Semantic Web Topic Models: Integrating Ontological Knowledge and Probabilistic Topic Models

by

Mehdi Allahyari

(Under the Direction of Krys Kochut)

Abstract

Currently we are coping with a plethora of text (more than 80% of web) generated and disseminated globally on the web, and thanks to new technologies such as smart devices and social networks it keeps growing exponentially every day. This tremendous amount of text mostly unstructured is easy to be processed and perceived by humans, but significantly hard for machines to understand. Needless to say, this volume of text is an invaluable source of information and knowledge. Thus, there is an increasing need to design methods and algorithms in order to effectively process this sheer volume of text and extract high quality information in an automatic fashion. Probabilistic topic models are a class of latent variable models for textual data that can be used to produce interpretable summarization of documents in the form of their constituent topics. However, because topic models

are entirely unsupervised, they may create topics that are not always meaningful and understandable to humans. In this dissertation, we develop novel topic models that combine probabilistic topic modeling with domain knowledge in the form of ontologies within a single framework. These models effectively enhance topic modeling process and produce the topics that are best aligned with user modeling goals. We first describe the ontology-based topic model, OntoLDA, in which a document is a mixture of topics where as topics are distributions over the ontology concepts and concepts are multinomial distributions over the words. We demonstrate the utility of this model in order to automatically generate labels for the topics. We next propose the sOntoLDA topic model which combines the DBpedia ontology with the topic modeling and use this model for semantic tagging of web documents. For all these models, we develop learning algorithms and show their usefulness with experiments conducted on real-world datasets.

INDEX WORDS: Semantic Web, Ontologies, DBpedia, Topic models, Statistical learning, Domain knowledge

Semantic Web Topic Models: Integrating Ontological Knowledge and Probabilistic Topic Models

by

Mehdi Allahyari

B.S., University of Kashan, 2005

A Dissertation Submitted to the Graduate Faculty
of The University of Georgia in Partial Fulfillment
of the
Requirements for the Degree

Doctor of Philosophy

Athens, Georgia

2016

Semantic Web Topic Models: Integrating Ontological Knowledge and Probabilistic Topic Models

by

Mehdi Allahyari

Approved:

Major Professor: Krys Kochut

Committee: John Miller
Will York

Electronic Version Approved:

Suzanne Barbour
Dean of the Graduate School
The University of Georgia
August 2016

Semantic Web Topic Models: Integrating Ontological Knowledge and Probabilistic Topic Models

Mehdi Allahyari

June 2016

For my parents and my family:

Parisa, and Ryan

Acknowledgments

Over the past six years I have received support from a great many individuals. I owe my appreciation to all these people without whom this dissertation could not have been completed.

First and foremost I want to express my sincere appreciation and gratitude to my advisor Dr. Krys Kochut who has been a mentor, colleague and friend. His willingness to give me the freedom to explore research on topics for which I have been truly passionate, and at the same time his guidance to recover when I faltered made my Ph.D. experience productive and rewarding.

I am also deeply grateful to other professors in my advising committee, Dr. John Miller and Will York, for their encouragement and valuable advice during the years I was striving to advance my research.

Most importantly, I owe my deepest gratitude to my wife Dr. Parisa Darkhal and my parents for their unflagging love and continuous support throughout my life and during my studies to whom this dissertation is dedicated.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Extracting high quality and useful information from massive and constantly growing collections of text documents has become a challenging task and has gained a great deal of attention in recent years. This tremendous amount of text data, which is often unstructured, is created in a variety of forms such as social networks, web and other type of information-centric applications. Understanding and modeling the content of documents can be very beneficial in many applications like information retrieval, natural language processing, document classification, document summarization, etc. Consequently, there is an increasing need to design methods and algorithms in order to effectively process this avalanche of text in a wide variety of text applications. Probabilistic topic models are being widely used to address these complex problems by virtue of their sound theoretical foundations in statistics and for their capability to be extended and combined with other models in a systematic manner.

Probabilistic topic models such as Latent Dirichlet Allocation[17] are powerful techniques to analyze the content of documents and extract the underlying topics represented in the collection. Topic models usually assume that individual documents are mixtures of one or more topics, while topics are probability distributions over the words. These models have been extensively used in a variety of text processing tasks, such as word sense disambiguation [47, 20], relation extraction [121], text classification [42, 65], and information retrieval [117]. Thus, topic models provide an effective framework for extracting the latent semantics from the unstructured text collection. In addition to modeling textual data such as webpages, news articles, emails, scientific and medical publications, etc. [86, 67, 108, 94, 100, 111], topic models have demonstrated to be useful in modeling non-textual data like image collections [13, 7, 119].

However, due to the fact that topic models are entirely unsupervised, purely statistical and data driven, they may produce topics that are not always meaningful and understandable to humans. In other words, the discovered topics may not always correspond to what the user had in mind. We introduce a mechanism to cope with this issue. We develop topic models that allow the user to impact and guide the learned topics, while still maintaining the statistical pattern discovery abilities, which makes the topic models a powerful tool.

In this dissertation, we first propose an ontology-based method for automatic

document classification into dynamically defined topics of interest. We investigate what benefits can be obtained by taking advantage of ontologies compare to using traditional supervised machine learning algorithms. We next explore how to integrate prior knowledge in the form of ontology to the topic modeling framework. We propose *knowledge-based* topic models that incorporate domain knowledge to guide the topic identification process. We particularly develop topic models that combine the ontological knowledge bases such as DBpedia ontology and Linked Open Data (LOD) with probabilistic topic models to benefit the best of the two worlds. Integration of prior knowledge with topic models enhances the effectiveness of topic modeling and can be very helpful by directing the model towards the topics that are best aligned with user modeling goals, when multiple candidate topic decompositions exist for a given corpus of documents.

The main contributions of this research can be summarized as follows:

- We identify the restrictions of using traditional supervised learning techniques for document classification task. We introduce an ontology-based method for automatic text document classification into dynamically defined topics. We show the benefits of our method over traditional methods and demonstrate its effectiveness through comprehensive evaluation.

- We introduce a new ontology-based topic model called OntoLDA topic model for automatic topic labeling task. It captures the relationships between the ontology concepts and the learned topics from text corpora, and generates labels for the topics relying on the ontological concepts.

- We develop a collapsed Gibbs sampling inference algorithm for the OntoLDA topic model.

- We evaluate the effectiveness of ontology-based topic models by conducting extensive experiments of different datasets.

- We develop a knowledge-based topic model called sOntoLDA topic model, which creates semantic tags for Web resources and online documents. It systematically combines the prior knowledge from the DBpedia ontology with the statistical topic models in a principled manner. Furthermore, it captures the distributions of DBpedia categories (concepts) over the words of the document collection.

- We create a collapsed Gibbs sampling inference algorithm for the sOntoLDA topic model.

- We show how sOntoLDA topic model functions though illustrative examples and demonstrate the utility of it by running comprehensive evaluations on multiple datasets.

1.1 Outline

The reminder of this dissertation is organized as follows:

- We formally define Latent Dirichlet Allocation (LDA) topic model and inference algorithms for topic models in Chapter 2. This chapter also describes several primary concepts associated with the Semantic Web.

- We survey a wide variety of related work from extensions and modifications researchers have carried out on the standard LDA model, to some of the recent research on exploiting prior knowledge in topic models in Chapter 3.

- In Chapter 4, we describe a motivating example to answer questions such as "How ontologies can benefit probabilistic topic models?" and "how domain knowledge from the ontologies can be integrated with unsupervised topic models?"

- Chapter 5 describes an ontology-based method for text classification into dynamically defined set of topics. In this method, ontology is the effectively the classifier and not only it does not require a training set, but also allows the user to change the topics of interest without re-training the classifier.

- Chapter 6 and 7 describe different mechanisms for the integration of ontological domain knowledge with unsupervised topic models. Chapter 6 introduces an ontology-based topic model for the task of automatic topic modeling, and illustrates the theory behind it and how it functions. This chapter additionally demonstrates experiments showing usefulness of ontology-based topic models. In Chapter 7, we describe a knowledge-based topic model for tagging documents in an automatic way. We discuss the generative process of this model and provide the inference algorithm relying on the collapsed Gibbs sampling. Moreover, we conduct extensive experiments showing the utility and robustness of this model.

- Chapter 8 concludes the dissertation, summarizing the contributions and

describing directions for further research building on the foundations established in this work.

Chapter 2

Background

In this section, we formally describe some of the related concepts and notations. We begin with the formal definition of several of fundamental Semantic Web concepts. We then, formally define Latent Dirichlet Allocation (LDA), the state-of-art probabilistic topic modeling technique.

2.1 Semantic Web

The Semantic Web is considered as an extension to the current Web through standards by the World Wide Web Consortium (W3C)[1] and was initiated by Tim Berners-Lee and formally introduced to the world by the May 2001 *Scientific American* article "The Semantic Web". The Semantic Web brings structure to the Web content, making the information not only human readable but also representing it in a form that is machine-processable [9]. Tim Berners-Lee articulated

[1] `www.w3.org`

a Web that apart from being an infrastructure for documents and their links, it would express a *Web of data* (i.e.,Web with resources with relations), which enables the information to be shared and reused across applications. For example, resources could represent objects such as organizations, people, locations, etc and links between then describe the relationships among them. In order to bring structure to Web and allow information exchange across applications, Semantic Web requires a few technologies. In the following section, we outline a few fundamental Semantic Web technologies that are necessary for achieving the functionality previously mentioned.

2.1.1 Ontology

Ontologies have been designed as a way to express knowledge about a domain in the Semantic Web. Tom Gruber defines an ontology as an "explicit and formal specification of a conceptualization" [39]. We define the concept of *ontology* using the definition presented in W3C's OWL Use Case and Requirements Documents[2] as follows:

An ontology $\mathcal{O}$ formally defines a common set of terms that are used to describe and represent a domain. An ontology defines the terms used to describe and represent an area of knowledge.

According to the definition above, we should mention a few points about on-

[2] `http://www.w3.org/TR/webont-req/`

tology: *1*) Ontology is domain specific, i.e., it is used to describe and represent *an area of knowledge* such as area in education, medicine, etc [122]. *2*) Ontology consists of terms and relationships among these terms. Terms are often called *classes* or concepts and relationships are called *properties*. By virtue of introduction of standard languages and recent advancements in ontology creation, working with ontologies has become a lot easier, which has significantly impacted the knowledge and data integration, exchange and collaborative work. Ontologies can be broadly classified into two categories: (a) Domain-specific ontologies that are important source of knowledge in those particular domains. Biomedical ontologies such as "Gene Ontology (GO)" and "Ontology for Biomedical Investigations[3]" are examples of this type where the first one is an ontology for describing the function of genes and gene products and the latter one is an integrated ontology for the description of life-science and clinical investigations. (b) In contrast, general-scope ontologies that cover multiple domains and include concepts from numerous areas. DBpedia [5], which is an encyclopedic ontology, derived from Wikipedia contains knowledge from biology, science, art, music and many more domains.

2.1.2 Resource Description Framework (RDF)

RDF was originally created in early 1999 by W3C as a standard for encoding metadata. RDF is the Semantic Web's data model, which is designed to make statements about resources, especially web resources, in the form of ⟨subject, predicate, object⟩ expressions. These expressions are called *triples*. For example,

[3] `http://www.obofoundry.org/`

we can express the fact "Barack Obama is the president of the United States." as an RDF triple: ⟨`Barack Obama, isPresidentOf, United States`⟩, where its graph structure is represented in Figure 2.1. RDF allows the structured and semi-structured data to be mixed and shared across different applications.

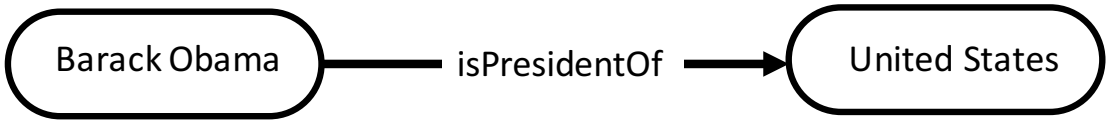


Figure 2.1: Graph structure of the example RDF statement

2.1.3 RDF Schema

RDFS is a set of classes and properties used to describe and encode RDF triples. According the W3C[4] RDFS is formally defined as:

"*RDFS is a recommendation from W3C and it is an extensible knowledge representation language that one can use to create a vocabulary for describing classes, sub-classes and properties of RDF resources.*"

Using this definition, RDFS is the RDF's vocabulary description language. For example, we can define a common vocabulary for various classes (types) of `laptops` and their properties.

[4] `http://www.w3.org/TR/rdf-schema/`

2.1.4 Linked Open Data (LOD)

Linked Data is about creating typed links between data from various sources [10]. In other words, linked Data is a method of publishing structured data in such a way that is interlinked with other data sources. Linked Data is based on the standard Web technologies such as `HTTP`, `RDF` and `URI`.

Tim Berners-Lee illustrated a set of rules for publishing linked data on the web as follows:

1. Use URIs as the identifiers for things.
2. Use HTTP so that the things can be looked up.
3. Provide useful information when people look up a URI, using standards such as RDF, SPARQL, etc.
4. Include links to other URIs, so that they can find more things.

Since Linked Data has been introduced, many publishers have published their datasets in the Linked Data format. These datasets are in different forms such as XML, RDF, CSV, text, etc, and cover multiple domains. As of 2014, the number of datasets on the LOD is over 1014[5]. Figure 2.2 illustrates the most recent image representing the datasets in the Linked Open Data cloud. One of the most important and primary datasets of LOD is `DBpedia` [5, 11]. DBpedia is an ontology containing structured information extracted from the Wikipedia and is publicly available on the Web. The English version of DBpedia knowledge base

[5] `http://linkeddatacatalog.dws.informatik.uni-mannheim.de/state/`

describes 4.58 million things, out of which 4.22 million are classified in a consistent ontology, including 1,445,000 persons, 735,000 places (including 478,000 populated places), 411,000 creative works (including 123,000 music albums, 87,000 films and 19,000 video games), 241,000 organizations (including 58,000 companies and 49,000 educational institutions), 251,000 species and 6,000 diseases[6].

DBpedia knowledge base is very useful and provides many advantages: it covers many domains; because it's extracted from Wikipedia, it automatically evolves as Wikipedia changes; it is multilingual and provides localized versions in 125 languages. Altogether it contains 3 billion pieces of information (RDF triples) out of which 580 million were extracted from the English edition of Wikipedia; because DBpedia is structured, it allows us to ask quite complex queries against Wikipedia. Hence, it should be feasible to leverage this invaluable knowledge in a given data/text mining task. In fact, the rich knowledge sources such as ontologies in the Semantic Web have been extensively utilized in a variety of data mining and knowledge discovery tasks [90].

[6] `http://dbpedia.org/about`

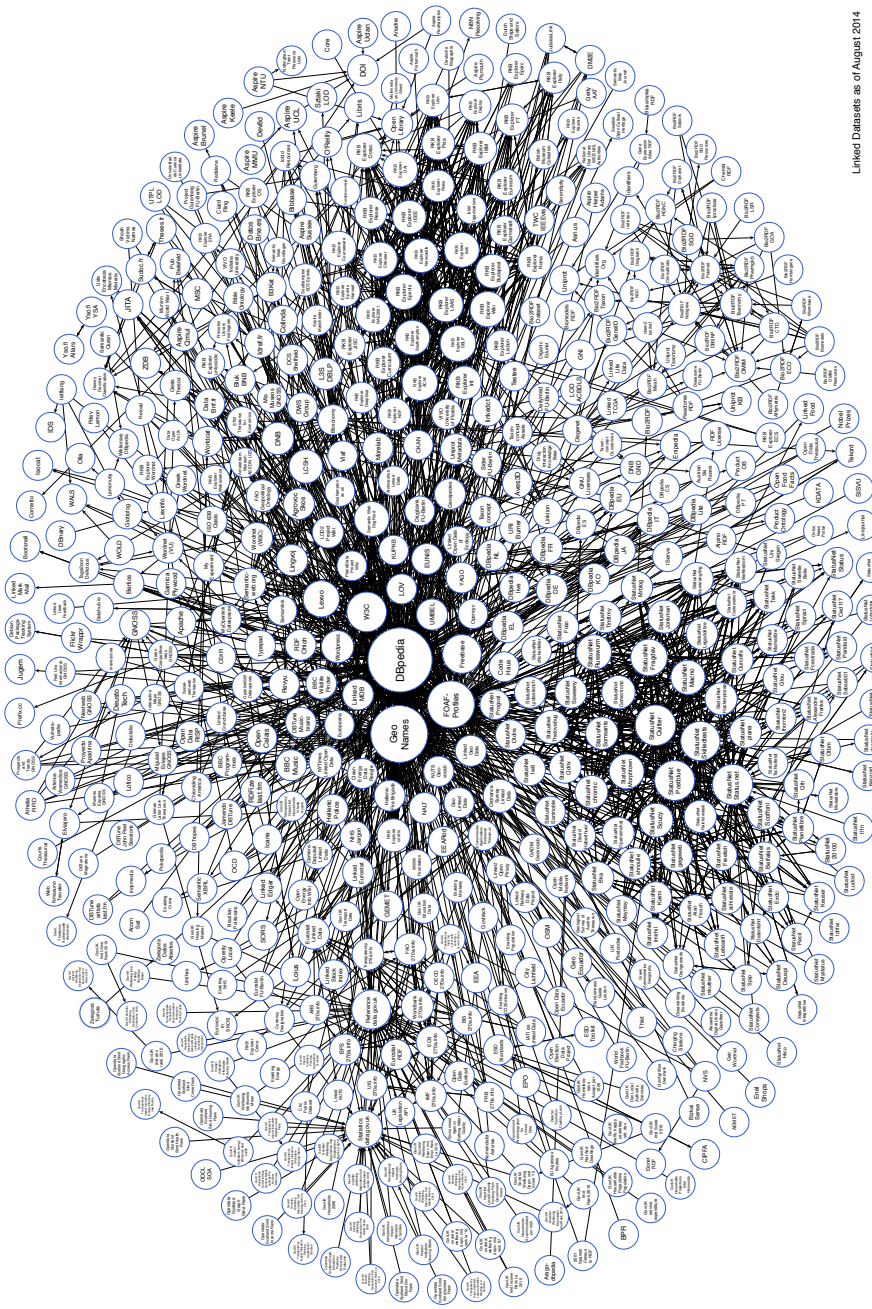


Figure 2.2: Linked Open Data Cloud

2.2 Probabilistic Topic Models

Probabilistic topic models are a set of algorithms that are used to uncover the hidden thematic structure from a collection of documents. The main idea of topic modeling is to create a probabilistic generative model for the corpus of text documents. In topic models, documents are mixture of topics, where a topic is a probability distribution over words. The two main topic models are Probabilistic Latent Semantic Analysis (pLSA) [44] and Latent Dirichlet Allocation (LDA) [17]. Hofmann (1999) introduced pLSA for document modeling. pLSA model does not provide any probabilistic model at the document level which makes it difficult to generalize it to model new unseen documents. Blei et al. [17] extended this model by introducing a Dirichlet prior on mixture weights of topics per documents, and called the model Latent Dirichlet Allocation (LDA). In this section we describe the LDA method.

The Latent Dirichlet Allocation (LDA) [17] is a generative probabilistic model for extracting thematic information (topics) of a collection of documents. LDA assumes that each document is made up of various topics, where each topic is a probability distribution over words.

