Spring 2011

Topic modeling and applications in Web 2.0

Ha Thuc Viet
University of Iowa

TOPIC MODELING AND APPLICATIONS IN WEB 2.0

by

Viet Ha Thuc

An Abstract

Of a thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy
degree in Computer Science
in the Graduate College of
The University of Iowa

May 2011

Thesis Supervisor: Professor Padmini Srinivasan

ABSTRACT

Along with the exponential growth of text data on the Web, particularly of the user-generated content, comes an increasing need for hierarchically organizing documents, retrieving documents accurately, and discovering evolutionary trends of various popular topics from the data. However, all of these are challenging due to the diversity, heterogeneity, noisiness and time-sensitivity of Web 2.0 data. Motivated by this, we tackle the challenges at a fundamental level, by proposing a novel topic modeling method with ontological guidance. It may be used to discover topic language models formalizing various terms relevant to given topics using the Web data. The topic model takes into account both the ontological relationships amongst the topics defined in a topic taxonomy and also word co-occurrence patterns in the data to automatically identify the portions in the data relevant to the topics. Then, it estimates language models for these topics from these relevant portions. At an application level, we use the topic model to propose novel approaches for three different tasks, namely hierarchical text classification without labeled data, information retrieval with pseudo-relevance feedback, and discovering topic evolutionary trends. Our classification experiment on the IPTC (International Press and Telecommunications Council) taxonomy, containing more than 1100 topics, shows that our approach achieves a performance of 67% in terms of the hierarchical version of the F-1 measure, without using any labeled data. Our retrieval experiments on five benchmark datasets show that compared to baseline retrieval (without pseudo-relevance feedback), our approach improves on average 39% in terms of mean average precision. Finally, for the last task, using blog data, our approach discovers meaningful insights on how the crowd responds to various news topics such as the language used to discuss each topic, how this language drifts over time, and when the crowd's focus on a topic increases, reaches a peak, and declines.

Abstract Approved: ______________________________

Thesis Supervisor

____

Title and Department

____

Date

TOPIC MODELING AND APPLICATIONS IN WEB 2.0

by

Viet Ha Thuc

A thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy
degree in Computer Science
in the Graduate College of
The University of Iowa

May 2011

Thesis Supervisor: Professor Padmini Srinivasan

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

PH.D. THESIS

This is to certify that the Ph.D. thesis of

Viet Ha Thuc

has been approved by the Examining Committee for the thesis requirement for the Doctor of Philosophy degree in Computer Science at the May 2011 graduation.

Thesis Committee: ______________________________
Padmini Srinivasan, Thesis Supervisor

____
Alberto Maria Segre

____
James Cremer

____
Kasturi Varadarajan

____
Nick Street

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor, Padmini Srinivasan, for all of the support since my very first day in Iowa. Padmini has given me a perfect combination of technical advice and research freedom during the past five years. I am grateful to all other committee members, Alberto Segre, Jim Cremer, Kasturi Varadarajan and Nick Street, who have given me a lot of help and useful comments on my research. In particular, I would also like to acknowledge Jim, who recommended me to the PhD program here.

I am very glad to have chances to work with many great peer students and researchers, Yelena Mejova, Christopher Harris, Sanmitra Bhattacharya, Hung Viet Tran, Brian Almquist (Text Mining and Retrieval Group, The University of Iowa), Jean-Michel Renders and Nicola Cancedda (Xerox Research Centre Europe). I have learnt a lot from them. I thank all of them. Especially, I would like to thank Yelena for building the retrieval systems and her collaboration on the research on pseudo-relevance feedback and topic tracking, thank Christopher for his help on building the event taxonomy and his judgement on event trends, and thank Jean-Michel for his collaboration on the research on hierarchical text classification.

Last but not least, I would like to thank my family for their encouragement and tremendous support. Without them, it would have been impossible for me to finish this five-year long journey.

ABSTRACT

Along with the exponential growth of text data on the Web, particularly of the user-generated content, comes an increasing need for hierarchically organizing documents, retrieving documents accurately, and discovering evolutionary trends of various popular topics from the data. However, all of these are challenging due to the diversity, heterogeneity, noisiness and time-sensitivity of Web 2.0 data. Motivated by this, we tackle the challenges at a fundamental level, by proposing a novel topic modeling method with ontological guidance. It may be used to discover topic language models formalizing various terms relevant to given topics using the Web data. The topic model takes into account both the ontological relationships amongst the topics defined in a topic taxonomy and also word co-occurrence patterns in the data to automatically identify the portions in the data relevant to the topics. Then, it estimates language models for these topics from these relevant portions. At an application level, we use the topic model to propose novel approaches for three different tasks, namely hierarchical text classification without labeled data, information retrieval with pseudo-relevance feedback, and discovering topic evolutionary trends. Our classification experiment on the IPTC (International Press and Telecommunications Council) taxonomy, containing more than 1100 topics, shows that our approach achieves a performance of 67% in terms of the hierarchical version of the F-1 measure, without using any labeled data. Our retrieval experiments on five benchmark datasets show that compared to baseline retrieval (without pseudo-relevance feedback), our approach improves on average 39% in terms of mean average precision. Finally, for the last task, using blog data, our approach discovers meaningful insights on how the crowd responds to various news topics such as the language used to discuss each topic, how this language drifts over time, and when the crowd's focus on a topic increases, reaches a peak, and declines.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

CHAPTER 1
INTRODUCTION

It is clear that every day, we are becoming increasingly enmeshed in the Web. Millions of individuals are communicating with each other using essentially a few clicks of a mouse. These discourses, supported by a growing number of online media - forums, Wikipedia, Twitter, Facebook and blogs - are engaging millions of individuals globally. Within the political and social realm, a recent Pew project reports that close to a fifth of US Internet users have posted online or used a social networking site for civic or political engagement [75]. Another Pew study found that 55% of the adult US population went online in 2008 in order to get involved in the political process or to seek information about the last US election [76].

Among data components on the Web, user-generated data or social data has grown very quickly and become a key component. The movement away from static webpages to user-generated and shareable content and social networking has been widely recognized as the second generation of the World Wide Web, commonly referred to as Web 2.0[1]. A key characteristic of social data is that it is generated by a large number of people from different cultures, locations, age groups, religions, income categories etc. This is in contrast to the case of traditional media such as newswire generated by a small group of journalists. Because of its popularity and diversity, social data is an excellent source for understanding the perspectives of large

[1] en.wiktionary.org/wiki/Web_2.0

groups of people, i.e., crowds, on a variety of news topics such as iPhone 4 release, US presidential elections and healthcare reform as expressed on the Web. Moreover, another interesting aspects of social media such as blog space is that blogging may itself influence the events that are being discussed. The fact that bloggers, at least elite bloggers, influence mainstream media is well acknowledged [28]. Bloggers have first-mover advantages in offering opinions, have important “local” knowledge, and incur low cost in publishing their immediate reactions in real-time. Because of the rapidity with which bloggers can operate, they can have agenda setting and framing effects on issues presented in the mainstream media. For example, it took only 5 days of intense blogging before mainstream media paid attention to Trent Lott’s statements at Storm Thurmond’s 100th birthday party. Lott resigned in the end [83]. Without a doubt, social media is an increasingly popular and important information source.

Due to the increasing importance and growing size of the data available on the Web, it has been more and more important to effectively manage and retrieve the data on the Web and discover important knowledge from the data. Given these broad motivations, this thesis focuses on three key problems including hierarchically classifying Web documents, retrieving Web documents, and discovering topic evolutionary trends reflected on the Web. The first problem allows effectively managing a huge number of documents. Moreover, it is a crucial step of many Web applications such as vertical search and information extraction on the Web. Document retrieval is obviously one of the most important applications for tackling the enormous size of the Web data. Finally, discovering topic evolutionary trends allows understanding the crowds' perspectives on various news topics. Specifically, this would provide meaningful insights on how the crowd responds to the topics such as the language used to discuss each topic, how this language drifts over time, and when the crowds focus on a topic increases, reaches a peak, and declines.

However, all of the problems are challenging because of the unique features of Web 2.0 data, compared to traditional data such as newswire, academic writing collections and enterprise corpora. First, social data is informal, diverse and user-centric [41]. So it is very likely that different individuals make different word (and syntax) choices while discussing the same topics. Second, topics of interest are often discussed sparsely relative to the total body of posts made on a given date. Moreover, each post may have several different themes. It is crucial to exclude the "noisy" portions of social data. The third issue is that of time. People's focus on topics can be very time-sensitive. A topic may attract much attention on one day and not on the next. Moreover the language people use to discuss a topic could drift dynamically over time. The final issue is one of scalability. Social data sets are often very large, so it is extremely expensive to scan over them multiple times. Solutions to problems such as modelling and tracking topics using a single pass though the data are therefore important.

To overcome these issues, in this thesis, we tackle the challenges of working with social media at a fundamental level by proposing a novel topic modeling method with ontological guidance. The method may be used to discover topic language models formalizing various terms relevant to given topics using the Web data. The topic model takes into account both the ontological relationships amongst the topics defined in a topic taxonomy and also word co-occurrence patterns in the data. These are used to automatically identify the portions in the data that are relevant to the topics. Then, it estimates language models for these topics from these relevant portions. So, the approach is robust to noise in the data. Moreover, our topic model could be adapted dynamically to reflect the evolution of the crowds's discourse on various topics.

At an application level, we use the topic model to propose novel approaches for all of the three different tasks, including hierarchical text classification without labeled data, information retrieval with pseudo-relevance feedback, and discovering topic evolutionary trends. Our classification experiment on the IPTC (International Press and Telecommunications Council) taxonomy, containing more than 1100 topics, shows that our approach achieves a performance of 67% in terms of the hierarchical version of the F-1 measure, without using any labeled data, compared to performances of 48% and 59% of naive Bayes and hierarchical naive Bayes. Our retrieval experiments on five benchmark datasets show that compared to baseline retrieval (without pseudo-relevance feedback), our approach improves on average 39% in terms of mean average precision. Our approach also achieves significantly better results on these datasets compared to relevance-based language models, a popular pseudo-relevance approach. Finally, for the last task, using blog data, our approach discovers meaningful insights on how the crowd responds to various news topics such as the language used to discuss each topic, how this language drifts over time, and when the crowd's focus on a topic increases, reaches a peak, and declines.

The organization of the thesis is as follows. In Chapter 2, we review probabilistic topic modelling, a framework on which we build our models and also discuss the limitations of this framework when applied to extracting language models for news topics in Web 2.0 data. In Chapter 3, we introduce our proposed hierarchical topic models with ontological guidance to overcome these limitations. Chapter 4 presents how the extracted language models can be used to retrieve documents relevant to the corresponding topics, and we also show results on retrieval experiments. Chapter 5 describes a hierarchical text classification approach exploiting the language models to categorize documents into the topics and events without using any human labelled data for training. In Chapter 6, we present how the hierarchical topic model is used for discovering topic evolutionary trends. Finally, in Chapter 7, we conclude our research.

CHAPTER 2
FOUNDATION AND LITERATURE REVIEW

This section presents an overview of the theoretical background on probabilistic topic models, which is closely related to our fundamental model presented in Chapter 3. Then, we detail the limitations of this previous work. The work related to the specific problems we address are presented at the appropriate points in Chapters 4-6.

2.1 Notation

A *Vocabulary* (dictionary) V is a set of W distinct *words* (*terms*): $V = \{word_1, word_2 \ldots word_W\}$. A *token* is a specific occurrence of one of the W words in a *document*. Document d is a sequence of N_d tokens, where N_d denotes the length of d. A corpus $C = \{(w_1, d_1), (w_2, d_2) \ldots (w_N, d_N)\}$, where $N = \sum N_d$, w_i and d_i are the word index and document index of the i^{th} token. A *topic* z is represented by a multinomial distribution over the vocabulary denoted by Φ_z, where $\Phi_{z,w} = p(w|z), \forall w \in W$. These distributions are also referred as topic *language models* that indicate the language used to converse about the corresponding topics. Each document d in the corpus is generated by a mixture of multiple topics, and the mixture is denoted by Θ_d, where $\Theta_{d,z} = p(z|d), \forall z$.

2.2 Topic modeling

Probabilistic topic model was first introduced by Hofmann [44]. The probabilistic topic model is based on the idea that documents are generated by mixtures of topics, where a topic is, as mentioned before, a multinomial distribution over words. One limitation of Hofmann's model is that it is not clear how the mixing proportions for topics in a document are generated. To overcome this limitation, Blei et al. [9] propose Latent Dirichlet Allocation (LDA). In LDA, the topic proportion of every document is a K-dimensional hidden variable randomly drawn from the same Dirichlet distribution, where K is the number of topics. Thus, generative semantics of LDA are complete, and LDA is acknowledged as the most popular approach for building topic models [9, 33, 89, 13, 70, 59, 18]

LDA is a generative model for documents: it specifies a simple probabilistic process for generating documents. The process is as follows. First, one chooses a set of multinomial distributions for all topics. Then, to generate a new document, one chooses a distribution over topics (i.e. topic mixture). After that, for each token in the document, one picks a topic randomly according to the distribution, and draws a word from the multinomial distribution of that topic. Given this assumption on how documents are generated, statistical inference techniques could be used to invert the process, i.e. to infer the set of topic models, the topic mixtures for all documents, and the topic generating each token.

LDA takes a collection of documents and a number K as input and infers a set of K topics mentioned in the collection. A topic is intuitively a cluster of

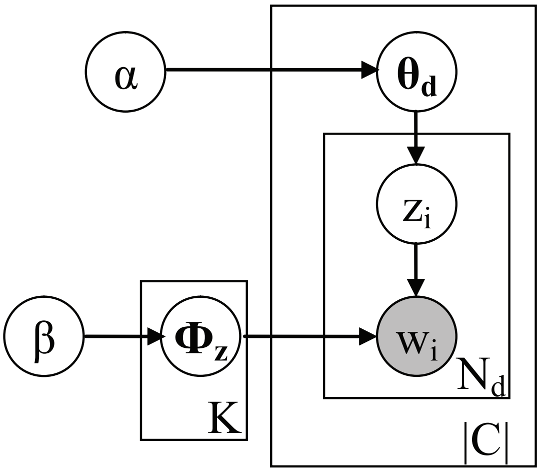


Figure 2.1: Standard Probabilistic Topic Models

tokens that tend to co-occur in the same subset of documents. Formally, LDA is a generative model describing how documents in a corpus are generated by K latent topics. Tokens generated by the same latent topic form the representation for that topic. The generative process is as follows:

1. Pick a multinomial distribution Φ_z over words for each topic z in topic set $\{1 \ldots K\}$ from a W-dimensional Dirichlet distribution with parameter β, $W\text{-}Dir(\beta)$.

2. For each document d in the corpus:

2.1 Pick a multinomial Θ_d for topic mixing proportion from K-dimensional distribution with parameter α, $K\text{-}Dir(\alpha)$.

2.2 For each token in the document d:

2.2.1 Pick a topic z from the distribution Θ_d

2.2.2 Pick a word w from the distribution Φ_z

In the generative process described above, α and β are hyper-parameters of Dirichlet distributions. In most of the previous work, values of α and β are pre-defined constants. For example, α is set to $50/K$ and β is set to 0.01 in [9, 33]. The graphical model for LDA is described in Fig. 2.1 by plate notation. The numbers in the right-lower corner of the plates (boxes) indicate the number of repetitions of the corresponding plates. The shaded circles represent observed variables, and un-shaded circles represent latent variables.

Given a corpus C (observed variables) and a value for K (a pre-specified number of latent topics), an inference algorithm based on Gibbs sampling or Expectation Maximization could be used to infer which latent topics generate which tokens (i.e., latent variables z_i for each token), the topic mixture proportions for each document (i.e., latent variables Θ_d), and the language model for each topic (i.e., latent variables Φ_z)

More recently, at a fundamental level, LDA has been extended in several ways. Rosen-Zvi et al. [70] propose the idea of incorporating authorship information in the topic modeling process. In [5], the authors propose a way to incorporate domain knowledge in LDA. In [11, 52, 69, 68], the authors extend the standard LDA by using document label information. Mimno et al. [64] propose the idea of exploiting various features such as authorship, date, venues etc. to determine hyper-parameters of Dirichlet distributions. Similarly, Zhu et al. [99] propose a conditional model to exploit various word co-occurrence relationships such as co-occurrence in sentences, in paragraphs etc. instead of just in documents to infer topic language models. To improve efficiency of LDA, parallel inference algorithms are proposed in [93, 6, 66, 77], and online inference algorithm is proposed in [43]. In another direction, LDA has also been extended for mining topics from multilingual collections [12, 47, 65, 67, 97]. Finally, instead of using bag-of-word assumption, Wallach [82] proposes a model for topic modeling that takes word order into account.

At an application level, LDA and its variants have been applied in many applications such as finding scientific topics [33], E-community discovery [98], mixed membership analysis [27], representing document language model for ad-hoc retrieval [89], resolving word ambiguity [13, 14], modeling citation influences [21], modeling author influences [63, 30] and matching papers with reviewers [62].

2.3 Hierarchical Topic Modeling

Blei et al. [8] and Mimno et al. [61] extend the “flat” topic models into hierarchical versions for extracting hierarchies of topics from text collections. Given a parameter L indicating the depth of the hierarchy, each document is assumed to be generated by a mixture of L topics on a path from the root to a leaf. To generate a document, one chooses a path from the root to a leaf. Then one draws a distribution over the topics on the path (i.e. topic mixture). After that, for each token in the document, one picks a topic randomly according to the distribution, and draws a word from the multinomial distribution of that topic. The tree structure and the random L-level path are generated by a random process called *nested Chinese restaurant process* [8].

More formally, the generative process for generating a document d in this hierarchical approach is described as follows:

1. Pick a path from the root to a leaf.

2. Pick a multinomial Θ_d for topic mixing proportion from L-dimensional distribution with parameter α, $L\text{-}Dir(\alpha)$.

2.2 For each token in the document d:

2.2.1 Pick a topic z from the distribution Θ_d

2.2.2 Pick a word w from the distribution Φ_z

Given the generation process for documents and an observable document collection, statistical inference techniques like Gibbs sampling could be used to invert the process, i.e. to infer the topic hierarchy.

2.4 Topic Tracking

LDA can also be extended to temporally track topic language models in text streams [60, 87, 3, 10, 32, 88, 85]. One way is that after LDA is applied to the whole text stream to discover topic language models, we may use an “intensity” measure to compute topic intensity over a sliding window of time. This intensity measure is based on the number of tokens in the window of time inferred by the LDA model to be generated by the corresponding topic [87]. Another approach is to first divide the text stream into segments by time, and then to apply LDA independently within each of the segments. This leads to the discovery of K topic language models in each time segment. Then, models across time segments are aligned by some heuristics [60].

Each final aligned model represents a topic's evolution. This is done for each of the discovered language models.

In order to track the drifts of topic language models over time, recently Al-Sumait et al. [3] propose an online LDA. Their method divides the whole text stream into temporal chunks (segments) and extracts topic language models in each chunk by using the results in previous chunk as priors.

Topic tracking task defined by Topic Detection and Tracking (TDT) organizers [2, 1] is also to track topics or events. In this task, each *given* topic or event is specified by its relevant documents. Given such input, the goal is to identify which unseen documents in the text stream belong to the topics or events. Solutions offered for this task typically consider documents as either wholly relevant or non-relevant to a particular event/topic. This is in contrast with LDA-based tracking framework, which is, as described earlier, able to extract portions in documents relevant to given topics (or events) and uses these portions to discover the temporal trends of these topics.

2.5 Discussion

A common problem with the previous approaches reviewed here is that the topic language models discovered by LDA or hierarchical LDA are synthetic. The topic language models discovered by LDA or hierarchical LDA are actually clusters of words that tend to co-occur in a subset of documents. So, there is no guarantee that the topic language models would correspond to actual topics in the real world or the topics that a user may have in mind.

LDA also assumes that *each* token in *every* document is generated by one of the K topic language models, whereas many documents (or their portions) may have nothing to do with any of the K topics. For instance, a blog post could be about a personal story and contain little or nothing about the K most common topics discussed in the blog collection. So, learning topic language models from all tokens in all documents could cause these language models to over-fit.

The aspects that define the limitations in LDA are the ones that motivate many interesting real-world applications. Tracking specific topics that a particular user is interested in is one example. Another example is document classification, where the labels (topics) are given upfront. These applications cannot be solved using LDA or hierarchical LDA directly. To overcome these limitations, we propose a novel framework for extracting the crowd's language used to discuss topics. This model is presented in the next chapter.

CHAPTER 3
HIERARCHICAL TOPIC MODELLING WITH ONTOLOGICAL GUIDANCE

3.1 Problem Statement and Motivations

As mentioned in the previous chapter, the topic language models extracted by Latent Dirichlet Allocation are synthetic and might not correspond to the actual topics[1] in the real world. To overcome the issue, we propose a novel model that takes into account a taxonomy a.k.a an ontological knowledge base[2]. The taxonomy succinctly represents the knowledge on topics existing in the real world or about topics a particular user has in mind. Examples of such taxonomy are the event hierarchy in Figure 6.3 (Chapter 6) and International Press Telecommunications Council (IPTC) topic hierarchy described in Section 5.3.1. Such ontological knowledge includes the titles of nodes in the taxonomy such as "sport" and "soccer", and their parent-child relationships. The ontological knowledge is used to guide the modelling and discovery process in our approach. As a result, each of the extracted language models will correspond to a topic or event defined in the taxonomy, and therefore correspond to a topic in the real world.

