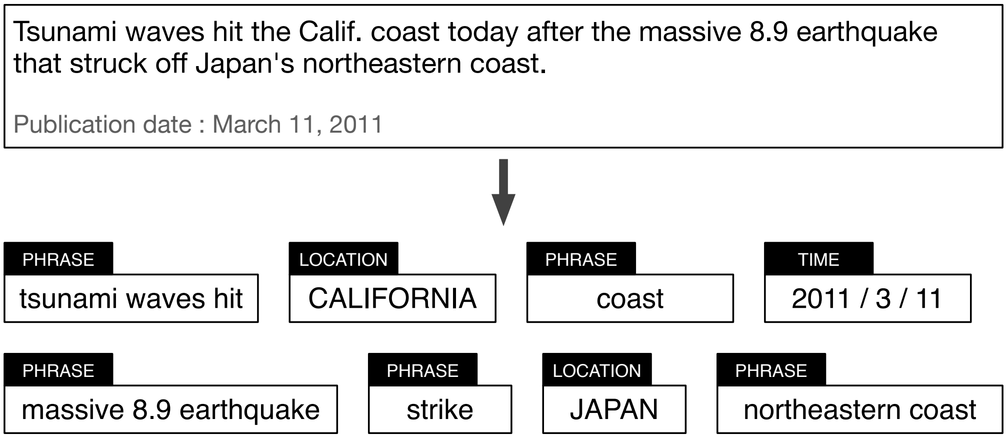

Tsunami waves hit the Calif. coast today after the massive 8.9 earthquake that struck off Japan's northeastern coast.
Publication date : March 11, 2011
PHRASE
tsunami waves hit
LOCATION
CALIFORNIA
PHRASE
coast
TIME
2011 / 3 / 11
PHRASE
massive 8.9 earthquake
PHRASE
strike
LOCATION
JAPAN
PHRASE
northeastern coast

Figure 6.2: The representation of a document as a sequence of bases. NLP tools and a phrase mining algorithm are used to segment documents.

the document as the fixed reference time. For example, the word “today” in Figure 6.2 is mapped to “2011/3/11” because of the publication date. We informally refer to the extracted normalized time expressions as time.

2. Location: Locations are geo-political entities such as city, state, and country. They are extracted and normalized to their surface forms. For example, the word “Calif.” is mapped to “CALIFORNIA” in Figure 6.2.

3. Person Extracted persons are not only public figures, but also private figures who are mentioned in news articles. For example, Jun-seok Lee, who was the captain of the sunken Sewol Ferry, is extracted. Coreferences are also resolved within a document such that Captain Lee is mapped to Jun-seok Lee.

4. Organization Companies, governments, and other organizations are extracted. An abbreviation of an organization is mapped to its full name if they are mentioned in the same document. For example, TEPCO is mapped to Tokyo Electric Power Company.

5. (Thematic) phrases: A phrase is a sequence of contiguous words that represents a meaningful semantic unit. Recently developed phrase mining algorithms such as ToPMine [103] and SegPhrase [110] perform fast, pruning-based frequent contiguous pattern mining and then statistically reason about the significance of the contiguous co-occurrence while applying context constraints to discover meaningful phrases. We use ToPMine [103] to mine quality phrases representing the above dimensions as well as thematic phrases

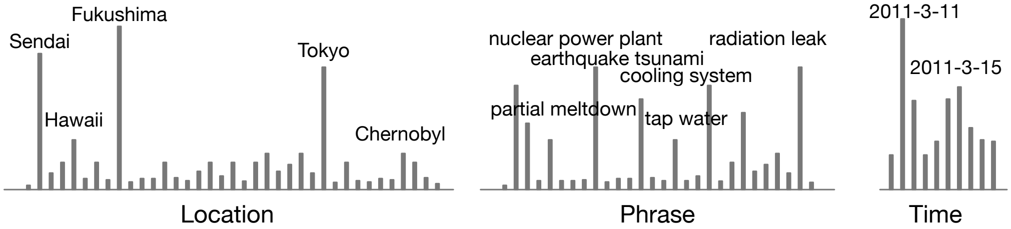


Figure 6.3: Statistical power of comparable news corpus: key information can be easily discovered by counting the occurrences of basis

that form a thematic dimension as shown in Fig. 6.2.

For simplicity, we refer to any phrase, time, location, person, or organization as a *basis*. A document d is a sequence of segments $\langle d_1, d_2, \ldots, d_k \rangle$, where d_i corresponds to a basis. The order of segments corresponds to the order of original word tokens in the document. For a given comparable news corpus, we want to discover events defined as follows:

Definition 7 (Event)**.** *An event, z, is a real-world occurrence represented as a 5-tuple $\langle \eta_z, \phi_z^L, \phi_z^T, \phi_z^O, \phi_z^P \rangle$, where η_z is the distribution over all phrases, ϕ_z^L is that (distribution) over all locations, ϕ_z^T is that over all time, ϕ_z^O is that over all organizations, and ϕ_z^P is that over all persons.*

6.2.2 Comparable News Corpus

A comparable news corpus is a collection of news articles that cover related events. The definition of a comparable news corpus is the same as that of a comparable corpus [111] frequently used in natural language processing tasks like translation, except that each document in a comparable news corpus has the same news events instead of the same topics. We can collect such corpora easily, for example, using keyword search on a news database. A comparable news corpus contains a lot of partially repeated information and common phrases for important events. These fragmented, but overlapping pieces of information can complement each other in a collective analysis.

Here we briefly illustrate the potential of a collective analysis on a comparable news corpus, with two simple but incomplete analysis methods. Counting the occurrences of bases gives key information for each dimension, such as locations, phrases, and time as shown in Fig. 6.3. By counting redundant

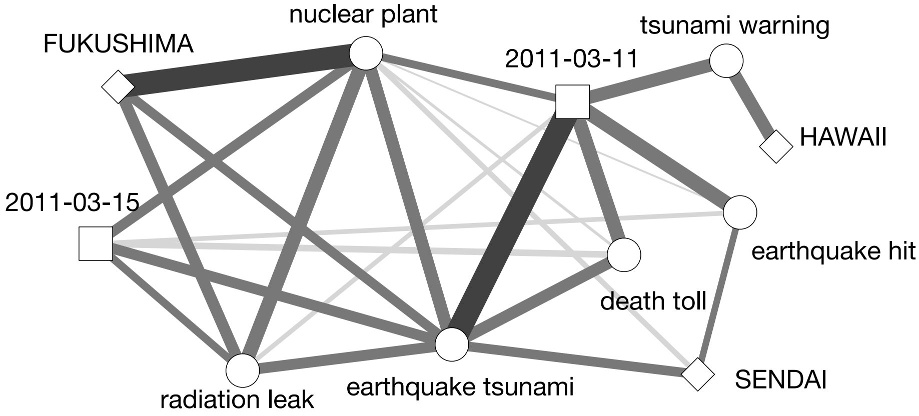


Figure 6.4: An example of proximity networks from the Japan Tsunami corpus. There are three types of nodes: ◯: a phrase node, ◊: a location node, and □: a time node. Line thickness indicates the weight of the corresponding edge in log scale.

information across the news articles about Japan tsunami in 2011, the peaks show important information in each dimension.

Unfortunately, such peaks, generated from document-level co-occurrences of key dimensions, cannot be used to extract events. This is because events mined from document-level co-occurrences can be inaccurate. For example, a hydrogen explosion in a nuclear power plant happened in Fukushima on March 14, 2011. The phrase “hydrogen explosion”, however, has high co-occurrence with “2011-03-11” because most of the news articles mentioned the earthquake on March 11, 2011 to address the cause of the damaged nuclear power plant. To avoid these problems, events should be resolved by considering the *proximity* of bases within documents.

6.2.3 Proximity Network

Proximity is a measure of how close two terms occur in a document or a passage. This measure has been successfully adopted in many different tasks including word association [112, 113], document retrieval [114, 115], named entity retrieval and expert finding [116, 117].

Proximity is an important cue for estimating the strength of association between two bases, in which a strong association between two bases indicates they belong to the same event.

For example, in the Japan Tsunami news corpus, we find time expressions of `2011/03/11` near `earthquake hit` phrase frequently: This is the time when a massive earthquake hit Japan. In addition, we find location mentions of Fukushima around the phrase `radiation leak` much more than any other locations: Similarly, Fukushima is the city where crippled nuclear power plants had radiation leaks.