Let $\mathcal{D} = \{d_1, d_2, \ldots, d_{|\mathcal{D}|}\}$ is the corpus and $\mathcal{V} = \{w_1, w_2, \ldots, w_{|\mathcal{V}|}\}$ is the vocabulary of the corpus. A topic $z_j, 1 \leq j \leq K$ is represented as a multinomial probability distribution over the $|\mathcal{V}|$ words, $p(w_i|z_j), \sum_i^{|\mathcal{V}|} p(w_i|z_j) = 1$. LDA generates the words in a two-stage process: words are generated from topics and topics are generated by documents. More formally, the distribution of words given the

document is calculated as follows:

$$p(w_i|d) = \sum_{j=1}^{K} p(w_i|z_j)p(z_j|d) \tag{2.1}$$

The graphical model of LDA is shown in Figure 7.1 and the generative process for the corpus $\mathcal{D}$ is as follows:

1. For each topic $k \in \{1, 2, \ldots, K\}$, sample a word distribution $\phi_k \sim \text{Dir}(\beta)$
2. For each document $d \in \{1, 2, \ldots, \mathcal{D}\}$,
   (a) Sample a topic distribution $\theta_d \sim \text{Dir}(\alpha)$
   (b) For each word w_n, where $n \in \{1, 2, \ldots, N\}$, in document d,
      i. Sample a topic $z_i \sim \text{Mult}(\theta_d)$
      ii. Sample a word $w_n \sim \text{Mult}(\phi_{z_i})$

The joint distribution of the model (hidden and observed variables) is:

$$P(\phi_{1:K}, \theta_{1:\mathcal{D}}, z_{1:\mathcal{D}}, w_{1:\mathcal{D}}) = \prod_{j=1}^{K} P(\phi_j|\beta) \prod_{d=1}^{|\mathcal{D}|} P(\theta_d|\alpha) \left(\prod_{n=1}^{N} P(z_{d,n}|\theta_d) P(w_{d,n}|\phi_{1:K}, z_{d,n}) \right)$$

2.2.1 Inference and Parameter Estimation for LDA

In the LDA model, the word-topic distribution $p(w|z)$ and topic-document distribution $p(z|d)$ are learned entirely in an unsupervised manner, without any prior knowledge about what words are related to the topics and what topics are related to individual documents.

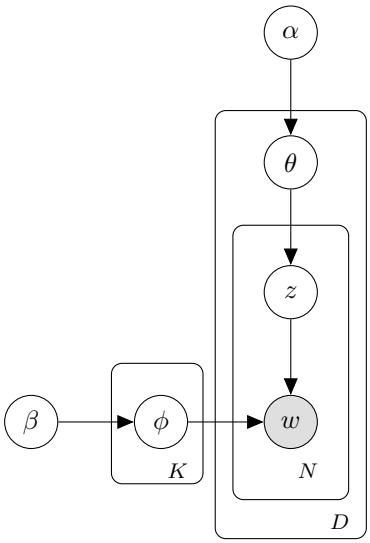


Figure 2.3: LDA Graphical Model

We now need to compute the *posterior* distribution of the hidden variables (topics), given the observed documents. Thus, the posterior is:

$$P(\phi_{1:K}, \theta_{1:\mathcal{D}}, z_{1:\mathcal{D}}|w_{1:\mathcal{D}}) = \frac{P(\phi_{1:K}, \theta_{1:\mathcal{D}}, z_{1:\mathcal{D}}, w_{1:\mathcal{D}})}{P(w_{1:\mathcal{D}})} \tag{2.2}$$

This distribution is intractable to compute [17] due to the denominator (probability of seeing the observed corpus under any topic model).

While the posterior distribution (exact inference) is not tractable, a wide variety of approximate inference techniques can be used, including variational inference [17] and Gibbs sampling [38]. Gibbs sampling is a Markov Chain Monte Carlo [37] algorithm, trying to collect sample from the posterior to approximate it with an

empirical distribution. Gibbs sampling begins with random assignment of words to topics, then the algorithm iterates over all the words in the training documents for a number of iterations (usually order of 100). In each iteration, it samples a new topic assignment for each word using the conditional distribution of that word given all other current word-topic assignments. After the iterations are finished, the algorithm reaches a steady state, and the word-topic probability distributions can be estimated using word-topic assignments.

Gibbs sampling computes the posterior over topic assignments for every word as follows:

$$P(z_i = k | w_i = w, \mathbf{z}_{-i}, \mathbf{w}_{-i}, \alpha, \beta) = \frac{n^{(d)}_{k,-i} + \alpha}{\sum_{k'=1}^{K} n^{(d)}_{k',-i} + K\alpha} \times \frac{n^{(k)}_{w,-i} + \beta}{\sum_{w'=1}^{W} n^{(k)}_{w',-i} + W\beta} \quad (2.3)$$

where $z_i = k$ is the topic assignment of word i to topic k, z_{-i} refers to the topic assignments of all other words. $n^{(k)}_{w,-i}$ is the number of times word w assigned to topic k excluding the current assignment. Similarly, $n^{(d)}_{k,-i}$ is the number of times topic k is assigned to any words in document d excluding the current assignment. For a theoretical overview on Gibbs sampling see [22, 41].

Chapter 3

Related Work

In this chapter, the the most important existing research related to the use of prior knowledge in the topic models are reviewed. The LDA is a well defined probabilistic topic model, which has allowed the researchers to exploit it as a building block for creating customized and richer topic models. We explore a variety of extensions to the standard unsupervised LDA topic model, which use some types of additional information or structure to learn more informative models. We first review some of the existing topic models that deal with various aspects of documents by modeling additional observed information beyond the text of the documents. We then, describe the prior works that have utilized domain knowledge in the topic models.

3.1 LDA-based Topic Models for Modeling Observed Features of Documents

In this section, we discuss extensions to standard LDA model that not only model the text documents, but also incorporate additional observed data in the topic models. This additional data varies from document labels, images associated with documents or links between the documents. Intuitively, the learned topics have to "explain" these additional features as well as the document text. These models have various applications such as labeling documents, annotating images with tags or predicting an unseen links between documents. [15] introduces a supervised LDA (sLDA) for labeled documents. sLDA pairs each document with a response variable y_d, which can be categorical or continuous. This approach jointly models the documents and the responses and can predict responses for an unlabeled test document by calculating the latent topics. Ramage et al. [89] propose a supervised topic model, Labeled LDA (L-LDA), for multi-label corpora. The L-LDA topic model assigns a K-dimensional binary vector of labels Λ to each document. K is the total number of unique labels as well as the number of topics in the Labeled LDA. For example, a document can be tagged with multiple labels such as "business" and "politics". Each one of these labels is associated with its own topic and can only be used in the documents that have that label. Thus, L-LDA incorporates the supervised information by restricting the topic model to pick only those topics that correspond to a document's observed label set. Blei and Jordan [13] propose a Correspondence LDA that jointly models images and their associated captions.

This model finds relationships between the image regions and words. Each latent topic z is a multivariate Gaussian distribution over the image regions and a multinomial distribution over the words for generating captions. Intuitively, this model captures the notion that the image is first created and the caption annotate the image. This model allows interesting applications such as automatic image annotation, automatic region annotation and text-based image retrieval. [109] develops a model for jointly modeling the image, its label and its annotations. It should be noted that considering topic-style topic models applications for vision tasks needs fundamental research literature of their own [34, 113, 116], which is not the focus of this dissertation.

There are existing prior topic models that deal with various aspects of document metadata. For example, in [92], authors integrate the authorship information into the topic model and discover a topic mixture over the documents and authors. In this model, each topic is generated by first sampling an author a, and then sampling a topic z from the topic mixture distribution specific to the author a. Incorporating the authorship with the topics can be used for different applications such as finding the affinity of a reviewer to a paper, assigning reviewers to scientific papers [79] or mining a developer contributions to a given code [68]. There are many types of documents that are inherently inter-linked, such as bibliographic information, citations with scientific articles, weblogs and comments or webpages and links. [24] proposes a topic model for modeling documents and links between them. This model can be used to summarize a network of documents, predict links between them or predict words within them. Liu et al. [69] develop a topic

model that combines topic models and author community discovery in a unified framework. [84] introduces a topic model that combines contentions (agreements) in discussion forums with discussion topics.

Recall that in standard LDA, document-topic distribution θ is independently drawn from a Dirichlet distribution α for each document. This assumption ignores the likely correlations between topics. Another line of related work is the set of topic models that are primarily concerned with the correlations that may exist among topics and the document-topic associations. Hierarchical LDA (hLDA) [16], Correlated Topic Models (CTM) [12], Pachinko Allocation Machines (PAM) [66, 81] are examples of topic models that alter the document-topic sampling process differently in order to capture the correlations among topics. CTM replaces the Dirichlet distribution θ with the logistic normal distribution, which gives a more realistic model of latent topic structure where existence of a topic may correlate with another topic. hLDA learns the hierarchical structures of the topics from data. This model assumes that there is a tree-structured hierarchy over the topics. Each document is generated by choosing a path through the topic tree from the root to the leaf, and each word is assigned to a topic at one of the levels of that path. hLDA expresses the data more accurately by organizing topics into a hierarchy and reflects the underlying the semantic notions of generality and specificity. PAM is another approach to representing the organization of topics into a hierarchy in which each document is a distribution over a single set of topics from root to leaf, using a directed acyclic graph (DAG) to represent topic co-occurrences. Every topic in PAM is a distribution over the sub-topics and a distribution over

the vocabulary. Encoding the connections between topics in the topic models have been exploited in interesting applications such as entity disambiguation [53] and document summarization [23] tasks. Mimno and McCallum [80] propose a Dirichlet-multinomial regression topic model (DMR) that combines text data with document metadata. DMR replaces the document-topic mixture θ witha log-linear prior on document-topic distributions, which is a function of observed features of the document such as, authors, references, publication venues and dates. For each document d, there is a feature vector $\boldsymbol{x}_d$ that encodes metadata values. The advantage of DMR over previous works in metadata-rich topic modeling is that DMR incorporate arbitrary types of features including continuous and categorical ones with no additional coding and with fairly simple inference algorithms.

There is also prior work that integrates *timestamps* (e.g. scientific articles publications dates) with the topic models. It make sense to try to incorporate temporal information into the topic modeling if we can learn something about the evolution of topics and topic trends over time. Blei and Lafferty [14] propose a dynamic topic model (DTM) that analyzes the time evolution of topics in a sequentially organized corpus of documents. DTM assumes that data is divided by time slice, for instance by year. It models the documents of each slice by a K-dimensional topic model, where topics related to slice t are evolved from the topics associated with slice $t-1$. DTM substitutes document-specific topic proportions θ with logistic normal distributions. One obvious restriction of DTM is that it requires the time to be discretized. Wang et al. [110] develop continuous time dynamic topic model (cDTM) replaces the discrete Gaussian evolution with

its continuous generalization, Brownian motion [61]. Wang and McCallum [114] develop a topic over time (TOT) topic model that unlike prior works that rely on Markov assumption or discretized time, the model itself generates the timestamps. In other words, this topic model jointly model the time with word co-occurrences in an explicit way. TOT topic model can be used to predict a timestamp given the words in a document. Another interesting application that TOT model can be used is that by obtaining a distribution over topics, it allows us to see topic occurrence patterns over time.

Another line of related work are topic models that combine topic modeling with the network structure of the data (TMN) using a graph-based regularization framework. Mei et al. [76] develop a method that regularizes statistical topic models with a harmonic regularizer based on the graph structure in the data. TMN leverages the power of both topic modeling and discrete regularization, which optimizes the likelihood of the generation of topics and topic smoothness on the graph together. The regularization framework that TMN utilizes to model topics with the network structure of the data is quite natural: vertices that are connected to one another should have similar weight of topics ($f(\theta, v)$), where f is a weighting function of a topic θ on vertex v. TMN enables a wide variety of applications such as mapping topics onto networks, topical community discovery and spatial text mining. The limitation of TMN is that it can merely integrate with homogeneous information network. [32] proposes a topic model with biased propagation (TMBP) which integrates the heterogeneous information network (i.e.,network with multi-typed objects) with topic modeling in a single framework. TMBP is effectively

applied to object clustering, document modeling, link prediction in multi-relational and heterogeneous networks [120] and user behavior learning in social networks [123].

3.2 LDA-based Topic Models exploiting Prior Knowledge

Standard LDA is entirely unsupervised, purely statistical and data driven, which may produce topics that are not meaningful and understandable to humans. Recently, several *knowledge-based* topic models haven been proposed to cope with this issue. Boyd-Graber et al. [20] develop a LDA-based topic model with WordNet (LDAWN) where the sense of the word is a hidden variable that is inferred from data. Thus, it discovers both the topics of the corpus and the meaning assigned to each of its words. LDAWN replaces the multinomial topic-word distributions with a WordNet-Walk, where WordNet-Walk is a probabilistic process of word generation that relies on the hyponomy relationship in WordNet [78]. LDAWN is used for word sense disambiguation tasks with the basic intuition that words in a topics have similar meanings and therefore share paths within WordNet. Chemudugunta et al. [25] describe a Concept-Topic model (CTM), which combines human-defined concepts with LDA. The key idea in their framework is topics from the statistical topic models and concepts of the ontology are both represented by a set of focused words and they use this similarity in their model. Thus, CTM essentially extends the number of topics by including the

human-generated concepts as special topics. A concept c is represented by a finite subset of W_c words from the vocabulary, where constraints the CTM model to set the probability of the words that are not a priori mentioned in the concept to 0, i.e.,$p(w_i|c) = 0, w_i \notin W_c$. These subsets of the vocabulary can be provided by some source of external knowledge such as Open Directory Project (ODP)[1], a human-edited hierarchical directory of the web. Since CTM assigns concepts at the word level, it allows many appealing applications such as creating summaries for documents at different levels of granularity or labeling documents. In [26], the authors extended the CTM model and propose HTCM, Hierarchical Concept-Topic model, in order to leverage the known hierarchical structure among concepts. HTCM incorporates this hierarchical information to propagate the words upwards in the concept tree, therefore each internal concept node is associated not only with its own words but also is associated with all the words of its children. Andrzejewski et al. [3] propose a topic model, DF-LDA, that integrates the domain knowledge in the form of must-links and cannot-links into LDA. A must-link indicated that two words should be in the same topic, whereas a cannot-link stated that two words should not be in the same topic. DF-LDA encodes the set of must-links and cannot-links associated with the domain knowledge using a Dirichlet Forest prior, replacing the Dirichlet prior over the topic-word multinomial distributions. It allows the user to control the strength of the domain knowledge. In [4], authors propose First-Order Logic LDA topic model (Fold.all), which allows the user to specify general domain knowledge in First-Order Logic (FOL). The primary idea

[1] `http://www.dmoz.org`

is that a domain expert specifies the domain knowledge as First-Order Logic rules, and Fold.all model will automatically incorporate them into LDA and produce the topics influenced by both data and rules. This approach enables domain experts to concentrate on high-level modeling goals rather than low-level issues involved in creating a custom topic model. Patterson et al. [88] leverage word features as side information to boost topic cohesion. The intuition is to treat word information as *features* instead of explicit restriction and to modify the smoothing prior over the topic distributions for words in such a way that correlation is stressed. In this way, we can learn the prior probability of how words are distributed over different topics based on how similar they are. Jagarlamudi et al. [49] introduce topic models that utilize prior knowledge in the form of *seed words* to learn topics of specific interest to a user. Seed words are user provided words that represent the topics underlying the corpus. For example, {"gas", "oil", "products", "petrol"} is a set of seed words representing a seed topic. Seed topic information can be utilized to improve the topic-word probability distributions or it can be first transfered to the document level based on the document words and then be used for enhancing document-topic distributions, or it can be combined at topic and then document level. Interactive Topic Modeling (ITM) [47] allows the user to incorporate knowledge interactively during the topic modeling process. Chen et al. [30] develop LDA with Multi-Domain Knowledge (MK-LDA) topic model that exploits multiple domains knowledge to enhance topic coherency in a new domain. The knowledge is called s-set (semantic-set) and refers to a set of words sharing the same semantic meaning in the domain. Each document is mixture of topics whereas each topic is

a distribution over s-sets. MK-LDA can handle multiple senses because the latent variable for s-sets allows to choose the right sense represented by an s-set. In [28], authors propose GK-LDA, general knowledge-based model, which exploits general knowledge of *lexical semantic relations* in topic model. The general knowledge includes synonyms, antonyms and adjective attributes and is domain independent. GK-LDA utilizes the knowledge of lexical relations in dictionaries in order to deal with the wrong knowledge (i.e.,meaning of a word that is not suitable or correct for a domain). The lexical knowledge is extracted from dictionaries to form a general knowledge and can be applied to any domain without user involvement. Other related works are [31] and [29], which develop topic models that are built upon GK-LDA topic model and have been used for aspect extraction task in sentiment analysis.

Chapter 4

Motivating Example

The standard LDA topic model is entirely *unsupervised*, i.e.,it divides the corpus into latent topics based on a completely data-driven objective function such as maximum likelihood. Thus, topic modeling may produce topics that are list of words that do not bear useful information for human use or not aligned with the user goals. For instance, Table 4.1 presents a few of the topics learned from a collection of news articles along with their high-probability words [87]. The first row is a topic that has associated *Carolina* with *Korea* through the words "north" and "south". The topic in the second row represents "comparisons" and is learned from a large collection of MEDLINE abstracts. The last row shows a topic that consists of names, days of the week and months of the year.

Purely unsupervised LDA topic model may not be able to capture important structure or extract meaningful, interpretable and coherent topics. Thus, researchers have extended LDA and developed richer topic models to address this

Table 4.1: Example learned topics that are less useful or meaningful to the user.

High probability words					Issue
north	south	carolina	korea	korean	north and south combination
effect	significant	increase	decrease	significantly	comparisons
weekend	december	monday	scott	wood	combination of names

issue. Incorporating ontological knowledge allows these topic models to uncover hidden semantic themes (topics) in more effective way. In the following section, we describe how to integrate ontology concepts with the standard LDA topic model to discover more coherent topics and automatically generate semantic labels for them.

4.1 Combining Ontologies with Topic Models

Let's presume that we are given a collection of news articles and told to extract the common themes present in this corpus. Manual inspection of articles is the simplest approach, but it is not practical for large collection of documents. We can make use of topic models to solve this problem by assuming that a collection of text documents comprises of a set of hidden themes, called *topics*. Each topic z is a multinomial distribution $P(w|z)$ over the words w of the vocabulary. Similarly, each document is made up of these topics, which allows multiple topics to be present in the same document. We estimate both the topics and document-topic mixtures from the data simultaneously. When the topic proportions of documents are estimated, they can be used as the themes (high-level semantics) of the doc-

uments. Top-ranked words in a topic-word distribution indicate the meaning of the topic. For example, Table 4.2 shows a sample of four topics with their top-10 words learned from a corpus of news articles.

Table 4.2: Top-10 words for topics from a document set.

Topic 1	**Topic 2**	**Topic 3**	**Topic 4**
company	film	drug	republican
mobile	show	drugs	house
technology	music	cancer	senate
facebook	year	fda	president
google	television	patients	state
apple	singer	reuters	republicans
online	years	disease	political
industry	movie	treatment	campaign
video	band	virus	party
business	actor	health	democratic

However, even though the topic-word distributions are usually meaningful, it is very challenging for the users to accurately interpret the meaning of the topics based only on the word distributions extracted from the corpus, particularly when they are not familiar with the domain of the corpus. Standard LDA model does not *automatically* provide the *labels* of the topics. Essentially, for each topic it gives a distribution over the words. A label is one or a few phrases that sufficiently explain the meaning of the topic. For instance, as shown in Table 4.2, topics do not have any labels, therefore they must be manually assigned. Topic labeling task can be labor intensive particularly when dealing with hundreds of topics. Table 4.3 illustrates the same topics that have been labeled (second row in the table) manually by a human.