The framework of our approach takes a taxonomy and a social Web stream as input. Each node in the taxonomy is a simple label of an actual topic or event

[1] In this thesis, we use the term topic to refer either to a concept such as political election or to an event happening at a specific time and place such as the 2008 US presidential election

[2] The work in this chapter is initially proposed in [39](Ha-Thuc et al., CIKM'08 PhD Workshop) and a refined version is published in [37] (Ha-Thuc et al. WSDM'11)

in the form of a simple title, such as “2008 US election”. The taxonomy could be entered by a user, similar to the way in which a user enters queries to commercial tracking systems such as Google Insight and Blog Pulse (which will be described in more details in Chapter 6), or the taxonomy could be extracted from a source such as Wikipedia or International Press Telecommunications Council consortium. Our modeling framework outputs language models for the nodes in the taxonomy.

3.2 Overall Framework

The overall framework is described in Figure 3.1. The social Web stream of interest is crawled and harvested. Then, the collection is parsed and indexed by a search engine.

Given a taxonomy, where each node is a simple title as mentioned earlier, first we exploit the hierarchy to construct an enriched and context-aware query for each category (i.e. each node in the hierarchy). Basically, for each category, we use its ancestors to define a context for the category and (partially) resolve possible ambiguities. We also exploit its children as special cases to enrich the query. This query is then submitted to a search engine. We take the top retrieved documents and assume they are relevant to the category. These documents are referred as pseudo-relevant documents. We present this phase in more detail in Section 3.3.

Second, given the training documents (i.e. pseudo-relevant documents) for all categories, we extract a language model (multinomial distribution over words) for each of these categories. Note that in the previous phase, even though we use

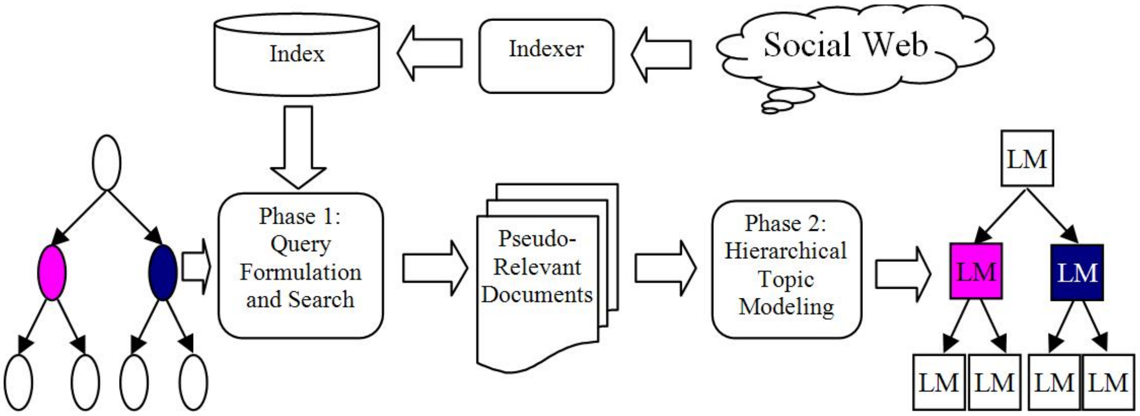

Index
Indexer
Social Web
Phase 1: Query Formulation and Search
Pseudo-Relevant Documents
Phase 2: Hierarchical Topic Modeling
LM
LM
LM
LM
LM
LM
LM

Figure 3.1: An Overall Framework

enriched and context-aware queries, the retrieved documents are still very likely to contain noise. Therefore, the challenge in the second phase is to exclude noise (non-relevant parts) and identify really relevant parts in training documents. Then, the category language models are estimated from the relevant parts only. To achieve this, we propose a hierarchical topic model extracting a language model for each category by using not only its training documents but also its position in the hierarchy and relationships with other categories. The details of this second phase are described in Section 3.4.

3.3 Phase 1: Retrieving Training Documents

For each category, we construct a query, and then submit the query to the search engine. We take the top k retrieved documents and temporarily consider these documents as relevant examples of the category.

When constructing the queries, we exploit the hierarchical relationships be-

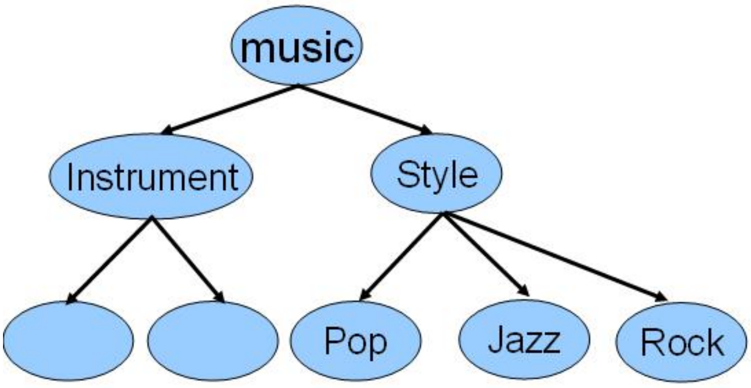


Figure 3.2: Query Construction Example

tween the categories. For each category, for instance "*security*" [3] (*economic sector/computing and information technology/security*), the upper level categories (e.g. "*computing and information technology*") specify the global context of the category; as such, they are useful to disambiguate with respect to categories having similar (or even the same) titles, for example "*securities*" [4] (*market and exchange/securities*).

Additionally, we find that given a category, for instance "*style*" (Figure 3.2), its sub-categories (i.e. the children in the hierarchy) such as "*Pop*", "*Jazz*"and "*Rock*" are also useful. The sub-categories are special cases of the parent category. So, they could be used to enrich the corresponding query. Given the two observations above, we construct a query for each category by combining the title of itself with the ones of its parent and children. For example, the query of category "*style*" in Figure 3.2 is "*music style (Pop OR Jazz OR Rock)*"

[3] This corresponds to the code=20000229 of the IPTC taxonomy used in our experiments.

[4] IPTC code=20000394

3.4 Phase 2: Extracting Language Models

Given training sets for categories in the hierarchy (pseudo-relevant documents obtained in the previous phase), we will estimate a language model $p(word|category)$ for each of these categories. The challenge in estimating the language models from training documents is that these training documents could also contain portions that are non-relevant to the category. For example, a training document about a "show of a rock band in London" for category "*rock and roll music*" could also contain terms relevant to more general categories such as "*music*" and "*art and entertainment*". It could also contain terms specific to the local context of the document such as *London* or proper names of the bar as well as the band members. Not removing the general terms could make the language model for "*rock music*" highly overlap with the language models for its sibling categories such as "*folk music*" or "*country music*". On the other hand, not excluding all document-specific terms could make the language model for the category over-fit its training set.

It is worth noting that enriching the search query for each category by taking information of its parent and children into account as described in previous section is necessary to reduce ambiguities. On the other hand, this enrichment makes the queries and consequently the training sets of linked categories highly overlapping. So, in the phase of extracting language models from these documents, it is crucial to exclude general terms, especially for low-level categories.

We address this challenge by proposing a hierarchical topic model with ontological guidance for extracting these language models. The approach takes into account the fact that although a document d may be relevant to a category c in the hierarchy, it could still have non-relevant portions. Specifically, a training document d is hypothesized to be generated by a mixture of multiple topics: the category c itself, its ascendant categories explaining general terms (including a "background" topic at the root of the hierarchy representing the general English vocabulary), and a document-specific topic $t_o(d)$ responsible for generating terms on other themes also mentioned in the document. These terms are specific to the document context and not relevant to c or its ascendant categories. The contributions of these topics in training documents are automatically inferred and only the truly relevant portions (the ones generated by c itself) will contribute to the estimated language model for c. The model description and the inference algorithm are described in detail in the next subsections.

3.4.1 Hierarchical Topic Modelling

we proposed a Hierarchical Topic Model with Ontological Guidance that is a generative model describing the process of generating relevant documents for topics in a given hierarchy. Let us denote by W, the number of words in the vocabulary, and by L_c, the level of topic c in the hierarchy ($L_b = 0$ for the background (root) topic). The multinomial distributions (i.e. the language models of the different topics, including the background) are denoted by Φ followed by a subscript that refers to the topic. These multinomial distributions are sampled from a W-dimensional Dirichlet distribution with hyper-parameters β, denoted by $W\text{-}Dir(\beta)$. As any pseudo-relevant document d (for category c) will be modelled as a mixture of multinomial distributions for topics in the path from the root to c itself and a document-specific topic $t_o(d)$, we denote the corresponding mixture weights by Θ_d. Θ_d is sampled from a Dirichlet distribution with hyper-parameters α. The generative process is formally described as follows:

1. Pick a multinomial distribution Φ_b for the background language model from $W\text{-}Dir(\beta)$

2. For each topic c in the hierarchy:

2.1 Pick a multinomial distribution Φ_c from $W\text{-}Dir(\beta)$

2.2 For each document d (pseudo-)relevant to c:

2.2.1 Pick a multinomial $\Phi_{t_o(d)}$ from $W\text{-}Dir(\beta)$

2.2.2 Pick a mixing proportion vector Θ_d for $(L_c + 2)$ topics $T = \{background \ldots c, t_0(d)\}$ from $(L_c + 2)\text{-}Dir(\alpha)$

2.2.3 For each token in d

2.2.3.1 Pick a topic z in set T from Θ_d

2.2.3.2 Pick a word w from Φ_z

Figure 3.3(b) presents the graphical model describing the generative process of training documents of the categories in a 3-level hierarchy shown in Figure 3.3(a) (the graphical models for higher-level hierarchies are straight forward extensions of this one). Assume that there are K nodes at the second level, and each node C_i has H_i children: $C_{i1}, C_{i2} ... C_{iH_i}$. The number at the low-right corner of each box (plate) indicates the number of iterations of that box. $|D_{C_i}|$ is the number of training

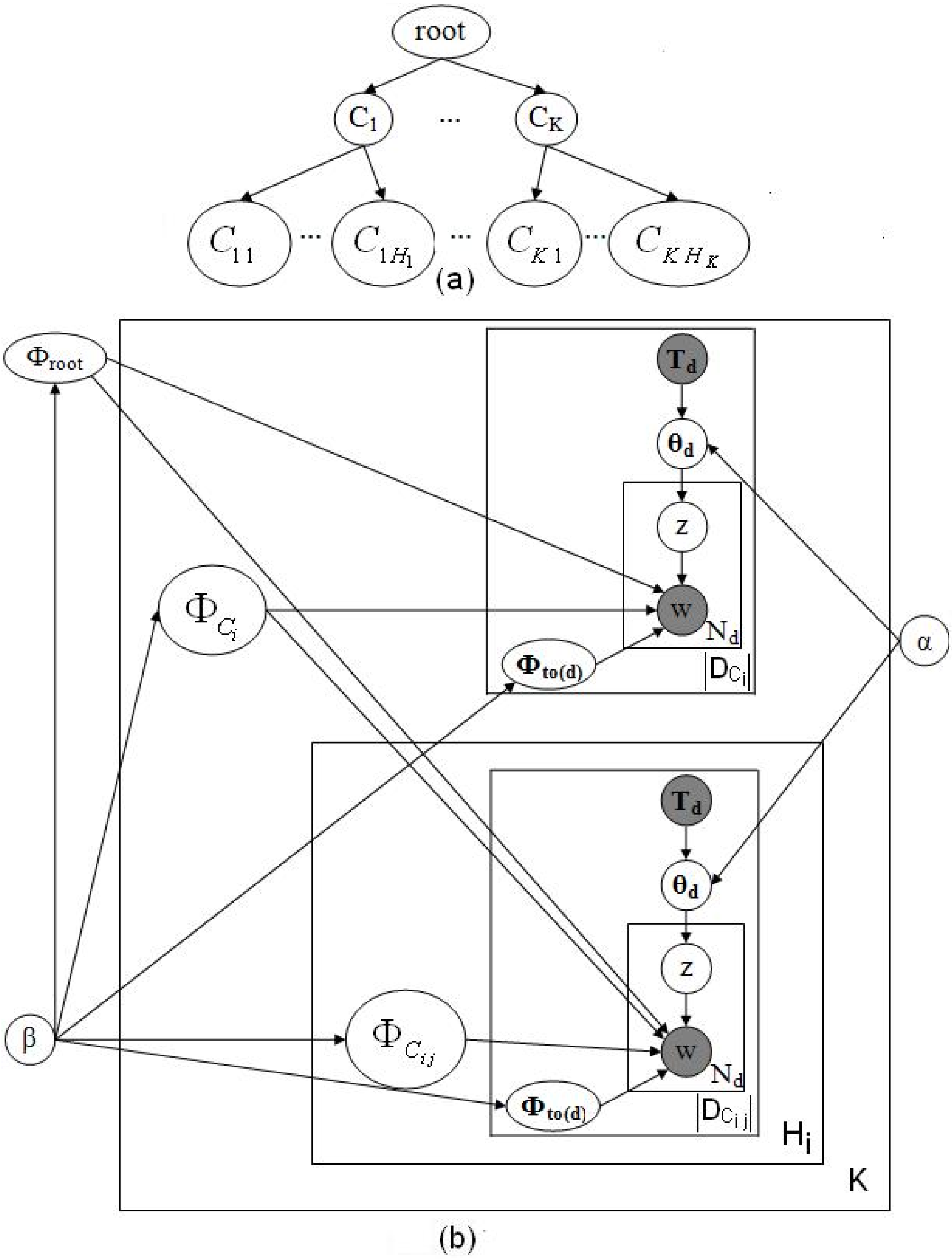


Figure 3.3: Graphical Model of the Hierarchic Topic Model (for a 3-level hierarchy).

documents related to the current category C_i, while N_d is the total number of tokens in the current document d. Each document d belonging to class C_i is generated by a mixture of topics $T_d = \{background, C_i, t_o(d)\}$. Similarly, each document d belonging to class C_{ij} is generated by a mixture of topics $T_d = \{background, C_i, C_{ij}, t_o(d)\}$. In this training phase, word tokens w and T_d are observed variables and denoted by shaded circles. Latent variables are denoted by unshaded circles.

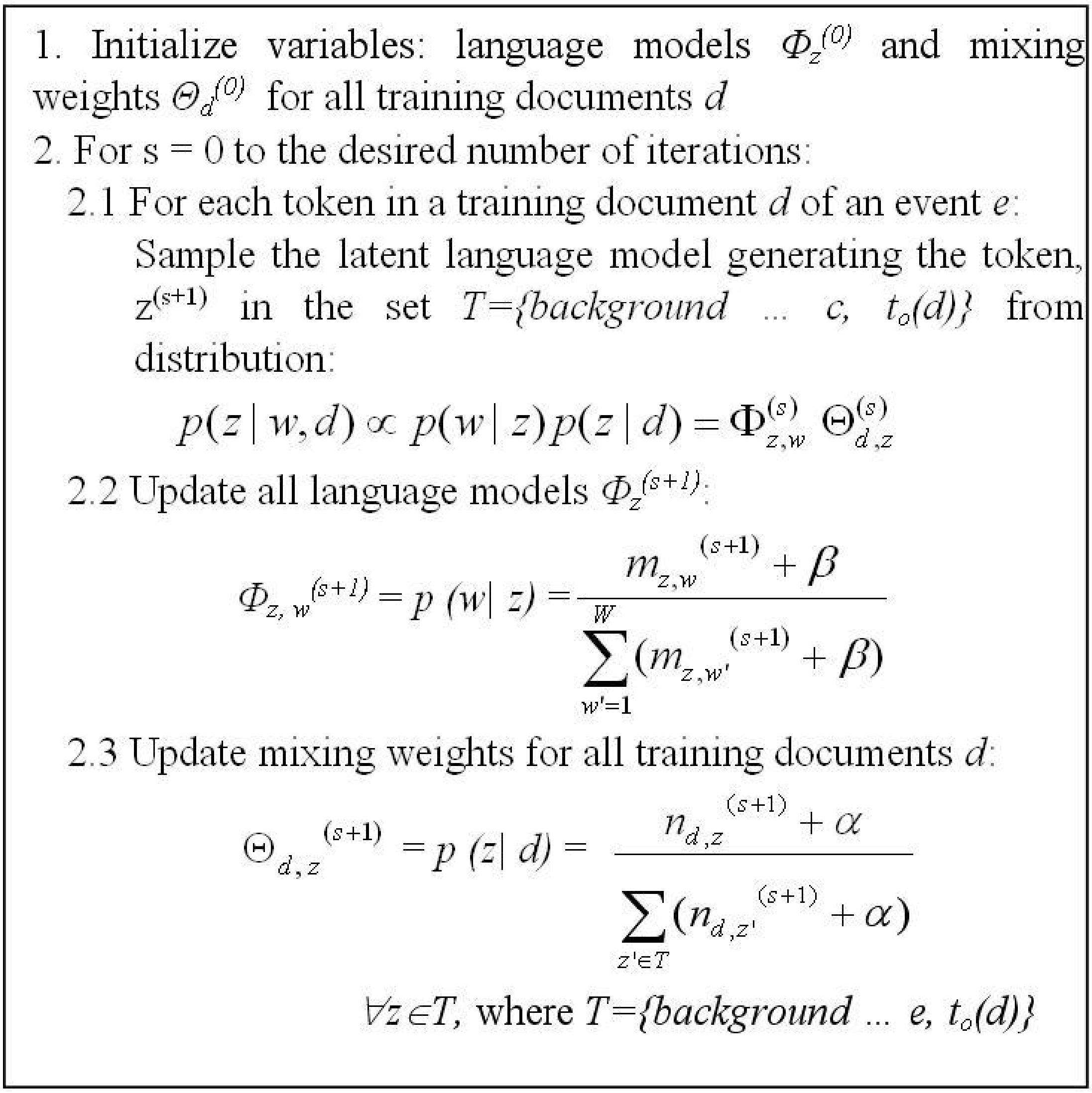
Observe that the scope of the background topic (root) is common to all training documents. The scope of a topic c in the hierarchy covers documents in the corresponding sub-tree (i.e. training documents associated with the category itself and its descendants, if any). The scope of $t_o(d)$ includes only document d. Therefore, the background category will explain words commonly appearing in all training documents of all categories (e.g. stop words). Each topic c generates words relevant to the top level of the sub-tree it represents (more general words are explained by its ascendants, more specific words are explained by its descendants or by document-specific ($t_o(d)$) topics). In each document d, $t_o(d)$ generates words specific to the context of the document but not relevant to any category from the root to category c to which the document belongs. So, the semantic meaning of a category is not only determined by its training documents but also by its relationships to other categories in the tree. All multinomial distributions for categories and category mixing proportions in documents are automatically inferred by the inference algorithm presented in the next subsection.

3.4.2 Inference

For inference, we adopt Gibbs sampling technique [84, 17, 16, 4, 7], which has widely been used for inference in Bayesian networks, for our particular model. Given observable data (i.e. tokens in documents), our algorithm infers all latent variables (multinomial distributions and mixing proportions in documents).

The algorithm is formally presented in the Figure 3.4. In Step (1), the algorithm initializes multinomial distributions for all topics z, and topic mixing proportions in all documents d. Specifically, the multinomial distribution for a category c in the hierarchy, Φ_c, is initialized by maximum likelihood estimate from the training documents belonging to the sub-tree rooting at c. Each multinomial distribution for a document-specific topic $t_o(d)$, $\Phi_{t_o(d)}$, is initialized by its maximum likelihood estimate from document d. Topic mixing proportions in all documents are initialized uniformly. In each iteration of Step (2), we sample latent topic generating each token from its posterior (Step(2.1)). After sampling for all tokens, we update the multinomial distributions and mixing proportions (Steps (2.2) and (2.3)), where $m_{z,w}$ is the number of times word w is assigned to topic z, and $n_{d,z}$ is the number of times topic z is assigned to a token in document d. These sampling and updating steps are repeated until convergence. In practice, we set a value for the maximum number of iterations.

The computational complexity of the inference algorithm is $O(T * D * N * S)$, where T is the number of topics in the hierarchy, D is the number of pseudo-relevant documents for each topic, N is the average length of a document, and S is the number



```
1. Initialize variables: language models $\Phi_z^{(0)}$ and mixing weights $\Theta_d^{(0)}$ for all training documents $d$
2. For s = 0 to the desired number of iterations:
   2.1 For each token in a training document $d$ of an event $e$:
       Sample the latent language model generating the token, $z^{(s+1)}$ in the set $T=\{background \ldots c, t_o(d)\}$ from distribution:
       $p(z \mid w, d) \propto p(w \mid z) p(z \mid d) = \Phi_{z,w}^{(s)} \, \Theta_{d,z}^{(s)}$
   2.2 Update all language models $\Phi_z^{(s+1)}$:
       $\Phi_{z,w}^{(s+1)} = p(w \mid z) = \frac{m_{z,w}^{(s+1)} + \beta}{\sum_{w'=1}^{W} (m_{z,w'}^{(s+1)} + \beta)}$
   2.3 Update mixing weights for all training documents $d$:
       $\Theta_{d,z}^{(s+1)} = p(z \mid d) = \frac{n_{d,z}^{(s+1)} + \alpha}{\sum_{z' \in T} (n_{d,z'}^{(s+1)} + \alpha)}$
       $\forall z \in T$, where $T=\{background \ldots e, t_o(d)\}$
```

Figure 3.4: Inference Algorithm

of samples generated (i.e. the number of iterations). Typically, S is a constant. So, the computational complexity of the inference algorithm is linear to the size of training data (i.e. pseudo-relevant documents of the topics).