We want to collect such evidence or associations between bases in an efficient way by constructing an information network, called a *proximity network*. We define a proximity network that has different types of nodes and edges between them. The set of nodes in the proximity network is the set of bases in a given corpus $\mathcal{C}$, and the weight of an edge between two nodes is based on proximity between the nodes as follows.

$$e_{x,y} = \sum_{d \in \mathcal{C}} \sum_{1 \leq i < j \leq N_d} \delta_d(i,x)\delta_d(j,y)k(i,j),$$

where $\delta_d : \mathbb{N} \times \mathcal{B} \to \{0,1\}$ is an indicator function and $k(i,j)$ is a proximity kernel such that

$$\delta_d(i,x) = \begin{cases} 1 & \text{if the segment at position } i \text{ in } d \text{ corresponds to } x \\ 0 & \text{otherwise} \end{cases}$$

$$k(i,j) = exp\left[\frac{-(i-j)^2}{2\sigma^2}\right]$$

Note that σ is a constant that controls the propagation scope of each segment. A proximity network with small σ captures very different information from one with large σ. We will use two proximity networks with different σs to model different information as discussed in the next section.

An example of proximity networks is shown in Figure 6.4, generated from the Japan Tsunami corpus with $\sigma = 1$. It shows strong proximity between `FUKUSHIMA` and `nuclear plant` and between `earthquake tsunami` and `2011-03-11`. If one tries to cluster the nodes in the figure based on the edges, such clustering may yield three clusters as follows:

1. { `2011-03-15`, `FUKUSHIMA`, `radiation leak`, `earthquake tsunami`, `nuclear plant` }
2. { `2011-03-11`, `SENDAI`, `earthquake tsunami`, `death toll`, `earthquake hit` }

3. { `2011-03-11`, `HAWAII`, `tsunami warning` }

There exist some latent parameters that form clusters of nodes, and we model such parameters by events as addressed in the following section.

A proximity network constructed from the corpus could be noisy and dense without post-processing. Since our corpus has partially repeated news articles and important links get greater weights, we use link minimum support (l_{minsup}) to remove infrequent links (i.e., whose weights are less than l_{minsup}). This truncation not only removes noises in the network, but also makes the network sparse, where modeling becomes more efficient in time and space.

6.2.4 Proximity Network Generative Models

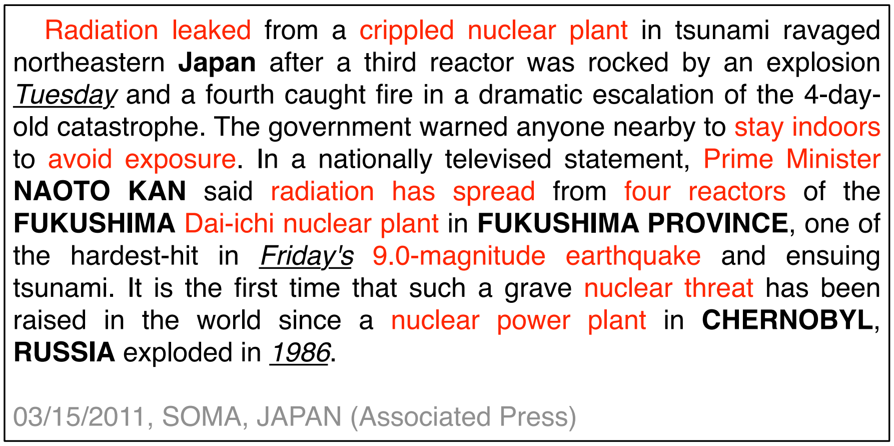
In this section, we describe *Proximity Network Generative Model* (ProxiModel). Proximity networks have pairwise proximity information among bases. Unlike previous studies that use heuristic proximity metrics [112, 113, 114, 115, 117], we learn latent parameters from proximity to model events. Specifically, we design a generative model for proximity networks to model events, in which edges in the networks are generated under some assumptions. In order to model events with descriptors, attributes, and connections, we construct two proximity networks N_s and N_l, with small σ_s and large σ_l values, from an input corpus.

Proximity Network N_s: σ_s is set smaller than σ_l to capture proximity within a smaller propagation scope. This proximity network is mainly used to learn event descriptors and attributes. It only has edges with at least one phrase end node. In other words, it only has edges consisting of `phrase-phrase`, `phrase-time`, `phrase-location`, `phrase-organization`, and `phrase-person`.

Proximity Network N_l: σ_l is set greater than σ_s to capture proximity within a larger propagation scope. This proximity network is mainly used to learn event connections. It only has edges with two phrase end nodes. In other words, it has only edges of `phrase-phrase`.

Our Assumptions: In the generative model, we encode our assumptions as follows:

1. Two phrases for the same event have high proximity in N_s.



Radiation leaked from a crippled nuclear plant in tsunami ravaged northeastern **Japan** after a third reactor was rocked by an explosion *Tuesday* and a fourth caught fire in a dramatic escalation of the 4-day-old catastrophe. The government warned anyone nearby to stay indoors to avoid exposure. In a nationally televised statement, Prime Minister **NAOTO KAN** said radiation has spread from four reactors of the **FUKUSHIMA** Dai-ichi nuclear plant in **FUKUSHIMA PROVINCE**, one of the hardest-hit in *Friday's* 9.0-magnitude earthquake and ensuing tsunami. It is the first time that such a grave nuclear threat has been raised in the world since a nuclear power plant in **CHERNOBYL**, **RUSSIA** exploded in *1986*.

03/15/2011, SOMA, JAPAN (Associated Press)

Figure 6.5: An example news article to discuss our assumptions. Phrases are in red, named entities are in bold, and temporal expressions are in italic and underlined.

2. A phrase and an event attribute for the same event have high proximity in N_s.
3. Two phrases from different events have high proximity in N_l if the events are connected
4. Each event has a few event attributes of the same type.
5. There are only a few event connections.

Note that two phrases for the same event have high proximity in N_l as well as in N_s because of the Gaussian kernel.

We first address the assumptions with an example news article in Figure 6.5. The news article mainly reports the leaked radiation from a crippled nuclear power plant in Fukushima, Japan, which happened in March 15, 2011.

The article also mentions a main cause of the damages in the nuclear power plant—a massive earthquake hit Japan in March 11, 2011 which caused strong tsunamis that damaged the nuclear power plant. For example, `radiation leaked` and `crippled nuclear plant` have high proximity in N_s as an example of Assumption 1. `9.0-magnitude earthquake` and `Friday` have high proximity in N_s as an example of Assumption 2. In addition, `9.0-magnitude earthquake` and `crippled nuclear plant` have high proximity in N_l.

6.2.5 Generative Process

In our generative model, we convert the edge weights in N_s and N_l to multigraphs as follows: The number of edges between two nodes is equal to the integer part of the weight in the original network. We denote the total number of edges in N_s and N_l by n_s and n_l respectively.

We define a generative process for edges in N_s and N_l as shown in Algorithm 3. In N_s, each edge belongs to one event, indicating two end points belong to the event. In N_l, end points of each edge can belong to different events as well as the same event. See Figure 6.6 for a graphical representation of the model.

In this generative model, we can derive the distribution of the number of edges between any two nodes in N_s. Generating an edge between a phrase-i node and an attribute-j node of type t in event z can be modeled as a Bernoulli trial with a success probability of $\theta_z \rho^t \eta_{z,i} \phi^t_{z,j}$. When n_s is large, the total number of successes, $e^{s,t}_{i,j,z}$ asymptotically follows a Poisson distribution [118] as follows:

$$e^{s,t}_{i,j,z} \sim Poisson(n_s \theta_z \rho^t \eta_{z,i} \phi^t_{z,j}).$$

Due to the additive property of Poisson distribution, we can derive that the observed variable $e^{s,t}_{i,j}$ follows a Poisson distribution as follows:

$$e^{s,t}_{i,j} = \sum_z e^{s,t}_{i,j,z} \sim Poisson(\sum_z n_s \theta_z \rho^t \eta_{z,i} \phi^t_{z,j}).$$

Thus, given the model parameters, the probability of all observed edges in N_s is

$$\mathcal{L}_s = p(\{e^{s,t}_{i,j}\}|\theta, \rho, \eta, \phi) = \prod_{i,j,t} \frac{(\mu_s^{i,j,t})^{e^{s,t}_{i,j}} exp(-\mu_s^{i,j,t})}{e^{s,t}_{i,j}!},$$

where $\mu_s^{i,j,t} = \sum_z n_s \theta_z \rho^t \eta_{z,i} \phi^t_{z,j}$.