Table 4.3: Top-10 words for topics from a document set. The second row presents the manually assigned labels.

Topic 1	**Topic 2**	**Topic 3**	**Topic 4**
"Technology"	"Entertainment"	"Health"	"Politics"
company	film	drug	republican
mobile	show	drugs	house
technology	music	cancer	senate
facebook	year	fda	president
google	television	patients	state
apple	singer	reuters	republicans
online	years	disease	political
industry	movie	treatment	campaign
video	band	virus	party
business	actor	health	democratic

Automatic topic labeling has recently attracted increasing attention [115, 75, 71, 60, 48]. However, all previous works have basically focused on the topics learned via LDA topic model (i.e.,topics are multinomial distribution over words). For example, Mei et al. [75] proposed an approach to automatically label the topics by converting the labeling problem to an optimization problem. Thus, for each topic a candidate label is chosen that has the minimum Kullback-Leibler (KL) divergence and the maximum mutual information with the topic. Table 4.4 presents sample results of topic labeling method described in [75] along with the top-5 generated labels.

We believe that the knowledge in the ontology can be integrated with the topic models to automatically generate topic labels that are semantically relevant,

Table 4.4: Sample topics with top-10 words and top-5 generated labels Mei et al. method.

Topic 1	**Topic 3**	**Topic 7**	**Topic 8**
Label	**Label**	**Label**	**Label**
rice production	cell lineage	hockey league	mobile devices
southeast asia	cell interactions	western conference	ralph lauren
rice fields	somatic blastomeres	national hockey	gerry shih
crop residues	cell stage	stokes editing	huffington post
weed species	maternal effect	field goal	analysts average
Top Words	**Top Words**	**Top Words**	**Top Words**
soil	cell	game	company
control	cells	team	million
organic	heading	season	billion
crop	expression	players	business
heading	al	left	executive
production	figure	time	revenue
crops	protein	games	shares
system	genes	sunday	companies
water	gene	football	chief
biological	par	pm	customers

understandable for humans and highly cover the discovered topics. In other words, our aim is to incorporate the semantic graph of concepts in an ontology (e.g. DBpedia) and their various properties within unsupervised topic models, such as LDA.

We introduce an ontology-based topic model, called *OntoLDA* model, which incorporates an ontology into the topic model in a systematic manner. For complete theoretical backgrounds and learning inference algorithms, see Chapter 6.

The basic intuition behind our model is that topics are distributions over ontology concepts, i.e.,topics are represented using concepts in the ontology, where concepts are distributions over words. Having the *concept* latent variable as another layer between topics and words benefits us in several ways: (1) it gives us much more information about the topics; (2) it allows us to illustrate topics more specifically, based on ontology concepts rather than words, which can be used to label topics; (3) it automatically integrates topics with knowledge bases. Table 4.5 shows the top-5 labels generated by our proposed method for the same topics that are already illustrated in Table 4.4. As can be seen, the labels generated by our proposed model are more understandable, semantically relevant and highly cover the topics.

Table 4.5: Sample topics with top-10 words and top-5 generated labels by our topic labeling method.

Topic 1	**Topic 3**	**Topic 7**	**Topic 8**
Label	**Label**	**Label**	**Label**
agriculture	structural proteins	national football league teams	investment banks
tropical agriculture	autoantigens	washington redskins	house of morgan
horticulture and gardening	cytoskeleton	sports clubs established in 1932	mortgage lenders
model organisms	epigenetics	american football teams in maryland	jpmorgan chase
rice	genetic mapping	green bay packers	banks established in 2000

Chapter 5

Ontology-Based Text Classification[1]

[1]Mehdi Allahyari, Krys Kochut and Maciej Janik. "Ontology-based Text Classification into Dynamically Defined Topics". 2014 IEEE International Conference on Semantic Computing (ICSC), Pages: 273 - 278.
Reprinted here with permission of the publisher.

Abstract

We present a method for the automatic classification of text documents into a dynamically defined set of topics of interest. The proposed approach requires only a domain ontology and a set of user-defined classification topics, specified as contexts in the ontology. Our method is based on measuring the semantic similarity of the thematic graph created from a text document and the ontology sub-graphs resulting from the projection of the defined contexts. The domain ontology effectively becomes the classifier, where classification topics are expressed using the defined ontological contexts. In contrast to the traditional supervised categorization methods, the proposed method does not require a training set of documents. More importantly, our approach allows dynamically changing the classification topics without retraining of the classifier. In our experiments, we used the English language Wikipedia converted to an RDF ontology to categorize a corpus of current Web news documents into selection of topics of interest. The high accuracy achieved in our tests demonstrates the effectiveness of the proposed method, as well as the applicability of Wikipedia for semantic text categorization purposes.

5.1 Introduction

Text categorization is a task of assigning one or more predefined categories to the analyzed document, based on its content. People categorize text documents based on their general knowledge and their interest that determines which facts are treated as more important. While reading a news document we can capture most important actors, facts and places, connecting them into a one coherent event. Computers equipped with proper knowledge represented by an ontology that is comprehensive enough, can spot the same actors and facts in the document. Furthermore, using predefined semantic relationships between recognized entities and knowledge from the ontology, they can construct a model of a presented event, augmenting it with important background facts that are not directly present in the document. The relative importance of facts and entities is determined by the defined ontological contexts (topics) of interest. Leveraging the ontological knowledge in the categorization process not only allows us to eliminate the training step in building a categorizer but also to dynamically change the topics of the categorization without any retraining when the user's interests change.

Traditional supervised text categorization methods use machine learning to perform the task. Most of them learn category definitions and create the categorizer from a set of training documents pre-classified into a number of fixed categories. Such methods, including Support Vector Machines [97], Naïve Bayes [97], decision trees [97], and Latent Semantic Analysis [58] are effective, but they require a set of pre-classified documents to train the categorizer.

In this paper, we propose to use an ontology and dynamically defined ontologi-

cal contexts as classification categories. The novelty of our categorization method is that it does not require a training set of documents divided into a fixed set of categories and relies exclusively on the knowledge represented in the ontology: (1) named entities, relationships between them, entity classification and the class hierarchy and (2) dynamically definable ontology contexts, representing the topics of interest (classification categories).

Since our categorization method relies exclusively on a supplied ontology, in a way, the ontology itself can be regarded as a classifier. Using a general, encyclopedic knowledge-based ontology, such as one derived from Wikipedia, allows us to recognize and classify entities from numerous domains. Furthermore, having the ability to dynamically define classification categories turns such a classifier into a universal text classifier, as we can define our topics of interest as combinations of any existing domains in the ontology.

5.2 Ontology-based Text Categorization

We argue that automatic text classification can be accomplished by relying on the semantic similarity between the information included in a text document and a suitable fragment of the ontology. Our argument is based on the assumption that entities occurring in the document text along with relationships among them can determine the document's categorization, and that the entities classified into the same or similar domains in the ontology are semantically closely related to each other. In order to be able to achieve meaningful results, we require that the

ontology (i) cover the categorization domain(s), (ii) include a rich instance base of named entities and meaningful relationships among them, (iii) have proper labels for named entities that enable their recognition in categorized documents, and (iv) have the entities classified according to a class taxonomy included in the ontology.

A Wikipedia-derived ontology fulfills most of the requirements for text categorization purposes. Its major advantages are in the richness of represented domains, high number of entities, and in the included categorization scheme. Wikipedia was already successfully used for the supervised text categorization [36] and predicting document topics [96]. We successfully used an RDF ontology created from Wikipedia in our previous ontology-based text categorization experiments described in [50] and [51]. A related task of predicting concepts that characterize sets of documents using ontology created based on Wikipedia is presented in [101].

A conversion of Wikipedia into a Wikipedia-based ontology has been done by the DBpedia project [11]. We used a modified method of creating a DBpedia-like ontology to (1) pre-process different ways an entity can be connected to its name variants and (2) introduce properties to represent them. The ontology-based categorization method proposed in this paper can be adjusted to use any encyclopedic-type ontology.

Motivating Example

Let us present a fragment of a recent news article to illustrate the process of ontology-based categorization:

Fiat has **completed** its **buyout** of **Chrysler**, making the **U.S.** business a wholly-

owned **subsidiary** of the **Italian carmaker** as it **gears** up to **use** their **combined resources** to **turn around** its loss-making **operations** in **Europe**. The **company announced** on January 1 that it had **struck** a $4.35 **billion deal** - cheaper than analysts had **expected** - to **gain** full **control** of **Chrysler**, **ending** more than a year of **tense** talks that had obstructed **Chief Executive Sergio Marchionne**'s efforts to create the world's seventh-largest **auto maker** [. . .]

Marchionne said at the **Detroit car show** last **week** that a **listing** of the **combined entity** was on the **agenda** for this year. While **New York** is the most **liquid market**, **Hong Kong** is also an option, the CEO said, pledging to **stay** at the **helm** of the **merged group** for at least three years. The first big **test** for the **merged Fiat-Chrysler** will be a three-year **industrial plan Marchionne** is expected to unveil in May, in which he will **outline planned investments** and **models**. [. . .]

Fiat has said its **new strategy** will **focus** on revamping its **Alfa Romeo** brand and **keeping production** of the **sporty marque** in **Italy** as it **seeks** to utilize **plants** operating below **capacity**, protect **jobs** and **compete** in the higher-margin **premium segment** of the **market**.

To categorize the above article, we identify the entities (underlined) and using the DBpedia-based ontology, induce relationships among them, and introduce the initial semantic graph of connected entities that were recognized in the document. Note that several matched entities do not belong to the main subject of the document and even some can be matched ambiguously (initial entity recognition is based on matching their names occurring in the text). Disambiguation issues are addressed in the subsequent analysis of the graph.

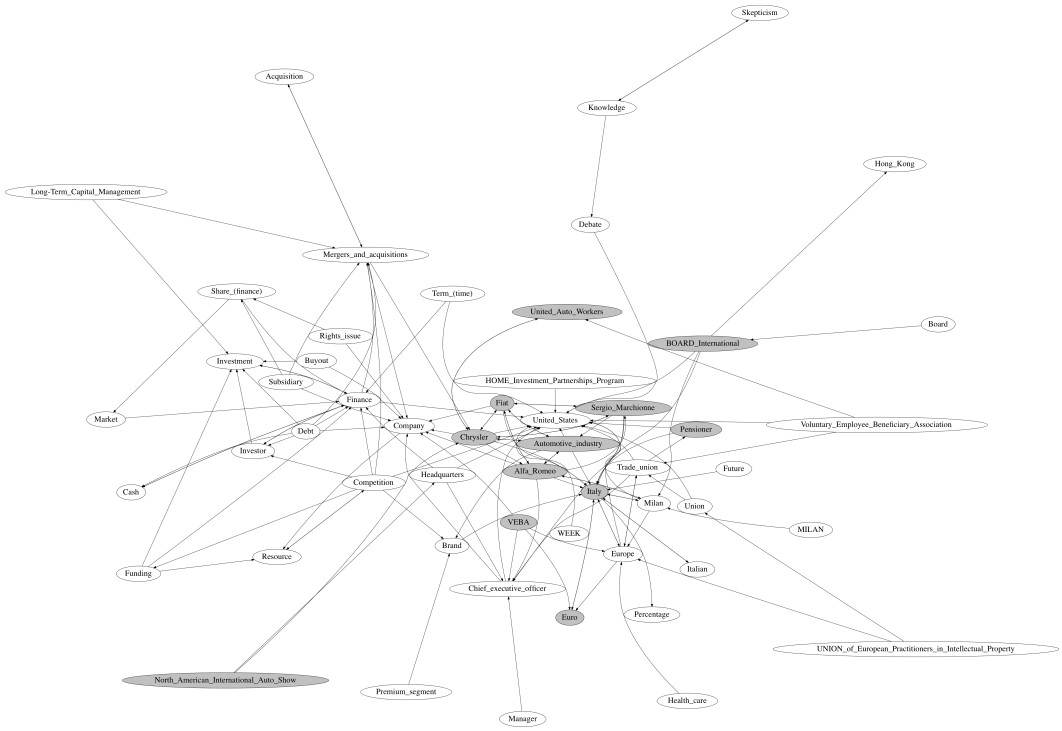


Figure 5.1: Thematic graph from the example text

Distinct connected components in the semantic graph are associated with different domains of interest, as we assume that entities from the same domain are closely connected. The selection of the most important component (a sub-graph of the semantic graph), based on its size and weights of the included entities, not only effectively eliminates entities assigned to different domain(s), but also helps to disambiguate multiple entities that were matched by the same phrase in the text by choosing a specific interpretation context.

Even after the elimination of less important connected components, the thematic graph may contain multiple sub-domains or different interpretations of en-

tities and relationships among them. In order to establish the focus of the categorization process, we choose the core entities in the *thematic graph* (shaded entities in Figure 1). These are the entities discovered as (1) the best hubs and authorities by the HITS algorithm [54], (2) the best entities describing the graph taking into account global information recursively computed from the entire graph by using TextRank algorithm [77] as well as (3) the most central entities in the analyzed thematic graph. They are the starting points of the categorization. The categorization of the thematic graph to classification topics, defined as compositions of ontology contexts, requires measuring the semantic similarity between the supplied context definition and the thematic graph. This similarity measure shows how close the thematic graph is to the selected context. Continuing with the example article, our categorization assigns a number of most likely Wikipedia categories to the analyzed document. The top 5 assigned categories are: "*Stock Market*", "*Automotive Industry*", "*Debating*", "*Corporate Finance*" and "*Legal Entities*".

5.3 Classification Categories

Our text classification method allows dynamic specification of contexts as classification categories. Our definition of a context is to some extent based on the previous research on views in semi-structured databases and ontologies. We define it in terms of an RDF/RDFS ontology.

Definition 1. The *hierarchical distance* between an instance entity e from a de-

scription base R and a class c from an RDFS schema S, denoted as $dist_H(e, c)$, is defined as the length of the shortest path formed by one *rdf:type* and zero or more *rdfs:subClassOf* properties connecting e and c. In case the entity e is not an instance of class c (directly or via the *rdfs:subClassOf* properties), $dist_H(e, c)$ is set to 0.

By extension of Def. 1, the hierarchical distance between an instance entity e and a set of classes C, denoted as $distH(e, C)$, is defined as the minimum, positive value among all $distH(e, c)$, where $c \in C$ and $distH(e, C)$ is set to 0 otherwise.

Definition 2. Let C be a set of schema classes included in an RDFS schema S. A *projection* of classes C onto an RDF description base R is a set of instance entities in R paired with their corresponding hierarchical distances to C, defined as:

$$\Pi(C, R) = \{e(k) : e \in R \wedge k = dist_H(e, C) \wedge k > 0\} \tag{5.1}$$

Definition 3. A *categorization context (topic)* defined by a set of schema classes C is a projection of C onto R.

Definition 4. Given two categorization contexts m_1 and m_2, the following *context expressions* are also categorization contexts:

$$m_1 \cap m_2 \equiv \text{intersection of contexts}$$

$$\equiv \{e(k) : e(k_1) \in m_1 \wedge e(k_2) \in m_2 \wedge k = min(k_1, k_2)\} \tag{5.2}$$

$$m_1 \cup m_2 \equiv \text{union of contexts}$$

$$\equiv \{e(k) : (e(k_1) \in m_1 \vee e(k_2) \in m_2) \wedge k = min(k_1, k_2)\} \tag{5.3}$$

$$m_1 \backslash m_2 \equiv \text{difference of contexts}$$

$$\equiv \{e(k) : e(k) \in m_1 \wedge \forall k_2 > 0 : e(k_2) \notin m_2\} \tag{5.4}$$

We say that an instance entity e, which is a member of a categorization context, is *covered* by the context.

Composition of Contexts

Topic definitions based on ontology context projections may not offer sufficient flexibility in defining classification topics. More specifically, a classification topic should capture user's interest in a specific area or even a *combination* of areas. Hence, we extend the definition of a classification topic to include a linear combination of a number of selected categorization contexts.

A combination of categorization contexts gives the user much greater flexibility and precision in defining a category of interest. The use of a linear combination of contexts enables us to define categories involving multiple contexts, but which cannot be expressed as an intersection, union or difference of these contexts.

As an example, consider contexts defining "*business*" and "*sports*". Different

topics represented as a combination of the two contexts are presented in Figure 5.2. Topic (A), defined as a union of the two contexts, will match documents that belong to "*business*", "*sports*" or both. Topic (B), defined as an intersection of the two contexts, will match documents with entities that belong at the same time both to "*business*" and "*sports*". Using only context expressions introduced in Def. 4, we are not able to specify a topic of documents that fall into both contexts, meaning that the document belongs to "*business*" and to the "*sports*" category (for example, business activities of football teams, or a football league). Such documents must include entities both from "*business*" context (area 1) and "*sports*" context (area 2), but not necessarily entities from their intersection. Intuitively, we name such documents as belonging to the intersection of "*sports*" and "*business*", but at the instance level such topic (C) should be defined as a linear combination of both contexts. It must contain entities from each of the included contexts, whereas the union of contexts is too wide and the intersection too narrow. Even using symmetric difference of contexts still does not guarantee that entities from both contexts will be represented in a document graph. The following is an extension of Def. 4.

Definition 5. Let $C_i, 1 \leq i \leq n$, be categorization contexts. A *composition of categorization contexts* is defined as vector of pairs $(C_i, a_i), 1 \leq i \leq n$, where the coefficients a_i, indicating relative importance of the contexts, are normalized.

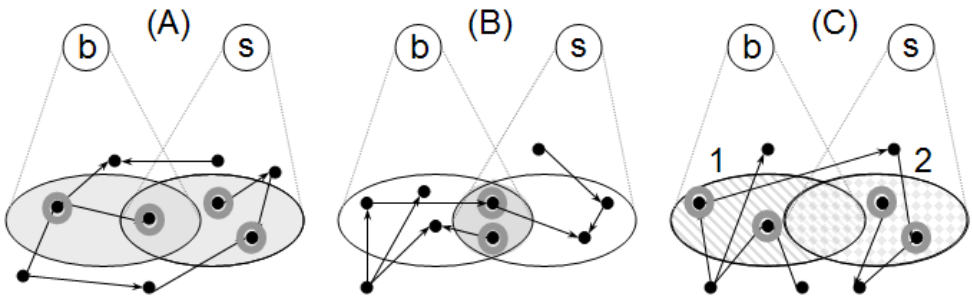


Figure 5.2: Instanced graph with selected matched entities for topics defined as union, intersection, and a combination of contexts.

5.4 Categorization Algorithm

DBpedia offers a tool for converting Wikipedia into an RDF/S format. We used it with our modifications to facilitate more precise discovery of named entities in a document. Literals (name, alias, redirection or disambiguation names) associated with an entity are the key information for the matching process. Each of the literal types has a different confidence in identification of the entity. We associated such literals with each entity using our own relations to distinguish their confidence level during the matching process.

Our categorization algorithm consists of three main steps: (1) construction of the semantic graph, (2) selection and analysis of the thematic graph, and (3) categorization of the selected thematic graph. The categorization topics are defined as ontology contexts, introduced in the previous section. They can be perceived as *ontology views* that specify user's contexts of interest. Classification topics are defined *dynamically* and *independently* from the document corpora.

Semantic Graph Construction

Document's semantic graph is constructed from the named entities identified in the document. An entity in the ontology has one or more associated literals that can be used for its identification. For each such literal, we assign a confidence level that reflects how uniquely it can identify the entity. Note, that one literal can be associated with multiple entities and produce ambiguous entity matches, which is discussed later.