CHAPTER 4
INFORMATION RETRIEVAL BY PSEUDO-RELEVANCE FEEDBACK

4.1 Problem Statement and Motivations

The central problem of information retrieval is to retrieve documents relevant to a user's information need, which is typically represented by a query. However, the language used in texts is often very diverse, and this is particularly true in the case of the social Web, where the content is generated by a large number of people. Different people could use different vocabulary to write about the same topic. That makes it challenging for users to comprehensively describe their information need and often causes a vocabulary mismatch problem between users' queries and relevant documents. Pseudo-relevance feedback is a popular approach to alleviate this problem [54, 15, 92, 80, 74, 71]. The basic idea is as follows. Given a query, for instance, "2008 US election", the retrieval system takes the top-ranked documents from the initial retrieval result with the original query. It assumes these documents to be relevant. Then, the system extracts new terms from these documents that are also useful to describe the information need, but not in the original query such as "Obama", "McCain", "presidential". The system uses these terms to expand the query, and does retrieval again with the expanded query. Since the expanded query is more comprehensive, the second retrieval run is more likely to retrieve more relevant documents.

In previous research, relevance-based language models are a popular formal model for pseudo-relevance feedback [54, 55, 57]. Given a query and its training documents (i.e. pseudo relevant documents), relevance-based language models estimate the probability distribution $p(w|Relevance)$, and take the top terms ranked by this probability for query expansion. The distribution is estimated by using a set of training documents. Nonetheless, these relevance-based language models have a limitation; they make an overly-strict assumption that all tokens in each training document are generated by a single topic (query) to which the document belongs. This assumption is obviously not true in practical cases. The example bellow is an excerpt of a Wall Street Journal article considered to be relevant to the topic "machine translation" (TREC topic 63). As we can see, many portions of it are non-relevant to the topic.

> "Buried among the many trade issues that bedevil the U.S. and Japan is the 1 billion dollars of translation work done every year in Japan that could be done better and more efficiently in the U.S. And in the next two years, the dollar value of Japanese-to-English translations is expected to double. Think about it. Every car, video cassette recorder, boom box or stereo imported into the U.S. from Japan has operating and assembly instructions ..."

In this chapter, we propose a novel approach for pseudo-relevance feedback based on the topic models presented in Chapter 3 [1]. As analysed earlier, our topic models are able to automatically rule out the non-relevant parts and infer the language

[1] The work in this chapter appears in [40] (Ha-Thuc et al. AIRS'10)

models from the relevant parts only. So, the resulting event language models are robust to noise in these pseudo-relevant documents. Thus, the proposed approach overcomes the limitation of the relevance-based language models in previous work. The proposed approach is described in more detail in Section 4.2. We also conduct experiments to empirically demonstrate the effectiveness of our approach over the relevance-based language models in Section 4.3. Section 4.4 reviews various related work in this area. Finally, Section 4.5 presents our concluding remarks of this chapter.

4.2 Proposed Approach

This section presents an approach applying topic models in Chapter 3 for pseudo-relevance feedback. The overall description of the approach is shown in Figure 4.1. In the degenerate case where the input is a flat list of queries, the topic taxonomy contains only two levels. The top level contains a common root (background topic), and there is a topic for each query in the second level. As described in Chapter 3, in the first step, the system retrieves top M documents for each query, and assumes these documents to be relevant. Then, the system applies topic models to rule out the non-relevant parts in the pseudo-relevant documents and infers a language model $p(word|topic)$ for each topic from the corresponding relevant parts. In the last step, the system takes the top N terms (words) ranked by $p(word|topic)$ to expand the corresponding query and does retrieval again with the expanded query.

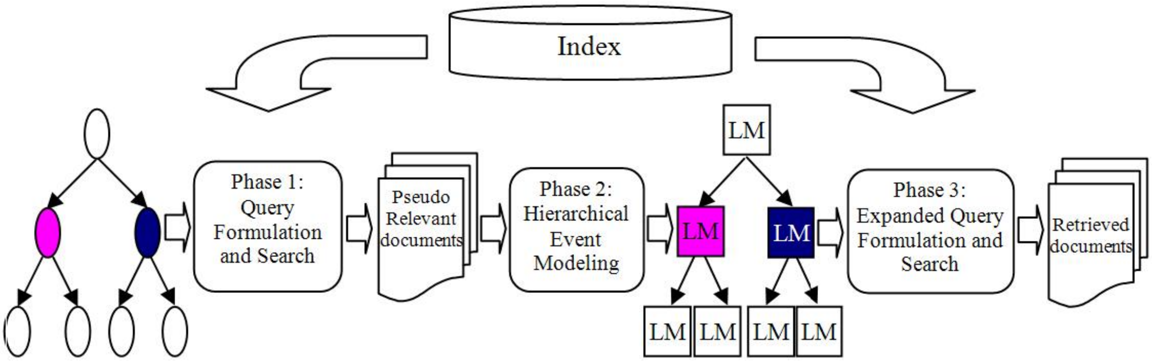

Index
Phase 1: Query Formulation and Search
Pseudo Relevant documents
Phase 2: Hierarchical Event Modeling
LM
LM
LM
LM
LM
LM
LM
Phase 3: Expanded Query Formulation and Search
Retrieved documents

Figure 4.1: Information Retrieval with Pseudo Relevance Feedback

4.3 Experiments

4.3.1 Query Expansion

In this section, we evaluate the effectiveness of our topic model on the task of pseudo-relevance feedback compared to the standard relevance-based language model (Rel-based LMs). Our experiments are done using five corpora (Table 4.1). The first two corpora are social text collections. The 20 Newsgroup dataset contains discussion posts. Each of the posts is labeled by one of 20 topics. We use the 20 topic titles as queries. TREC 2009 Blog collection contains about 24 million blog post spanning from early 2008 to early 2009. We pick a test topic set of 219 New York Times headlines. Each headline has at east 5 relevant posts (based on TREC judgements). Besides these social collections, we also conduct experiments on three other popular datasets. AP and WSJ contain newswire articles. For these corpora, we use 100 topics (title only) and partial judgements for these topics provided by TREC. Finally, Cora contains abstracts of computer science research papers. These papers are also manually assigned to topics. We use 20 topics for this corpus.

Corpus	the number of documents	the number of topics (queries)
20 Newsgroups	19 956	20
TREC 2009 Blog Collection	28 488 766	219
TREC Associate Press (AP)	242 918	100
TREC Wall Street Journal (WSJ)	173 252	100
Cora	25 705	20

Table 4.1: Corpora used for pseudo-relevance feedback experiments.

For each of the topics associated with 20 Newsgroup, AP, WSJ and Cora, we take the top 50 retrieved documents from the initial retrieval run as training documents. For 219 headline queries of the TREC 2009 Blog dataset, we first identify blog posts containing the links to the corresponding articles (article URLs). The appearances of the links are considered as evidence indicating the relevance to the corresponding headlines. Then, we collect the text around each link within a window of +/-800 characters. This social data is used as training data.

We apply the topic model to estimate language models $p(word|topic)$ for all topics from the training data. We compare our topic model approach with relevance-based language models **??** applied on the same training data. Relevance-based language models, a popular approach for pseudo-relevance feedback [94, 53, 20, 46, 79, 56], also expand a human-generated query (topic) to a multinomial distribution over a finite set of words. Specifically, given a topic t and a set of its pseudo-relevant documents D_t, the model estimates the multinomial distribution $p(w|t)$ of observing a word w in documents relevant to t as in Equation 4.1.

$$p(w|t) = \sum_{d \in D_t} p(w|d)p(d|D_t) \tag{4.1}$$

In the equation, first factor $p(w|d)$ is a document language model of d, that could be estimated by normalized term frequencies. The second factor $p(d|D_t)$ could be estimated as $p(d|D_t) = 1/|D_t|$ [42]. A key difference between relevance-based language model approach and our approach is that the former does not explicitly exclude the non-relevant terms in pseudo-relevant documents when estimating $p(w|t)$. Our experiments will show the benefit of excluding the non-relevant terms.

In terms of efficiency, assume there are T topics, each topic has D pseudo-relevant documents, each document on average has a length of N, the running time of relevance-based language models is *O(T*D*N)*. As analysed in Chapter 3, the running time of our approach is *O(T*D*N*S)*, where S is the number of iterations in the inference algorithm, which is typically a constant (30 in our experiments). So the running time of our approach is S times longer than relevance-based language model approach.

For each topic t, the top 50 words ranked by $p(word|t)$ provided by each approach are used to expand the corresponding topic. The parameter value (50) has been tuned for relevance-based language model approach in previous work [56]. We also use this value for our approach. Tuning the parameter specifically for our approach could result in a better performance. We leave it for future work. The expanded queries are submitted to the search engines associated with the corresponding dataset.

We measure retrieval performances by Mean Average Precision (MAP), a standard measure in information retrieval [73]. Given a ranked list of documents for a query, Average Precision (AP) is the average of precision values at the points at which each relevant document appears. Then, MAP is defined as the mean of all Average Precisions over all queries. So, MAP takes into account both precision and recall and emphasizes on ranking relevant documents higher.

The performances of the initial retrieval and retrieval with pseudo-relevance feedback by the two approaches are shown for each dataset in Table 4.2. In the table, α and β indicate statistical significance over the baseline and Rel-based LMs (p-value<0.05 by the paired t-test), respectively. On the five datasets, the improvements against the baseline for the Rel-based LMs are generally in the range of 0.3% to 43% (19% on average), while for Topic Models are in 5% to 100% (39% on average). In all the cases, Topic Models are always significantly better than Rel-based LMs. The best improvement is observed in the 20 Newsgroup dataset (40%), compared to Rel-based LMs.

These results support our contention that a) pseudo-relevant documents may contain portions that are not relevant to the topic of interest and b) it is possible to build more robust relevance models using the Topic Models framework.

4.3.2 Perplexity

The goal of both relevance-based language models and our topic model is to estimate the unknown true relevance distribution $p(w|t)$ of some topic of interest t.

	20 Newsgroups	Blog 2009	AP	WSJ	Cora
Baseline (Initial Retrieval)	0.1783	0.2326	0.1948	0.2340	0.2307
Rel-based LMs	0.2548^{α}	0.2334	0.2409^{α}	0.2817^{α}	0.2549^{α}
Proposed Topic Model	$0.3568^{\alpha,\beta}$	$0.2444^{\alpha,\beta}$	$0.2650^{\alpha,\beta}$	$0.3118^{\alpha,\beta}$	$0.2844^{\alpha,\beta}$

Table 4.2: Retrieval Performance in terms of

Mean Average Precision (MAP).

A traditional measure for comparing these two estimations is perplexity. Perplexity indicates how well the estimated distributions predict a new sequence of tokens drawn from the true distribution. Better estimations of the true distribution tend to give higher probabilities to test tokens. As a result, they have lower perplexity, which means that they are less surprised by these tokens.

In our experiment such ideal test data is not available. Instead, for each topic (query) t, we approximate the new sequence of relevant tokens by using a held out set of 50 actual relevant documents that do not appear in the training set. We remove stop words from a standard list and also rare words in these relevant documents. Then, we use the remaining tokens as test data. Given the estimated distributions $p_{Rel-basedLMs}(w|t)$ and $p_{TopicModel}(w|t)$ obtained from the previous experiment, we compute Perplexity (PPX) with respect to the test data for each topic as in Equation 4.2, where N is the number of tokens in the test data. Table 4.3 shows the average perplexity over the 20 topics of Cora and 20 Newsgroup datasets. The asterisk symbol ($*$) means that the difference between the two results is statistically significant (i.e. p-value<0.05 by the paired t-test). We experiment on Cora and 20 Newsgroup datasets since each topic of these corpora has hundreds of relevant documents. As we see, the perplexity of relevance distributions estimated by the proposed model is significantly lower than distributions estimated by relevance-based language models. This indicates that our topic model is better at predicting unseen test data from the true distribution as compared to Rel-based LMs. Again, the key difference here is that our model considers each document to be generated by a mixture of topics and not just the relevant topic alone.

$$PPX(TestData|t) = exp(\frac{-1}{N} \sum_{w_i \in TestData} log(p(w_i|t))) \tag{4.2}$$

	Cora	20 Newsgroup
Rel-based LMs	1364	4976
Proposed Topic Models	942*	3134*

Table 4.3: Perplexity.

4.4 Related Work

Relevance-based language models [54], a popular approach for relevance modeling, expand a given topic (query) t to a multinomial distribution $p(w|t)$ of observing a word w in documents relevant to t. The probabilities are estimated by using a set of training documents. A limitation of the relevance-based language models is that they are based on a strict assumption that if a document D is relevant to a topic, all tokens in the document are equally relevant to that topic.

In [42, 96], three-component mixture relevance models are proposed. Besides the relevance component (R_t), the authors introduce two additional components to capture the background (b) and local features (d) in documents. However, the model's assumption that the mixing proportions of the three components $(\lambda_b, \lambda_{R_t}, \lambda_d)$ are known in advance and the same for all documents is not reasonable. For instance, in the case where we use top 50 retrieved documents for the query t as the training set, the first document is likely to contain more relevant portions than the 50^{th} document.

Another approach to alleviate the problem of noise in training documents is to build relevance model on passages (usually windows of text) instead of the whole documents (Liu et al. [57]). However, the way that documents are broken into passages is rather ad-hoc and corpus specific. Moreover, all tokens in each passage are still considered equally relevant. As in the WSJ example documents shown above, relevant and non-relevant terms appear together even within a sentence.

4.5 Summary

In this chapter, we propose a novel approach for information retrieval by pseudo-relevance feedback based on the topic models presented in Chapter 3. Crucially, our approach relaxes the strict assumption of relevance-based language models that if a document is relevant to a topic, the entire document is relevant to that topic. This is done by automatically identifying the non-relevant parts in the document and estimating the relevance models from the truly relevant parts only.

Our experiments on pseudo-relevance feedback show the effectiveness of the proposed model. The results obtained by our model are consistently better across all of the four corpora than the results of the baseline retrieval (23%-100% improvement in terms MAP) and relevance-based language models (10%-40%). Moreover, our experiment on perplexity re-affirms the advantages of our model over relevance-based language models in terms of estimating the true unknown relevance model.

CHAPTER 5
HIERARCHICAL TEXT CLASSIFICATION WITHOUT LABELLED DATA

5.1 Problem Statement and Motivations

With the exponential growth of text data, particularly on the Web, hierarchical organization of these documents is becoming increasingly important to manage the data. Along with the widespread use of the hierarchical data management, comes the need for automatic classification of documents to the categories in the hierarchy. Traditional supervised and semi-supervised approaches for hierarchical text classification often require labelled data for learning classifiers. However, when applied to large-scale classification which involves thousands of categories (topics), creating such labelled data, even just a few documents per category, is extremely expensive since typically the data is manually labelled by humans. Motivated by this, we propose a novel approach for large-scale hierarchical text classification which does not require any labelled data[1].

In this chapter, we explore another perspective on text classification where the meaning of a category is not defined by human-labelled documents, but by its descriptions and more importantly its relationships with other categories (e.g. its ascendants and descendants). Specifically, we take advantage of the ontological knowledge in all three phases of the whole process. First, we exploit the hierarchy to construct a

[1] The work in this chapter appears in [38] (Ha-Thuc et al. TMW'07) and in [37] (Ha-Thuc et al. WSDM'11).

context-aware query for each category. The query is then submitted to a web search engine to get pseudo-relevant documents for that category. Second, given pseudo-relevant documents for categories, we propose a hierarchical topic model approach to extract a language model (multinomial distribution over words) for each category. Note that in the previous phase, even though we use context-aware queries, the retrieved documents could still contain a lot of noise. In the second phase, our hierarchical topic model takes the relationships amongst categories defined in the hierarchy to exclude noise, to identify really relevant parts in training documents, and to estimate category language models from these relevant parts only. Finally, given extracted category language models, the hierarchical structure is again exploited to classify test documents into categories. We propose a novel classification algorithm using information propagated both top-down and bottom-up when making decisions.

We demonstrate the effectiveness of our approach through a series of experiments based on a recent taxonomy released by the IPTC (International Press and Telecommunications Council; see details on *www.iptc.org*), that is increasing being used by major news agencies all over the world as a standard for annotating news items and events. This taxonomy includes 1131 categories, organised in a hierarchical tree that contains up to 6 levels including the common root. We show the benefits of using the ontological knowledge at different stages both qualitatively and quantitatively. In particular, we emphasize that just by taking the simple ontological knowledge defined in the category hierarchy and not using any labelled data, we could automatically build a large-scale hierarchical classifier with reasonable performance.

Specifically, we get performance of 67% in terms of the hierarchical version of the F-1 measure (as described later), when classifying news items from popular sites (recall that in large-scale classification, particularly in our experiments, the system has to make decisions amongst more than one thousand possible choices).

5.2 Proposed Approach

In this section, we introduce an overview of our proposed approach. The overall framework is described in Figure 5.1. First, we exploit the hierarchy to construct an enriched and context-aware query for each category. Basically, for each category, we use its ancestors and its children to define a context for the category and (partially) resolve possible ambiguities. The query is then submitted to a web search engine to get pseudo-relevant documents for the category. Second, given pseudo-relevant documents for categories, we extract a language model for each category. Note that in the previous phase, even though we use enriched and context-aware queries, the retrieved documents are still very likely to contain noise. Therefore, the challenge in the second phase is to exclude noise (non-relevant parts) and identify really relevant parts in training documents. Then, the category language models are estimated from the relevant parts only. The first and second phases are essentially the fundamental model presented in Chapter 3.

Finally, given extracted category language models, we classify test documents into categories. We propose a novel top-down classification approach taking advantage of the hierarchical structure. To alleviate the risk of cascading error, which is

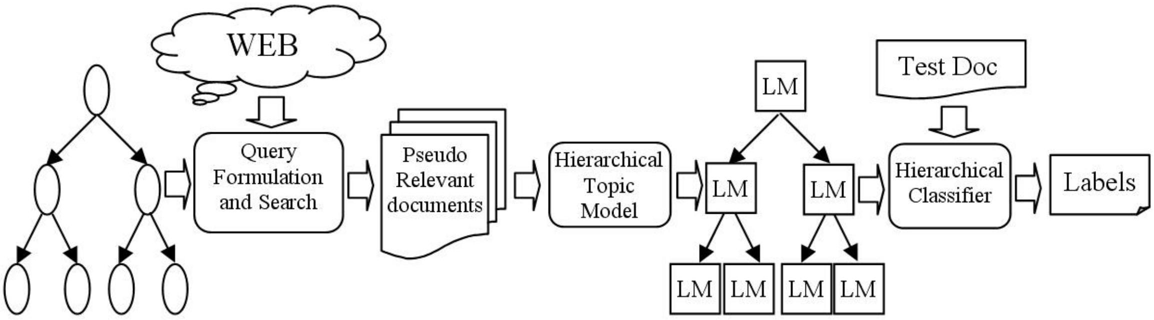


Figure 5.1: A Framework for Large-scale Text Classification without Labelled Data

common in previous top-down approaches [50, 78, 25], our approach softens its decisions at upper levels. Moreover, when making decisions at these levels, the approach also takes into account information propagating from lower levels (bottom-up). The approach is based on a hierarchical extension of the inference algorithm for topic models, that integrates the document context into word features to resolve the polysemy issue (e.g. word feature *race* is important, but with different senses with respect to category "*motorcycling*" and "*people*"). Finally, by taking the hierarchical structure into account, the algorithm could prune a large part of the hierarchy from consideration. Therefore, the algorithm scales well when the number of categories increases. The details of this third phase are described in the next subsection.

5.2.1 Hierarchical Classification

In this study, we consider the general case where a test document could be assigned to multiple categories at different levels of abstractions in the hierarchy. While this setting is more complicated than the case where each test document is

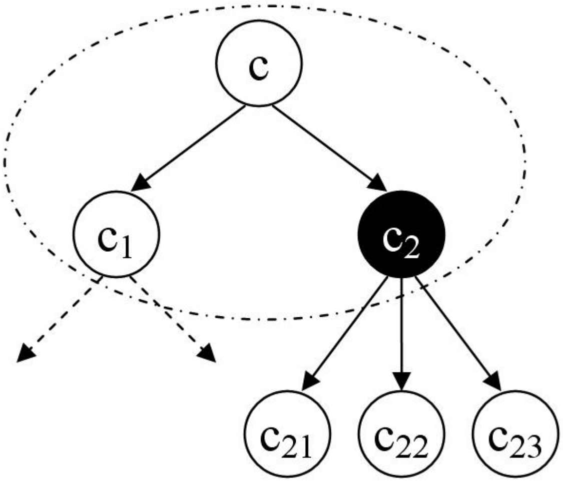


Figure 5.2: Sampling Example

assigned to only the leaf categories, it is more natural in practice. We assume each test document is generated by a mixture of all nodes in the hierarchy (if some category is totally irrelevant to the document, its mixture weight will be close to zero). So, the multi-labeled classification problem can be seen as the task of inferring mixture weights given the document and the language models of all nodes estimated in the previous phase. We solve this inference problem by a sampling approach, keeping the language models fixed. Specifically, we iteratively sample the latent topics generating the tokens in the test document. Then, we rank categories by their mixture weights $p(c|d)$, estimated from the samples.