Similarly, we can derive the distribution of the number of edges between any two nodes in N_l.

$$e^l_{i,j} = \sum_{z_1,z_2} e^l_{i,j,z_1,z_2} \sim Poisson(\sum_{z_1,z_2} n_l \varphi_{z_1,z_2} \eta_{z_1,i} \eta_{z_2,j}).$$

Thus, given the model parameters, the probability of all observed edges in N_l is

$$\mathcal{L}_l = p(\{e^l_{i,j}\}|\varphi, \eta) = \prod_{i,j} \frac{(\mu^{i,j}_l)^{e^l_{i,j}} exp(-\mu^{i,j}_l)}{e^l_{i,j}!},$$

where $\mu^{i,j}_l = \sum_{z_1,z_2} n_l \varphi_{z_1,z_2} \eta_{z_1,i} \eta_{z_2,j}$.

The overall probability of all observed edges in N_s and N_l is

$$\mathcal{L} = \mathcal{L}_s \cdot \mathcal{L}_l.$$

We encode Assumptions 1 and 2 in the generative process of N_s, and Assumption 3 in the generative process of N_l.

To model the assumptions that each event has only a few event attributes and there are only few event connections, we introduce sparse regularization on model parameters as their priors.

We impose an *a priori* probability on the parameters given by

$$\mathcal{L}' \propto \mathcal{L} \cdot p(\phi) \cdot p(\varphi), \tag{6.1}$$

where $p(\phi) = e^{-\sum_z \sum_t \alpha_t \mathcal{H}(\phi^t_z)}$, $p(\varphi) = e^{-\beta \mathcal{H}(\varphi)}$, $\mathcal{H}(x)$ is the Shannon's entropy of distribution x, and α_t and β are sparse prior weights. With higher values of α_t and β, event attributes and connections have lower entropies, i.e., sparser.

6.2.6 Parameter Learning

We learn the model parameters by the Maximum Likelihood (ML) principle. To deal with the normalization constants of the prior probabilities, the log-likelihood of Eq (6.1) must be augmented by appropriate Lagrange multipliers: $Q = \log \mathcal{L}' + \lambda_\theta \left(\sum_z \theta_z - 1\right) + \lambda_\rho \left(\sum_t \rho_t - 1\right) + \sum_z \lambda^z_\eta \left(\sum_i \eta_{z,i} - 1\right) + \sum_{t,z} \lambda^{t,z}_\phi \left(\sum_i \phi^t_{z,i} - 1\right) + \lambda_\varphi \left(\sum_{z_1,z_2} \varphi_{z_1,z_2} - 1\right)$

Then, we maximize Q using an Expectation-Maximization (EM) algorithm that iteratively infers the model parameters.

The E-step calculates the expected number of edges:

$$\hat{e}^{s,t}_{i,j,z} = e^{s,t}_{i,j} \frac{\theta_z \eta_{z,i} \phi^t_{z,j}}{\sum_k \theta_k \eta_{k,i} \phi^t_{k,j}} \tag{6.2}$$

$$\hat{e}^{l}_{i,j,z_1,z_2} = e^{l}_{i,j} \frac{\varphi_{z_1,z_2} \eta_{z_1} \eta_{z_2}}{\sum_{k_1,k_2} \varphi_{k_1,k_2} \eta_{k_1} \eta_{k_2}} \tag{6.3}$$

In the M-step, the update equations for θ_z, ρ_t, and $\eta_{z,i}$ are given by

$$\theta_z = \frac{\sum_{i,j,t} \hat{e}^{s,t}_{i,j,z}}{n_s}, \quad \rho_t = \frac{\sum_{i,j,z} \hat{e}^{s,t}_{i,j,z}}{n_s}, \tag{6.4}$$

$$\eta_{z,i} = \frac{\sum_{j,t} \hat{e}^{s,t}_{i,j,z} + \sum_{j,z_2} \hat{e}^{l}_{i,j,z,z_2}}{\sum_{k,j,t} \hat{e}^{s,t}_{k,j,z} + \sum_{k,j,z_2} \hat{e}^{l}_{k,j,z,z_2}} \tag{6.5}$$

In the M-step, maximization of Q with respect to ϕ and φ leads to different sets of equations due to their priors and Lagrange multipliers:

$$\frac{1}{\phi^t_{z,i}} \sum_j \hat{e}^{s,t}_{i,j,z} - n_s \theta_z + \alpha_t \log \phi^t_{z,i} + \alpha_t + \lambda^{t,z}_{\phi} = 0 \tag{6.6}$$

$$\frac{1}{\varphi_{z_1,z_2}} \sum_{i,j} \hat{e}^{l}_{i,j,z_1,z_2} - n_l + \beta \log \varphi_{z_1,z_2} + \beta + \lambda_{\varphi} = 0. \tag{6.7}$$

The above set of simultaneous transcendental equations for ϕ and φ can be solved using the Lambert's $\mathcal{W}$ function similar to [119].

$$\phi^t_{z,i} = \frac{-\sum_j \hat{e}^{s,t}_{i,j,z}/\alpha_t}{\mathcal{W}(-\sum_j \hat{e}^{s,t}_{i,j,z} e^{1-n_s\theta_z/\alpha_t + \lambda^{t,z}_{\phi}/\alpha_t}/\alpha_t)}, \tag{6.8}$$

where equations Eq. (6.6) and Eq. (6.8) form a set of fixed-point iterations for $\lambda^{t,z}_{\phi}$, and thus the M-step for finding $\phi^t_{z,i}$.

Similarly, we can get the following update equation for φ_{z_1,z_2}:

$$\varphi_{z_1,z_2} = \frac{-\sum_{i,j} \hat{e}^{l}_{i,j,z_1,z_2}/\beta}{\mathcal{W}(-\sum_{i,j} \hat{e}^{l}_{i,j,z_1,z_2} e^{1-n_l/\beta + \lambda_{\varphi}/\beta}/\beta)}. \tag{6.9}$$

Algorithm 3 Proximity Link Generative Models

```
for each edge $e_i$ in $N_s$ do
    Draw an event $z_i \sim Multi(\theta)$
    Draw a type $t_i \sim Multi(\rho)$
    Draw a phrase $p_i \sim Multi(\eta_{z_i})$
    Draw an attribute $x_i \sim Multi(\phi_{z_i}^{t_i})$
end for
for each edge $e_j$ in $N_l$ do
    Draw a pair of events $w_j \sim Multi(\varphi)$
    Draw a phrase $y_{j,1} \sim Multi(\eta_{w_{j,1}})$
    Draw a phrase $y_{j,2} \sim Multi(\eta_{w_{j,2}})$
end for
```

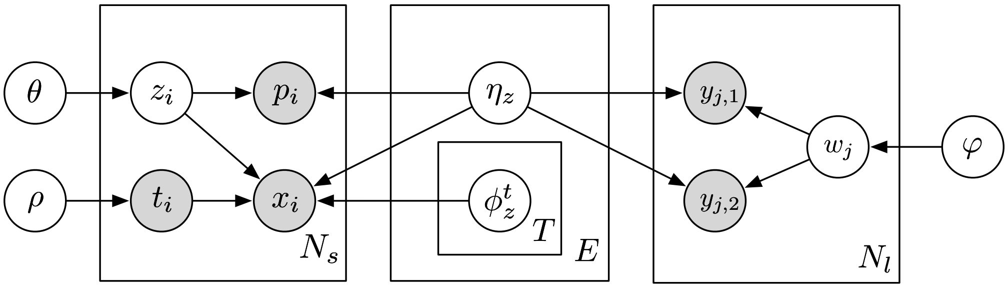


Figure 6.6: A generative model for σ_s-proximity network(N_s) and σ_l-proximity network(N_l)

6.3 Experiments

In this section, we evaluate ProxiModel on a variety of news article corpora. We begin by first describing the comparable news corpora we collected for our evaluation, then showing the quality of event descriptors and attributes generated by ProxiModel, when compared to those by other baselines. After evaluating the quality of our events, we focus on benchmarking the efficiency of our algorithm. We demonstrate the efficiency gains of constructing a compact network for a corpus (without document-level representation) as we increase the number of documents. In addition, we show how using a link minimum support threshold reduces the runtime while maintaining high-quality attributes. Since we have three technical parameters—noise reduction, proximity and sparsity—that affect the quality of event descriptors and attributes as well as method efficiency, we perform parameter studies by varying these parameters to highlight the effects of proximity and sparsity.