Definition 6. Given a document d, the *entity matching function*, $E(d)$, returns a set of ontology entities e, such that for each $e \in E(d)$, there exists a phrase in d matching one of e's identifying labels. Each entity in $E(d)$ is assigned a weight $w(e)$, given by the formula:

$$w(e) = 1 - \frac{1}{1 + \sum_{i=1}^{n} p_i * s(l_i, sp_i)} \tag{5.5}$$

where n is the number of occurrences in d of a phrase matching e, p_i is the confidence of the relationship (property) used for entity identification and $s(l_i, sp_i)$ is the similarity of the spotted phrase sp_i and the entity's identifying literal l_i. Function s measures the similarity between the spotted phrase sp in document d and the entity e's label (literal) l in the ontology, taking into account the removed stop words and/or stemming. For more details refer to [51]. Because we did not use stemming for spotted phrases, $s(l, sp)$ is set to 1.

Definition 7. A *semantic graph* of a document d, denoted $SG(d)$, is a labeled graph with a set of vertices $E(d)$ and a set of labeled edges $\{(e_i, e_j)$ with label r,

such that e_i, $e_j \in E(d)$ and e_i, and e_j are connected by a relationship (property) r in the ontology$\}$.

Even though the ontology relationships induced in $SG(d)$ are directed, from now on, we will consider $SG(d)$ as an undirected graph. Since the semantic graph of a document is created by forming associations among the identified entities based on the properties existing in the ontology, it can be seen as adding the background knowledge to the document in order to explain the associations between the entities.

Thematic Graph Selection

The selection of the thematic graph is based on the assumption that entities related to a single topic are closely associated in the ontology, while entities from different topics are placed far apart, or even not connected at all. As a result, the analyzed semantic graph may be composed of multiple connected components, as each set of connected entities represents a different topic recognized in the document.

Definition 8. A sub-graph of $SG(d)$ is called an *interpretation of a document* d, denoted $I(d)$, if the sub-graph does not contain any ambiguous entities.

Definition 9. A connected component of $I(d)$ is called a *thematic sub-graph*. In particular, if the whole $I(d)$ is a connected graph, it is also a thematic sub-graph.

In general, an interpretation of a document may have many thematic sub-graphs, one for each of its connected components. The importance of entities in a thematic graph of a document is determined not only by their initial weights

but also by their placement in the graph. We utilize the HITS algorithm with the assigned initial weights for both entities and relationships to find the authoritative entities in the semantic graph.

Definition 10. A thematic sub-graph with the largest number of nodes and the highest total of entity weights is selected as the *dominant thematic graph* for the document.

Selecting a dominant thematic graph sets a specific interpretation context and effectively disambiguates any incorrectly matched entities. Furthermore, we locate central entities in the graph (based on the geographical centrality measure), since they can be identified as the thematic landmarks of the graph. We also use TextRank algorithm [77] to find the best entities describing the graph.

Definition 11. The *core of the dominant thematic graph* is composed of k most authoritative, descriptive and most central entities.

From now on, we will simply write thematic graph when referring to the dominant thematic graph of a document.

Classification into Defined Ontological Contexts

Classification of a document into the defined ontological contexts (topics) is based on calculating a similarity of the document's thematic graph to each of the defined contexts. In general, the similarity is calculated based on the following objectives:

- The intersection of the context projection with the thematic graph should be maximized (coverage).

- The hierarchical distance of the entities in the thematic graph to the classes included in the context should be minimized (closeness).
- The highest number of the core entities should be covered.

To establish the similarity of the document's d thematic graph to each of the defined contexts $c_1, c_2, \ldots, c_n$ (topics), we perform the following steps:

1. Find the expanded core entities in d, which include the core entities in the thematic graph of d and all of their immediate neighbors.
2. Construct a taxonomy graph out of the Wikipedia categories network for the expanded core entities. It should be noted that we empirically restrict the hierarchy's height to 3, due to the fact that increasing the height further quickly leads to excessively general categories.
3. Compute the semantic associativity score of the categories located in step 2 to the expanded core entities.
4. Find the top-k categories based on their score as the best categories describing the document.
5. For every defined context (user defined topic)[2], execute Algorithm 1 and return the semantic relatedness score of document d to the defined context.

Steps 3 and 5 are described in the following sections, respectively.

[2] "Ontology context", "user defined category" and "topic" are interchangeable

Computing Category Semantic Associativity Score

To compute the *semantic associativity* of a document to a categorization context, we first calculate the *membership score* and the *coverage score*. We have adopted a modified Vector-based Vector Generation method (VVG) described in [99] to calculate the category *membership score*. Given Wikipedia as a directed graph $G = \{W, V, E\}$ and a Wikipedia concept w_i and category v_j, the *membership score* $mScore(w_i, v_j)$ of concept w_i to category v_j is defined as follows:

$$mScore(w_i, v_j) = \prod_{e_k \in E_l} m(e_k) \tag{5.6}$$

$$m(e_k) = \frac{1}{n} \tag{5.7}$$

where $m(e_k)$ is the weight of membership links (category links), e_k, from node v_i (or w_i) to category $v \in V$, n is the number of membership links, and $E_l = \{e_1, e_2, \ldots, e_m\}$ represents a set of all membership links forming the shortest path p from the concept w_i to category v_j.

The coverage score $cScore(c, e)$ of an entity e by a Wikipedia category c is computed by the following formula:

$$cScore(c, e) = \begin{cases} 1 & \text{if there is a path between } c \text{ and } e \\ 0 & \text{otherwise.} \end{cases} \tag{5.8}$$

The *semantic associativity* score between a category and a set of entities is

Algorithm 1: computeSemRelatedness(d,C)

Input : d is a document, $c_1, \ldots, c_n$ is a set of categorization contexts (topics), and t is a threshold

foreach $c_i, 1 \leq i \leq n$ **do**
  Find the intersection S of the top-k categories of d and c_i
  foreach category C_p in S **do**
    Find all the entities belonging to C_p, denoted as E_{c_p}
    Find the *maximum* of $sr(e_j, E_{c_p}), e_j \in E_{ee}$
  end
  Sort the maximums in descending order and compute, the average of the top-k of them, denoted as $\zeta_i(c_i)$
  if $\zeta_i(c_i) < t$ **then**
    $\zeta_i(c_i) \leftarrow 0$
  end
end
return $\zeta_1(c_1) + \zeta_2(c_2) + \ldots + \zeta_n(c_n)$

defined as follows:

$$cSAssScore(c, E_{ee}) = \beta * \sum_{e \in Eee} mScore(c, e) + (1 - \beta) * \sum_{e \in Eee} cScore(c, e) \quad (5.9)$$

where $E_{ee} = \{\text{expanded core entities}\}$, c is the Wikipedia category and β is the smoothing factor to control the influence weight of two scores. We used $\beta = 0.8$ in our experiments. It should be noted that expanding core entities is used to reach objective one, and *mScore* and *cScore* are used to satisfy second and third objectives respectively.

Document Categorization to Ontological Context

To find the categorization score of a document to an ontological context (topic), we start by measuring the semantic relatedness among Wikipedia entities (concepts). In order to do that, we adopted the Wikipedia Link-based Measure (WLM) introduced in [118]. Given two Wikipedia entities a and b, we define the *semantic relatedness* between them as follows:

$$sr(a,b) = \frac{log(max(|A|,|B|)) - log(|A \cap B|)}{log(W) - log(min(|A|,|B|))} \tag{5.10}$$

where A and B are the sets of Wikipedia entities that link to a and b respectively, and W is the set of all entities in Wikipedia. By extension, the *semantic relatedness* between a Wikipedia entity a and a categorization context C (a categorization context is a projection of C onto the background ontology, including entities $\{e_1, e_2, \ldots, e_t\}$) is defined as follows:

$$sr(a,C) = \frac{1}{t}\sum_{i=1}^{t} sr(a,e_i) \tag{5.11}$$

If $\zeta_i(c_i) \geq t$, we conclude that document d belongs to topic c_i. The threshold t is established empirically. Note, that in case a topic is defined as a composition of contexts, we determine the final score of a document based on the set operators used in the composition as follows:

- $c_i \cap c_j : \zeta = min\Big(\zeta_i(c_i), \zeta_j(c_j)\Big)$
- $c_i \cup c_j : \zeta = max\Big(\zeta_i(c_i), \zeta_j(c_j)\Big)$

- $!c_i : \zeta = 1 - \zeta_i(c_i)$

5.5 Experiments

In our experiments, we used an RDF ontology created from the full version of English Wikipedia XML dump from 2013-06-04. The created ontology contained 5,047,075 entities connected by 287,016,171 statements and 13,062,411 literals describing the entities. They were classified using 930,472 categories defined in Wikipedia. We used Virtuoso[3] for ontology storage (triple store) and querying. We evaluated our system on a text corpus obtained from the Reuters[4] RSS feed (2013-10-24 2014-01-30). We divided some of the main topics into fine-grained sub-topics in order to evaluate our classification method. The details of the text corpus, as well as the fine-grained categorization of the main topics are presented in Table 5.1 and Table 5.2, respectively.

Due to the nature of the analyzed news documents, we decided to exclude time related entities since they provided highly misleading connections among other entities from the categorization process.

Experiment Results

We conducted three experiments on our Reuters corpus. In the first experiment, we wanted to assess the basic topic categorization of our system. Here, we created

[3]http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/
[4]http://www.reuters.com/

Table 5.1: Category details of used text corpus

Reuters Category	**Number of Documents**
Sports	254
Technology	927
Business	786
Arts	94
Science	140
Health	864
Politics	807
Total	**3,872**

Table 5.2: Fine-grained categorization of main categories

Main Topics		
Sports	**Technology**	**Business**
Sub-topics	**Sub-topics**	**Sub-topics**
Baseball	Digital_Technology	Economics
Basketball	Space_Technology	Industry
National_Hockey-League	Mobile_Technology	Financial_Markets
Tennis	Telecommunications	
Golf		
National_Football_League		
Football (Soccer)		

Table 5.3: Highlevel ontological contexts categorization

TOPICS	**MAP**
Arts	96.80%
Science	92.10%
Health	90.60%
Politics	95.70%
Total	**93.80%**

categorization contexts consisting of high-level Wikipedia categories[5] to represent the topics best corresponding to those in the Reuters corpus. The defined contexts included Wikipedia categories with names directly corresponding to the Reuters' category names. Table 5.3 shows the micro averaged precision (MAP) of the first performed experiment.

In the second experiment, we evaluated the effectiveness of categorizing into topics composed of *unions of contexts*. Therefore we created topics for Sports, Technology and Business as the unions of their sub-topics, shown in Table 5.2, (e.g. Business = Economics $\bigcup$ Industry $\bigcup$ Financial_markets). Thus, we identified high-level topics of documents and specific sub-topics within them. The results are presented in Table 5.4.

In the third experiment, we assessed our system's ability to categorize documents into topics expressed as more complex context compositions. Consequently, we created *compositions of contexts* from the "technology", "business" and "politics" topics. The topics were defined as follows:

[5]We have experimented with YAGO, but due to its coarse-grained categorization we used a Wikipedia-based ontology.

Table 5.4: Categorization based on *unions* of sub-categories

TOPICS	**MAP**
Sports	97.80%
Technology	85.60%
Business	79.40%
Total	**87.60%**

- $(Digital \cap Telecom) \cap (!Mobile)$
- $Economics \cap (!Financial\_Markets)$
- $!(Economics \cup Industry \cup Financial\_Markets)$
- $Politics \cap (!Immigration)$

We chose random samples of 94, 68 and 236 documents from "technology", "business" and "politics" topics respectively, to measure the MAP. Table 5.5 represents the details of the third experiment.

Table 5.5: Categorization based on *unions* of sub-categories

TOPICS	**MAP**
$(Digital \cap Telecom) \cap (!Mobile)$	86.70%
$Economics \cap (!Financial\_Markets)$	80.00%
$!(Economics \cup Industry \cup Financial\_Markets)$	100%
$Politics \cap (!Immigration)$	90.40%
Total	**89.30%**

Results Analysis

Our ontology-based categorization method achieved very good results. These results are especially promising in view of the fact that our method did not rely on classifier training and that it can be readily applied to any other set of topics defined as classification contexts or their compositions.

The analysis of the incorrectly classified documents and the created semantic graphs revealed the following clues for possible causes of misclassifications:

(a) The prepared categorization contexts used the Wikipedia category hierarchy, which not always reflects the topics covered in Reuters news.

(b) In some cases, highly connected domains in Wikipedia favored a context different than the major one described in the document. This was caused by the imbalance between densely and sparsely populated domains in Wikipedia.

(c) In some cases, although the categories are entirely different, one is a subcategory of the other. For example, "American football" and "football (Soccer)" are two different sports, but the former is a subcategory of the latter one in Wikipedia, which results in categorizing the "American football" documents into "football (Soccer)" category, as well.

Despite the imprecise coverage of the news topics by the Wikipedia-based created categorization contexts and the imbalance in the coverage of some domains, our ontology-based training-less categorization method was able to achieve comparable results to the traditional, training-based categorization methods. We intend

to conduct a thorough evaluation of our categorization method and compare it with traditional methods, e.g. Naïve Bayes, SVM, etc.

An important aspect of the proposed method is that with the change of users topics of interest, the classification contexts can be easily redefined and the documents can be re-classified into newly defined contexts without the need for a new set of training documents and classifier re-training.

5.6 Conclusion and Future Work

We presented a novel approach to text categorization, relying only on the ontological knowledge. Categories of interest can be defined as context projections or their combinations. Our experiments proved the applicability of ontologies for automatic text categorization and demonstrated a significant value of knowledge represented in Wikipedia when applied to this problem. We intend to conduct additional testing of our method, especially involving the combination of classification contexts.

Chapter 6

OntoLDA: An Ontology-based Topic Model for Automatic Topic Labeling[1]

[1]Mehdi Allahyari and Krys Kochut. "*OntoLDA: An Ontology-based Topic Model for Automatic Topic Labeling*".
Submitted to the *Semantic Web Journal*.
A shorter, preliminary version of the paper, "*Automatic Topic Labeling Using Ontology-Based Topic Models*", has been published in 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), Pages 259 - 264

Abstract

Topic models, which frequently represent topics as multinomial distributions over words, have been extensively used for discovering latent topics in text corpora. Topic labeling, which aims to assign meaningful labels for discovered topics, has recently gained significant attention. In this paper, we argue that the quality of topic labeling can be improved by considering ontology concepts rather than words alone, in contrast to previous works in this area, which usually represent topics via groups of words selected from topics. We have created (1) a topic model that integrates ontological concepts with topic models in a single framework, where each topic is represented as a multinomial distribution over concepts and each concept is a multinomial distribution over words, and (2) a topic labeling method based on the ontological meaning of the concepts included in the discovered topics. In selecting the best topic labels, we rely on the semantic relatedness of the concepts and their ontological classifications. The results of our experiments conducted on two different data sets show that introducing concepts as additional, richer features between topics and words and describing topics in terms of concepts offers an effective method for generating meaningful labels for the discovered topics.

6.1 Introduction

Topic models such as Latent Dirichlet Allocation (LDA) [17] have gained considerable attention, recently. They have been successfully applied to a wide variety of text mining tasks, such as word sense disambiguation [47, 20], sentiment analysis [62], information retrieval [117] and others, in order to identify hidden topics in text documents. Topic models typically assume that documents are mixtures of topics, while topics are probability distributions over the vocabulary. When the topic proportions of documents are estimated, they can be used as the themes (high-level representations of the semantics) of the documents. Highest-ranked words in a topic-word distribution indicate the meaning of the topic. Thus, topic models provide an effective framework for extracting the latent semantics from unstructured text collections. For example, Table 6.1 shows the top words of a topic learned from a collection of computer science abstracts; the topic has been labeled by a human "relational databases".

However, even though the topic word distributions are usually meaningful, it is very challenging for the users to accurately interpret the meaning of the topics based only on the word distributions extracted from the corpus, particularly when they are not familiar with the domain of the corpus. It would be very difficult to answer questions such as "What does a topic inform about?" and "What is a good enough label for a topic?"

Topic labeling means finding one or a few phrases that sufficiently explain the meaning of the topic. This task, which can be labor intensive particularly when dealing with hundreds of topics, has recently attracted considerable attention.

Table 6.1: Example of a topic with its label.

Human Label: relational databases				
query	database	databases	queries	processing
efficient	relational	object	xml	systems

The aim of this research is to *automatically* generate *good* labels for the topics. But, what makes a label good for a topic? We assume that a good label: (1) should be semantically relevant to the topic; (2) should be understandable to the user; and (3) highly cover the meaning of the topic. For instance, "relational databases", "databases" and "database systems" are a few good labels for the example topic illustrated in Table 6.1.

Within the Semantic Web, numerous data sources have been published as ontologies. Many of them are inter-connected as Linked Open Data (LOD)[2]. Linked Open Data provides rich knowledge in multiple domains, which is a valuable asset when used in combination with various analyses based on unsupervised topic models, in particular, for topic labeling. For example, DBpedia [11] (as part of LOD) is a publicly available knowledge base extracted from Wikipedia in the form of an ontology of concepts and relationships, making this vast amount of information programmatically accessible on the Web.

The principal objective of the research presented here is to leverage and incorporate the semantic graph of concepts in an ontology, DBpedia in this work, and their various properties within unsupervised topic models, such as LDA. In

[2] `http://linkeddata.org/`

our model, we introduce another latent variable called, *concept*, i.e.,ontological concept, between topics and words. Thus, each document is a multinomial distribution over topics, where each topic is represented as a multinomial distribution over concepts, and each concept is defined as a multinomial distribution over words.

Defining the concept latent variable as another layer between topics and words has multiple advantages: (1) it gives us much more information about the topics; (2) it allows us to illustrate topics more specifically, based on ontology concepts rather than words, which can be used to label topics; (3) it automatically integrates topics with knowledge bases. We first presented our ontology-based topic model, OntoLDA model, in [1] where we showed that incorporating ontological concepts with topic models improves the quality of topic labeling. In this paper, we elaborate on and extend these results. We also extensively explore the theoretical foundation of our ontology-based framework, demonstrating the effectiveness of our proposed model over two datasets.

Our contributions in this work are as follows:

1. We propose an ontology-based topic model, OntoLDA, which incorporates an ontology into the topic model in a systematic manner. Our model integrates the topics to external knowledge bases, which can benefit other research areas such as information retrieval, classification and visualization.

2. We introduce a topic labeling method, based on the semantics of the concepts that are included in the discovered topics, as well as ontological relationships existing among the concepts in the ontology. Our model improves the labeling accuracy by exploiting the topic-concept relations and can automatically

generate labels that are meaningful for interpreting the topics.

3. We demonstrate the usefulness of our approach in two ways. We first show how our model can be exploited to link text documents to ontology concepts and categories. Then we illustrate automatic topic labeling by performing a series of experiments.

The paper is organized as follows. In section 2, we formally define our model for labeling the topics by integrating the ontological concepts with probabilistic topic models. We present our method for concept-based topic labeling in section 3. In section 4, we demonstrate the effectiveness of our method on two different datasets. Finally, we present our conclusions and future work in section 5.

6.2 Background

In this section, we formally describe some of the related concepts and notations that will be used throughout this paper.

6.2.1 Ontologies

Ontologies are fundamental elements of the Semantic Web and could be thought of knowledge representation methods, which are used to specify the knowledge shared among different systems. An ontology is referred to an "explicit specification of a conceptualization." [39]. In other words, an ontology is a structure consisting of a set of concepts and a set of relationships existing among them.