We exploit the hierarchical structure to decompose the sampling step for each token into sub-steps. The sampling algorithm starts from the root, $c = root$. Assume c has two children c_1 and c_2 (see Figure 5.2). The algorithm probabilistically decides if the token is generated by c or a node in one of the two sub-trees by sampling in the set $S = \{c, c_1^{subtree}, c_2^{subtree}\}$ (where $c_i^{subtree}$ is a pseudo-topic representing the whole sub-tree rooted at c_i) from posterior distribution as in Equation 5.1. In this equation, $p(z|d)$ indicates how much z contributes to the content of document d. These probabilities are estimated iteratively (as we will show later). $p(w|c)$ is estimated in the previous phase. $p(w|c_i^{subtree})$ is a multinomial distribution representing the language model of the whole sub-tree rooted at c_i . It is estimated from the multinomial distributions of all nodes belonging to the sub-trees including c_i itself (see Equations 5.2 and 5.3). When the algorithm samples the latent topic in the set S, if topic c is picked, then the latent topic for the token is determined. If one of the sub-trees, for instance $c_2^{subtree}$, is picked, i.e. the token is generated by a node in the sub-tree rooted at c_2, then the algorithm proceeds to this sub-tree (Figure 5.2) and repeats the process until the latent topic is determined.

$$p(z|w,d) \propto p(w|z)p(z|d), z \in \{c, c_1^{subtree}, c_2^{subtree}\} \tag{5.1}$$

$$\begin{align} p(w|c_i^{subtree}) &= \sum_{z \in c_i^{subtree}} p(w|z)p(z|c_i^{subtree}) \tag{5.2} \\ &\approx \frac{\sum_{z \in c_i^{subtree}} p(w|z)}{|c_i^{subtree}|} \tag{5.3} \end{align}$$

The classification algorithm is formally described in Figure 5.3. The mixture weights are initialized uniformly (Step 1) and will be updated iteratively. In Step 2.1, the algorithm samples latent topics of all tokens in the test document from the corresponding posterior distributions. To avoid the issue of cascading errors, the algorithm "softens" its behaviour by doing the sampling M times (Step 2.1.1). As described earlier, this sampling step is performed in a top-down manner starting from the root (Step 2.1.1.1) and averaging over these samples. When sampling (Step 2.1.2.3), the topic mixing proportions $p(z|d)$ are integrated in the posterior probabilities. This factor representing the context of document d aims to resolve word ambiguity. For example, if d is an article about a fishing resort, then terms in d like "*fish*", "*fishing*" or "*boat*" have high likelihood $p(word|topic)$ in both topics "*travel*" and "*fishing industry*". However, by taking the context of the document into account, the algorithm can recognize that these terms are not meant to be mentioned in the context of topic "*fishing industry*". After generating M samples for all tokens, the algorithm re-updates the mixture weights (Step 2.2). The whole process (including Steps (2.1) and (2.2)) is iterated N times. M and N are parameters.

In the hierarchical sampling process above (from Steps 2.1.1.1 to 2.1.1.4), a token is assigned to topic c only if it is also assigned to all sub-trees rooted at ancestors of c. On the other hand, when the algorithm decides to assign a token to a sub-tree, the algorithm takes information from all the nodes in the sub-tree into account (recall how $p(w|c_i^{subtree})$ is estimated in Equation 5.3). So, when sampling at a particular level in the hierarchy, the algorithm uses information propagated both top-down and bottom-up to alleviate possibly inaccurate estimations of probabilities $p(w|c)$ for some words w and categories c. Moreover, by hierarchically decomposing the sampling, the algorithm can prune a large part of the hierarchy from consideration in the sampling process. As a result, the number of nodes it considers is only $O(log(n))$, where n is the number of categories in the hierarchy. Therefore, it scales well when the number of categories increases (as in the case of large-scale classification).

5.3 Experiments

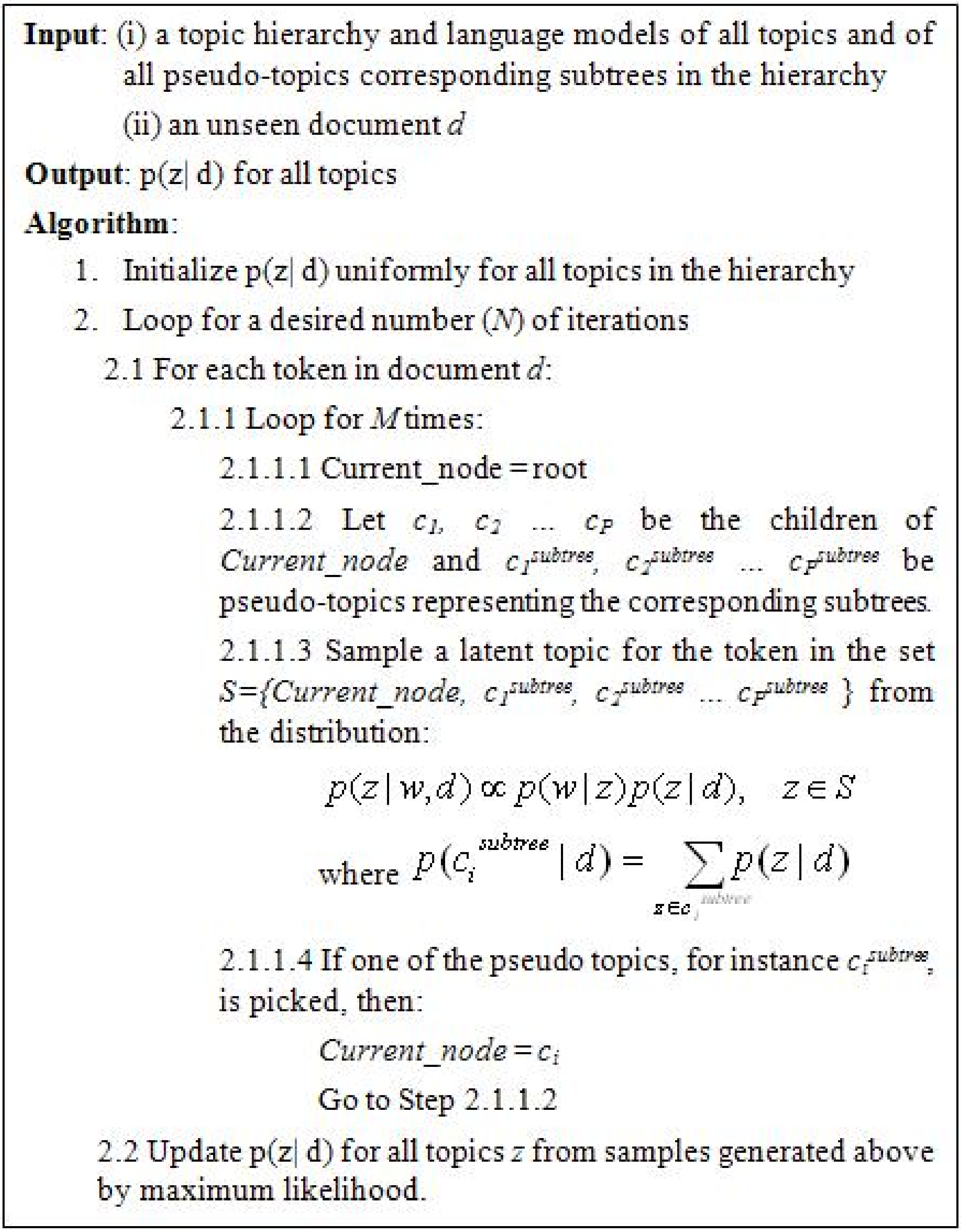
In this section, we demonstrate the effectiveness of our approach in estimating category language models and in classifying test documents. We first describe the topic hierarchy and test documents we use in our experiments. Then, we present performances of our approach in comparison to baselines.

5.3.1 Topic Hierarchy and Test Set

As already mentioned, the IPTC (*International Press and Telecommunications Council*) has recently released a taxonomy of codes, for annotating news items and events. It is becoming a standard for main news agencies and an important component of the `NewsML` standard as media-independent structural framework for multi-media news. This taxonomy contains 1131 categories, organised in a tree that contains up to 6 levels including the common background (root). The first level contains 17 main topics, covering domains such as business, economics, education, religion, crimes, disasters, weather, etc. The last level contains very specific topics, such as “assisted suicide” or “methodist christians”. The average number of children is around 3 in this hierarchy. Each category contains a title (typically two or three words), as well as a short description (25 words on average).

The evaluation set consists of a collection of 1130 news items[2], crawled on the

[2] The preprocessed, annotated collection is available on the web site of the SYNC3 Eu-



```
Input: (i) a topic hierarchy and language models of all topics and of
           all pseudo-topics corresponding subtrees in the hierarchy
       (ii) an unseen document $d$
Output: p(z| d) for all topics
Algorithm:
    1. Initialize p(z| d) uniformly for all topics in the hierarchy
    2. Loop for a desired number ($N$) of iterations
       2.1 For each token in document $d$:
           2.1.1 Loop for $M$ times:
               2.1.1.1 Current_node = root
               2.1.1.2 Let $c_1, c_2 \ldots c_P$ be the children of
                       Current_node and $c_1^{subtree}, c_2^{subtree} \ldots c_P^{subtree}$ be
                       pseudo-topics representing the corresponding subtrees.
               2.1.1.3 Sample a latent topic for the token in the set
                       $S=\{Current\_node, c_1^{subtree}, c_2^{subtree} \ldots c_P^{subtree}\}$ from
                       the distribution:
                           $p(z \mid w,d) \propto p(w \mid z)p(z \mid d), \quad z \in S$
                           where $p(c_i^{subtree} \mid d) = \sum_{z \in c_i^{subtree}} p(z \mid d)$
               2.1.1.4 If one of the pseudo topics, for instance $c_i^{subtree}$,
                       is picked, then:
                           $Current\_node = c_i$
                           Go to Step 2.1.1.2
       2.2 Update p(z| d) for all topics $z$ from samples generated above
           by maximum likelihood.
```

Figure 5.3: Hierarchical Classification Algorithm

web sites of 4 news agencies (CNN, Reuters, France24 and DW-World), during the first two weeks of June 2010. The preprocessing consisted in cleaning the html files (boilerplate removal, etc.), and removing stopwords. Two independent annotators (with a journalism background) labelled this set of 1130 news items: for each item, they were allowed to give as many labels as they wanted, provided that they used the most specific ones in the trees.

5.3.2 Extracting Category Language Models

In this subsection, we show the effectiveness of our approach in extracting topic language models for topics in the IPTC hierarchy. We compose a query for each topic as described earlier. We conduct two searches for each query. For the first one, we search on the Wikipedia site using Yahoo search engine, and take the top-10 retrieved documents. For the second one, we search on the general Web, and take the top-50 retrieved documents. The two results are merged and used as training documents for the topic. Then, we use the topic model with ontological guidance to extract a language model for each topic. We compare results of our approach with results of the standard maximum likelihood approach (where the language model of a category is derived from the count of the total number of occurrences of a particular word divided by the total number of tokens, when we consider the concatenation of all documents related to the category (topic)) applied on the same training documents.

Figures 5.4 and 5.5 show top terms of language models of categories in a seg-

ropean Project: www.sync3.eu

ment of the whole hierarchy extracted by the baseline and our approach. Comparing language models for topic "*music*" (at third level) extracted by the two approaches, we see that the one in Figure 5.4 contains too general terms like "*art*", "*entertainment*", "*news*" and "*search*" on top. On the other hand, most of the top terms in the language model extracted by our approach are strongly relevant to the topic "*music*" (in Figure 5.5).

At the fourth level, in Figure 5.4, general musical terms such as "*music*" and "*musical*" are ranked very high in the language models of categories "*musical style*", "*musical performance*" and "*musical instruments*". These terms, however, have little power to differentiate each of these categories with the others and their parents. Language model of category "*musical performance*" also contains non-relevant terms such as "*instruments*" and "*instrument*" on top. This is because training documents for this category contains noise that is about topic "*musical instruments*" instead, and the standard likelihood approach assumes all parts in the training documents are relevant. Our approach, on the other hand, exploits the relationships amongst the categories to automatically exclude non-relevant parts. As a result, the non-relevant terms do not appear on top of the language model extracted by our approach.

Similarly, at the lowest level, language models in Figure 5.4 contain general terms while the language models in Figure 5.5 focus on terms that are unique for the category at this level. Due to the space limit, we only show language models of topics in a segment of the hierarchy extracted by the two approach. But, we observed that the patterns described above hold consistently across the whole hierarchy.

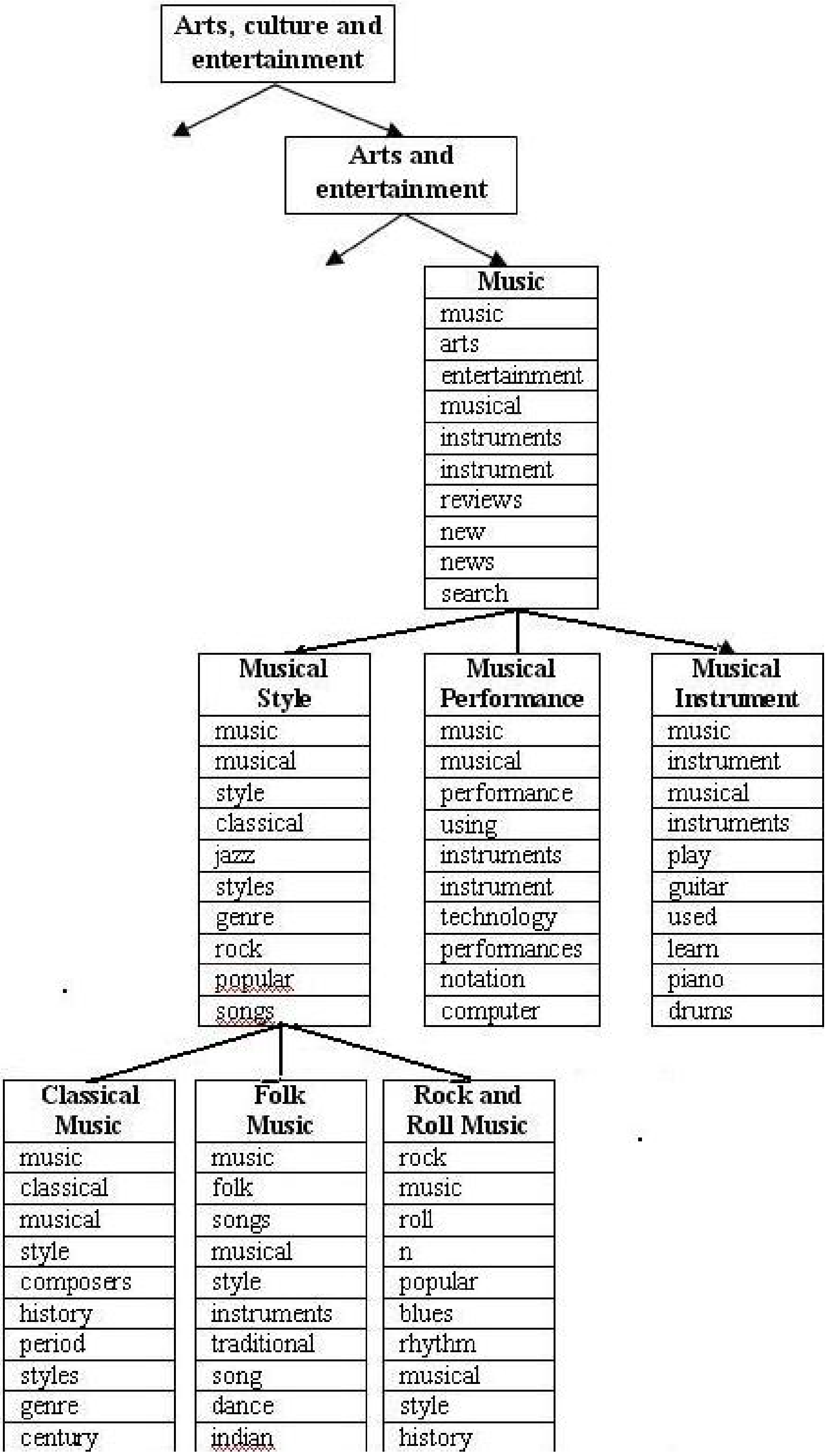


Figure 5.4: Topic Language Models extracted by standard maximum likelihood

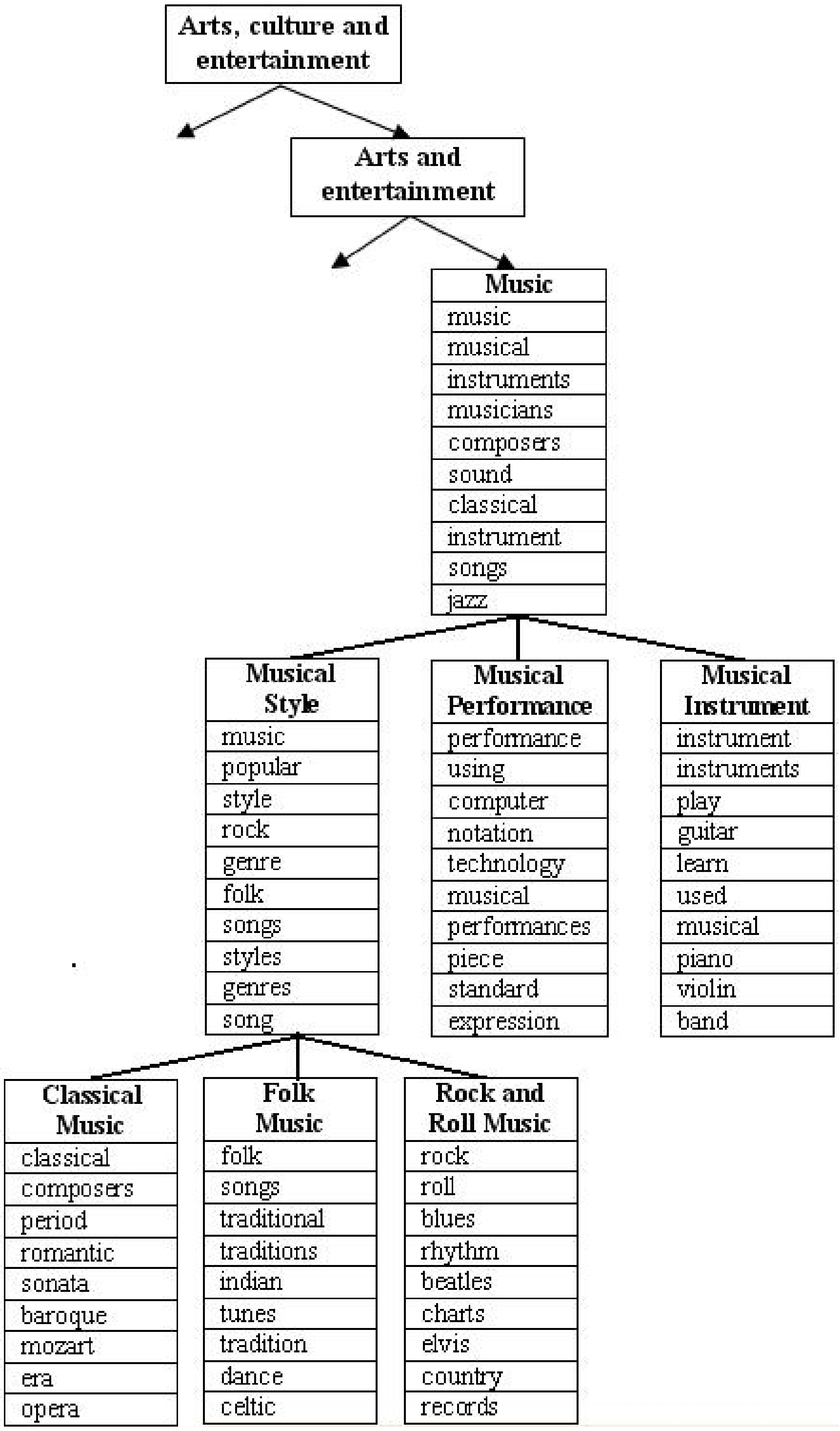


Figure 5.5: Topic Language Models extracted by the hierarchial topic models with ontological guidance

5.3.3 Classification

In this subsection, we empirically demonstrate the effectiveness of our hierarchical classification approach by comparing it with two baselines: the naive Bayes classifier and hierarchical naive Bayes classifier. We pick naive Bayes as a baseline because it has widely been shown effective for text classification, especially when training data is imperfect (Krithara et al., [51]). Hierarchical naive Bayes is an extension of naive Bayes [81]. Specifically, the language model of a category is smoothed by the language models of its ancestors (shrinkage technique).