Finally, by applying our methodology and extracting key event descriptors and event attributes, we demonstrate how one can construct an event storyline detailing the timeline of events.

6.3.1 Datasets

We evaluate each method on three news corpora, collected from a variety of news agencies through NewsBank [1], which cover different distinct topical content.

- **Sewol Ferry (2014)**: The sinking of Sewol ferry occured on April 16, 2014, en route from Incheon to Jeju. We searched articles with "Sewol Korea" keywords, and collected 1,520 articles published from April 15, 2014 to June 30, 2014.
- **Japan Tsunami (2011)**: A massive 8.9-magnitude earthquake shook Japan on March 11, 2011, causing a devastating tsunami to the coast of Japan. We searched articles with "Japan Tsunami" keywords, and collected 21,528 articles published from March 11, 2011 to April 11, 2011.
- **Multiple (2014)**: This dataset has multiple news stories, including Ebola outbreak, the 2014 Winter Olympics, Russian military intervention in Ukraine, missing MH370, Gamboru Ngala attack, Jos bombings, ISIS, Israel-Gaza conflict, and the MH17 tragedy. We searched articles with multiple keywords for each news story, and collected 100,472 articles published in 2014.

Table 6.1 summarizes the collected three datasets. The number of events and the other input parameters can be selected by using cross-validation with perplexity or Bayesian information criterion (BIC) [120]. In our study, we set the number of events as follows: 10 for Sewol Ferry, 30 for Japan Tsunami, and 60 for Multiple. As the default values, we set the proximity parameters σ_s and σ_l to 1 and 10, and the sparsity parameters α_L, α_T, α_O, α_P, and β to 1000, 1000, 10, 10, and 100, respectively.

[1] www.newsbank.com

Dataset	Articles	Words	TIME	GPE	ORG	PERSON
Sewol Ferry	1,520	5,706	67	190	164	235
Japan Tsunami	21,528	31,793	574	2,367	2,862	4,338
Multiple	100,472	133,540	3,565	10,907	15,417	39,093

Table 6.1: Statistics of the datasets: We count words and other entities that appear in at least 5 different news articles.

6.3.2 Baselines

For the comparative study, we have identified two, directly comparable methods and two variations of ProxiModel as baselines for each of our proposed hypotheses.

- **HISCOVERY**: This work [37] assumes each document describes a single event, and the event time is very close to the publication date of the news article. Because of the event time assumption, this model uses publication dates as extra information, which is not available to other baselines.
- **PhraseLDA**: PhraseLDA is proposed in [103]. This model extends Latent Dirichlet allocation to incorporate phrase generation. It utilizes the co-occurrence of phrases or attributes in documents, instead of using proximity. In addition, it has homogeneous outcomes from the generative process, in which all phrases and attributes are generated from a single distribution.
- **ProxiModel-NP**: This is a variation of our model which does not use the proximity information, but the co-occurrence information. It is a special case of ProxiModel, where the proximity parameters (σ) are set to a very large number. This model serves to see the effectiveness of the proximity information.
- **ProxiModel-NS**: This is our model without the sparse regularization. It is a special case of ProxiModel, where the sparsity parameters are 0. This baseline is designed to show the effects of sparse priors.

6.3.3 Event Descriptor Evaluation

In this section we apply a proposed user-study to evaluate the descriptors of the key events across each method.

Method	HISCOVERY	PhraseLDA	ProxiModel-NP	ProxiModel-NS	ProxiModel
Descriptors	ferry	**third mate**	**save life**	**people in need**	**people in need**
	ship	**abandon ship**	**third mate**	**two crew member**	**two crew member**
	people	begin list	**two crew member**	**third mate**	**third mate**
	captain	**senior prosecutor**	look whether	**arrest suspicion negligence abandon**	**arrest suspicion negligence abandon**
	passenger	30 minute	distress call	look whether	look whether
	crew	make sharp turn	**evacuation order**	**senior prosecutor**	**evacuation order**
	rescue	**arrest warrant**	sharp turn	**evacuation order**	**save life**
	sewol	**arrest suspicion negligence abandon**	coast guard	news agency	**arrest warrant**
	miss	look whether	inside ferry	**save life**	**senior prosecutor**
	official	**first mate**	high school	**arrest warrant**	**four crew member**
Attributes	SEOUL	INCHEON	INCHEON	MOKPO	MOKPO
	2014-04-20	2014-04-19	2014-04-19	2014-04-19	2014-04-19
	LEE JOON-SEOK	LEE JOON-SEOK	LEE JOON-SEOK	LEE JOON-SEOK	LEE JOON-SEOK
	YANG JUNG-JIN	YANG JUNG-JIN	YANG JUNG-JIN	YANG JUNG-JIN	YANG JUNG-JIN

Table 6.2: Top 10 descriptors from different methods for a key event in Sewol dataset: “captain arrested on suspicion of negligence” in 4/19/2014. The descriptors in bold are labeled as key descriptors for the event by at least one participant.

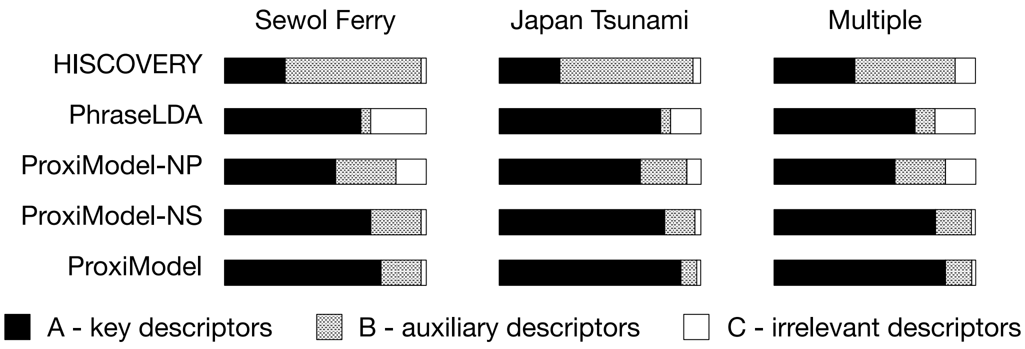


Figure 6.7: The evaluation of descriptors of the aligned key events generated by different methods

We select key events from each dataset that can be found across all the methods.

These alignments were performed by expert examination of the descriptors and attributes. One example of a key event used in our evaluation is: "captain arrested on suspicion of negligence" in 4/19/2014 which was reported in a news article: "senior prosecutor Yang Jung-jin said the ferry captain, Lee Joon-seok, 68, faces five charges including negligence of duty and violation of maritime law." [2]

We found 4 events in Sewol Ferry, 10 events in Japan Tsunami, and 16 events in Multiple datasets. For each key event, we collected the top 10 descriptors from each method, combined and shuffled them to make a method-blind list of descriptors. We then asked four participants, who are very familiar with each event and have first read multiple articles for further familiarity, to label each descriptors into the following categories A to C: (A) key descriptors, (B) vague or auxiliary descriptors, and (C) not related. The agreement of the labels by the four participants was measured as 0.67 in Fleiss' kappa [121], indicating substantial agreement.

We show an example of aligned key events in Sewol Ferry and the top 10 descriptors and the associated attributes from each methods in Table 6.2. Figure 6.7 shows the distribution of labels for each method. The phrase-based methods have smaller proportions of B labels than the word-based method, HISCOVERY. In addition, the results show that modeling proximity is important to find key descriptors for events.

[2] The article can be found in `http://goo.gl/0jW2dO`

6.3.4 Event Attribute Evaluation

We use a positive and negative set of event attributes for event attribute evaluation. We define a positive set of event attributes as follows: all attributes in a positive set related to one specific event.

We generated a candidate list of attribute sets and labelled them manually. Table 6.3 shows some of the annotated event attribute sets.

We compute the probability to generate a given set of attributes from one event as following:

$$Pr(\tau|\mathbf{M}) = \sum_{e} Pr(e|\mathbf{M}) \prod_{a \in \tau} Pr(a|e),$$

where $\mathbf{M}$ is a model, e is an event, τ is a given labeled set of attributes, and a is an attribute in τ.