Ontologies have been widely used as the background knowledge (i.e., knowledge bases) in a variety of text mining and knowledge discovery tasks such as text clustering [35, 46, 45], text classification [2, 70, 21], word sense disambiguation [19, 63, 64], and others. See [90] for a comprehensive review of Semantic Web in data mining and knowledge discovery.

6.2.2 Probabilistic Topic Models

Probabilistic topic models are a set of algorithms that are used to uncover the hidden thematic structure from a collection of documents. The main idea of topic modeling is to create a probabilistic generative model for the corpus of text documents. In topic models, documents are mixture of topics, where a topic is a probability distribution over words. The two main topic models are Probabilistic Latent Semantic Analysis (pLSA) [44] and Latent Dirichlet Allocation (LDA) [17]. Hofmann (1999) introduced pLSA for document modeling. pLSA model does not provide any probabilistic model at the document level. Each model is represented by a list of probability values $p(z|d)$, but these numbers are not generated from a probabilistic model, which makes generalizing pLSA difficult to model new unseen documents. Blei et al. [17] extended this model by introducing a Dirichlet prior on mixture weights of topics per documents, and called the model Latent Dirichlet Allocation (LDA). In this section we describe the LDA method.

The Latent Dirichlet Allocation (LDA) [17] is a generative probabilistic model for extracting thematic information (topics) of a collection of documents. LDA assumes that each document is made up of various topics, where each topic is a

probability distribution over words.

Let $\mathcal{D} = \{d_1, d_2, \ldots, d_D\}$ is the corpus and $\mathcal{V} = \{w_1, w_2, \ldots, w_V\}$ is the vocabulary of the corpus. A topic $z_j, 1 \leq j \leq K$ is represented as a multinomial probability distribution over the V words, $p(w_i|z_j), \sum_i^V p(w_i|z_j) = 1$. LDA generates the words in a two-stage process: words are generated from topics and topics are generated by documents. More formally, the distribution of words given the document is calculated as follows:

$$p(w_i|d) = \sum_{j=1}^{K} p(w_i|z_j)p(z_j|d) \tag{6.1}$$

The graphical model of LDA is shown in Figure 7.1 and the generative process for the corpus $\mathcal{D}$ is as follows:

1. For each topic $k \in \{1, 2, \ldots, K\}$, sample a word distribution $\phi_k \sim \text{Dir}(\beta)$
2. For each document $d \in \{1, 2, \ldots, D\}$,
   (a) Sample a topic distribution $\theta_d \sim \text{Dir}(\alpha)$
   (b) For each word w_n, where $n \in \{1, 2, \ldots, N\}$, in document d,
      i. Sample a topic $z_i \sim \text{Mult}(\theta_d)$
      ii. Sample a word $w_n \sim \text{Mult}(\phi_{z_i})$

The joint distribution of the model (hidden and observed variables) is:

$$P(\phi_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D}) = \prod_{j=1}^{K} P(\phi_j|\beta) \prod_{d=1}^{D} P(\theta_d|\alpha) \left(\prod_{n=1}^{N} P(z_{d,n}|\theta_d) P(w_{d,n}|\phi_{1:K}, z_{d,n}) \right) \tag{6.2}$$

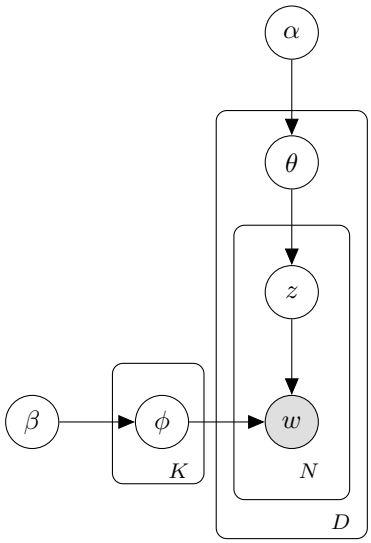


Figure 6.1: LDA Graphical Model

In the LDA model, the word-topic distribution $p(w|z)$ and topic-document distribution $p(z|d)$ are learned entirely in an unsupervised manner, without any prior knowledge about what words are related to the topics and what topics are related to individual documents. One of the most widely-used approximate inference techniques is Gibbs sampling [38]. Gibbs sampling begins with random assignment of words to topics, then the algorithm iterates over all the words in the training documents for a number of iterations (usually on order of 100). In each iteration, it samples a new topic assignment for each word using the conditional distribution of that word given all other current word-topic assignments. After the iterations are finished, the algorithm reaches a steady state, and the word-topic probability

distributions can be estimated using word-topic assignments.

6.3 Motivating Example

Let us presume that we are given a collection of news articles and told to extract the common themes present in this corpus. Manual inspection of the articles is the simplest approach, but it is not practical for large collection of documents. We can make use of topic models to solve this problem by assuming that a collection of text documents comprises of a set of hidden themes, called *topics*. Each topic z is a multinomial distribution $p(w|z)$ over the words w of the vocabulary. Similarly, each document is made up of these topics, which allows multiple topics to be present in the same document. We estimate both the topics and document-topic mixtures from the data simultaneously. When the topic proportions of documents are estimated, they can be used as the themes (high-level semantics) of the documents. Top-ranked words in a topic-word distribution indicate the meaning of the topic.

For example, Table 6.2 shows a sample of four topics with their top-10 words learned from a corpus of news articles. Although the topic-word distributions are usually meaningful, it is very difficult for the users to accurately infer the meanings of the topics just from the top words, particularly when they are not familiar with the domain of the corpus. The Standard LDA model does not *automatically* provide the labels of the topics. Essentially, for each topic it gives a distribution over the entire words of the vocabulary. A *label* is one or a few phrases that

Table 6.2: Example topics with top-10 words learned from a document set.

Topic 1	**Topic 2**	**Topic 3**	**Topic 4**
company	film	drug	republican
mobile	show	drugs	house
technology	music	cancer	senate
facebook	year	fda	president
google	television	patients	state
apple	singer	reuters	republicans
online	years	disease	political
industry	movie	treatment	campaign
video	band	virus	party
business	actor	health	democratic

sufficiently explain the meaning of the topic. For instance, as shown in Table 6.2, topics do not have any labels, therefore they must be manually assigned. Topic labeling task can be labor intensive particularly when dealing with hundreds of topics. Table 6.3 illustrates the same topics that have been labeled (second row in the table) manually by a human.

Automatic topic labeling which aims to to automatically generate meaningful labels for the topics has recently attracted increasing attention [115, 75, 71, 60, 48]. Unlike previous works that have essentially concentrated on the topics learned from LDA topic model and represented the topics by words, we propose an ontology-based topic model, OntoLDA, where topics are labeled by ontological concepts.

We believe that the knowledge in the ontology can be integrated with the topic models to automatically generate topic labels that are semantically relevant, understandable for humans and highly cover the discovered topics. In other words,

Table 6.3: Example topics with top-10 words learned from a document set. The second row presents the manually assigned labels.

Topic 1	**Topic 2**	**Topic 3**	**Topic 4**
"Technology"	"Entertainment"	"Health"	"U.S. Politics"
company	film	drug	republican
mobile	show	drugs	house
technology	music	cancer	senate
facebook	year	fda	president
google	television	patients	state
apple	singer	reuters	republicans
online	years	disease	political
industry	movie	treatment	campaign
video	band	virus	party
business	actor	health	democratic

our aim is to incorporate the semantic graph of concepts in an ontology (e.g., DBpedia) and their various properties with unsupervised topic models, such as LDA, in a principled manner and exploit this information to automatically generate meaningful topic labels.

6.4 Related Work

Probabilistic topic modeling has been widely applied to various text mining tasks in virtue of its broad application in applications such as text classification [42, 65, 94], word sense disambiguation [47, 20], sentiment analysis [62, 67], and others. A main challenge in such topic models is to interpret the semantic of each topic in

an accurate way.

Early research on topic labeling usually considers the top-n words that are ranked based on their marginal probability $p(w_i|z_j)$ in that topic as the primitive labels [17, 38]. This option is not satisfactory, because it necessitates significant perception to interpret the topic, particularly if the user is not familiar with the domain of the topic. For example, it would be very hard to infer the meaning of the topic shown in Table 6.1 only based on the top terms, if someone is not knowledgeable about the "database" domain. The other conventional approach for topic labeling is to manually generate topic labels [74, 114]. This approach has disadvantages: (a) the labels are prone to subjectivity; and (b) the method can not be scale up, especially when dealing with massive number of topics.

Recently, automatic topic labeling has been an area of active research. [115] represented topics as multinomial distribution over n-grams, so top n-grams of a topic can be used to label the topic. Mei et al. [75] proposed an approach to automatically label the topics by converting the labeling problem to an optimization problem. First they generate candidate labels by extracting either bigrams or noun chunks from the collection of documents. Then, they rank the candidate labels based on Kullback-Leibler (KL) divergence with a given topic, and choose a candidate label that has the minimum KL divergence and the maximum mutual information with the topic to label the corresponding topic. [71] introduced an algorithm for topic labeling based on a given topic hierarchy. Given a topic, they generate a label candidate set using Google Directory hierarchy and find the best label according to a set of similarity measures.

Lau et al. [59] introduced a method for topic labeling by selecting the best topic word as its label based on a number of features. They assume that the topic terms are representative enough and appropriate to be considered as labels, which is not always the case. Lau et al. [60] reused the features proposed in [59] and also extended the set of candidate labels exploiting Wikipedia. For each topic they first select the top terms and query the Wikipedia to find top article titles having these terms according to the features and consider them as extra candidate labels. Then they rank the candidate to find the best label for the topic.

Mao et al. [73] proposed a topic labeling approach which enhances the labeling by using the sibling and parent-child relations between topics. They first generate a set of candidate labels by extracting meaningful phrases using Ngram Testing [27] for a topic and adding the top topic terms to the set based on marginal term probabilities. And then rank the candidate labels by exploiting the hierarchical structure between topics and pick the best candidate as the label of the topic.

In a more recent work Hulpus et al. [48] proposed an automatic topic labeling approach by exploiting structured data from DBpedia[3]. Given a topic, they first find the terms with highest marginal probabilities, and then determine a set of DBpedia concepts where each concept represents the identified sense of one of the top terms of the topic. After that, they create a graph out of the concepts and use graph centrality algorithms to identify the most representative concepts for the topic.

Our work is different from all previous works in that we propose a topic model

[3] `http://dbpedia.org`

that integrates structured data with data-driven topics within a single general framework. Prior works basically focus on the topics learned via LDA topic model (i.e.,topics are multinomial distribution over words) whereas in our model we introduce another latent variable called *concept* between topics and words, i.e., each document is a multinomial distribution over topics where each topic is represented as a multinomial distribution over concepts and each concept is defined as a multinomial distribution over words.

The hierarchical topic models, which represent correlations among topics, are conceptually related to our OntoLDA model. Mimno et al. [81] proposed the hPAM model that models a document as a mixture of distributions over super-topics and sub-topics, using a directed acyclic graph to represent a topic hierarchy. The OntoLDA model is different, because in hPAM, distribution of each super-topic over sub-topics depends on the document, whereas in OntoLDA, distributions of topics over concepts are independent of the corpus and are based on an ontology. The other difference is that sub-topics in the hPAM model are still unigram words, whereas in OntoLDA, ontological concepts are n-grams, which makes them more specific and more meaningful, a key point in OntoLDA. [25, 26] introduced topic models that combine concepts with data-driven topics. The key idea in their frameworks is that topics from the statistical topic models and concepts of the ontology are both represented by a set of "focused" words, i.e.,distributions over words, and they use this similarity in their models. However, our OntoLDA model is different from these models in that they treat the concepts and topics in the same way, whereas in OntoLDA, concepts and topics form two distinct layers in

the model.

6.5 Problem Formulation

In this section, we formally describe our model and its learning process. We then explain how to leverage the topic-concept distribution to generate meaningful semantic labels for each topic, in section 4. The notation used in this paper is summarized in Table 6.5.

Most topic models like LDA consider each document as a mixture of topics where each topic is defined as a multinomial distribution over the vocabulary. Unlike LDA, OntoLDA defines another latent variable called *concept* between topics and words, i.e., each document is a multinomial distribution over topics where each topic is a represented as a multinomial distribution over concepts and each concept is defined as a multinomial distribution over words.

The intuition behind our model is that using words from the vocabulary of the document corpus to represent topics is not a good way to convey the meaning of the topics. Words usually describe topics in a broad way while ontological concepts express the topics in a more focused way. Additionally, concepts representing a topic are semantically more closely related to each other. As an example, the first column of Table 6.4 lists a topic learned by standard LDA and represented by top words, whereas the second column shows the same topic learned by the OntoLDA model, which represents the topic using ontology concepts. From the topic-word representation we can conclude that the topic is about "sports", but the topic-

concept representation indicates that not only the topic is about "sports", but more specifically about "American sports".

Table 6.4: Example of topic-word representation learned by LDA and topic-concept representation learned by OntoLDA.

LDA		**OntoLDA**	
Human Label: Sports		**Human Label:** American Sports	
Topic-word	**Probability**	**Topic-concept**	**Probability**
team	(0.123)	oakland raiders	(0.174)
est	(0.101)	san francisco giants	(0.118)
home	(0.022)	red	(0.087)
league	(0.015)	new jersey devils	(0.074)
games	(0.010)	boston red sox	(0.068)
second	(0.010)	kansas city chiefs	(0.054)

Let $\mathcal{C} = \{c_1, c_2, \ldots, c_C\}$ be the set of DBpedia concepts, and $\mathcal{D} = \{d_i\}_{i=1}^{D}$ be a collection of documents. We represent a document d in the collection $\mathcal{D}$ with a bag of words, i.e., $d = \{w_1, w_2, \ldots, w_V\}$, where V is the size of the vocabulary.

Definition 12. (Concept): A *concept* in a text collection $\mathcal{D}$ is represented by c and defined as a multinomial distribution over the vocabulary $\mathcal{V}$, i.e., $\{p(w|c)\}_{w\in\mathcal{V}}$. Clearly, we have $\sum_{w\in\mathcal{V}} p(w|c) = 1$. We assume that there are $|C'|$ concepts in $\mathcal{D}$ where $C' \subset C$.

Definition 13. (Topic): A *topic* ϕ in a given text collection $\mathcal{D}$ is defined as a multinomial distribution over the *concepts* $\mathcal{C}$, i.e., $\{p(c|\phi)\}_{c\in\mathcal{C}}$. Clearly, we have $\sum_{c\in\mathcal{C}} p(c|\phi) = 1$. We assume that there are K topics in $\mathcal{D}$.

Definition 14. (Topic representation): The *topic representation* of a document d, θ_d, is defined as a probabilistic distribution over K topics, i.e., $\{p(\phi_k|\theta_d)\}_{k\in K}$.

Table 6.5: Notation used in this paper

Symbol	Description
D	number of documents
K	number of topics
C	number of concepts
V	number of words
N_d	number of words in document d
α_t	asymmetric Dirichlet prior for topic t
β	symmetric Dirichlet prior for topic-concept distribution
γ	symmetric Dirichlet prior for concept-word distribution
z_i	topic assigned to the word at position i in the document d
c_i	concept assigned to the word at position i in the document d
w_i	word at position i in the document d
θ_d	multinomial distribution of topics for document d
ϕ_k	multinomial distribution of concepts for topic k
ζ_c	multinomial distribution of words for concept c

Definition 15. (Topic Modeling): Given a collection of text documents, $\mathcal{D}$, the task of *Topic Modeling* aims at discovering and extracting K topics, i.e., $\{\phi_1, \phi_2, \ldots, \phi_K\}$, where the number of topics, K, is specified by the user.

6.5.1 The OntoLDA Topic Model

The key idea of the OntoLDA topic model is to integrate ontology concepts directly with topic models. Thus, topics are represented as distributions over concepts, and concepts are defined as distributions over the vocabulary. Later in this paper, concepts will also be used to identify appropriate labels for topics.

The OntoLDA topic model is illustrated in Figure 6.2 and the generative process is defined as Algorithm 2.

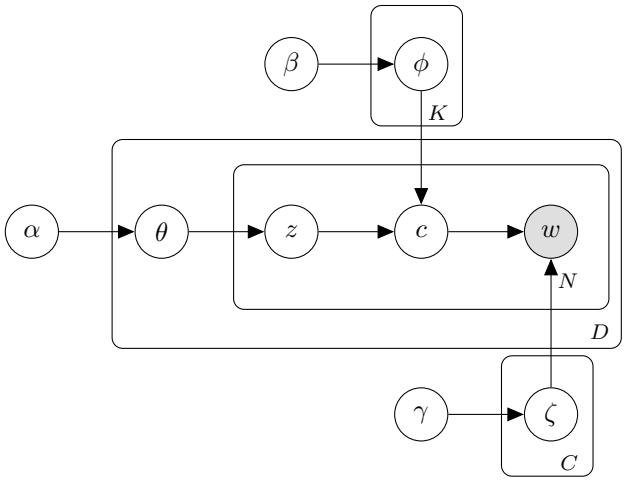


Figure 6.2: Graphical representation of OntoLDA model

Algorithm 2: OntoLDA Topic Model

foreach concept $c \in \{1, 2, \ldots, C\}$ **do**
  Draw a word distribution $\zeta_c \sim \text{Dir}(\gamma)$
end
foreach topic $k \in \{1, 2, \ldots, K\}$ **do**
  Draw a concept distribution $\phi_k \sim \text{Dir}(\beta)$
end
foreach document $d \in \{1, 2, \ldots, D\}$ **do**
  Draw a topic distribution $\theta_d \sim \text{Dir}(\alpha)$
  foreach word w of document d **do**
    Draw a topic $z \sim \text{Mult}(\theta_d)$
    Draw a concept $c \sim \text{Mult}(\phi_z)$
    Draw a word w from concept $c, w \sim \text{Mult}(\zeta_c)$
  end
end

Following this process, the joint probability of generating a corpus $D = \{d_1, d_2, \ldots, d_{|D|}\}$, the topic assignments $\mathbf{z}$ and the concept assignments $\mathbf{c}$ given the hyperparameters α, β and γ is:

$$P(\mathbf{w}, \mathbf{c}, \mathbf{z}|\alpha, \beta, \gamma) = \int_{\zeta} P(\zeta|\gamma) \prod_{d} \sum_{c_d} P(w_d|c_d, \zeta) \times \int_{\phi} P(\phi|\beta) \int_{\theta} P(\theta|\alpha) P(c_d|\theta, \phi) d\theta d\phi d\zeta \tag{6.3}$$

6.5.2 Inference using Gibbs Sampling

Since the posterior inference of the OntoLDA is intractable, we need to find an algorithm for estimating posterior inference. A variety of algorithms have been used to estimate the parameters of topic models, such as variational EM [17] and Gibbs sampling [38]. In this paper we will use collapsed Gibbs sampling procedure for OntoLDA topic model. Collapsed Gibbs sampling [38] is a Markov Chain Monte Carlo (MCMC) [91] algorithm which constructs a Markov chain over the latent variables in the model and converges to the posterior distribution after a number of iterations. In our case, we aim to construct a Markov chain that converges to the posterior distribution over $\mathbf{z}$ and $\mathbf{c}$ conditioned on observed words $\mathbf{w}$ and hyperparameters α, β and γ. We use a blocked Gibbs sampling to jointly sample $\mathbf{z}$ and $\mathbf{c}$, although we can alternatively perform hierarchical sampling, i.e., first sample $\mathbf{z}$ and then sample $\mathbf{c}$. Nonetheless, Rosen-Zvi [93] argue that in cases where latent variables are greatly related, blocked sampling boosts convergence of

the Markov chain and decreases auto-correlation, as well.