All of the three approaches take the category language models extracted by the hierarchical topic model approach and a test document as inputs; they then rank the categories in decreasing order of relevance. We measure performances by precision, recall and F-1 at different positions in the ranked list. Besides standard measures of precision, recall and F-1, we also use hierarchy-based extensions of these measures as proposed in [19]. The basic idea is that it is better to classify a document into a category near the correct one in the hierarchy, than to a totally unrelated category (i.e. the cost of error depends on the dissimilarity between the predicted category and the real ones). The dissimilarity of two categories is defined by their respective positions in the hierarchy. We average the performances over all test documents.

Figure 5.6(a) shows standard precision, recall and F-1 of the three approaches at ranks from 5 to 35. In terms of these standard measures, performances of the two baselines are similar. The proposed hierarchical classification approach is consistently better than naive Bayes and hierarchical naive Bayes in terms of both precision and

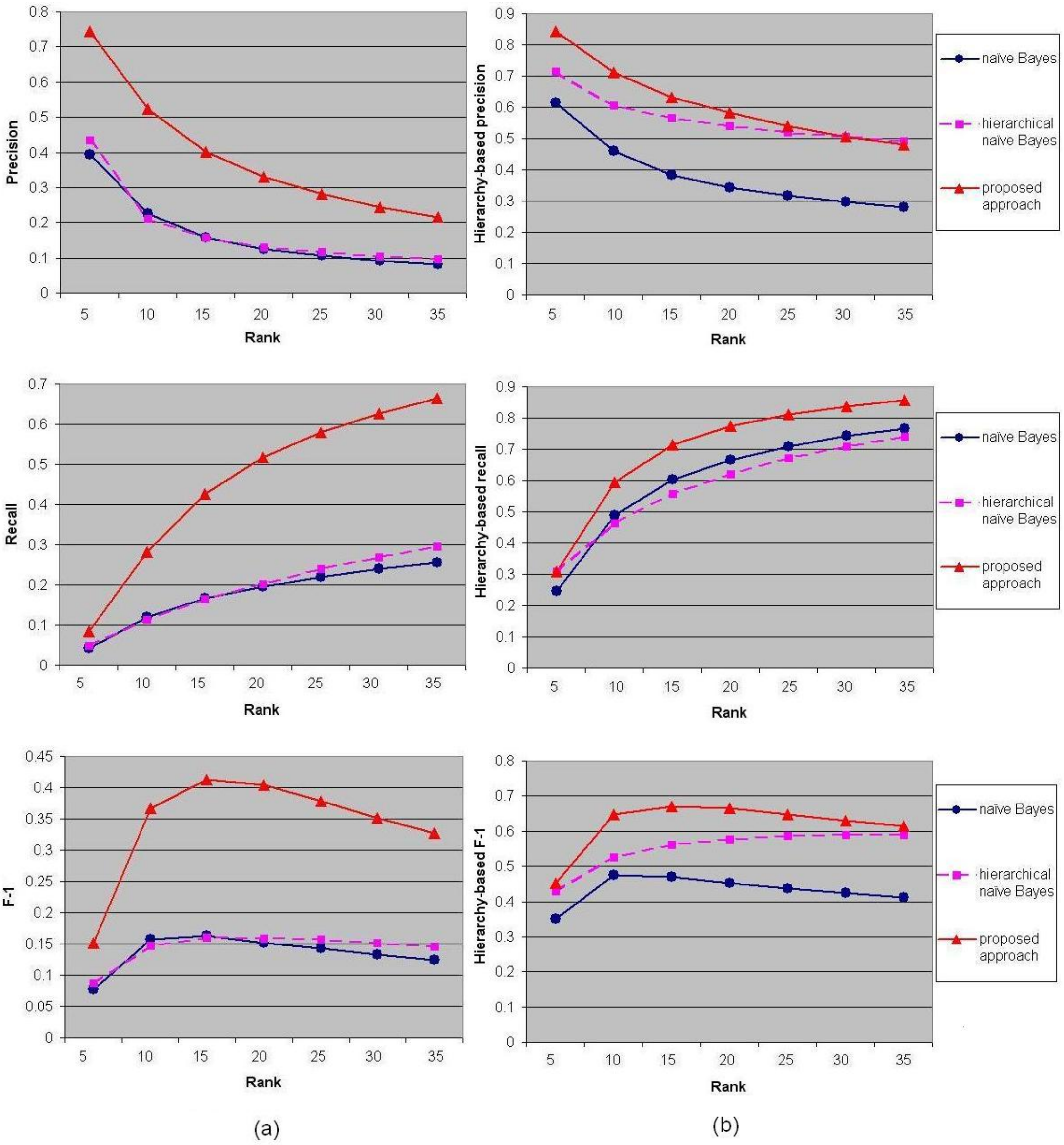


Figure 5.6: Classification Performances by (a) standard and (b) hierarchy-based measures

recall. In terms of F-1, the best performance of our approach is 41%, while the best performances of naive Bayes and hierarchical naive Bayes are 16%. Note that in this large-scale text classification problem, the classifiers have to make tough decision amongst more than 1100 possible choices.

When using hierarchy-based measures (Figure 5.6(b)), we could see in the figure that hierarchical naive Bayes is better than naive Bayes in terms of precision since the shrinkage smoothing technique could alleviate some imprecision in the estimation of category language models. However, the hierarchical naive Bayes is slightly worse than naive Bayes in terms of recall. This is due to the smoothing technique that makes language models of neighbour categories (i.e. categories that share some common ancestors) highly similar. Consequently, this results in a ranked list of categories for each test document that is less diverse. As in the previous case, our approach is generally better than the two baselines in terms of both precision and recall. In terms of F-1, our approach is around 13.4% and 41.2% better than hierarchical naive Bayes and naive Bayes, respectively.

5.4 Related Work

Our work in this study is related to several existing directions: information retrieval, document classification without labelled data, hierarchical text classification and topic modelling. We briefly review each of these directions.

The use of pseudo-positive documents as an important component to bootstrap process is common in the information retrieval community. But, unlike standard ad-hoc retrieval, the most popular form of information retrieval, which aims at retrieving relevant documents for individual queries separately, our retrieval approach exploits hierarchical relationships amongst queries to improve retrieval performance.

As far as text classification without labelled data is concerned, several works have been proposed recently for building flat text classifiers without labelled data [31, 86, 95, 49, 45]. Generally, instead of using labelled documents, their approach uses retrieval or bootstrapping technique to initially assign documents to topics represented by a title or a few keywords, then incrementally builds a classifier and refines the assignments through many iterations. This family of approaches adopts a strategy in "three phases" (initialization exploiting the prior knowledge; iterative refinement and final categorization), as our method does. However, when the topic representations are short and ambiguous, the initial assignment is likely to be inaccurate and that could mislead the whole process. Our approach, proposed for hierarchical classification, on the other hand, takes into account the hierarchical relationships to automatically enrich semantic representations of topics. As a result, performance of the initial retrieval phase is improved. Second, our learning approach on initially retrieved documents is robust to noise in these documents. So, the approach could reduce the risk of depending on the initial step. Finally, in the categorization step, our approach uses information cascaded top-down (from ascendant categories) and bottom-up (from descendant categories) to alleviate any noise in each category language model estimation.

In terms of hierarchical text classification based on languages models, our work has to be related to the methods proposed in [50, 78, 25, 29]. These papers all propose supervised approaches which rely on manually labelled data. There are also several previous works exploiting hierarchical structure to improve estimated topic language models. Specifically, [58] and [81] use top-down information, while [90] uses bottom-up information to smooth the estimates of $p(w|t)$ for all topics. However, in all of the work, the smoothing could have a side effect that makes distributions of similar topics which share common ancestors or descendants highly overlapping and less distinguishable. Our hierarchical classification approach uses both top-down and bottom-up information in an adaptive way to alleviate the problem of noise in topic language model estimations as well as maintain discriminative power of these language models. Moreover, our classification approach softens decision at early stages, so it could further alleviate the issue of cascading errors.

5.5 Summary

In this chapter, we propose a novel approach for automatic large-scale hierarchical text classification which does not require any labelled data. Instead of using human-labelled documents, we take advantage of the ontological knowledge defined in a category hierarchy to construct enriched and context-aware queries for these categories in the hierarchy and then use these queries to retrieve pseudo-relevant documents on the Web. Then, we propose a hierarchical topic model with ontological guidance, which exploits the relationships amongst categories to exclude noise, identify really relevant parts in the pseudo-relevant documents, and estimate language models for these categories. Finally, we present a novel algorithm using hierarchical structure for classifying test documents.

Our experiments on IPTC taxonomy containing 1131 categories demonstrate effectiveness of our approach. In estimating language models for categories, our experiments show that the hierarchical topic model with ontological guidance is robust to noise in pseudo-relevant documents and could be able to identify terms relevant to categories at different levels of abstraction. As a result, language models extracted by the proposed approach are more appropriate than ones extracted by the maximum likelihood. In the final phase, classifying test documents, the proposed hierarchical classification algorithm outperforms flat naive Bayes (150% and 41.2% improvement w.r.t to the standard and hierachy-based F-1, respectively) and a popular hierarchical classification approach, hierarchical naive Bayes (150% and 13.4% improvement). Overall, we show that just by taking the simple ontological knowledge defined in a category hierarchy, we could automatically build a large-scale hierarchical classifier with quite satisfying performance.

CHAPTER 6
EVOLUTIONARY TREND DISCOVERY

6.1 Problem Statement and Motivations

In this chapter, we apply our hierarchical topic model to the problem of retrospectively discovering evolutionary trends of a crowd's discourse on news events, such as the 2008 US election. Specifically, our goal is to discover when an event starts to get discussed, when the discussion reaches to its peak, when it declines and how the language the crowd talks about the event evolves over time. This aims to reveal important insights on how the crowd's interests on news events shift dynamically and provides a basis to predict what happens next[1].

Because of its importance, it is not surprising that many systems, for instance, Google (Google Trend[2], Google Insight Search[3]), Blog Pulse (Trend Search)[4] and Blog Scope[5] offer services to support discovery of evolutionary trends of particular news events (or of queries in general). However, these services tend to be "term-based". That is, given terms entered by a user to describe an event, generally these systems compute event popularity in a time period by counting the number of social media

[1] The work in this chapter appears in [34] (Ha-Thuc et al. SIGIR'09) and [35] (Ha-Thuc et al. ICSC'10) and is preprared for a journal publication [36] (Ha-Thuc et al. ACM Transactions on Intelligent Systems and Technology).

[2] http://www.google.com/trends

[3] http://www.google.com/insights/search/

[4] http://blogpulse.com/

[5] http://www.blogscope.net/

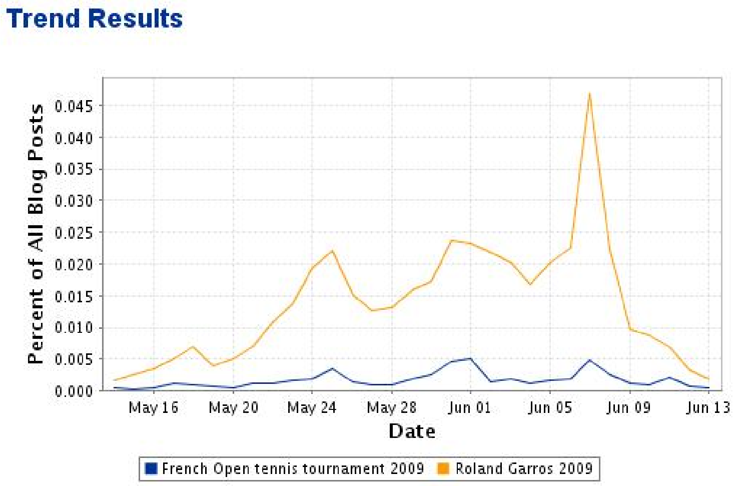


Figure 6.1: An Example of Event Evolutionary Trend Discovery from Social Data

documents (web query logs entries or blog posts) containing these terms.

However, the same news event could be discussed (and searched for) in different ways, i.e. using different vocabulary. So, it is difficult to identify the right query that will track all or most of the conversation about a topic. Figure 6.1 illustrates this point. The figure shows the temporal trends generated by Blog Pulse for two queries "French Open Tennis Tournament 2009" and "Roland Garros 2009" representing the same event. We see a drastic difference between the two results. Moreover, the language people use to discuss an event is dynamic. For instance, at the beginning of the 2008 US election, people were unlikely to use terms "Palin" or "Biden" in their discussions of the event, but at the end these terms were very likely to be mentioned. We observe that using fixed queries to track events could not accurately capture the event evolutionary trends.

In this chapter, we propose an approach to overcome these limitations. Again, we represent each event by a language model that captures various relevant terms and weights the relative importance of these terms with respect to the event. The event language models are also refined at different time steps to more accurately capture the crowd's vocabulary at that time. The refined language models are then used to estimate the event popularity. Tracking the event popularity over time shows the evolutionary trends of the events. The details of our approach and its effectiveness are presented in the next sections.

6.2 Proposed Approach

Our approach takes an event taxonomy and a data stream such as a weblog stream as input. Each node in the taxonomy is a simple label of an actual event in the form of an event "title". The framework outputs dynamic language models for the events in the taxonomy and their popularity in social media at each time step. The overall approach is described in Figure 6.2. The weblog stream of interest is crawled, harvested, parsed and then indexed. The taxonomy represents prior knowledge that a user typically has regarding the structure of news events of interest. The event taxonomy could be entered by a user, similar to the way in which a user enters queries into commercial tracking systems mentioned above, or the taxonomy could be extracted from a source such as Wikipedia. The specific event taxonomy we use in our experiments is given in Figure 6.3. The ontological knowledge defined in the taxonomy (brief event titles and event relationships) is used in all three phases of the framework. The first and the second phases are essentially the hierarchical topic model presented in Chapter 3. In the first phase, the system retrieves documents in the whole stream relevant to the events in the taxonomy. Then, in the second phase, the system takes these training documents and the hierarchical relationships amongst the events into account to estimate language models for the events. These event language models are time-independent and referred as static models.

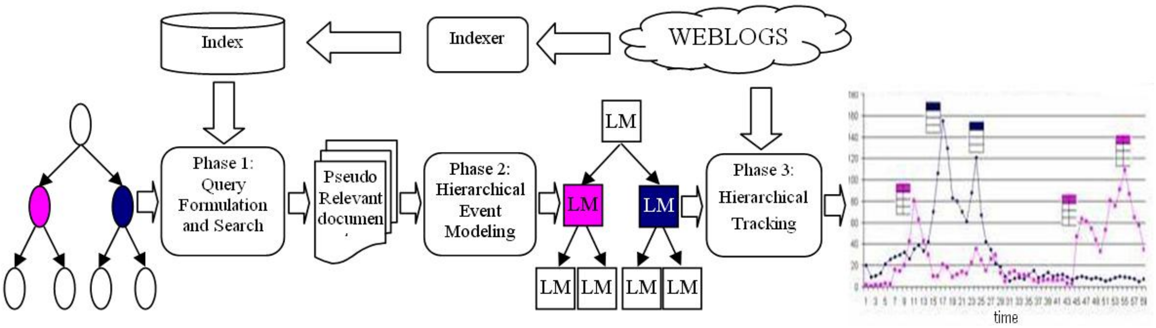


Figure 6.2: An Overall Framework for Modeling a Crowd's Perspectives on News Events

In the third phase, given the static event language models, we adapt these models in each time step. For this, we first divide the stream into temporal chunks and scan the stream chunk by chunk. We observe that the language used by bloggers to discuss the same events could evolve over time. So, for each chunk we refine the static event language models in order to make them better fit the blog data in the current chunk. The refined language models are then used to infer event popularity within that chunk. This popularity measure evaluates the extent to which the corresponding events are discussed in the blogsphere during the chunk time span. The next subsection focuses on the third phase.

6.2.1 Hierarchical Event Tracking

In this phase, given the static event language models discovered in the previous phase, we retrospectively track these events in the whole weblog stream. We refine the static language models in order to make them better fit the discussions of the events at different time steps, and use these refined models to estimate their social popularities at these time steps. Thus, this refinement dynamically captures “semantic drifts” over time associated with the events. As an example, for the US Presidential Election event, around the time of Democratic National Convention the focus is likely on Democratic Party. While around the time of Republican National Convention, the focus is likely on the Republican Party. Moreover, the dynamic event language models also allow more accurate estimation of the event social popularities.

The key challenges in this phase are as follows. First, the whole data stream is often too large to load into internal memory, so it is prohibitively expensive to make multiple passes over this large amount of data. Second, the data relevant to the events is likely to be sparse and also appear together with non-relevant data pieces in the same weblog posts.

To overcome the scalability issue, we divide the whole blog stream into temporal chunks and scan the whole stream chunk by chunk. In this study, a chunk is defined as the collection of all weblog posts in a day. To counter the problem of sparsity of relevant data in the weblog stream we use a simple heuristic to identify pseudo-relevant documents (weblog posts) of the events in the current chunk. Specifically, a document is considered *possibly* relevant to an event if it contains at least m out of top M most probable terms ranked by the corresponding event language model $p(terms|event)$. m and M are parameters, and of course $m \ll M$. It is also worth noting that the heuristic is used only to initially assign documents to events. Our approach will further exclude non-relevant portions of documents for each event automatically. In principle, the non-relevant portions could be anywhere between 0% to 100% of any training document. The dynamic event language models are estimated from the really relevant parts only.

As in phase 2, a training document d of an event e is also hypothesized to be generated by a mixture of multiple language models: the language models of the nodes from the root to e itself , and a document-specific language model $t_o(d)$. Which tokens are generated by which language models is automatically inferred by an inference algorithm. The inference algorithm is still similar to the one in previous phase except for a key difference: the static language models are now used to regulate the inference process. In particular, the static models are used for initialization (Step 1, Figure 3.4) and for regulating the update step (Step 2.2, Figure 3.4). The update step now becomes as in Equation 6.1. The numerator and each element in the denominator include two terms. The left term is the word count in the current chunk representing the likelihood, while the right terms represents the prior. The “scaling” parameter μ indicates their relative importance. In our experiment, we initially assign μ to a large value (10000) then gradually decrease this value after each iteration until 1000. The rationale for this is that at the beginning (first iterations) the information discovered from the current chunk (likelihood) is less reliable, so we strongly emphasize prior knowledge to prevent the language model from drifting in a wrong direction. As we get to the end, we rely more on the likelihood to make the language model fit the current data well.

$$\Phi_{z,w}^{(s+1)} = \frac{m_{z,w}^{(s+1)} + \mu * \Phi_{z,w}^{static}}{\sum_{w'=1}^{W}(m_{z,w'}^{(s+1)} + \mu * \Phi_{z,w'}^{static})} \tag{6.1}$$

After running the inference algorithm, we compute the popularity (i.e., intensity) of each event at each time point p with window size L as in Equation (6.5). The measure indicates how much event e is mentioned in the subset $C[p, p+L]$ of weblog posts written in the period $[p, p+L]$.

$$\begin{aligned}
Popularity(e,p) &= p(e|C[p,p+L]) && (6.2)\\
&= \frac{\sum_{d\in C[p,p+L]} p(e|d)p(d)}{p(C[p,p+L])} && (6.3)\\
&\propto \sum_{d\in C[p,p+L]} p(e|d) && (6.4)\\
&\propto \sum_{d\in C[p,p+L]} \theta_{d,e} && (6.5)
\end{aligned}$$

6.3 Experiments

6.3.1 Experiment Setup

The data we use for our experiments comes from the International Conference on Weblogs and Social Media (ICWSM) 2009. This data was provided by weblog indexing service Spinn3r[6]. It includes 60 million postings spanning August and September 2008. Using a language tag provided by Spinn3r we identified and used the 24 million posts that were in English. We indexed this blog dataset using Lucene[7].