Based on these probabilities, we rank the labelled sets of attributes to compute the area under the curve (AUC) of the receiver operator curve (ROC)—a curve showing the true positive rate against the false positive rate. This is a standard measure used in information retrieval to show the performance of a binary classifier as the discriminatory threshold is varied. We can see the performance of our model compared to other baselines in Table 6.4. While ProxiModel and ProxiModel-NS outperform the other baseline methods, ProxiModel has marginal improvement over ProxiModel-NS. We will address this difference between our sparse model, and non-sparse model in Section 6.3.8. Also, note that PhraseLDA has lower AUC than ProxiModel, especially in Organization and Person because of using a single distribution for attributes and phrases.

6.3.5 Parameter Studies

There are three main parameters in ProxiModel to control the noise reduction (link minimum support), proximity measures, and sparsity of learning parameters. In the following sections, we show how these parameters affect the model's performance.

	Time	Location	Phrase
+	2011-03-11	HAWAII	tsunami warning
+	2011-03-16	FUKUSHIMA	nuclear power plant
-	2011-03-11	CHERNOBYL	cooling system
-	2011-03-16	TOKYO	spend fuel pool

Positive and negative examples of Base

	Time	Location	Phrase	Org.
+	2011-03-11	SENDAI	relief effort	RED CROSS
+	2011-03-19	TOKYO	radiation level	TEPCO
-	2011-03-12	CHERNOBYL	cooling system	IAEA
-	2011-03-12	LIBYA	sweep away	UN

Positive and negative examples of Base + Organization

	Time	Location	Phrase	Person.
+	2011-03-11	FUKUSHIMA	stay indoors	NAOTO KAN
+	2011-03-17	FUKUSHIMA	storage pool	YUKIO EDANO
-	1979	UKRAINE	radioactive material	NAOTO KAN
-	2011-03-11	SENDAI	fuel rod	BARACK OBAMA

Positive and negative examples of Base + Person

Table 6.3: Examples of human annotated event attributes

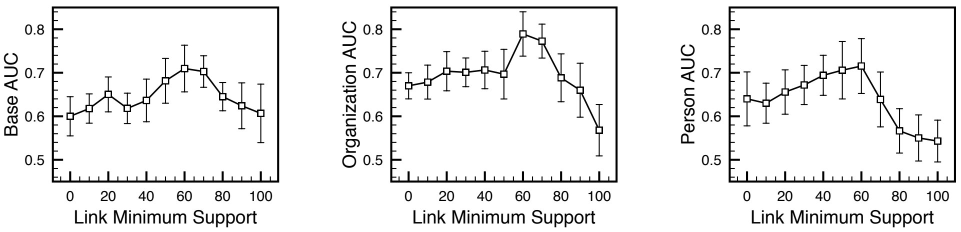
	HISCOVERY	PhraseLDA	ProxiModel-NP	ProxiModel-NS	ProxiModel
Sewol Ferry					
Base	0.5217	**0.7971**	0.6102	**0.8010**	**0.8103**
Org.	0.5190	0.6659	0.5111	**0.6983**	**0.6944**
Person	0.5144	0.6105	0.5308	**0.6385**	**0.6455**
Japan Tsunami					
Base	0.5149	0.6018	0.5212	**0.6854**	**0.6976**
Org.	0.4754	0.5018	0.5594	**0.7648**	**0.7688**
Person	0.6093	0.5334	0.5291	**0.6710**	**0.6948**
Multiple					
Base	0.5928	**0.7139**	0.6272	**0.7310**	**0.7351**
Org.	0.6254	0.6740	0.6170	**0.7564**	**0.7431**
Person	0.5409	0.6688	0.6504	**0.7605**	**0.7660**

Table 6.4: Event retrieval task evaluated using AUC: bold numbers indicate significantly better results than other methods.

6.3.6 Link Minimum Supports

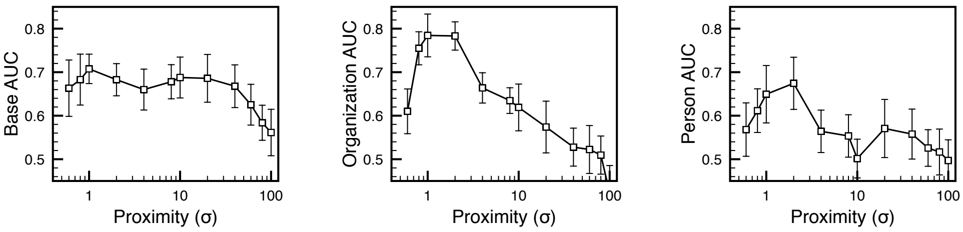
Because ProxiModel leverages data redundancy, it naturally places higher emphasis on larger link-weights. Taking this into consideration, we apply a minimum support to links in order to reduces the number of trivial links and thus enhance the efficiency of the algorithm. In the Japan tsunami dataset, more than 96% of links have less than 1.0 weight. By removing small weight links, we have comparable results in quality, but better efficiency.

In Figure 6.8, we analyze both our performance as a measure of area under curve of ROC and our runtime performance as we vary the link minimum support parameter. We show the performance of ProxiModel in AUC against



(a) AUC for Base (b) AUC for Org. (c) AUC for Person

Figure 6.8: Link Minimum Supports (AUC)



(a) AUC for Base (b) AUC for Org. (c) AUC for Person

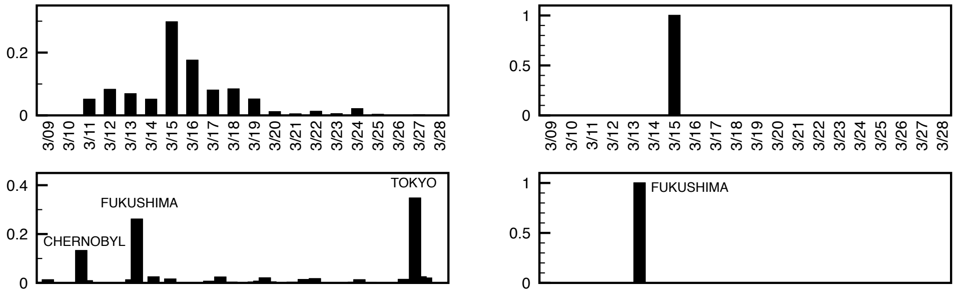
Figure 6.9: Different σ (AUC)

different values of link minimum support, l_{minsup}. When l_{minsup} is too large, the performance is degraded due to the loss of important information. For all our datasets, we set l_{minsup} to 10.

In Figure 6.11(a), as we increase the minimum support, proximity networks become sparser, leading to improved efficiency and better runtimes.

6.3.7 Proximity

In Section 6.3.4, experiments showed ProxiModel outperforming ProxiModel-NP (non-proximity). In this section we vary σ to control our proximity parameter and analyze its effect on retrieval performance. In Figure 6.9, we show the performance of our model in AUC with variants with different proximity parameters(σ) for N_s. We notice peaks around one in the all figures, but we have significant drops for Organization and Person performance when $\sigma > 2$. As we addressed in Section 6.2.3, proximity is related to the information propagation within a document. When σ is large, the proximity network captures long range information propagation. For smaller σ, only near-by information is propagated. Analyzing Figure 6.9, we can see indica-



(a) Non Sparse Distributions

(b) Sparse Distributions

Figure 6.10: Sparsity vs Non-sparsity: Location and Time

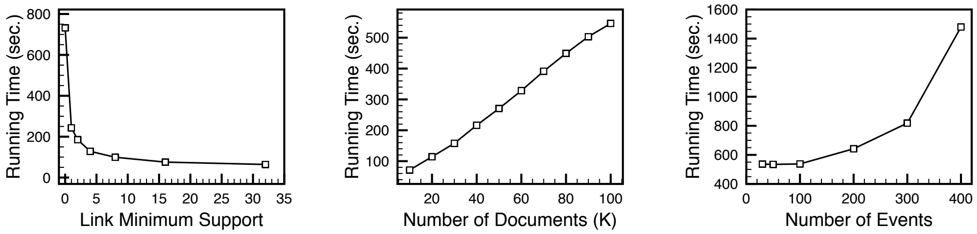
tion that for organizations and persons, information is generally propagated in relatively shorter range when compared to location and time information while enjoying long-range propagation. As such, this motivates setting the proximity parameters for each attribute.