We derive the posterior inference from Eq. 6.3 as follows:

$$\begin{aligned} P(\mathbf{z}, \mathbf{c}|\mathbf{w}, \alpha, \beta, \gamma) &= \frac{P(\mathbf{z}, \mathbf{c}, \mathbf{w}|\alpha, \beta, \gamma)}{P(\mathbf{w}|\alpha, \beta, \gamma)} \propto P(\mathbf{z}, \mathbf{c}, \mathbf{w}|\alpha, \beta, \gamma) \\ &\propto P(\mathbf{z})P(\mathbf{c}|\mathbf{z})P(\mathbf{w}|\mathbf{c}) \end{aligned} \tag{6.4}$$

where

$$P(\mathbf{z}) = \left(\frac{\Gamma(K\alpha)}{\Gamma(\alpha)^K}\right)^D \prod_{d=1}^{D} \frac{\prod_{k=1}^{K} \Gamma(n_k^{(d)} + \alpha)}{\Gamma(\sum_{k'}(n_{k'}^{(d)} + \alpha))} \tag{6.5}$$

$$P(\mathbf{c}|\mathbf{z}) = \left(\frac{\Gamma(C\beta)}{\Gamma(\beta)^C}\right)^K \prod_{k=1}^{K} \frac{\prod_{c=1}^{C} \Gamma(n_c^{(k)} + \beta)}{\Gamma(\sum_{c'}(n_{c'}^{(k)} + \beta))} \tag{6.6}$$

$$P(\mathbf{w}|\mathbf{c}) = \left(\frac{\Gamma(V\zeta)}{\Gamma(\zeta)^V}\right)^C \prod_{c=1}^{C} \frac{\prod_{w=1}^{V} \Gamma(n_w^{(c)} + \zeta)}{\Gamma(\sum_{w'}(n_{w'}^{(c)} + \zeta))} \tag{6.7}$$

where $P(\mathbf{z})$ is the probability of the joint topic assignments $\mathbf{z}$ to all the words $\mathbf{w}$ in corpus $\mathcal{D}$. $P(\mathbf{c}|\mathbf{z})$ is the conditional probability of joint concept assignments $\mathbf{c}$ to all the words $\mathbf{w}$ in corpus $\mathcal{D}$, given all topic assignments $\mathbf{z}$, and $P(\mathbf{w}|\mathbf{c})$ is the conditional probability of all the words $\mathbf{w}$ in corpus $\mathcal{D}$, given all concept assignments $\mathbf{c}$.

For a word token w at position i, its full conditional distribution can be written

as:

$$P(z_i = k, c_i = c | w_i = w, \mathbf{z}_{-i}, \mathbf{c}_{-i}, \mathbf{w}_{-i}, \alpha, \beta, \gamma) \propto$$
$$\frac{n^{(d)}_{k,-i} + \alpha_k}{\sum_{k'} (n^{(d)}_{k',-i} + \alpha_{k'})} \times \frac{n^{(k)}_{c,-i} + \beta}{\sum_{c'} (n^{(k)}_{c',-i} + \beta)} \times \frac{n^{(c)}_{w,-i} + \gamma}{\sum_{w'} (n^{(c)}_{w',-i} + \gamma)} \tag{6.8}$$

where $n^{(c)}_w$ is the number of times word w is assigned to concept c. $n^{(k)}_c$ is the number of times concept c occurs under topic k. $n^{(d)}_k$ denotes the number of times topic k is associated with document d. Subscript $-i$ indicates the contribution of the current word w_i being sampled is removed from the counts.

In most probabilistic topic models, the Dirichlet parameters α are assumed to be given and fixed, which still produce reasonable results. But, as described in [107], that asymmetric Dirichlet prior α has substantial advantages over a symmetric prior, we have to learn these parameters in our proposed model. We could use maximum likelihood or maximum a posteriori estimation to learn α. However, there is no closed-form solution for these methods and for the sake of simplicity and speed we use moment matching methods [83] to approximate the parameters of α. In each iteration of Gibbs sampling, we update

$$
\begin{aligned}
mean_{dk} &= \frac{1}{N} \times \sum_{d} \frac{n_k^{(d)}}{n^{(d)}} \\
var_{dk} &= \frac{1}{N} \times \sum_{d} (\frac{n_k^{(d)}}{n^{(d)}} - mean_{dk})^2 \\
m_{dk} &= \frac{mean_{dk} \times (1 - mean_{dk})}{var_{dk}} - 1 \\
\alpha_{dk} &\propto mean_{dk} \\
\sum_{k=1}^{K} \alpha_{dk} &= exp(\frac{\sum_{k=1}^{K} log(m_{dk})}{K-1})
\end{aligned} \tag{6.9}
$$

For each document d and topic k, we first compute the sample mean $mean_{dk}$ and sample variance var_{dk}. N is the number of documents and $n^{(d)}$ is the number of words in document d.

Algorithm 3 shows the Gibbs sampling process for our OntoLDA model.

After Gibbs sampling, we can use the sampled topics and concepts to estimate the probability of a topic given a document, θ_{dk}, probability of a concept given a topic, ϕ_{kc}, and the probability of a word given a concept, ζ_{cw}:

$$\theta_{dk} = \frac{n_k^{(d)} + \alpha_k}{\sum_{k'} (n_{k'}^{(d)} + \alpha_{k'})} \tag{6.10}$$

$$\phi_{kc} = \frac{n_c^{(k)} + \beta}{\sum_{c'} (n_{c'}^{(k)} + \beta)} \tag{6.11}$$

$$\zeta_{cw} = \frac{n_w^{(c)} + \gamma}{\sum_{w'} (n_{w'}^{(c)} + \gamma)} \tag{6.12}$$

6.6 Concept-based Topic Labeling

The intuition behind our approach is that entities (i.e., ontology concepts and instances) occurring in the text along with relationships among them can determine the document's topic(s). Furthermore, the entities classified into the same or similar domains in the ontology are semantically closely related to each other. Hence, we rely on the semantic similarity between the information included in the text and a suitable fragment of the ontology in order to identify good labels for the topics. Research presented in [2] use a similar approach to perform ontology-based text categorization.

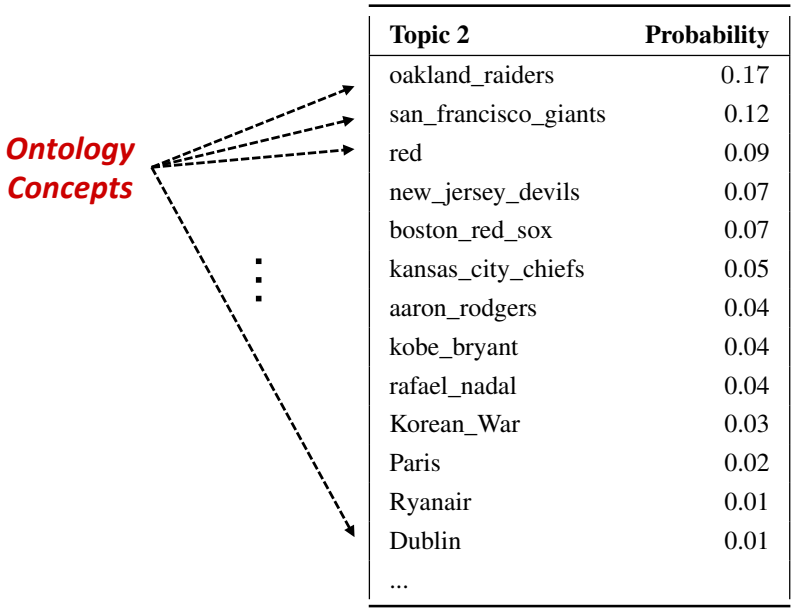
Definition 16. (Topic Label): A *topic label* ℓ for topic ϕ is a sequence of words which is semantically meaningful and sufficiently explains the meaning of ϕ.

Our approach focuses only on the ontology concepts and their class hierarchy as topic labels. Finding meaningful and semantically relevant labels for an identified

Algorithm 3: OntoLDA Gibbs Sampling

Input : A collection of documents D, number of topics K and α, β, γ
Output: $\zeta = \{p(w_i|c_j)\}$, $\phi = \{p(c_j|z_k)\}$ and $\theta = \{p(z_k|d)\}$,
i.e.,concept-word, topic-concept and document-topic distributions

```
/* Randomly, initialize concept-word assignments for all word tokens,
   topic-concept assignments for all concepts and document-topic
   assignments for all the documents                                  */
initialize the parameters φ, θ and ζ randomly;
if computing parameter estimation then
    initialize alpha parameters, α, using Eq. 6.9;
end
t ← 0;
while t < MaxIteration do
    foreach word w do
        c = c(w) // get the current concept assignment
        k = z(w) // get the current topic assignment
        // Exclude the contribution of the current word w
        n_w^(c) ← n_w^(c) − 1;
        n_c^(k) ← n_c^(k) − 1;
        n_k^(d) ← n_k^(d) − 1 // w is a document word
        (newk, newc) = sample new topic-concept and concept-word for
        word w using Eq. 6.8;
        // Increment the count matrices
        n_w^(newc) ← n_w^(newc) + 1;
        n_newc^(newk) ← n_newc^(newk) + 1;
        n_newk^(d) ← n_newk^(d) + 1;
        // Update the concept assignments and topic assignment vectors
        c(w) = newc;
        z(w) = newk;
        if computing parameter estimation then
            update alpha parameters, α, using Eq. 6.9;
        end
    end
    t ← t + 1;
end
```



Topic 2	Probability
oakland_raiders	0.17
san_francisco_giants	0.12
red	0.09
new_jersey_devils	0.07
boston_red_sox	0.07
kansas_city_chiefs	0.05
aaron_rodgers	0.04
kobe_bryant	0.04
rafael_nadal	0.04
Korean_War	0.03
Paris	0.02
Ryanair	0.01
Dublin	0.01
...	

Figure 6.3: Example of a topic represented by top concepts learned by OntoLDA.

topic ϕ involves four primary steps: (1) construction of the semantic graph from top concepts in the given topic; (2) selection and analysis of the thematic graph, a semantic graph's subgraph; (3) topic graph extraction from the thematic graph concepts; and (4) computation of the semantic similarity between topic ϕ and the candidate labels of the topic label graph.

6.6.1 Semantic Graph Construction

We use the marginal probabilities $p(c_i|\phi_j)$ associated with each concept c_i in a given topic ϕ_j and extract the $\mathcal{K}$ concepts with the highest marginal probability to construct the topic's semantic graph. Figure 6.3 shows the top-10 concepts of

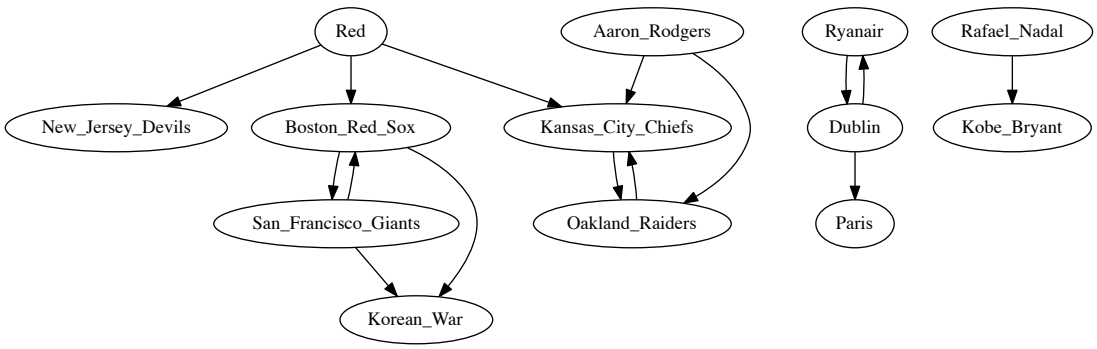


Figure 6.4: Semantic graph of the example topic ϕ described in Fig. 6.3 with $|V^{\phi}| = 13$

a topic learned by OntoLDA.

Definition 17. (Semantic Graph): A *semantic graph* of a topic ϕ is a labeled graph $G^{\phi} = \langle V^{\phi}, E^{\phi} \rangle$, where V^{ϕ} is a set of labeled vertices, which are the top concepts of ϕ (their labels are the concept labels from the ontology) and E^{ϕ} is a set of edges $\{\langle v_i, v_j \rangle$ with label r, such that $v_i, v_j \in V^{\phi}$ and v_i and v_j are connected by a relationship r in the ontology$\}$.

For instance, Figure 6.4 shows the semantic graph of the example topic ϕ in Fig. 6.3, which consists of three sub-graphs (connected components).

Although the ontology relationships induced in G^{ϕ} are directed, in this paper, we will consider the G^{ϕ} as an undirected graph.

6.6.2 Thematic Graph Selection

The selection of the thematic graph is based on the assumption that concepts under a given topic are closely associated in the ontology, whereas concepts from different topics are placed far apart, or even not connected at all. Due to the fact that topic models are statistical and data driven, they may produce topics that are not coherent. In other words, for a given topic that is represented as a list of $\mathcal{K}$ most probable concepts, there may be a few concepts which are not semantically close to other concepts and to the topic, accordingly. As a result, the topic's semantic graph may be composed of multiple connected components.

Definition 18. (Thematic graph): A *thematic graph* is a connected component of G^{ϕ}. In particular, if the entire G^{ϕ} is a connected graph, it is also a thematic graph.

Definition 19. (Dominant Thematic Graph): A thematic graph with the largest number of nodes is called the *dominant thematic graph* for topic ϕ.

Figure 6.5 depicts the dominant thematic graph for the example topic ϕ along with the initial weights of nodes, $p(c_i|\phi)$.

6.6.3 Topic Label Graph Extraction

The idea behind a topic label graph extraction is to find ontology concepts as candidate labels for the topic.

We determine the importance of concepts in a thematic graph not only by their initial weights, which are the marginal probabilities of concepts under the

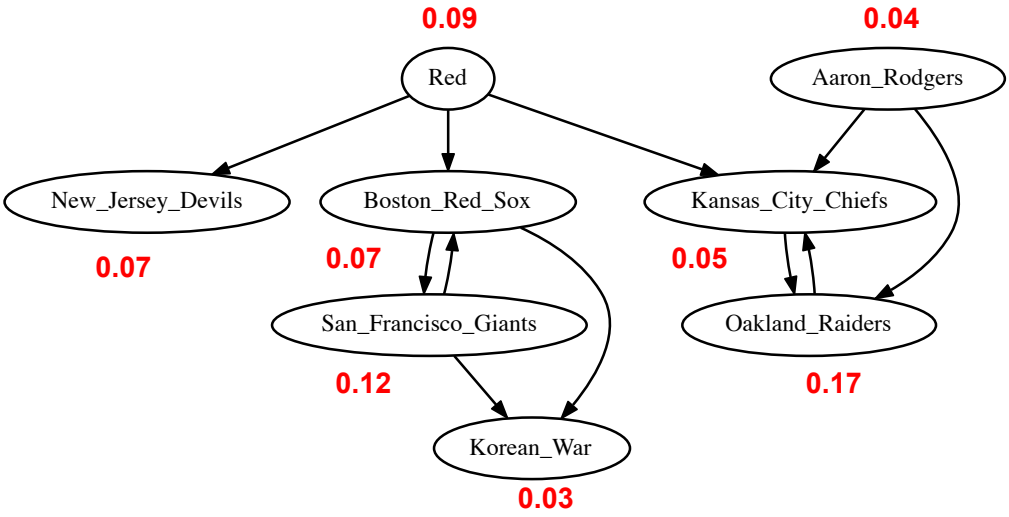


Figure 6.5: Dominant thematic graph of the example topic described in Fig. 6.4

topic, but also by their relative positions in the graph. Here, we utilize the HITS algorithm [54] with the assigned initial weights for concepts to find the *authoritative concepts* in the dominant thematic graph. Subsequently, we locate the *central concepts* in the graph based on the geographical centrality measure, since these nodes can be identified as the thematic landmarks of the graph.

Definition 20. (Core Concepts): The set of the the most authoritative and central concepts in the dominant thematic graph forms the *core concepts* of the topic ϕ and is denoted by CC^{ϕ}.

The top-4 core concept nodes of the dominant thematic graph of example topic ϕ are highlighted in Figure 6.6. It should be noted that "Boston_Red_Sox" has not been selected as a core concept, because it's score is lower than that of the

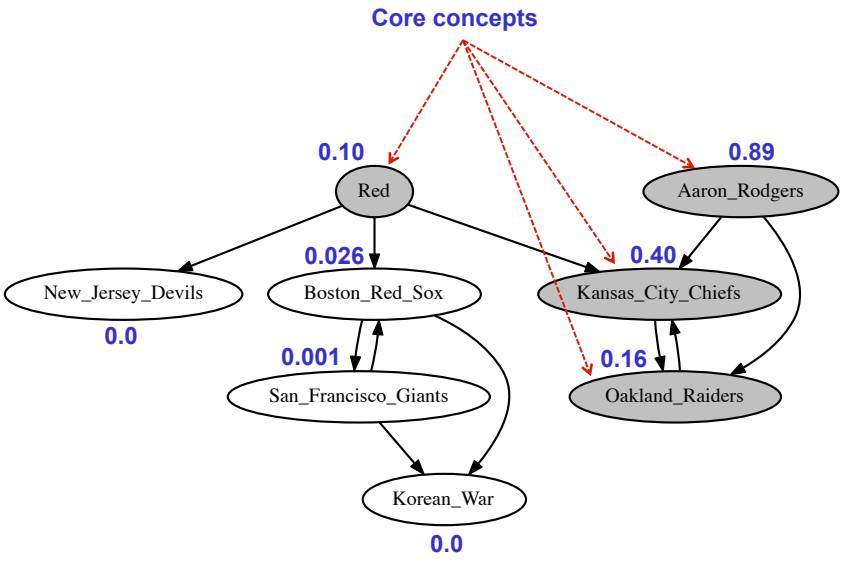


Figure 6.6: Core concepts of the Dominant thematic graph of the example topic described in Fig. 6.5

concept "Red" based on the HITS and centrality computations ("Red" has far more relationships to other concepts in DBpedia).

From now on, we will simply write thematic graph when referring to the dominant thematic graph of a topic.

To extract the topic label graph for the core concepts CC^{ϕ}, we primarily focus on the ontology class structure, since we can consider the topic labeling as assigning class labels to topics. We introduce definitions similar to those in [48] for describing the label graph and topic label graph.

Definition 21. (Label Graph): The *label graph* of a concept c_i is an undirected graph $G_i = \langle V_i, E_i \rangle$, where V_i is the union of $\{c_i\}$ and a subset of ontology classes

(c_i's types and their ancestors) and E_i is a set of edges labeled by *rdf:type* and *rdfs:subClassOf* and connecting the nodes. Each node in the label graph excluding c_i is regarded as a *label* for c_i.

Definition 22. (Topic Label Graph): Let $CC^{\phi} = \{c_1, c_2, \ldots, c_m\}$ be the core concept set. For each concept $c_i \in CC^{\phi}$, we extract its *label graph*, $G_i = \langle V_i, E_i \rangle$, by traversing the ontology from c_i and retrieving all the nodes laying at most three hops away from C_i. The *union* of these graphs $\boldsymbol{G}_{cc^{\phi}} = \langle \boldsymbol{V}, \boldsymbol{E} \rangle$ where $\mathbf{V} = \bigcup V_i$ and $\mathbf{E} = \bigcup E_i$ is called the *topic label graph*.