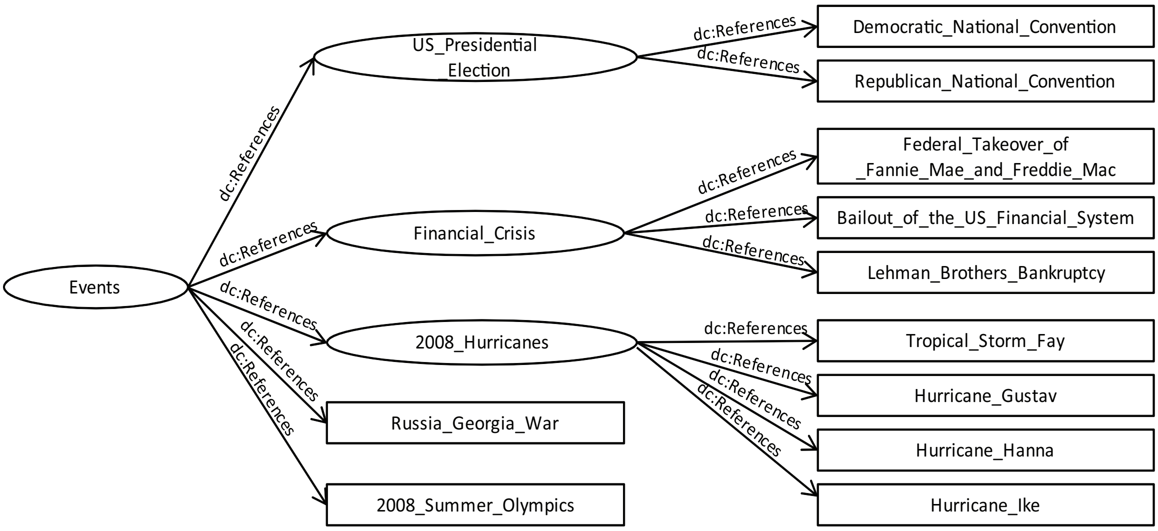


Figure 6.3: Event Taxonomy

Figure 6.3 shows the event taxonomy used in our experiments. The procedure that we used to build the taxonomy is as follows. We used the English version of Wikipedia[8] to get a list of events during the time span of the dataset, then we picked the most prominent ones and organized them into a hierarchical structure. The parent-child relationship is defined as follows. An event A subsumes a sub-event B if

[6] http://spinn3r.com

[7] http://lucene.apache.org/

[8] http://en.wikipedia.org

(i) the time span of A covers the time span of B, and (ii) a document that is relevant to B will also be relevant to A.

6.3.2 Extracting Static News Event Models

In this subsection, we demonstrate the effectiveness of our proposed hierarchical event modeling approach in estimating static topic language models. Our approach takes the event taxonomy in Figure 6.3 as a guide and generates a representation of the discourse in the blogsphere of these events. In this experiment, we use standard LDA as a baseline. LDA has been widely used to discover event or topic language models in other domains such as academic writing [33] and newswire [60]. One of the key differences compared to our approach is that LDA does not have a mechanism to use ontological knowledge to guide the modeling process.

First, we investigate the extracted language models for events at the top level of the taxonomy (events in the general domain). For LDA, to extract language models in the top level of abstraction, we run LDA on a random subset of 10,000 documents[9] in the general domain. To make it comparable, the number of events for the two approaches are set to be the same ($K = 5$). Then given the language models extracted by LDA and by our approach, we rank words by $p(w|event)$ w.r.t. each event as in previous work [26][33][60]. Since LDA does not have a "built-in" mechanism to lower the roles of background words as in our approach, for LDA, we additionally rank words by $p(event|w)$, estimated as in Equation (6.7). We evaluate

[9] 10,000 documents is reasonably sufficient compared to previous work on LDA

the language models extracted by the two approaches using human knowledge and judgement as the gold standard.

$$\begin{aligned} p(event|w) &\propto \frac{p(w|event)}{p(w)} && (6.6) \\ &\propto \frac{p(w|event)}{log(df(w))} && (6.7) \end{aligned}$$

Tables 6.1 and 6.2 show the top important terms of the language models for news events extracted by standard LDA and by our approach respectively. One may observe that none of the language models extracted by LDA are really meaningful and unequivocally associated with actual news events that happened in that time span. For example, there are general terms non-relevant to any of the top events in the models extracted by LDA such as *said*, *style*, *news*, *great*, *time*, *online*, *list* even when we rank words by $p(event|word)$[10]. On the other hand, the language models discovered by our approach are clearly meaningful and strongly associated with the news events at the top level of the taxonomy. There are hardly any trivial words (except perhaps for *lot*). So, with some simple additional input (i.e., short titles for news events in the taxonomy), our approach provides much more useful results over LDA.

We remind the reader that our approach begins with a short label (title) for each event. In comparison with the information in the titles alone, our language

[10] We also empirically tried some stricter ways to lower the roles of backgrounds words (e.g. using raw $df(w)$ instead of $log(df)$ for the prior $p(w)$). Top words ranked by $p(event|word)$ still do not show semantic coherence or any association with the news events

Event 1		Event 2		Event 3		Event 4		Event 5	
w	$p(w \mid z)$	w	$p(w \mid z)$	w	$p(w \mid z)$	w	$p(w \mid z)$	w	$p(w \mid z)$
said	0.006	don't	0.012	day	0.009	new	0.013	other	0.006
span	0.006	know	0.012	time	0.008	video	0.006	online	0.0048
new	0.006	think	0.01	great	0.006	please	0.005	work	0.0042
style	0.006	really	0.009	week	0.006	free	0.005	buy	0.0039
top	0.005	people	0.008	home	0.006	power	0.005	time	0.0037
news	0.005	want	0.007	night	0.006	page	0.004	need	0.0037
color	0.005	ve	0.007	first	0.006	call	0.004	people	0.0036
obama	0.004	time	0.007	good	0.005	black	0.004	help	0.0034
state	0.004	love	0.007	today	0.005	email	0.004	use	0.0033
year	0.004	things	0.006	next	0.005	system	0.004	only	0.0032

Event 1		Event 2		Event 3		Event 4		Event 5	
w	$p(z \mid w)$	w	$p(z \mid w)$	w	$p(z \mid w)$	w	$p(z \mid w)$	w	$p(z \mid w)$
span	0.03	don	0.024	day	0.02	new	0.021	online	0.0162
style	0.022	know	0.023	week	0.016	video	0.019	buy	0.0134
img	0.018	think	0.02	night	0.015	power	0.015	other	0.0113
color	0.017	really	0.019	great	0.015	page	0.015	company	0.0108
obama	0.016	people	0.017	home	0.014	please	0.015	self	0.0097
mccain	0.016	love	0.017	time	0.014	free	0.015	blog	0.0095
top	0.016	want	0.016	water	0.013	vmware	0.015	work	0.0094
said	0.015	ve	0.016	next	0.012	orlistat	0.014	help	0.0092
news	0.014	things	0.015	first	0.011	black	0.013	information	0.0091
palin	0.013	re	0.015	today	0.011	web	0.012	list	0.0088

Table 6.1: Event language models in the general domain discovered by LDA. (a): ranked by $p(w|z)$ and (b): ranked by $p(z|w)$

models offer a richer representation of the news events. For example, the language model for the news item with title "US Presidential Election" shows many other important terms including significant names involved in the election race - *Obama*, and *McCain*, as well as other keywords dealing with the event - *vote*, *elect*, and *poll*. For the "2008 Summer Olympics", our model also finds other semantically related terms such as *Beijing*, *China*, *game*. Similarly, for the "Russia-Georgia War" event, our model also finds relevant terms including *South* and *Ossetia*, which together comprise a common synonym for the conflict. So, the extracted event language models do capture a variety of terms relevant to the news events. Moreover, these relevant terms are also assigned weights reflecting their relative importance with respect to the

US Presidential Election		Financial Crisis		2008 Summer Olympics		Russia-Georgia War		2008 Hurricanes, Tropical Storms	
w	$p(w\|z)$	w	$p(w\|z)$	w	$p(w\|z)$	w	$p(w\|z)$	w	$p(w\|z)$
election	0.058	financial	0.085	olympics	0.088	georgia	0.071	hurricane	0.08988
presidential	0.048	crisis	0.070	summer	0.050	russia	0.063	storm	0.08785
obama	0.022	bank	0.021	beijing	0.042	war	0.049	tropical	0.05444
vote	0.021	market	0.013	ceremony	0.016	russian	0.041	atlantic	0.01594
candidate	0.015	economy	0.012	gold	0.016	georgian	0.028	season	0.01467
mccain	0.013	economics	0.008	game	0.015	south	0.025	ocean	0.01368
barack	0.011	wall	0.008	china	0.014	nato	0.022	warm	0.01295
democrat	0.010	global	0.007	picture	0.013	ossetia	0.021	gustav	0.01172
november	0.008	street	0.007	team	0.013	military	0.019	wind	0.01095
lot	0.007	obama	0.005	medal	0.013	saakashvili	0.014	forecast	0.01057

Table 6.2: Event language models in the general domain discovered by the proposed model

event. As we will show in a later section on language drifts, our approach is capable of automatically adjusting these weights over time as discussion on a topic evolves.

Second, we investigate extracted language models for the three events within the special domain of financial crisis (this is at the second level of the taxonomy). For LDA, to extract language models in the domain of financial crisis, we run LDA on 10000 documents belonging to this domain [11] with the same value of *K=3* to make the results comparable. Tables 6.3 and 6.4 show the language models produced by the standard LDA and by our approach for the sub-events in the domain of “Financial Crisis”. Table 6.3 presents both $p(w|z)$ and $p(z|w)$ results. Notice in table 6.3(a) that the language models produced by LDA again rank very general terms like *said* and *know* quite high. Also ranked very high are terms belonging to the super event such as *crisis*, *financial*. The later terms are still identified as relevant to the sub-events but in reality these are not that important as they do not help to distinguish between

[11] These documents are the top ranked ones in the document set returned by our Lucene search engine when we submit the domain title as query

sub-events of the same domain. When ranking words by $p(event|word)$, the result in Table 6.3(b) shows that the top words are still not as semantically meaningful as in our approach. Only the top words in the first column of Table 6.3(b) seem to correspond to the sub-event "Federal Takeover of Fannie Mae Freddie Mac", the other columns are not really meaningful. Our approach, on the other hand, highlights terms that specifically represent the meaning of the sub-events. Our experimental results on the other specific domains ("US Presidential Election" and "2008 Hurricanes") reveal similar findings. Due to the space limit, these results are not shown here.

To summarize, the findings in this section are two-fold. First, although LDA has been shown to be able to discover meaningful topic language models in other domains, it fails to do so in social media such as blogs. We hypothesize that the reason may be because news events appear sparsely in the blogsphere. Moreover, blog posts tend to be mixed in their content. Event discussions are often mentioned in combination with other topics (e.g some personal story). Second, this section confirms that our approach is able to rule out terms in training documents that are on other topics. Consequentially, our approach with some simple ontological guidance extracts much more meaningful and interpretable language models compared to LDA. The models are also by far stronger than the minimal label/title assigned in the taxonomy in terms of conveying details about the events. Our recent work [40] shows benefits of the models over the original description in terms of retrieving relevant documents.

Event 1		Event 2		Event 3	
w	$p(w \mid z)$	w	$p(w \mid z)$	w	$p(w \mid z)$
financial	0.018	said	0.006	mccain	0.019
government	0.009	new	0.006	obama	0.014
market	0.008	world	0.005	crisis	0.01
banks	0.008	financial	0.005	said	0.008
crisis	0.008	year	0.004	john	0.007
money	0.007	percent	0.004	people	0.006
mortgage	0.006	global	0.004	campaign	0.006
credit	0.006	crisis	0.003	president	0.006
billion	0.006	other	0.003	bush	0.005
fannie	0.006	state	0.003	house	0.005

Event 1		Event 2		Event 3	
w	$p(z \mid w)$	w	$p(z \mid w)$	w	$p(z \mid w)$
banks	0.01	percent	0.009	mccain	0.023
fannie	0.009	oil	0.007	obama	0.019
government	0.009	global	0.006	campaign	0.009
market	0.008	world	0.006	palin	0.009
mortgage	0.008	year	0.006	john	0.008
freddie	0.008	said	0.006	debate	0.008
money	0.008	state	0.006	senator	0.008
billion	0.007	energy	0.005	bush	0.007
debt	0.007	inflation	0.005	president	0.007
loans	0.007	new	0.005	democrats	0.007

Table 6.3: Language models for sub-events in the Financial Crisis domain discovered by LDA. (a): ranked by $p(w|z)$ and (b): ranked by $p(z|w)$

6.3.3 Extracting Dynamic News Event Model

Given the static topic language models, which are built from the whole dataset, our system refines the models dynamically to reflect the evolution of the topics. Table 6.5 shows top terms ranked by the static language model (output of phase 2 in the overall framework) and by dynamic ones at different points of time for the event 2008 US election. The numbers indicate ranks of words in the static models. Words

Federal Takeover of Fannie Mae Freddie Mac		Lehman Brothers Bankruptcy		Bailout of the US Financial System	
w	$p(w\|z)$	w	$p(w\|z)$	w	$p(w\|z)$
fannie	0.061	lehman	0.11085	bailout	0.05834
freddie	0.058	bankruptcy	0.05339	system	0.03265
mae	0.047	brother	0.04602	financial	0.02706
mac	0.046	file	0.0328	plan	0.01484
mortgage	0.03	bank	0.01981	republican	0.01301
federal	0.022	investment	0.01353	congress	0.01225
takeover	0.021	merril	0.01202	paulson	0.012
company	0.017	barclays	0.01198	bill	0.00947
house	0.016	chapter	0.01196	proposal	0.00885
government	0.013	protect	0.01141	house	0.00852

Table 6.4: Language models for sub-events in the Financial Crisis domain discovered by the proposed model

marked with an asterisk are those that appear in the top 15 positions of that version's language model and not in the static model. These mirror the special characteristics of the discussion about the event in the corresponding time period. On August 23 our model gained terms *joe*, *biden*, *run*, *mate*, and *pick*, which is the day the Obama campaign announced that Biden would become Barack Obama's running mate. A similar pattern occurs in the August 27 version of the language model when Sarah Palin became the official running mate of John McCain. Likewise, during the Republican National Convention (September 1 to 4) we gain *convention*, and notably just for this period of time, *mccain* is first on the list, outranking *obama*.

Similar drifts occur in the *2008 Hurricanes and Tropical Storms* event language model (Table Dynamic Language Models for 2008 Hurricanes). Each time a new storm

static		Aug 23, 2008		Aug 27, 2008		Aug 30, 2008		Sept 3, 2008	
1	election	3	obama	3	obama	6	mccain	6	mccain
2	presidential	*	biden	8	democrat	*	palin	3	obama
3	obama	8	democrat	6	mccain	3	obama	*	palin
4	vote	15	senator	7	barack	8	democrat	14	republican
5	candidate	7	barack	*	clinton	14	republican	*	john
6	mccain	6	mccain	4	vote	*	john	8	democrat
7	barack	2	presidential	*	convention	*	president	*	sarah
8	democrat	*	joe	*	support	*	sarah	*	president
9	november	*	campaign	*	hillary	5	candidate	5	candidate
10	lot	*	run	5	candidate	*	vote	*	campaign
11	moore	5	candidate	14	republican	2	presidential	2	presidential
12	poll	*	mate	*	party	7	barack	*	convention
13	voter	*	president	*	speech	*	party	1	election
14	republican	*	pick	2	presidential	*	pick	*	politics
15	senator	14	republican	*	president	*	politics	7	barack

Table 6.5: Language models discovered during various stages of temporal tracking for event 2008 Presidential Election

approaches, its name rises in the top 15 ranked terms. Other more specific terms also appear describing the geographical regions affected by the storms at the time such as *florida* on August 19 and *jamaica* on August 28. These terms can help us not only identify the events but also determine their location.

6.3.4 Discovering Event Evolutionary Trends

Tracking event social popularities over time could depict the evolutionary trends of the corresponding events. These may yield insights into event evolution such as when discussions tend to start, reach their peak and decline. In this subsection, we demonstrate the effectiveness of our framework in discovering event evolutionary trends. Our framework discovers trends for an event by tracking event popularity (as in the previous sections) for different time points (dates). We compare our approach to a baseline method that follows ideas used by commercial systems mentioned ear-

static		Aug 19, 2008		Aug 28, 2008		Sept 5, 2008		Sept 13, 2008	
1	hurricane	2	storm	2	storm	2	storm	1	hurricane
2	storm	*	fay	8	gustav	1	hurricane	14	ike
3	tropical	1	hurricane	1	hurricane	3	tropical	2	storm
4	atlantic	*	florida	3	tropical	14	ike	9	wind
5	season	3	tropical	*	gulf	4	atlantic	*	coast
6	ocean	9	wind	11	hit	9	wind	*	gulf
7	warm	5	season	*	coast	*	hanna	12	weather
8	gustav	*	rain	12	weather	*	coast	*	area
9	wind	11	hit	10	forecast	*	category	7	warm
10	forecast	12	weather	*	jamaica	*	across	*	water
11	hit	10	forecast	*	mexico	*	cause	15	track
12	weather	*	area	9	wind	*	toward	10	forecast
13	hanna	*	cover	*	katrina	*	gulf	*	name
14	ike	*	water	15	track	8	gustav	*	damage
15	track	7	warm	*	force	11	hit	11	hit

Table 6.6: Language models discovered during various stages of temporal tracking for event 2008 Hurricanes

lier. We do this comparison both qualitatively and quantitatively. Specifically, the baseline computes popularity of an event over a sliding window of time. It does so by counting the number of blog posts in the whole corpus that are relevant to the event normalized by the total number of blog posts in the window. This number of relevant blog posts is determined by the search engine when we submit the event title as a query.

Evaluation of news popularity trends is a challenge. To the best of our knowledge, gold standard judgments on event evolutionary trend discovery are not available. In this experiment, we create reference trends as follows. For each event, we manually compose a query combining different synonyms of the event. We aim to create a good query for the event. For example, the query for the event Summer Olympics is *"Summer Olympics" OR "Beijing Olympics"*. Then, we submit the query to Google Insight to get the event evolutionary trend extracted from Google's massive query logs during the same time span. We use human knowledge gained with the help of news sources to verify the sensibility of returned results. The results are finally normalized to make them comparable with the results extracted by the baseline and our approach.

Figures 6.17 and 6.6 show the evolutionary trends of two of the fourteen events of our study: tropical storm Fay and the Russia-Georgia war (the trends of all other events are shown in the appendix of this thesis). The first shows a case where the trend discovered by our approach is most similar to the reference. As we can see, the two trends are very close. They both peak at around August 18, when the storm made landfall, while the baseline fails to discover the pattern. The second event, the Russia-Georgia war, shows the case where the result of our approach and the reference are most dissimilar. They both peak at August 08, around the time the war began. However, the trend discovered by our approach also peaks at around August 16 which is when a ceasefire agreement was signed. Our trend shows another peak around August 26 when the formal recognition of the independence of South Ossetia and Abkhazia were declared, again the reference trend does not peak at the points of time. We hypothesize this is because of the difference in nature between two datasets on this event.

To quantitatively evaluate our event trend discovery methods, we compute the distances between trends discovered by our proposed approach (or the baseline approach) and the corresponding reference trends. The smaller the distance, the better the approach. We use two distance metrics including traditional Euclidean distance and Dynamic Time Warping (DTW) distance. DTW distance has been widely used for comparing time-series [48]. Compared to Euclidean distance, DTW distance takes into account the fact that there might be some "temporal shifts" in terms of latency related to when events are reflected in various data sources such as weblogs and query logs. In our experiment, we take a conservative approach and constrain the shifts to be less than two days. So, DTW distance more precisely capture the similarity (or dissimilarity) between two trends. Specifically, $DTW(i, j)$, the DTW distance between two trends $t[1...i]$ and $s[1...j]$, is recursively defined as bellows. In the base case, $DTW(0, 0) = 0$.

$$DTW(i,j)=\begin{cases} d(s[i],t[j])+min\{DTW(i,j-1),DTW(i-1,j),DTW(i-1,j-1)\} & \text{if } |i-j| \leq 2 \\ \infty & \text{otherwise} \end{cases}$$

Events	dis(baseline, Reference)	dis(proposedApproach, Reference)
US Presidential election	0.504393731	0.256800119
Financial crisis	0.320325168	0.189976168
Summer Olympics	0.684288949	0.20566907
Russia-Georgia War	0.704218415	0.332715014
Storms and Hurricanes	0.721529741	0.217904874
DNC	0.906344955	0.110265006
RNC	0.91617548	0.179111625
Fannie Mae and Freddie Mac	0.447639249	0.117994975
Lehman Brothers Bankcruptcy	0.132858788	0.106173182
Bailout	0.905596806	0.197111701
Gustav	0.39547927	0.099005976
Hanna	0.792851962	0.125015559
Ike	0.444859403	0.104423943
Fay	0.72392359	0.067348439
Average	**0.614320393**	**0.164965404***

Table 6.7: Euclidean Distance.

Tables 6.7 and 6.8 show the results using normalized Euclidean and DTW

Events	dis(baseline, Reference)	dis(proposedApproach, Reference)
US Presidential election	0.454958575	0.168409836
Financial crisis	0.294147283	0.082851767
Summer Olympics	0.64550821	0.111524527
Russia-Georgia War	0.674162936	0.230288386
Storms and Hurricanes	0.688124456	0.111258279
DNC	0.883033618	0.042674039
RNC	0.898826874	0.102139132
Fannie Mae and Freddie Mac	0.417838573	0.066133343
Lehman Brothers Bankcruptcy	0.124195878	0.02824768
Bailout	0.884227296	0.077429229
Gustav	0.36535711	0.03250454
Hanna	0.746231607	0.050411593
Ike	0.387236497	0.055568469
Fay	0.688225901	0.022491851
Average	**0.582291058**	**0.084423762***

Table 6.8: DTW Distance.

distances, respectively. In these tables, the asterisk symbol (*) indicates statistical significance by paired t-test with p-value < 0.001. We could see that our approach outperforms the baseline in all of the events on both metrics. The average Euclidean distance between evolutionary trends discovered by approach and the reference is 0.165, which is significantly better than the average distance between evolutionary trends discovered by the baseline and the reference (0.614). Similarly, when DTW distance is computed, the average distance between our approach and the reference is 0.084, which is also significantly better than the average distance between the baseline and the reference (0.582).