6.3.8 Sparsity

As mentioned previously, ProxiModel demonstrated marginal improvement over ProxiModel-NS, which was shown to not be statistically significant in Table 6.4. While objectively performance is marginal, we observe however that sparsity affects the interpretability of the learned parameters. For example, Figure 6.10 shows the learned parameters – location distribution and time distributions – for the fire explosion that occurred in the Fukushima nuclear power plant on March 15th. Unlike non-sparse models which display many peaks and thus conflicting information, ProxiModel appears sparse displaying single peaks in the location distribution and time distribution. These are significantly more human-interpretable.

6.3.9 Efficiency Analysis

To understand the run-time efficiency of our methodology, we measure the run-time of ProxiModel using our Multiple dataset, which has approximately 100k documents combined from a variety of sources. We measure runtime as we incrementally increase corpus size. Figure 6.11(b) demonstrates empirically run-time is linear in terms of the number of documents. We then vary



(a) Link Min. Support (b) Documents (c) Events

Figure 6.11: Running Time

the number of events parameter and observe run-time performance. From Figure 6.11(c) we can see that runtime is quadratic in relation to number of events. As this parameter is usually small (a small number of events), this is less significant than linearity with respect to corpus size.

6.3.10 Visualization

We use ProxiModel to learn Japan Tsunami events and their connections, and visualize them in Figure 6.1. An event is represented by a circle with a radius proportional to its probability in event distribution θ. Each event is described by a list of top 6 event descriptors from η_z in conjunction with top event attributes (e.g., time, location, organization, persons) from ϕ. For some events, there could be no relevant event attributes for a certain type. When $\sum_i \hat{e}^{s,t}_{i,j,z} < 1$ for a top event attribute j of an event z, the top event attribute is ignored and shown as –. This combination of human-interpretable, multi-word phrases with event attributes help to understand each event.

In addition, links between events are drawn based on φ. Since we impose sparsity on φ, there are only a few non-trivial links between events. Each line width is proportional to its probability in event link distribution φ. The links between events help chain related events together naturally forming an easy-to-interpret branching timeline story. By systematically traversing this event graph, one can naturally construct a storyline of the significant events surrounding the Japan nuclear disaster.

CHAPTER 7

CONCLUSION

I have investigated the topics of schema conversion for constructing information networks from documents.

For schema conversion, I investigate the relationship between topic models and information networks, and demonstrate to use a novel Entity Topic Model (ETM) to build a information networks with documents, words, entities and topics, which can explicitly model the word co-occurrences in pairs of a topic and entity, with smartly designed priors. Having shared asymmetric Dirichlet priors, our model reduces the size of its parameter space while learning a large number of parameters. A Gibbs sampling-based algorithm is proposed to learn the model.

In addition to topic models, I propose a new syntactic feature set of *k-ee* subtrees as nodes in information networks with authors, documents, and discriminative writing styles. To mine *k-ee* subtrees, I developed a direct discriminative *k-ee* subtree mining algorithm via a *branch-and-bound* approach. Our novel algorithm could perform a discriminative score based feature selection procedure to mine discriminative patterns in one step, not iteratively.

I investigate clustering redundant nodes in information networks. A link-based similarity function called SynRank is proposed to capture similarities between nodes in an iterative way. Experiments on real-world datasets have shown the performance of SynRank.

Finally, for building event information networks, I propose a novel event mining framework (ProxiModel) to integrate phrases, named entities, and time expressions to construct then cluster proximity networks to identify these hidden events. A key aspect of the approach involves utilizing proximity of information consistently found in a comparable corpus in order to model and propagate event information.

REFERENCES

[1] Y. Sun, Y. Yu, and J. Han, "Ranking-based clustering of heterogeneous information networks with star network schema," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2009, pp. 797–806.

[2] Y. Sun, J. Han, P. Zhao, Z. Yin, H. Cheng, and T. Wu, "Rankclus: integrating clustering with ranking for heterogeneous information network analysis," in *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology.* ACM, 2009, pp. 565–576.

[3] M. Ji, J. Han, and M. Danilevsky, "Ranking-based classification of heterogeneous information networks," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2011, pp. 1298–1306.

[4] B. Zhao, B. I. Rubinstein, J. Gemmell, and J. Han, "A bayesian approach to discovering truth from conflicting sources for data integration," *Proceedings of the VLDB Endowment*, vol. 5, no. 6, pp. 550–561, 2012.

[5] Y. Sun, J. Han, C. C. Aggarwal, and N. V. Chawla, "When will it happen?: relationship prediction in heterogeneous information networks," in *Proceedings of the fifth ACM international conference on Web search and data mining.* ACM, 2012, pp. 663–672.

[6] D. Davis, R. Lichtenwalter, and N. V. Chawla, "Multi-relational link prediction in heterogeneous information networks," in *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on.* IEEE, 2011, pp. 281–288.

[7] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks," *Proceedings of the VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, 2011.

[8] P. Zhao, J. Han, and Y. Sun, "P-rank: a comprehensive structural similarity measure over information networks," in *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 2009, pp. 553–562.

[9] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, "Dbpedia-a crystallization point for the web of data," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 7, no. 3, pp. 154–165, 2009.

[10] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. ACM, 2008, pp. 1247–1250.

[11] J. Zhu, Z. Nie, X. Liu, B. Zhang, and J.-R. Wen, "Statsnowball: a statistical approach to extracting entity relationships," in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 101–110.

[12] M. J. Cafarella, D. Downey, S. Soderland, and O. Etzioni, "Knowitnow: Fast, scalable information extraction from the web," in *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2005, pp. 563–570.

[13] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka Jr, and T. M. Mitchell, "Toward an architecture for never-ending language learning." in *AAAI*, vol. 5, 2010, p. 3.

[14] T. Weninger, M. Danilevsky, F. Fumarola, J. Hailpern, J. Han, T. J. Johnston, S. Kallumadi, H. Kim, Z. Li, D. McCloskey et al., "Winacs: Construction and analysis of web-based computer science information networks," in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*. ACM, 2011, pp. 1255–1258.

[15] H. Ji and R. Grishman, "Refining event extraction through cross-document inference." in *ACL*, 2008, pp. 254–262.

[16] A. Ritter, O. Etzioni, S. Clark et al., "Open domain event extraction from twitter," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 1104–1112.

[17] S. Riedel and A. McCallum, "Fast and robust joint models for biomedical event extraction," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 1–12.

[18] S.-P. Choi and S.-H. Myaeng, “Simplicity is better: revisiting single kernel ppi extraction,” in *Proceedings of the 23rd International Conference on Computational Linguistics.* Association for Computational Linguistics, 2010, pp. 206–214.

[19] T. Hofmann, “Probabilistic latent semantic indexing,” in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval.* ACM, 1999, pp. 50–57.

[20] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[21] D. Zhang, C. Zhai, and J. Han, “Topic cube: Topic modeling for olap on multidimensional text databases,” in *Proceedings of the 2009 SIAM International Conference on Data Mining.* SIAM, 2009, pp. 1124–1135.

[22] X. Wang and A. McCallum, “Topics over time: a non-markov continuous-time model of topical trends,” in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2006, pp. 424–433.

[23] D. M. Blei and J. D. Lafferty, “Dynamic topic models,” in *Proceedings of the 23rd international conference on Machine learning.* ACM, 2006, pp. 113–120.

[24] S. Sizov, “Geofolk: latent spatial semantics in web 2.0 social media,” in *Proceedings of the third ACM international conference on Web search and data mining.* ACM, 2010, pp. 281–290.

[25] Z. Yin, L. Cao, J. Han, C. Zhai, and T. Huang, “Geographical topic discovery and comparison,” in *Proceedings of the 20th international conference on World wide web.* ACM, 2011, pp. 247–256.

[26] A. Mccallum, “Multi-label text classification with a mixture model trained by em,” in *AAAI ’99 Workshop on Text Learning*, 1999.

[27] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, “The author-topic model for authors and documents,” in *Proceedings of the 20th conference on Uncertainty in artificial intelligence.* AUAI Press, 2004, pp. 487–494.