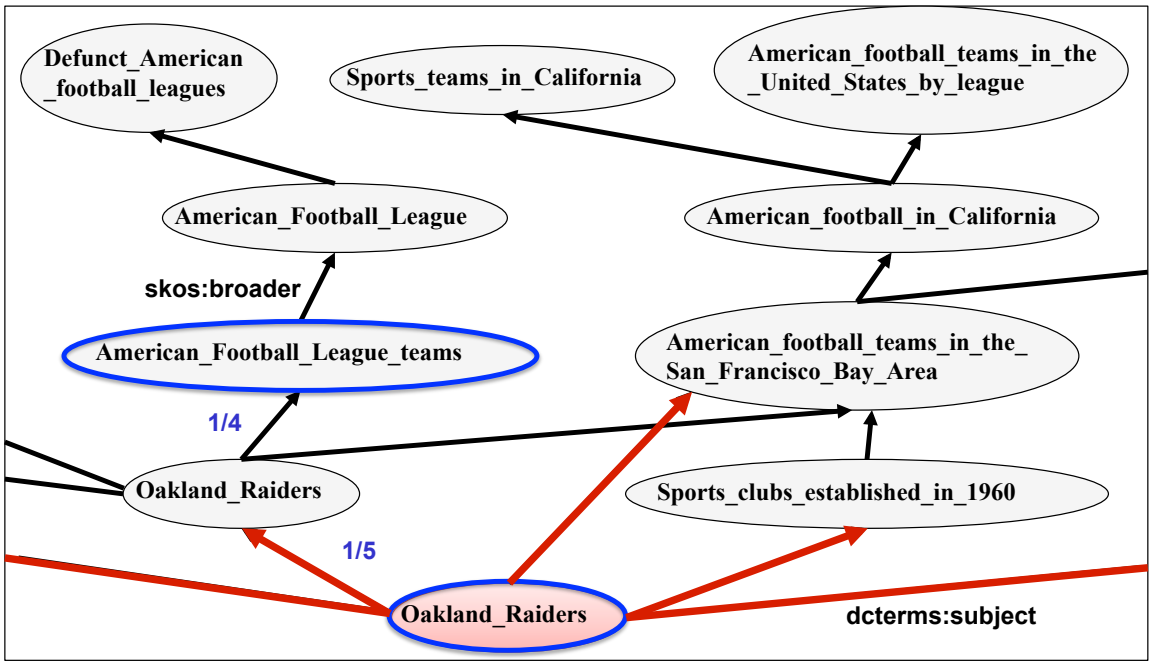
It should be noted that we empirically restrict the ancestors to three levels, due to the fact that increasing the distance further quickly leads to excessively general classes.

6.6.4 Semantic Relevance Scoring Function

In this section, we introduce a semantic relevance scoring function to rank the candidate labels by measuring their semantic similarity to a topic.

Mei et al. [75] describe that the semantics of a topic should be interpreted based on two parameters: (1) distribution of the topic; and (2) the context of the topic. Our topic label graph for a topic ϕ is extracted, taking into account the topic distribution over the concepts as well as the context of the topic in the form of semantic relatedness between the concepts in the ontology.

In order to find the semantic similarity of a label ℓ in $\mathbf{G}_{cc^{\phi}}$ to a topic ϕ, we compute the semantic similarity between ℓ and all of the concepts in the core concept set CC^{ϕ}, rank the labels and then select the best labels for the topic.



$$mScore(Oakland\_Raider, American\_Football\_League\_teams) = \frac{1}{5} \times \frac{1}{4} = 0.05$$

Figure 6.7: Label graph of the concept *"Oakland_Raiders"* along with its *mScore* to the category *"American_Football_League_teams"*.

A candidate label is scored according to three main objectives: (1) the label should cover *important concepts* of the topic (i.e.,concepts with higher marginal probabilities); (2) the label should be specific (lower in the class hierarchy) to the core concepts; and (3) the label should cover the highest number of core concepts in $\mathbf{G}_{cc^{\phi}}$.

To compute the semantic similarity of a label to a concept, we first calculate the *membership score* and the *coverage score*. We have adopted a modified Vector-based Vector Generation method (VVG) described in [99] to calculate the membership score of a concept to a label.

In the experiments described in this paper, we used DBpedia, an ontology created out of Wikipedia. All concepts in DBpedia are classified into DBpedia categories and categories are inter-related via subcategory relationships, including *skos:broader*, *skos:broaderOf*, *rdfs:subClassOf*, *rdfs:type* and *dcterms:subject*. We rely on these relationships for the construction of the label graph. Given the topic label graph $\mathbf{G}_{cc^{\phi}}$ we compute the similarity of the label ℓ to the core concepts of topic ϕ as follows.

If a concept c_i has been classified to N DBpedia categories, or similarly, if a category C_j has N parent categories, we set the weight of each of the membership (classification) relationships e to:

$$m(e) = \frac{1}{N} \tag{6.13}$$

The *membership score*, $mScore(c_i, C_j)$, of a concept c_i to a category C_j is

defined as follows:

$$mScore(c_i, C_j) = \prod_{e_k \in E_l} m(e_k) \tag{6.14}$$

where $E_l = \{e_1, e_2, \ldots, e_m\}$ represents the set of all membership relationships forming the shortest path p from concept c_i to category C_j. Figure 6.7 illustrates a fragment of the label graph for the concept *"Oakland_Raiders"* and shows how its membership score to the category *"American_Football_League_teams"* is computed.

The *coverage score*, $cScore(c_i, C_j)$, of a concept c_i to a category C_j is defined as follows:

$$cScore(w_i, v_j) = \begin{cases} \dfrac{1}{d(c_i, C_j)} & \text{if there is a path from } c_i \text{ to } C_j \\ 0 & \text{otherwise.} \end{cases} \tag{6.15}$$

The *semantic similarity* between a concept c_i and label ℓ in the topic label graph $\mathbf{G}_{cc^\phi}$ is defined as follows:

$$SSim(c_i, \ell) = w(c_i) \cdot \Big[\lambda \cdot mScore(c_i, \ell) + (1 - \lambda) \cdot cScore(c_i, \ell)\Big] \tag{6.16}$$

where $w(c_i)$ is the weight of the c_i in $\mathbf{G}_{cc^\phi}$, which is the marginal probability of concept c_i under topic $\phi, w(c_i) = p(c_i|\phi)$. Similarly, the semantic similarity between a set of core concept CC^ϕ and a label ℓ in the topic label graph $\mathbf{G}_{cc^\phi}$ is

Table 6.6: Example of a topic with top-10 concepts (first column) and top-10 labels (second column) generated by our proposed method

Topic 2	**Top Labels**
oakland_raiders	National_Football_League_teams
san_francisco_giants	American_Football_League_teams
red	American_football_teams_in_the_San_Francisco_Bay_Area
new_jersey_devils	Sports_clubs_established_in_1960
boston_red_sox	National_Football_League_teams_in_Los_Angeles
kansas_city_chiefs	American_Football_League
nigeria	American_football_teams_in_the_United_States_by_league
aaron_rodgers	National_Football_League
kobe_bryant	Green_Bay_Packers
rafael_nadal	California_Golden_Bears_football

defined as:

$$SSim(CC^{\phi}, \ell) = \frac{\lambda}{|CC^{\phi}|} \sum_{i=1}^{|CC^{\phi}|} w(c_i) \cdot mScore(c_i, \ell) + (1-\lambda) \sum_{i=1}^{|CC^{\phi}|} w(c_i) \cdot cScore(c_i, \ell) \tag{6.17}$$

where λ is the smoothing factor to control the influence of the two scores. We used $\lambda = 0.8$ in our experiments. It should be noted that $SSim(CC^{\phi}, \ell)$ score is not normalized and needs to be normalized. The scoring function aims to satisfy the three criteria by using concept *weight*, *mScore* and *cScore* for first, second and third objectives respectively. This scoring function ranks a label node higher, if the label covers more important topical concepts, if it is closer to the core concepts,

and if it covers more core concepts. Top-ranked labels are selected as the labels for the given topic. Table 6.6 illustrates a topic along with the top-10 generated labels using our ontology-based framework.

6.7 Experiments

In order to demonstrate the effectiveness of our OntoLDA method, utilizing ontology-based topic models, we compared it to one of the state-of-the-art traditional, text-based approaches described in [75]. We will refer to that method as Mei07.

We selected two different data sets for our experiments. First, we extracted the top-2000 bigrams using the N-gram Statistics Package [6]. Then, we tested the significance of the bigrams using the Student's T-Test, and extracted the top 1000 candidate bigrams $\mathcal{L}$. For each label $\ell \in \mathcal{L}$ and topic ϕ, we computed the score s, defined by the authors as:

$$s(\ell, \phi) = \sum_{w} \Big(p(w|\phi) PMI(w, \ell|D) \Big) \tag{6.18}$$

where PMI is the point-wise mutual information between the label ℓ and the topic words w, given the document corpus D. We selected the top-6 labels as the labels of the topic ϕ generated by the Mei07 method.

6.7.1 Data Sets and Concept Selection

The experiments in this paper are based on two text corpora and the DBpedia ontology. The text collections are: the British Academic Written English Corpus

(BAWE) [85], and a subset of the Reuters[4] news articles. BAWE contains 2,761 documents of proficient university-level student writing that are fairly evenly divided into four broad disciplinary areas (Arts and Humanities, Social Sciences, Life Sciences and Physical Sciences) covering 32 disciplines. In this paper, we focused on the documents categorized as LIFE SCIENCES (covering Agriculture, Biological Sciences, Food Sciences, Health, Medicine and Psychology) consisting of $D = 683$ documents and 218,692 words. The second dataset is composed of $D = 1,414$ Reuters news articles divided into four main topics: *Business*, *Politics*, *Science*, and *Sports*, consisting of 155,746 words.

Subsequently, we extracted 20 major topics from each dataset using OntoLDA and, similarly, 20 topics using Mei07.

The DBpedia ontology created from the English language subset of Wikipedia includes over 5,000,000 concepts. Using the full set of concepts included in the ontology is computationally very expensive. Therefore, we selected a subset of concepts from DBpedia that were relevant to our datasets. We identified 16,719 concepts (named entities) mentioned in the BAWE dataset and 13,676 in the Reuters news dataset and used these concept sets in our experiments.

6.7.2 Experimental Setup

We pre-processed the datasets by removing punctuation, stopwords, numbers, and words occurring fewer than 10 times in each corpus. For each concept in the two concept sets, we created a bag of words by downloading its Wikipedia page and

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

collecting the text, and eventually, constructed a vocabulary for each concept set. Then, we created a $W = 4,879$ vocabulary based on the intersection between the vocabularies of BAWE corpus and its corresponding concept set. We used this vocabulary for experiments on the BAWE corpus. Similarly, we constructed a $W = 3,855$ vocabulary by computing the intersection between the Reuters news articles and its concept set and used that for the Reuters experiments. We assumed symmetric Dirichlet prior and set $\beta = 0.01$ and $\gamma = 0.01$. We ran the Gibbs sampling algorithm for 500 iterations and computed the posterior inference after the last sampling iteration.

6.7.3 Results

Tables 6.7 and 6.8 present sample results of our topic labeling method, along with labels generated from the Mei07 method as well as the top-10 words for each topic. For example, the columns with title "Topic 1" show and compare the top-6 labels generated for the same topic under Mei07 and the proposed OntoLDA method, respectively. We compared the top-6 labels and the top words for each topic are also shown in the respective Tables. We believe that the labels generated by OntoLDA are more meaningful than the corresponding labels created by the Mei07 method.

In order to quantitatively evaluate the two methods, we asked three human assessors to compare the labels. We selected a subset of topics in a random order and for each topic, the judges were given the top-6 labels generated by the OntoLDA method and Moi07. The labels were listed randomly and for each label the

assessors had to choose between "Good" and "Unrelated".

We compared the two different methods using the *Precision@k*, taking the top-1 to top-6 generated labels into consideration. Precision for a topic at top-k is defined as follows:

$$Precision@k = \frac{\#\text{ of "Good" labels with rank} \leq k}{k} \tag{6.19}$$

We then averaged the precision over all the topics. Figure 6.8 illustrates the results for each individual corpus.

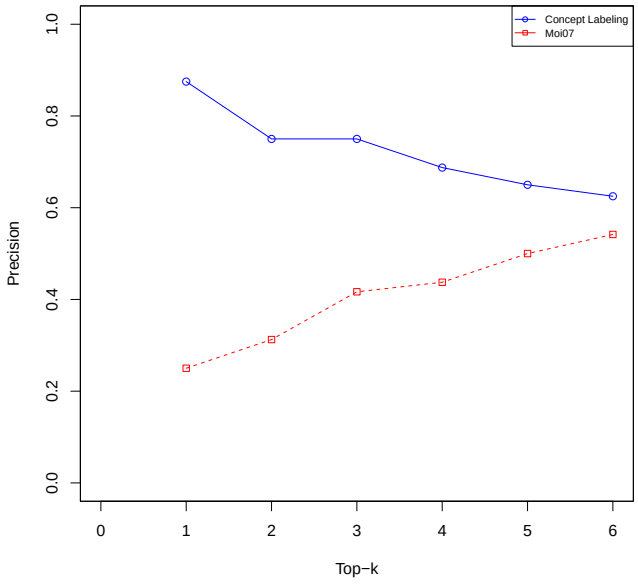
The results in Figure 6.8, reveal two interesting observations: (1) in Figure 6.8(a), the precision difference between the two methods illustrates the effectiveness of our method, particularly for up to top-3 labels, and (2) the average precision for the BAWE corpus is higher than for the Reuters corpus. Regarding (1), our method assigns the labels that are more specific and meaningful to the topics. As we select more labels, they become more general and likely too broad for the topic, which impacts the precision. For the BAWE corpus as shown in 6.8(b), the precision begins to rise as we select more top labels and then starts to fall. The reason for this is that OntoLDA finds the labels that are likely too specific to match the topics. But, as we choose further labels ($1 < k \leq 4$), they become more general but not too broad to describe the topics, and eventually ($k > 4$) the labels become too general and consequently not appropriate for the topics. Regarding observation (2), the BAWE documents are educational and scientific, and phrases used in scientific documents are more discriminative than in news articles. This makes the constructed semantic graph include more inter-related concepts and ul-

Table 6.7: Sample topics of the BAWE corpus with top-6 generated labels for the Mei method and OntoLDA + Concept Labeling, along with top-10 words

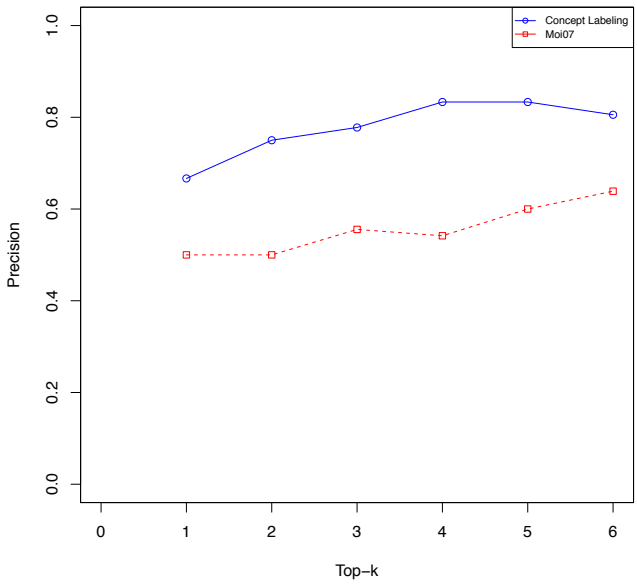
Mei07				
Topic 1	**Topic 3**	**Topic 12**	**Topic 9**	**Topic 6**
rice production	cell lineage	nuclear dna	disabled people	mg od
southeast asia	cell interactions	eukaryotic organelles	health inequalities	red cells
rice fields	somatic blastomeres	hydrogen hypothesis	social classes	heading mr
crop residues	cell stage	qo site	lower social	colorectal carcinoma
weed species	maternal effect	iron sulphur	black report	cyanosis oedema
weed control	germline blastomeres	sulphur protein	health exclusion	jaundice anaemia
OntoLDA + Concept Labeling				
Topic 1	**Topic 3**	**Topic 12**	**Topic 9**	**Topic 6**
agriculture	structural proteins	bacteriology	gender	aging-associated diseases
tropical agriculture	autoantigens	bacteria	biology	smoking
horticulture and gardening	cytoskeleton	prokaryotes	sex	chronic lower respiratory
model organisms	epigenetics	gut flora	sociology and society	inflammations
rice	genetic mapping	digestive system	identity	human behavior
agricultur in the united kingdom	teratogens	firmicutes	sexuality	arthritis
Topic top-10 words				
Topic 1	**Topic 3**	**Topic 12**	**Topic 9**	**Topic 6**
soil	cell	bacteria	health	history
water	cells	cell	care	blood
crop	protein	cells	social	disease
organic	dna	bacterial	professionals	examination
land	gene	immune	life	pain
plant	acid	organisms	mental	medical
control	proteins	growth	medical	care
environmental	amino	host	family	heart
production	binding	virus	children	physical
management	membrane	number	individual	information

Table 6.8: Sample topics of the Reuters corpus with top-6 generated labels for the Mei method and OntoLDA + Concept Labeling, along with top-10 words

Mei07				
Topic 20	**Topic 1**	**Topic 18**	**Topic 19**	**Topic 3**
hockey league	mobile devices	upgraded falcon	investment bank	russel said
western conference	ralph lauren	commercial communications	royal bank	territorial claims
national hockey	gerry shih	falcon rocket	america corp	south china
stokes editing	huffington post	communications satellites	big banks	milk powder
field goal	analysts average	cargo runs	biggest bank	china sea
seconds left	olivia oran	earth spacex	hedge funds	east china
OntoLDA + Concept Labeling				
Topic 20	**Topic 1**	**Topic 18**	**Topic 19**	**Topic 3**
national football league teams	investment banks	space agencies	investment banking	island countries
washington redskins	house of morgan	space organizations	great recession	liberal democracies
sports clubs established in 1932	mortgage lenders	european space agency	criminal investigation	countries bordering the philippine sea
american football teams in maryland	jpmorgan chase	science and technology in europe	madoff investment scandal	east asian countries
american football teams in virginia	banks established in 2000	organizations based in paris	corporate scandals	countries bordering the pacific ocean
american football teams in washington d.c.	banks based in new york city	nasa	taxation	countries bordering the south china sea
Topic top-10 words				
Topic 20	**Topic 1**	**Topic 18**	**Topic 19**	**Topic 3**
league	company	space	bank	china
team	stock	station	financial	chinese
game	buzz	nasa	reuters	beijing
season	research	earth	stock	japan
football	profile	launch	fund	states
national	chief	florida	capital	south
york	executive	mission	research	asia
games	quote	flight	exchange	united
los	million	solar	banks	korea
angeles	corp	cape	group	japanese



(a) Precision for Reuters Corpus



(b) Precision for BAWE Corpus

Figure 6.8: Comparison of the systems using human evaluation

timately leads to the selection of concepts that are good labels for the scientific documents, which is also discussed in [75].

Topic Coherence. In our model, the topics are represented over concepts. Hence, in order to compute the word distribution for each topic t under OntoLDA, we can use the following formula:

$$\vartheta_t(w) = \sum_{c=1}^{\mathcal{C}} \Big(\zeta_c(w) \cdot \phi_t(c) \Big) \tag{6.20}$$

Table 6.9 shows three example topics from the BAWE corpus. Each "topic" column illustrates the top words from LDA and OntoLDA, respectively.