6.4 Summary

In this chapter, we propose an approach for discovering evolution of a crowd's discourse on news events. We also conduct experiments to demonstrate the effectiveness of the approach. In terms of discovering the evolution of language models, our approach discovers meaningful semantics drifts of the events defined in the taxonomy. We also show that our approach is significantly better than the query-based baseline for discovering event evolutionary trends. Another interesting finding is that there is strong agreement between the trends we discover from the blogsphere and from Google query logs (which is the basis for Google Insight) even though the two social data sources are quite different in nature. They are also very likely to be created by significantly non-overlapping crowds. The blog is made to express people's thinking, while queries are created when people search for information.

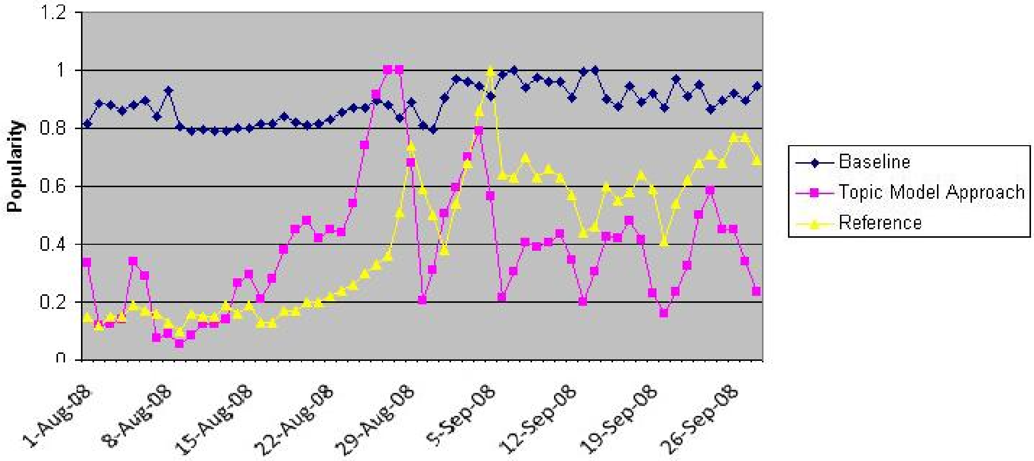


Figure 6.4: US election.

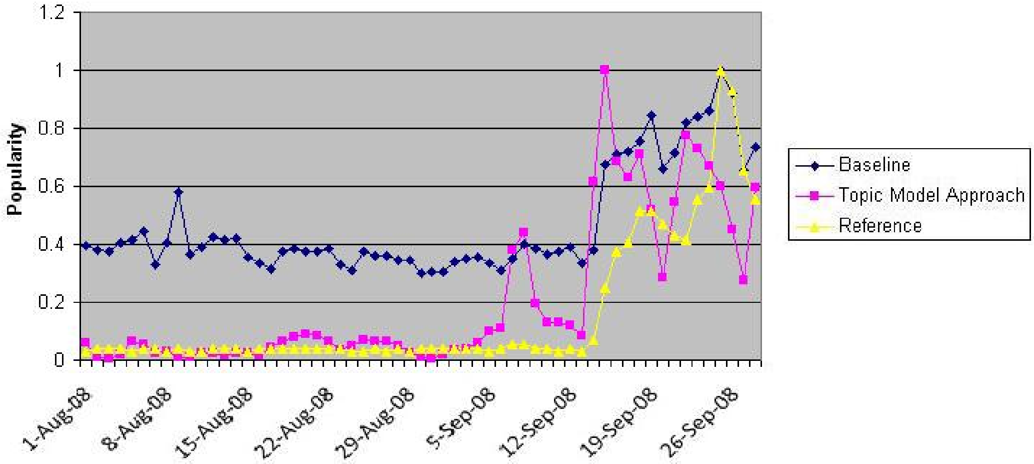


Figure 6.5: Financial crisis.

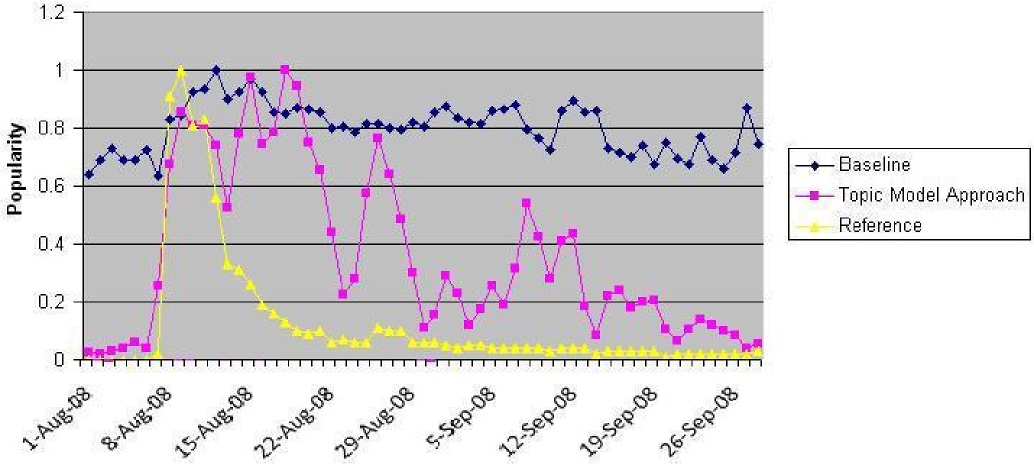


Figure 6.6: Russia-Georgia war.

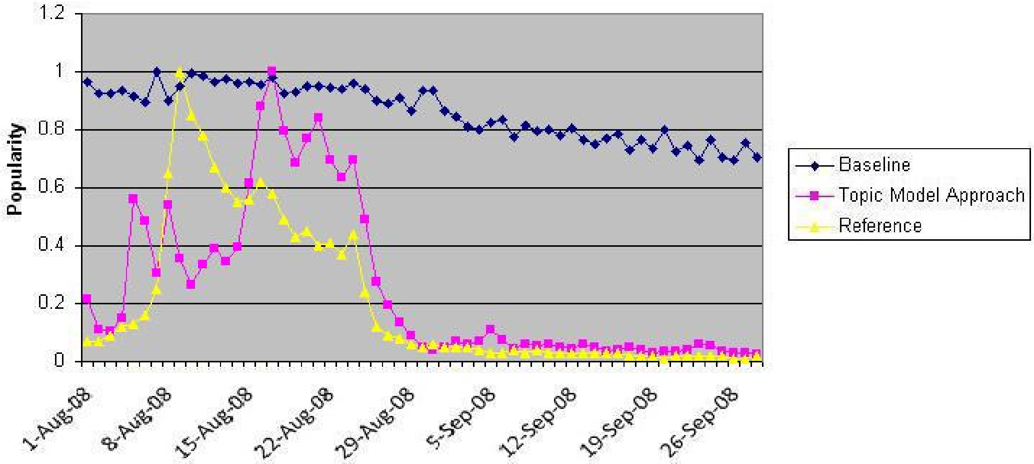


Figure 6.7: 2008 Summer Olympics.

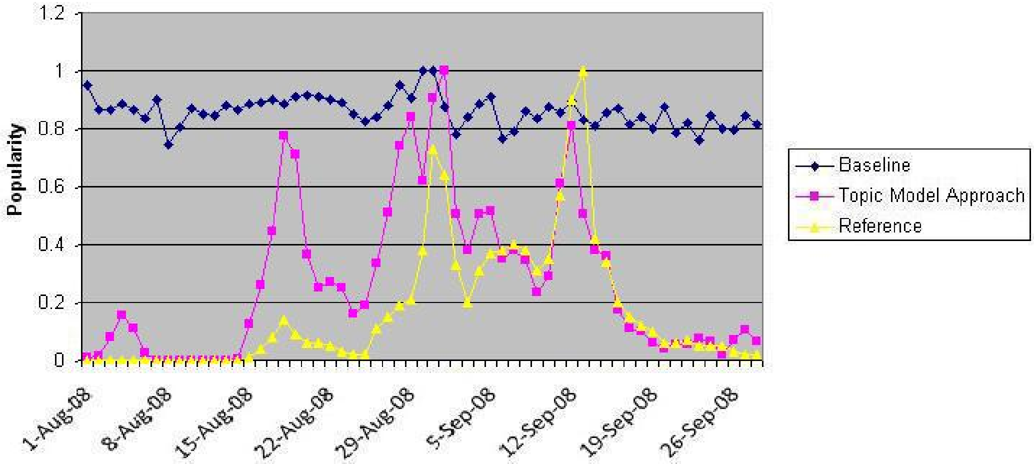


Figure 6.8: 2008 hurricane storms.

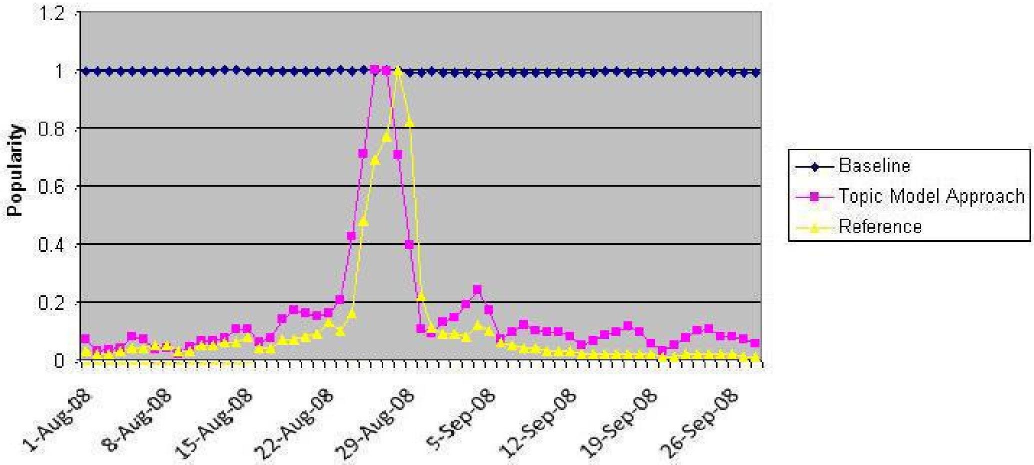


Figure 6.9: Democratic national convention.

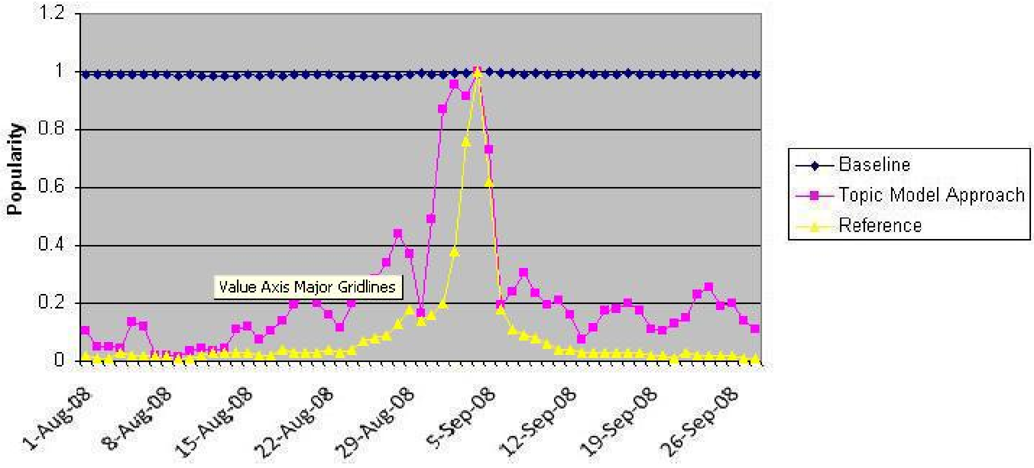


Figure 6.10: Republican national convention.

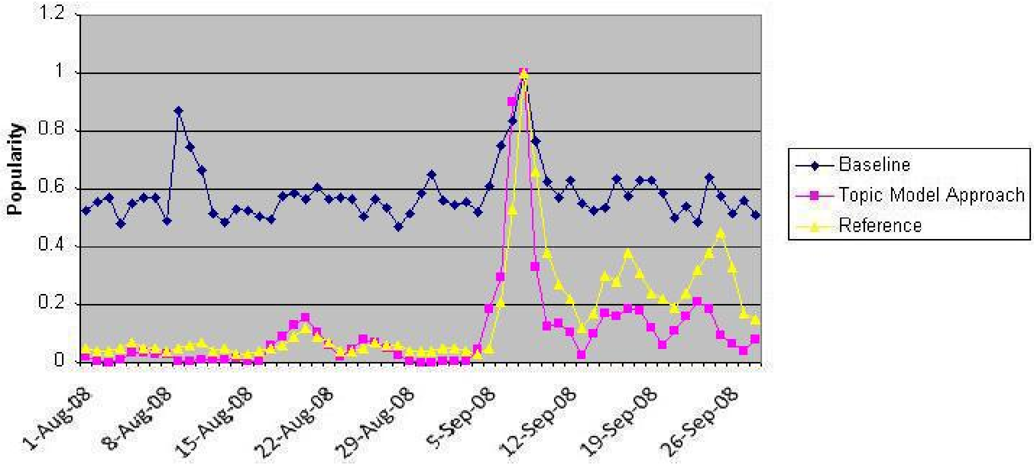


Figure 6.11: Fannie Mae Freddie Mac.

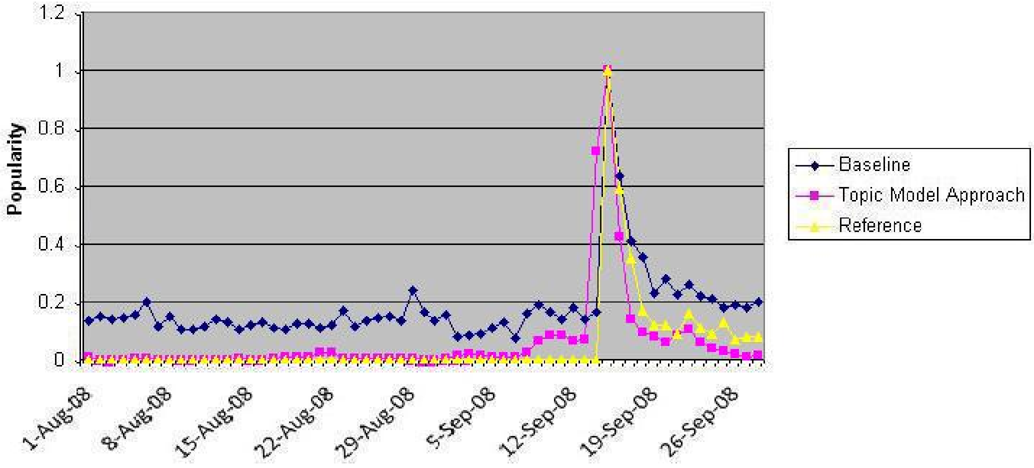


Figure 6.12: Lehman Bothers bankcruptcy.

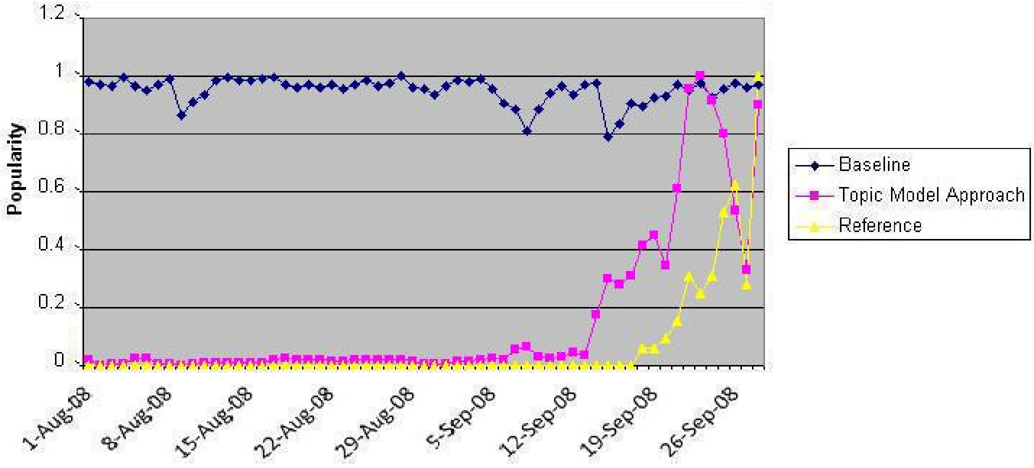


Figure 6.13: US bailout.

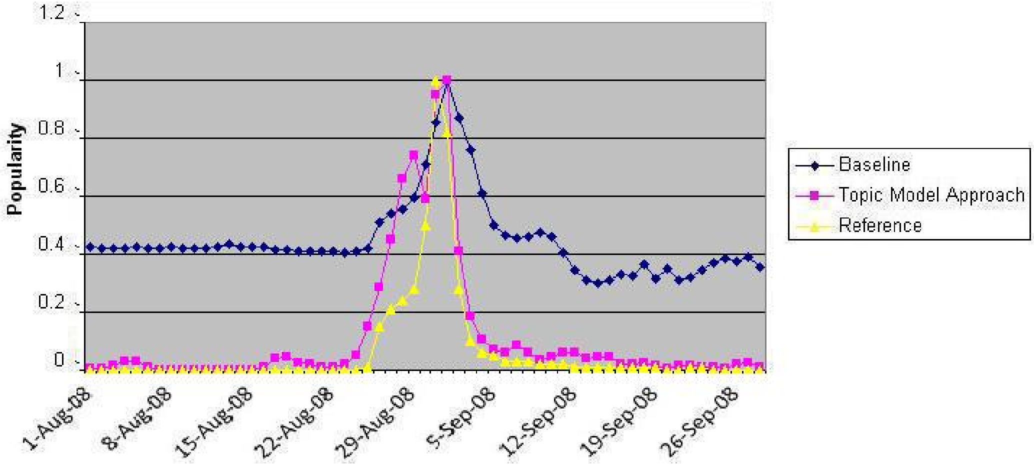


Figure 6.14: Hurricane Gustav.

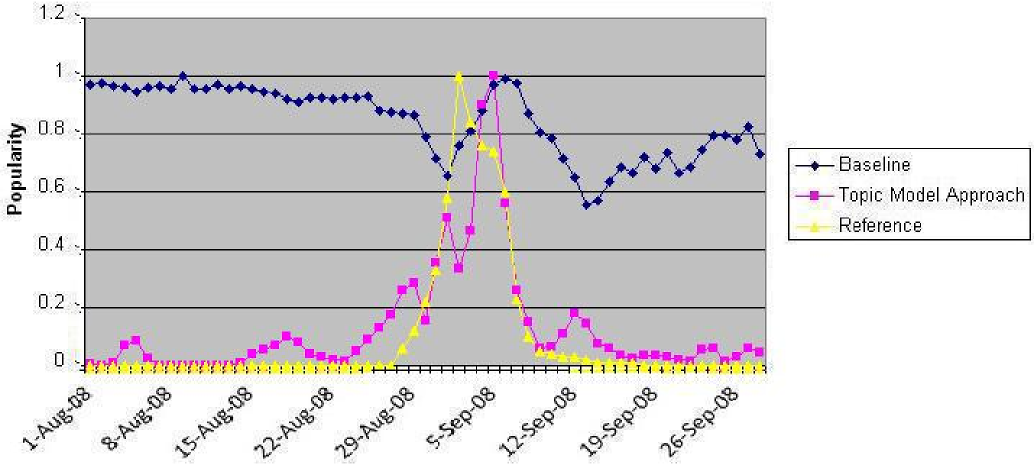


Figure 6.15: Hurricane Hanna.

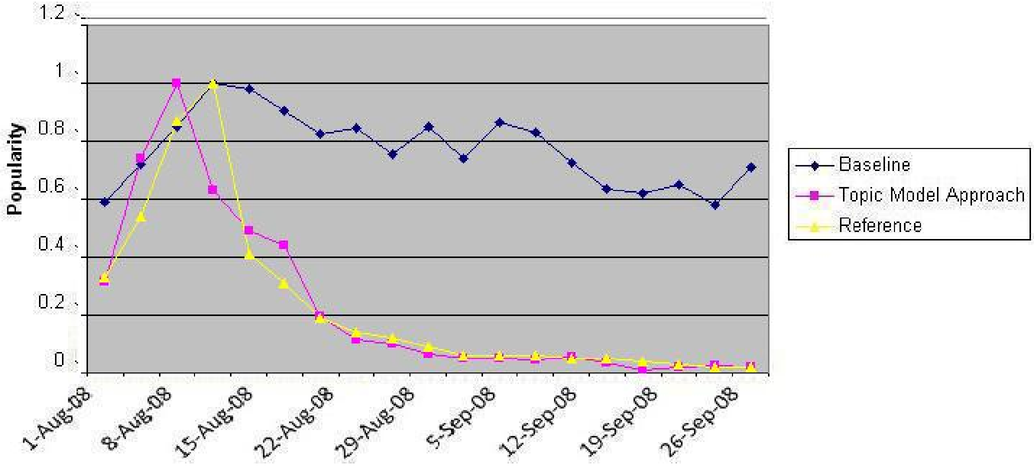


Figure 6.16: Hurricane Ike.