[28] E. Erosheva, S. Fienberg, and J. Lafferty, “Mixed-membership models of scientific publications,” *Proceedings of the National Academy of Sciences*, vol. 101, no. suppl. 1, pp. 5220–5227, 2004. [Online]. Available: http://www.pnas.org/cgi/content/abstract/101/suppl_1/5220

[29] R. Balasubramanyan and W. W. Cohen, “Block-lda: Jointly modeling entity-annotated text and entity-entity links,” in *Proceedings of the 2011 SIAM International Conference on Data Mining.* SIAM, 2011, pp. 450–461.

[30] H. Deng, J. Han, B. Zhao, Y. Yu, and C. X. Lin, “Probabilistic topic models with biased propagation on heterogeneous information networks,” in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2011, pp. 1271–1279.

[31] H. Tong, C. Faloutsos, and J.-Y. Pan, “Fast random walk with restart and its applications,” 2006.

[32] G. Jeh and J. Widom, “Simrank: a measure of structural-context similarity,” in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2002, pp. 538–543.

[33] X. Yin, J. Han, and S. Y. Philip, “Object distinction: Distinguishing objects with identical names,” in *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on.* IEEE, 2007, pp. 1242–1246.

[34] D. McClosky, M. Surdeanu, and C. D. Manning, “Event extraction as dependency parsing,” in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1.* Association for Computational Linguistics, 2011.

[35] M. Miwa, R. Sætre, J.-D. Kim, and J. Tsujii, “Event extraction with complex event classification using rich features,” *Journal of bioinformatics and computational biology*, vol. 8, no. 01, pp. 131–146, 2010.

[36] C. Aone and M. Ramos-Santacruz, “Rees: a large-scale relation and event extraction system,” in *Proceedings of the sixth conference on Applied natural language processing.* Association for Computational Linguistics, 2000.

[37] Z. Li, B. Wang, M. Li, and W.-Y. Ma, “A probabilistic model for retrospective news event detection,” in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval.* ACM, 2005, pp. 106–113.

[38] M. Naughton, N. Kushmerick, and J. Carthy, “Clustering sentences for discovering events in news articles,” in *Advances in Information Retrieval.* Springer, 2006, pp. 535–538.

[39] M. Naughton, N. Kushmerick, and J. Carthy, "Event extraction from heterogeneous news sources," in *proceedings of the AAAI workshop event extraction and synthesis*, 2006.

[40] H. Sayyadi, A. Sahraei, and H. Abolhassani, "Event detection from news articles," in *Advances in Computer Science and Engineering.* Springer, 2009, pp. 981–984.

[41] W. Lam, H. Meng, K. Wong, and J. Yen, "Using contextual analysis for news event detection," *International Journal of Intelligent Systems*, vol. 16, no. 4, pp. 525–546, 2001.

[42] W. W. Cohen and S. Sarawagi, "Exploiting dictionaries in named entity extraction: combining semi-markov extraction processes and data integration methods," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2004, pp. 89–98.

[43] O. Etzioni, M. Banko, S. Soderland, and D. S. Weld, "Open information extraction from the web," *Communications of the ACM*, vol. 51, no. 12, pp. 68–74, 2008.

[44] C. Zhai and J. Lafferty, "A study of smoothing methods for language models applied to information retrieval," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 2, pp. 179–214, 2004.

[45] H. M. Wallach, D. M. Mimno, and A. McCallum, "Rethinking lda: Why priors matter," in *Advances in neural information processing systems*, 2009, pp. 1973–1981.

[46] P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer, "Dbpedia spotlight: shedding light on the web of documents," in *I-Semantics '11*, 2011, pp. 1–8.

[47] H. M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno, "Evaluation methods for topic models," in *Proceedings of the 26th annual international conference on machine learning.* ACM, 2009, pp. 1105–1112.

[48] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical dirichlet processes," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. pp. 1566–1581, 2006.

[49] R. M. Neal, "Slice sampling," *Annals of Statistics*, vol. 31, pp. 705–767, 2003.

[50] T. C. Mendenhall, "The characteristic curves of composition," *Science*, vol. 11, no. 214, pp. 237–246, 1887.

[51] F. Mosteller and D. L. Wallace, *Inference & Disputed Authorship: The Federalist*. Addison Wesley, 1964.

[52] R. Zheng, J. Li, H. Chen, and Z. Huang, "A framework for authorship identification of online messages: Writing-style features and classification techniques," *Journal of the American Society for Information Science and Technology*, vol. 57, no. 3, pp. 378–393, 2006.

[53] S. Argamon, M. Šarić, and S. S. Stein, "Style mining of electronic messages for multiple authorship discrimination: first results," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2003, pp. 475–480.

[54] S. Burrows, A. L. Uitdenbogerd, and A. Turpin, "Application of information retrieval techniques for source code authorship attribution," in *International Conference on Database Systems for Advanced Applications*. Springer, 2009, pp. 699–713.

[55] J. Diederich, J. Kindermann, E. Leopold, and G. Paass, "Authorship attribution with support vector machines," *Applied Intelligence*, vol. 19, no. 1-2, pp. 109–123, 2003.

[56] Y. Zhao and J. Zobel, "Effective and scalable authorship attribution using function words," in *Asia Information Retrieval Symposium*. Springer, 2005, pp. 174–189.

[57] S. Argamon and S. Levitan, "Measuring the usefulness of function words for authorship attribution," in *ACH/ALLC*, 2005.

[58] A. M. García and J. C. Martín, "Function words in authorship attribution studies," *Literary and Linguistic Computing*, vol. 22, no. 1, pp. 49–66, 2007.

[59] H. Baayen, H. van Halteren, and F. Tweedie, "Outside the cave of shadows: using syntactic annotation to enhance authorship attribution," *Literary and Linguist Computing*, vol. 11, no. 3, pp. 121–132, 1996.

[60] M. Gamon, "Linguistic correlates of style: authorship classification with deep linguistic analysis features," in *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics, 2004, p. 611.

[61] Y. Zhao, J. Zobel, and P. Vines, "Using relative entropy for authorship attribution," in *Asia Information Retrieval Symposium*. Springer, 2006, pp. 92–105.

[62] Y. Zhao and J. Zobel, "Searching with style: authorship attribution in classic literature," in *Proceedings of the thirtieth Australasian conference on Computer science*, 2007.

[63] G. Hirst and O. Feiguina, "Bigrams of syntactic labels for authorship discrimination of short texts," *Literary and Linguistic Computing*, vol. 22, no. 4, pp. 405–417, 2007.

[64] D. Lo, H. Cheng, J. Han, S.-C. Khoo, and C. Sun, "Classification of software behaviors for failure detection: a discriminative pattern mining approach," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2009, pp. 557–566.

[65] H. Cheng, X. Yan, J. Han, and S. Y. Philip, "Direct discriminative pattern mining for effective classification," in *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on.* IEEE, 2008, pp. 169–178.

[66] A. Zimmermann and B. Bringmann, "Ctc-correlating tree patterns for classification," in *Data Mining, Fifth IEEE International Conference on.* IEEE, 2005, pp. 4–pp.

[67] X. Yan, H. Cheng, J. Han, and P. S. Yu, "Mining significant graph patterns by leap search," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data.* ACM, 2008, pp. 433–444.

[68] H. Kim, S. Kim, T. Weninger, J. Han, and T. Abdelzaher, "Ndpmine: Efficiently mining discriminative numerical features for pattern-based classification." Springer, 2010, pp. 35–50.

[69] H. Cheng, X. Yan, J. Han, and C.-W. Hsu, "Discriminative frequent pattern analysis for effective classification," in *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on.* IEEE, 2007, pp. 716–725.

[70] J. Han, J. Pei, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu, "Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth," in *proceedings of the 17th international conference on data engineering*, 2001, pp. 215–224.

[71] L. Zou, Y. Lu, H. Zhang, R. Hu, and C. Zhou, "Mining frequent induced subtrees by prefix-tree-projected pattern growth," in *Web-Age Information Management Workshops, 2006. WAIM'06. Seventh International Conference on.* IEEE, 2006, pp. 18–18.

[72] S. Nijssen, T. Guns, and L. De Raedt, "Correlated itemset mining in roc space: a constraint programming approach," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2009, pp. 647–656.

[73] W. Fan, K. Zhang, H. Cheng, J. Gao, X. Yan, J. Han, P. Yu, and O. Verscheure, "Direct mining of discriminative and essential frequent patterns via model-based search tree," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2008, pp. 230–238.

[74] D. Harman, "Overview of the second text retrieval conference (trec-2)," *Information Processing & Management*, vol. 31, no. 3, pp. 271–289, 1995.