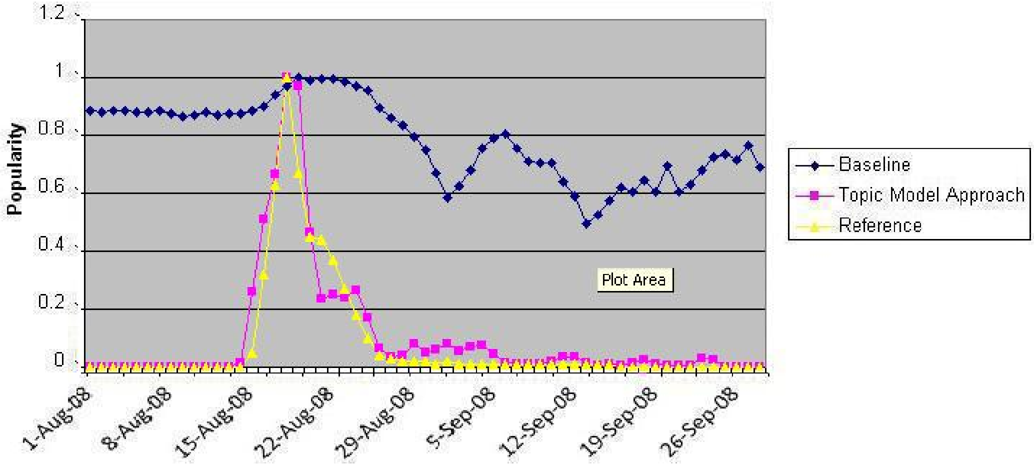


Figure 6.17: Tropical Storm Fay.

CHAPTER 7
CONCLUSIONS

In this dissertation, we propose a novel framework that uses ontological guidance for topic modeling in the Web data. The key advantages of the proposed framework are as follows. First, the framework is able to extract language models that explicitly correspond to the actual topics in the real world. Second, the discourse on each topic is formally modeled by a language model, so it is able to capture the diversity of the language the crowd uses to discuss the topic. Third, the framework overcomes the challenges of sparseness of relevant data by automatically identifying the relevant parts and ruling out the non-relevant parts. Fourth, our approach is adaptive as these language models are estimated dynamically at each time period. Thus, it is able to reflect topical drifts in the discussions about various topics. This advantage is cylic: since the adaptive language models better fit the data in the current time period, they are better able to find portions of documents (e.g. blog posts) relevant to the topic.

This framework can be applied for three crucial problems including retrieving Web documents, hierarchically classifying Web documents, and discovering topic evolutionary trends reflected on the Web. For the first problem, we apply the topic model to extract a language model for each query. Then we take the top terms ranked by the language model to expand the query and submit the expanded query to the search engine again. Our retrieval experiments on five benchmark datasets show that compared to baseline retrieval (without pseudo-relevance feedback), our approach improves on average 39% in terms of mean average precision.

For hierarchical classification, we apply the hierarchical topic model to build a hierarchical classifier from the Web documents, without using any human-labeled documents. Our classification experiment on IPTC (International Press and Telecommunications Council) taxonomy, containing more 1100 topics, shows that our approach achieves a performance of 67% in terms of the hierarchical version of the F-1 measure, without using any labeled data.

Finally, for the task of retrospective discovering topic trends, we again apply the hierarchical topic model to estimate static language models for all topics. Then, we propose an approach to adapt these static topic language models in each time step to make them better fit the data in the current time step. The adaptive language models are then used to infer topic popularity within that time step. This popularity measure indicates the extent to which the corresponding topics are discussed during the time span. In our experiments, using blog data, our approach discovers meaningful insights on how the crowd responds to various news topics such as the language used to discuss each topic, how this language drifts over time, and when the crowd's focus on a topic increases, reaches a peak, and declines.

For future work, we would like to apply the adaptive topic language models to real-time information retrieval. Since the adaptive models capture what happened within the topics dynamically, they could better retrieve documents in real time. Similarly, the adaptive models could also be used for real-time classification. Another direction we also plan to explore is to apply the framework to text applications on other domains such as biomedicine [72]. Finally, given language models for topics in taxonomies, we could measure the similarity between the topics. This information could then be used for modifying a taxonomy and mapping topics between two taxonomies [24, 23, 91, 22].

APPENDIX A
DIRICHLET PROBABILITY DISTRIBUTION

A Dirichlet distribution often denoted by Dir(α), is a continuous multivariate probability distributions over K-dimensional vectors x. Each entry x_i, $(1 \leq i \leq K)$ in vector $\boldsymbol{x}$ is a real number in the interval (0,1), and $||\mathbf{x}||_1 = 1$. The vector x itself could be a K-dimensional multinomial distribution. A Dirichlet distribution is parametrized by a vector α=$(\alpha_1, \alpha_2 \ldots \alpha_K)$ of positive real numbers. One example use of the Dirichlet distribution is if one wants to cut strings (each has initial length of 1.0) into K pieces with different lengths, where each piece has a designated average length, but allowing some variation in the relative sizes of the pieces. Another example is if one represents language models by multinomial distributions, then the multinomial distributions could be assumed to be sampled from a Dirichlet distribution.

The probability density distribution of Dir(α) is in Equation A.1, where the normalizing constant B(α) is the Beta function.

$$f(x_1, \ldots, x_K, \alpha_1, \ldots, \alpha_K) = \frac{1}{B(\alpha)} \prod x_i^{\alpha_i - 1} \tag{A.1}$$

A common special case of the Dirichlet distribution is the symmetric Dirichlet distribution, where all of the elements of the parameter vector α are the same. Symmetric Dirichlet distributions are often used when there typically is no prior knowledge favoring one component over another. Since all elements of the parameter vector are the same, the distribution alternatively can be parametrized by a single scalar value. If this value is 1, the symmetric Dirichlet distribution is equivalent to a uniform distribution.

Dirichlet distributions are very often used as prior distributions in Bayesian statistics, and in fact the Dirichlet distribution is the conjugate prior of the multinomial distribution in the sense that if the prior is a Dirichlet distribution, the likelihood is a multinomial distribution, then the posterior is also a Dirichlet distribution.

REFERENCES

[1] J. Allan. Topic detection and tracking: Event-based information organization, 2002.

[2] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang. Topic detection and pilot study: Final report. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, 1998.

[3] L. AlSumait, D. Barbara, and C. Domeniconi. On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking. In *Proceedings of IEEE ICDM*, 2008.

[4] Christophe Andrieu, Nando de Freitas, Arnaud Doucet, and Michael I. Jordan. An introduction to mcmc for machine learning. *Machine Learning*, 50(1-2):5–43, 2003.

[5] David Andrzejewski, Xiaojin Zhu, and Mark Craven. Incorporating domain knowledge into topic modeling via dirichlet forest priors. In *ICML*, pages 25–32, 2009.

[6] Arthur Asuncion, Padhraic Smyth, and Max Welling. Asynchronous distributed learning of topic models. In *NIPS*, pages 81–88, 2008.

[7] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics).* 1 edition, 2007.

[8] David Blei, Thomas Griffiths, Michael Jordan, and Joshua Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *Advances in Neural Information Processing Systems*. MIT Press, 2004.

[9] David Blei, Andrew Y. Ng, and Michael Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[10] David M. Blei and John D. Lafferty. Dynamic topic models. In *ICML*, 2006.

[11] David M. Blei and Jon D. McAuliffe. Supervised topic models. In *NIPS*, 2007.

[12] Jordan Boyd-Graber and David M. Blei. Multilingual topic models for unaligned text. In *UAI*.

[13] Jordan Boyd-Graber, David M. Blei, and Xiaojin Zhu. A topic model for word sense disambiguation. In *EMNLP*, 2007.

[14] Jun Fu Cai, Wee Sun Lee, and Yee Whye Teh. Improving word sense disambiguation using topic features. In *SEMEVAL*, 2007.

[15] Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *SIGIR*, pages 243–250, 2008.

[16] George Casella. Empirical bayes gibbs sampling. *Biostatistics*, 4(3):485–500, 2001.

[17] George Casella and Edward I. George. explaining gibbs sampler. *The American Statistician*, 46(3):167–174, 1992.

[18] C. Chemudugunta, P. Smyth, and M. Steyvers. Modeling general and specific aspects of documents with a probabilistics topic model. In *Advances in Neural Information Processing Systems 19*, 2007.

[19] E. Costa, A. Lorena, A. Carvalho, and A. Freitas. A review of performance evaluation measures for hierarchical classifiers. In *Evaluation Methods for Machine Learning II: papers from the AAAI-2007 Workshop*, pages 1–6. AAAI Press, 2007.

[20] W. Bruce Croft, Stephen Cronen-Townsend, and Victor Lavrenko. Relevance feedback and personalization: A language modeling perspective. In *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*, 2001.

[21] Laura Dietz, Steffen Bickel, and Tobias Scheffer. Unsupervised prediction of citation influences. In *ICML*, 2007.

[22] AnHai Doan and Alon Y. Halevy. Semantic integration research in the database community: A brief survey. *AI Magazine*, 26(1):83–94, 2005.

[23] AnHai Doan, Jayant Madhavan, Robin Dhamankar, Pedro Domingos, and Alon Y. Halevy. Learning to match ontologies on the semantic web. *VLDB J.*, 12(4):303–319, 2003.

[24] AnHai Doan, Jayant Madhavan, Pedro Domingos, and Alon Y. Halevy. Ontology matching: A machine learning approach. pages 385–404, 2004.

[25] Susan Dumais. Hierarchical classification of web content. In *Proceedings of the 23th SIGIR*, pages 256–263. ACM Press, 2000.

[26] E. Erosheva, S. Fienberg, and J. Lafferty. Mixed-membership models of scientific publication. In *Proceedings of National Academy of Science (PNAS)*, 2004.

[27] Elena Erosheva, Stephen Fienberg, and John Lafferty. Mixed membership models of scientific publications. *PNAS*, 101(Suppl. 1):5220–5227, 2004.

[28] H. Farrell and D. Drezner. Power and politics of blogs, 2008.

[29] E. Gaussier, C. Goutte, K. Popat, and F. Chen. A hierarchical model for clustering and categorising documents. In *Proceedings of ECIR*, 2002.

[30] Sean Gerrish and David M. Blei. A language-based approach to measuring scholarly impact. In *ICML*, 2010.

[31] Alfio Gliozzo, Carlo Strapparava, and Ido Dagan. Improving text categorization bootstrapping via unsupervised learning. *ACM Transactions on Speech and Language Processing (TSLP)*, 6(1), 2009.

[32] Andre Gohr, Alexander Hinneburg, Rene Schult, and Myra Spiliopoulou. Topic evolution in a stream of documents. In *SDM*, pages 859–870, 2009.

[33] Thomas Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101:5228–5235, 2004.

[34] Viet Ha-Thuc, Yelena Mejova, Christopher Harris, and Padmini Srinivasan. Relevance-based topic models for news event tracking. In *Proceeding of the 32th ACM SIGIR*, 2009.

[35] Viet Ha-Thuc, Yelena Mejova, Christopher Harris, and Padmini Srinivasan. News event modeling and tracking in the social web with ontological guidance. In *Proceedings of IEEE International Conference on Semantic Computing*, 2010.

[36] Viet Ha-Thuc, Yelena Mejova, and Padmini Srinivasan. Modeling the crowd's perspectives on news events. *Special Issue of the ACM Transactions on Intelligent Systems and Technology on Computational Models of Collective Intelligence in the Social Web (under review)*, 2011.

[37] Viet Ha-Thuc and Jean-Michel Renders. Large-scale hierarchical text classification without labelled data. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*, 2011.

[38] Viet Ha-Thuc and Padmini Srinivasan. A robust learning approach for text classification. In *Proceeding of the 6th SIAM SDM Workshop on Text Mining*, 2007.

[39] Viet Ha-Thuc and Padmini Srinivasan. Topic models and a revisit of text-related applications. In *Proceeding of the 17th ACM Conference on Information and Knowledge Management - PhD Workshop*, 2008.

[40] Viet Ha-Thuc and Padmini Srinivasan. A latent dirichlet framework for relevance modeling. In *Proceeding of the 5th Asia Information Retrieval Symposium (LNCS)*, 2009.

[41] M. Hearst, M. Hurst, and S. Dumais. What should blog search look like? In *Proceedings of the 2008 ACM Workshop on Search in social media*, 2008.

[42] Djoerd Hiemstra, Stephen Robertson, and Hugo Zaragoza. Parsimonious language models for information retrieval. In *Proceeding of the 27th ACM SIGIR*, 2004.

[43] Matthew Hoffman, David M. Blei, and Francis Bach. Online learning for latent dirichlet allocation. In *NIPS*, 2010.

[44] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of Uncertainty in Artificial Intelligence*, pages 289–296, 1999.

[45] Chen-Ming Hung and Lee-Feng Chien. Web-based text classification in the absence of manually labeled training documents. *JASIST*, 58(1):88–96, 2007.

[46] Jagadeesh J, Prasad Pingali, and Vasudeva Varma. A relevance-based language modeling approach to duc 2005. In *In Document Understanding Conference*. Thomas, 2005.

[47] Jagadeesh Jagarlamudi and Hal Daum III. Extracting multilingual topics from unaligned comparable corpora. pages 444–456, 2010.

[48] Eamonn Keogh. Data mining and machine learning in time series databases. In *Tutorials in ACM SIGKDD*, 2004.

[49] Youngjoong Ko and Jungyun Seo. Learning with unlabeled data for text categorization using bootstrapping and feature projection techniques. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, 2004.

[50] Daphne Koller and Mehran Sahami. Hierarchically classifying documents using very few words. In *Proceedings of ICML*, 1997.

[51] Anastasia Krithara, Massih Amini, Jean michel Renders, and Cyril Goutte. Semi-supervised document classification with a mislabeling error model. In *Proceedings of ECIR*, 2008.

[52] Simon Lacoste-Julien, Fei Sha, and Michael I. Jordan. Disclda: Discriminative learning for dimensionality reduction and classification. In *NIPS*, 2008.

[53] John Lafferty and Chengxiang Zhai. Probabilistic relevance models based on document and query generation. In *Language Modeling and Information Retrieval*, pages 1–10. Kluwer Academic Publishers, 2002.

[54] V. Lavrenko and B. Croft. Relevance-based language models. In *Proceedings of the 24th ACM SIGIR*, 2001.

[55] V. Lavrenko and B. Croft. Relevance models in information retrieval, 2003.

[56] Kyung-Soon Lee, W. Bruce Croft, and James Allan. A cluster-based resampling method for pseudo-relevance feedback. In *SIGIR*, pages 235–242, 2008.

[57] Xiaoyong Liu and W. Bruce Croft. Passage retrieval based on language models. In *Proceedings of the 11th ACM CIKM*, pages 375–382, 2002.

[58] Andrew Mccallum, Ronald Rosenfeld, Tom Mitchell, and Andrew Ng. Improving text classification by shrinkage in a hierarchy of classes, 1998.

[59] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai. Topic sentiment mixture: Modeling facets and opinions in weblogs. In *Proceedings the 16th ACM WWW*, 2007.

[60] Q. Mei and C. Zhai. Discovering evolutionary theme patterns from text - an exploration of temporal text mining. In *Proceedings the 11th ACM SIGKDD*, 2005.

[61] David Mimno, Wei Li, and Andrew McCallum. Mixtures of hierarchical topics with pachinko allocation. In *ICML '07: Proceedings of the 24th International Conference on Machine Learning*. ACM, 2007.

[62] David Mimno and Andrew McCallum. Expertise modeling for matching papers with reviewers. In *KDD*, 2007.

[63] David Mimno and Andrew McCallum. Mining a digital library for influential authors. In *JCDL*, 2007.

[64] David Mimno and Andrew McCallum. Topic models conditioned on arbitrary features with dirichlet-multinomial regression. In *UAI*, 2008.

[65] David Mimno, Hanna Wallach, Jason Naradowsky, David A. Smith, and Andrew McCallum. Polylingual topic models. In *EMNLP*, 2009.

[66] Ramesh Nallapati, William Cohen, and John Lafferty. Parallelized variational em for latent dirichlet allocation: An experimental evaluation of speed and scalability. In *ICDM workshop on high performance data mining*, 2007.

[67] Xiaochuan Ni, Jian-Tao Sun, Jian Hu, and Zheng Chen. Mining multilingual topics from wikipedia. In *WWW*, 2009.

[68] Daniel Ramage, Susan Dumais, and Dan Liebling. Characterizing microblogs with topic models. In *ICWSM*, 2010.

[69] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *EMNLP*, 2009.

[70] Michal Rosen-Zvi, Tom Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *UAI*, 2004.

[71] Gerard Salton, editor. *The SMART Retrieval System - Experiments in Automatic Document Processing.* Prentice Hall, Englewood, Cliffs, New Jersey, 1971.

[72] Viet Ha-Thuc Sanmitra Bhattacharya and Padmini Srinivasan. Mesh: A window into full-text for document summarization. In *Proceedings of ISMB/ECCB*, 2011.

[73] Amit Singhal. Modern information retrieval: A brief overview. 2001.

[74] Alan F. Smeaton and C. J. van Rijsbergen. The retrieval effects of query expansion on a feedback document retrieval system. *Comput. J.*, 26(3):239–246, 1983.

[75] A. Smith, K. Schlozman, S. Verba, and H. Brady. The internet and civic engagement, 2009.

[76] Aaron Smith. The internet's role in campaign 2008, 2009.

[77] Alexander Smola and Shravan Narayanamurthy. An architecture for parallel topic models. In *VLDB*, 2010.

[78] Aixin Sun and Ee-Peng Lim. Hierarchical text classification and evaluation. In *Proceedings of ICDM*, 2001.

[79] Bin Tan, Atulya Velivelli, Hui Fang, and Chengxiang Zhai. Term feedback for information retrieval with language models. In *In Proceedings of the 30 th ACM SIG International Conference on Research and Development in Information Retrieval (SIGIR*, 2007.

[80] Tao Tao and ChengXiang Zhai. Regularized estimation of mixture models for robust pseudo-relevance feedback. In *SIGIR*, pages 162–169, 2006.

[81] Kristina Toutanova and Francine Chen. Text classification in a hierarchical mixture model for small training sets. In *Proceedings of CIKM*, pages 105–113. ACM Press, 2001.

[82] Hanna M. Wallach. Topic modeling: beyond bag-of-words. In *ICML*, 2006.

[83] Kevin Wallsten. Mixedthe blogosphere's influence on political discourse: Is anyone listening. In *Proceedings of Annual meeting of the Midwest Political Science Association*, 2009.

[84] B. Walsh. *Markov Chain Monte Carlo and Gibbs Sampling.* 2004.

[85] Chong Wang, David M. Blei, and David Heckerman. Continuous time dynamic topic models. In *UAI*, 2008.

[86] Pu Wang and Carlotta Domeniconi. Towards a universal text classifier: Transfer learning using encyclopedic knowledge. In *Proceedings of ICDM Workshops*, 2009.

[87] X. Wang, C. Zhai, X. Hu, and R. Sproat. Mining correlated bursty topic patterns from coordinated text streams. In *Proceedings the 13th ACM SIGKDD*, 2007.

[88] Xuerui Wang and Andrew McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *KDD*, 2006.

[89] X. Wei and B. Croft. Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th ACM SIGIR*, 2006.

[90] Robert Wetzker, Tansu Alpcan, Christian Bauckhage, Winfried Umbrath, and Sahin Albayrak. An unsupervised hierarchical approach to document categorization. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pages 52–58, 2007.

[91] Michael L. Wick, Khashayar Rohanimanesh, Andrew McCallum, and AnHai Doan. A discriminative approach to ontology mapping. In *Proceedings of NTII 2008*, pages 16–19, 2008.

[92] Jinxi Xu and W. Bruce Croft. Query expansion using local and global document analysis. In *SIGIR*, pages 4–11, 1996.

[93] Feng Yan, Ningyi Xu, and Yuan Qi. Parallel inference for latent dirichlet allocation on graphics processing units. In *NIPS*, 2009.

[94] Chengxiang Zhai and John Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *In Proceedings of Tenth International Conference on Information and Knowledge Management*, pages 403–410, 2001.

[95] Congle Zhang, Gui-Rong Xue, and Yong Yu. Knowledge supervised text classification with no labeled documents. In *Proceedings of PRICAI*. Springer, 2008.

[96] Yi Zhang and Jamie Callan. Novelty and redundancy detection in adaptive filtering. In *Proceeding of the 25th ACM SIGIR*, pages 81–88, 2002.

[97] Bing Zhao and Eric P. Xing. Bitam: Bilingual topic admixture models for word alignment. In *ACL*, 2006.

[98] D. Zhou, E. Manavoglu, J. Li, L. Giles, and H. Zha. Probabilistic models for discovering e-communities. In *Proceedings of of the 15th ACM WWW*, 2006.

[99] Jun Zhu and Eric P. Xing. Conditional topic random fields. In *ICML*, 2010.