[75] R. Mitton, "Spelling checkers,spelling correctors and the misspellings of poor spellers," *Information Processing and Management*, vol. 23, no. 5, pp. 495–505, 1987.

[76] C. Manning and P. Raghavan, *Introduction to information retrieval.* Cambridge university press, vol. 1.

[77] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

[78] D. Wang, S. Zhu, T. Li, and Y. Gong, "Multi-document summarization using sentence-based topic models," in *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers.* Association for Computational Linguistics, 2009, pp. 297–300.

[79] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002.

[80] R. Baeza-Yates, B. Ribeiro-Neto et al., *Modern information retrieval.* ACM press, 1999, vol. 463.

[81] A. Moschitti and R. Basili, "Complex linguistic features for text classification: A comprehensive study," *Advances in Information Retrieval*, pp. 181–196, 2004.

[82] M. Mitra, C. Buckley, A. Singhal, and C. Cardie, "An analysis of statistical and syntactic phrases," in *Computer-Assisted Information Searching on Internet.* LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, 1997, pp. 200–214.

[83] Y. Miao, V. Kešelj, and E. Milios, “Document clustering using character n-grams: a comparative evaluation with term-based and word-based clustering,” in *Proceedings of the 14th ACM international conference on Information and knowledge management*. ACM, 2005, pp. 357–358.

[84] R. Bekkerman, R. El-Yaniv, N. Tishby, and Y. Winter, “Distributional word clusters vs. words for text categorization,” *Journal of Machine Learning Research*, vol. 3, no. Mar, pp. 1183–1208, 2003.

[85] R. Barzilay and N. Elhadad, “Sentence alignment for monolingual comparable corpora,” in *Proceedings of the 2003 conference on Empirical methods in natural language processing*. Association for Computational Linguistics, 2003, pp. 25–32.

[86] A. Ibrahim, B. Katz, and J. Lin, “Extracting structural paraphrases from aligned monolingual corpora,” in *Proceedings of the second international workshop on Paraphrasing-Volume 16*. Association for Computational Linguistics, 2003, pp. 57–64.

[87] L.-A. Tang, Q. Gu, X. Yu, J. Han, T. L. Porta, A. Leung, T. Abdelzaher, and L. Kaplan, “Intrumine: Mining intruders in untrustworthy data of cyber-physical systems,” in *Proceedings of the 2012 SIAM International Conference on Data Mining*. SIAM, 2012, pp. 600–611.

[88] D. Gildea and D. Jurafsky, “Automatic labeling of semantic roles,” *Computational Linguistics*, vol. 28, no. 3, pp. 245–288, 2002.

[89] J. Christensen, S. Soderland, O. Etzioni et al., “Semantic role labeling for open information extraction,” in *Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading*. Association for Computational Linguistics, 2010, pp. 52–60.

[90] D. Shen and M. Lapata, “Using semantic roles to improve question answering.” in *EMNLP-CoNLL*, 2007, pp. 12–21.

[91] C. Fillmore, “Frame semantics and the nature of language,” *Annals of the New York Academy of Sciences*, vol. 280, no. 1, pp. 20–32, 1976.

[92] M. Palmer, D. Gildea, and P. Kingsbury, “The proposition bank: An annotated corpus of semantic roles,” *Computational Linguistics*, vol. 31, no. 1, pp. 71–106, 2005.

[93] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, “Natural language processing (almost) from scratch,” *Journal of Machine Learning Research*, vol. 12, no. Aug, pp. 2493–2537, 2011.

[94] C. Fellbaum, "Wordnet," *Theory and Applications of Ontology: Computer Applications*, pp. 231–243, 2010.

[95] R. Mihalcea, C. Corley, and C. Strapparava, "Corpus-based and knowledge-based measures of text semantic similarity," in *AAAI*, 2006.

[96] C. Leacock and M. Chodorow, "Combining local context and wordnet similarity for word sense identification," *WordNet: An electronic lexical database*, 1998.

[97] M. Ester, H.-P. Kriegel, J. Sander, X. Xu et al., "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[98] D. Dueck and B. J. Frey, "Non-metric affinity propagation for unsupervised image categorization," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on.* IEEE, 2007, pp. 1–8.

[99] D. Lin and X. Wu, "Phrase clustering for discriminative learning," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2.* Association for Computational Linguistics, 2009, pp. 1030–1038.

[100] X. He, D. Cai, H. Liu, and W.-Y. Ma, "Locality preserving indexing for document representation," in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval.* ACM, 2004, pp. 96–103.

[101] Q. Mei, C. Liu, H. Su, and C. Zhai, "A probabilistic approach to spatiotemporal theme pattern mining on weblogs," in *Proceedings of the 15th international conference on World Wide Web.* ACM, 2006, pp. 533–542.

[102] C. Wang, M. Danilevsky, N. Desai, Y. Zhang, P. Nguyen, T. Taula, and J. Han, "A phrase mining framework for recursive construction of a topical hierarchy," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2013, pp. 437–445.

[103] A. El-Kishky, Y. Song, C. Wang, C. Voss, and J. Han, "Scalable topical phrase mining from text corpora," *Proceedings of the VLDB Endowment*, vol. 8, no. 3, 2014.

[104] J. Leskovec, L. Backstrom, and J. Kleinberg, "Meme-tracking and the dynamics of the news cycle," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2009, pp. 497–506.

[105] C. Suen, S. Huang, C. Eksombatchai, R. Sosic, and J. Leskovec, "Nifty: a system for large scale information flow tracking and clustering," in *Proceedings of the 22nd international conference on World Wide Web.* ACM, 2013.

[106] R. Li, K. H. Lei, R. Khadiwala, and K.-C. Chang, "Tedas: A twitter-based event detection and analysis system," in *Data Engineering (ICDE), 2012 IEEE 28th International Conference on.* IEEE, 2012.

[107] E. J. Wagner, J. Liu, L. Birnbaum, and K. D. Forbus, "Rich interfaces for reading news on the web," in *Proceedings of the 14th international conference on Intelligent user interfaces.* ACM, 2009, pp. 27–36.

[108] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors," in *Proceedings of the 19th international conference on World wide web.* ACM, 2010, pp. 851–860.

[109] J. Pustejovsky, J. M. Castano, R. Ingria, R. Sauri, R. J. Gaizauskas, A. Setzer, G. Katz, and D. R. Radev, "Timeml: Robust specification of event and temporal expressions in text." *New directions in question answering*, vol. 3, pp. 28–34, 2003.

[110] J. Liu, J. Shang, C. Wang, X. Ren, and J. Han, "Mining quality phrases from massive text corpora," in *Proc. 2015 ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'15)*, 2015.

[111] M. Baker, "Corpora in translation studies: An overview and some suggestions for future research," *Target*, pp. 223–243, 1995.

[112] J. Washtell, "Co-dispersion: A windowless approach to lexical association," in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics.* Association for Computational Linguistics, 2009.

[113] J. Washtell and K. Markert, "A comparison of windowless and window-based computational association measures as predictors of syntagmatic human associations," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2.* Association for Computational Linguistics, 2009.

[114] Y. Lv and C. Zhai, "Positional language models for information retrieval," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval.* ACM, 2009, pp. 299–306.

[115] J. Zhao, J. X. Huang, and B. He, "Crter: using cross terms to enhance probabilistic information retrieval," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval.* ACM, 2011, pp. 155–164.

[116] K. Balog, L. Azzopardi, and M. de Rijke, "A language modeling framework for expert finding," *Information Processing & Management*, vol. 45, no. 1, pp. 1–19, 2009.

[117] D. Petkova and W. B. Croft, "Proximity-based document representation for named entity retrieval," in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management.* ACM, 2007, pp. 731–740.

[118] R. J. Serfling, "Some elementary results on poisson approximation in a sequence of bernoulli trials," *Siam review*, vol. 20, no. 3, pp. 567–579, 1978.

[119] M. Shashanka, B. Raj, and P. Smaragdis, "Sparse overcomplete latent variable decomposition of counts data," in *Advances in neural information processing systems*, 2008, pp. 1313–1320.

[120] G. Schwarz et al., "Estimating the dimension of a model," *The annals of statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[121] J. L. Fleiss, "Measuring nominal scale agreement among many raters." *Psychological bulletin*, p. 378, 1971.