Based on Table 6.9, we can draw an interesting observation. Although both LDA and OntoLDA represent the top words for each topic, the ***topic coherence*** under OntoLDA is qualitatively better than LDA. For each topic we italicized and marked in red the wrong topical words. We can see that OntoLDA produces much better topics than LDA does. For example, "Topic 3" in Table 6.9 shows the top words for the same topic under standard LDA and OntoLDA. LDA did not perform well, as some words in most of the topics were considered as not relevant to the topic.

We performed quantitative comparison of the coherence of the topics created using OntoLDA and LDA, computing the *coherence score* based on the formula presented in [82]. This has become the most commonly used topic coherence evaluation method. Given a topic ϕ and its top T words $V^{(\phi)} = (v_1^{(\phi)}, \cdots, v_T^{(\phi)})$

Table 6.9: Example topics from the two document sets (top-10 words are shown). The third row presents the manually assigned labels

BAWE Corpus						**Reuters Corpus**			
Topic 1		**Topic 2**		**Topic 3**		**Topic 7**		**Topic 8**	
Agriculture		Medicine		Gene Expression		Sports-Football		Financial Companies	
LDA	**OntoLDA**	**LDA**	**OntoLDA**	**LDA**	**OntoLDA**	**LDA**	**OntoLDA**	**LDA**	**OntoLDA**
soil	soil	*list*	history	cell	cell	game	league	company	company
control	water	history	blood	cells	cells	team	team	million	stock
organic	crop	patient	disease	*heading*	protein	season	game	billion	buzz
crop	organic	pain	examination	expression	dna	players	season	business	research
heading	land	examination	pain	*al*	gene	left	football	executive	profile
production	plant	diagnosis	medical	*figure*	acid	time	national	revenue	chief
crops	control	*mr*	care	protein	proteins	games	york	shares	executive
system	environmental	*mg*	heart	genes	amino	*sunday*	games	companies	quote
water	production	problem	physical	gene	binding	football	los	chief	million
biological	management	disease	treatment	*par*	membrane	*pm*	angeles	customers	corp

Table 6.10: Topic Coherence on top T words. A higher coherence score means the topics are more coherent

	BAWE Corpus			**Reuters Corpus**		
T	**5**	**10**	**15**	**5**	**10**	**15**
LDA	−223.86	−1060.90	−2577.30	−270.48	−1372.80	−3426.60
OntoLDA	**−193.41**	**−926.13**	**−2474.70**	**−206.14**	**−1256.00**	**−3213.00**

ordered by $P(w|\phi)$, the coherence score is defined as:

$$C(\phi; V^{(\phi)}) = \sum_{t=2}^{T} \sum_{l=1}^{t-1} \log \frac{D(v_t^{(\phi)}, v_l^{(\phi)}) + 1}{D(v_l^{(\phi)})} \tag{6.21}$$

where $D(v)$ is the document frequency of word v and $D(v, v')$ is the number of documents in which words v and v' co-occurred. It is demonstrated that the coherence score is highly consistent with human-judged topic coherence [82]. Higher coherence scores indicates higher quality of topics. The results are illustrated in Table 6.10.

As we mentioned before, OntoLDA represents each topic as a distribution over concepts. Table 6.11 illustrates the top-10 concepts of highest probabilities in the topic distribution under the OntoLDA framework for the same three topics ("topic 1", "topic2" and "topic3") of Table 6.9. Because concepts are more informative than individual words, the interpretation of topics is more intuitive in OntoLDA than that of standard LDA.

Table 6.11: Example topics with top-10 concept distributions in OntoLDA model

Topic 1		**Topic 2**		**Topic 3**	
rice	0.106	hypertension	0.063	actin	0.141
agriculture	0.095	epilepsy	0.053	epigenetics	0.082
commercial agriculture	0.067	chronic bronchitis	0.051	mitochondrion	0.067
sea	0.061	stroke	0.049	breast cancer	0.066
sustainable living	0.047	breastfeeding	0.047	apoptosis	0.057
agriculture in the united kingdom	0.039	prostate cancer	0.047	ecology	0.042
fungus	0.037	consciousness	0.047	urban planning	0.040
egypt	0.037	childbirth	0.042	abiogenesis	0.039
novel	0.034	right heart	0.024	biodiversity	0.037
diabetes management	0.033	rheumatoid arthritis	0.023	industrial revolution	0.036

6.8 Conclusions

In this paper, we presented OntoLDA, an ontology-based topic model, along with a graph-based topic labeling method for the task of topic labeling. Experimental results show the effectiveness and robustness of the proposed method when applied on different domains of text collections. The proposed ontology-based topic model improves the topic coherence in comparison to the standard LDA model by integrating ontological concepts with probabilistic topic models into a unified framework.

There are many interesting future extensions to this work. It would be interesting to define a global optimization scoring function for the labels instead of Eq. 6.17. Furthermore, how to incorporate the hierarchical relations as well as *lateral* relationships between the ontology concepts into the topic model, is also an interesting future direction.

Chapter 7

Combining Topic Models with Wikipedia Category Network for Semantic Tagging [1]

[1] Mehdi Allahyari and Krys Kochut. "Semantic Tagging Using Topic Models exploiting Wikipedia Category Network".
Submitted to the *Semantic Web Journal*.
A shorter, preliminary version of the paper, "*Semantic Tagging Using Topic Models exploiting Wikipedia Category Network*" , has been published in 2016 IEEE 10th International Conference on Semantic Computing (ICSC), Pages 63-70

Abstract

The volume of documents and online resources has been increasing significantly on the Web for many years. Effectively, organizing this huge amount of information has become a challenging problem. Tagging is a mechanism to aggregate information and a great step towards the Semantic Web vision. Tagging aims to organize, summarize, share and search the Web resources in an effective way. One important problem facing tagging systems is to automatically determine the most appropriate tags for Web documents. In this paper, we propose a probabilistic topic model that incorporates DBpedia knowledge into the topic model for tagging Web pages and online documents with topics discovered in them. Our method is based on integration of the DBpedia hierarchical category network with statistical topic models, where DBpedia categories are considered as topics. We have conducted extensive experiments on two different datasets to demonstrate the effectiveness of our method.

7.1 Introduction

The advent of the World Wide Web has made a huge volume of online resources and documents freely accessible. The big challenge is to effectively organize this tremendous amount of information. Tagging is the process of assigning labels to Web resources with the purpose to organize, share, discover and recover them easily. One of the important steps towards the Semantic Web is the automatic tagging of documents and Web pages with ontology concepts, which is also called

ontology-based semantic tagging. A *semantic tag* is a phrase (sequence of words) belonging mainly to the *topic* which describes a tagged resource. Semantic tagging of textual content can significantly benefit information access tasks, for example, by enhancing the development of tools for classification and retrieval of documents, and has attracted significant attention in recent years. In this paper, we address this issue and propose an approach that integrates prior knowledge (i.e., ontological concepts) with unsupervised topic models into a unified probabilistic framework. We use the DBpedia's [11] hierarchical category network as our background knowledge, which includes the categories organized into a hierarchical structure and a set of articles from Wikipedia. We need to note that the DBpedia knowledge base is extracted from Wikipedia in the form of an ontology of concepts and relationships, which includes the Wikipedia classification schema. Thus, we refer to the DBpedia category network and Wikipedia category network interchangeably throughout this paper.

Probabilistic topic models such as Latent Dirichlet Allocation (LDA) [17] are powerful techniques which are widely used for discovering topics or semantic content from a large collection of documents. Topic models typically assume that documents are mixtures of topics, while topics are probability distributions over the words. When the proportions of topics in a document are estimated, the top-proportion topics can be used as the themes (high-level representations of the semantics) of the document. Similarly, top-ranked words in a topic-word distribution indicate the meaning of the topic. Thus, topic models provide an effective framework for extracting the latent semantics from unstructured text collections.

For example, Table 7.1 shows the top words of five topics learned by LDA from a collection of news articles; the topics have been labeled by a human "*Healthcare*", "*Farming*", "*Finance*", "*Disease*" and "*Olympic Sports*", respectively.

Table 7.1: Examples of five topics with their labels.

Healthcare	Farming	Finance	Disease	Olympic Sports
health	farm	bank	disease	world
care	food	fed	blood	games
social	products	banks	infection	year
patient	organic	financial	cells	brazil
people	consumers	central	damage	cup
patients	market	fund	system	sochi
client	quality	market	due	olympic
mental	business	markets	host	team
individual	production	billion	type	olympics
family	product	funds	brain	sports

We consider the Wikipedia categories as the ***topics*** in the probabilistic model, i.e., each document is a multinomial distribution over the Wikipedia categories. Thus, we combine the ontology concepts and data-driven topics, which enables us to semantically tag the documents with Wikipedia categories, after the topic mixtures of documents are estimated.

Our proposed method for semantic tagging is entirely different from supervised text categorization techniques. Supervised text categorization methods are typically based on a set of predefined categories and a set of documents with preassigned categories, which is used as a training set. A classifier is created based on the training set and then is used to predict the categories of previously unseen documents. In the work presented here, we assign categories (topics) from Wikipedia

to text documents for which there are no predefined or known categories. We learn the probability distribution of each category over the words using the statistical topic models taking into account the prior knowledge from Wikipedia about the words and their associated probabilities in various categories. For instance, in Wikipedia, the words "rule", "reasoning" and "triple" have likely higher weights (see section 7.4.2) under the "Knowledge Representation" category and, similarly, the words "democracy", "debate", and "campaign" are more related to the "Politics" category.

We should point out that there exist several knowledge bases such as DBpedia [11] (constructed based on the content of Wikipedia [101]), YAGO [43], and Freebase [18] that could be exploited as the prior knowledge in this work. DBpedia provides different classification schemes, including the Wikipedia and YAGO categorization systems. For this research, we selected DBpedia as arguably more frequently used for Semantic Web tasks, but our approach could be used with other knowledge bases, as well.

In recent years, several attempts have been made for annotating Web pages and online documents. For example, [95] uses linguistic techniques to address annotation of Web resources. [102, 33] utilize various natural language processing and information extraction techniques and [57] employs regular expression patterns for semantic annotation. Our approach differs from previous works in that they are primarily focused on entities mentioned in the documents, whereas we take all the words into consideration. Furthermore, our method tags (annotates) the whole document as a unit, as opposed to annotating entities and other phrases

occuring within the document.

Some other related works include [96] where the author uses article titles and categories of Wikipedia to identify document topics. In that method, the author first finds all the related Wikipedia articles to a document by matching their titles with the words of the document. Then, he selects categories assigned to these articles and ranks them, and finally chooses the categories with the highest weights as the topics of the document. [40] proposes a method that constructs a category-term matrix C from Wikipedia exploiting categories and articles text. Then, for the input document a document-term matrix D is constructed. They eventually, calculate the document-categories similarity matrix $S = DC^T$ in order to find the relevant topics of a document. Our method is different from the aforementioned works in that we use a probabilistic model that incorporates ontological concepts with data-driven topics in a unified framework.

Several other publications focused on combining ontological concepts with statistical topic models. In [25], the authors describe the Concept-Topic Model (CTM), which combines human-defined concepts with LDA. The key idea in their framework is that both topics from the statistical topic models and concepts of the ontology are similarly represented by a set of "focused" words and they use this representation similarity as the key idea in their model. In [26], the authors extended their previous work and proposed the Hierarchical Concept-Topic Model (HCTM), in order to leverage the known hierarchical structure among concepts. Our method is somewhat similar to [25, 26] in terms of exploiting ontologies in the topic models, yet it differs from them in that they model concepts where they

are directly associated with words, whereas in our model, the concepts (Wikipedia categories) are not associated directly with words but associated with documents. Moreover, our method learns the probability distributions of Wikipedia categories over the vocabulary, exploiting the information provided by the background knowledge.

In this paper, we propose a probabilistic approach that exploits prior knowledge from the ontology concepts and integrates it with statistical topic modeling. DBpedia is used as the background knowledge, as it is a rich source of semantically related concepts organized into a category network. Concepts (categories) are directly associated with the documents, not words and Wikipedia articles with their assigned categories provide *labeled features* from which we can infer a concept-word distribution that is later used to tag other documents, such as Web pages, news articles, and other online documents.

7.2 Related Work

Recently, automatic semantic tagging and annotation of documents has attracted a great deal of attention. Semantic annotation or ontology-based semantic tagging is an important component in Semantic Web that can certainly bring significant benefits to many text mining tasks, such as information retrieval [98] and text classification [112]. Thus, several attempts have been made to address this issue.

Most of the existing approaches for semantic annotation of documents have primarily focused on tagging entities and phrases appearing in the textual con-

tent, using a variety of techniques, such as Natural Language Processing (NLP), information extraction, and probabilistic methods. For example, [102] uses a Conditional Random Fields (CRF) approach for semantic annotation. [33] introduces a system called SemTag to perform semantic tagging utilizing NLP techniques. In more recent works, [95] uses linguistic patterns and learning methods to discover entities in text and associate them to classes of an ontology. In [57], authors employ regular expression patterns for semantic annotation of documents.

Wikipedia's category network has previously been used for document topic identification. In [96], the authors propose a method that uses Wikipedia article titles as well as the category network to identify topics of documents. [40] introduces a method where they first, construct a category-term matrix C from the Wikipedia categories and articles text. Then, they construct a document-term matrix D for the input document and as the final step, calculate the document-category similarity matrix $S = DC^T$, in order to find the relevant topics of a document. Our work, presented in this paper, is different from all previous works, because we combine the ontological concepts with the probabilistic topic models within a unified framework.

Several authors have published their research results on methods that integrate concepts of an ontology with statistical modeling. As we already mentioned, [25] proposes a Concept-Topic Model which combines human-defined concepts with topic models. Topics from the statistical models and concepts of the ontology both represent sets of "focused" words that relate to some abstract notions. In [26], the authors describe a Hierarchical Concept-Topic Model that extends the

CTM in [25] to integrate the hierarchical relations between the concepts. Our work presented in this paper, differs from the aforementioned works in that those previous works model the concepts that are associated directly with words of the documents, whereas we associate the concepts with documents and incorporate supervised data provided by concepts features into the LDA-based model to infer concept-word probability distribution. This is a transition from unsupervised topic models to a supervised setting, where labeled information for the concepts exists.

There are also prior works that use probabilistic model for tag recommendation [56, 8, 105, 55]. [56, 55] build an LDA model that uses resources and their associated tags previously assigned by users. Then, they represent each resource with the tags from topics discovered by LDA. The basic idea in [8, 105] is that each document is represented as set of textual features and its assigned tags. Then, the authors train a model from training data and use that model to output a ranked list of tags for new documents. Our proposed method is different from these prior works in that we do not use documents previously associated tags in our model to assign tags to documents. We rely on the documents texts and prior knowledge from the ontology to identify most appropriate semantic tags for them.

The rest of this paper is organized as follows. We begin by presenting a brief overview of Latent Dirichlet Allocation (LDA), the state-of-art probabilistic topic modeling technique. In section 7.4, we describe our proposed *sOntoLDA* model and compare it with the standard LDA. We demonstrate the effectiveness of our proposed method in section 7.5. We present our conclusion and future work in section 7.7.

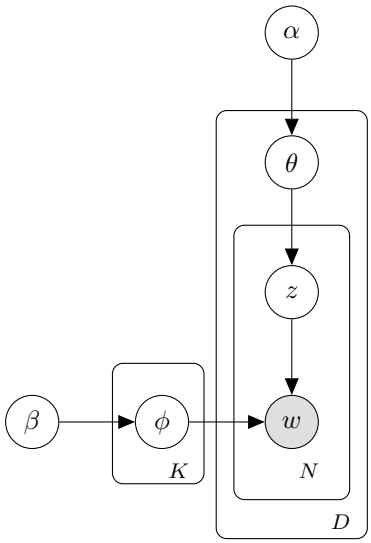


Figure 7.1: Graphical representation of LDA model

7.3 Probabilistic Topic Models

The Latent Dirichlet Allocation (LDA) [17] is a generative probabilistic model which has been extensively used for discovering topics or semantic content from a large collection of documents. LDA assumes that each document is made up of various topics, where each topic is a probability distribution over words. The graphical model of LDA is shown in Figure 7.1 and the generative process is as follows:

1. For each topic $k \in \{1, 2, \ldots, K\}$,

   (a) Draw a word distribution $\phi_k \sim \text{Dir}(\beta)$

2. For each document $d \in \{1, 2, \ldots, D\}$,

   (a) Draw a topic distribution $\theta_d \sim \text{Dir}(\alpha)$

   (b) For each word w_i of document d,

      i. Draw a topic $z_i \sim \text{Mult}(\theta_d)$

      ii. Draw a word w_i from topic $z_i, w \sim \text{Mult}(\phi_{z_i})$

where α and β are parameters of the symmetric Dirichlet prior. In LDA, the words are generated from the topics and topics are generated from documents. In other words, the probability of a word w given a document d is defined as:

$$P(w|d) = \sum_{j=1}^{K} P(w|z_j)P(z_j|d) \tag{7.1}$$

In the standard LDA model, the topic-word probability distributions $P(w|z)$ and document-topic distributions $P(z|d)$ are learned in an entirely unsupervised manner, without integrating any prior knowledge into the statistical framework. In the following section, we describe our sOntoLDA model, where we incorporate prior knowledge of an ontology, that is the DBpedia's category network, into the LDA model.

7.4 Ontology-based Topic Models for Semantic Tagging

In this section, we formally introduce our model. We then describe how to integrate the prior knowledge from the DBpedia's category network into the topic model.

Our objective is to tag (annotate) a corpus of documents with DBpedia (Wikipedia) categories to indicate their semantic content. We assume that documents are not assigned to any predefined categories. Consequently, we do not rely on or require such information in our sOntoLDA model. Our goal is to assign k categories to each document as the topics of the document. This is fundamentally different from the supervised text classification task where a classifier is trained based on a training set of documents that have already been assigned to a fixed set of predefined categories and then used to predict the categories of previously unseen documents.

7.4.1 The sOntoLDA Topic Model

sOntoLDA is a generative topic model for semantic tagging of Web pages and other online documents. The key idea of our model is to integrate prior knowledge from the ontology concepts directly with topic models. The intuition is that the presence of words in documents can be described by both learned topics and human prior knowledge about the words. In standard LDA, word proportions of a topic are drawn from a symmetric Dirichlet distribution. However, in our proposed model, we modify the Dirichlet priors of topic-word distribution by encoding the background knowledge derived from the DBpedia (Wikipedia) hierarchical cate-

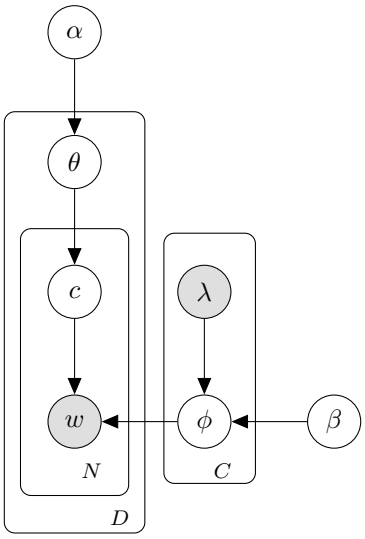


Figure 7.2: Graphical representation of sOntoLDA model

gory network in the form a λ matrix. The graphical representation of sOntoLDA model is illustrated in Figure 7.2 and the generative process is as follows:

1. For each Wikipedia category $c \in \{1, 2, \ldots, C\}$,

   (a) Draw a word distribution $\phi_c \sim \text{Dir}(\lambda_c \times \beta_c)$

2. For each document $d \in \{1, 2, \ldots, D\}$,

   (a) Draw a category distribution $\theta_d \sim \text{Dir}(\alpha)$

   (c) For each word w_i of document d,

      i. Draw a category $c_i \sim \text{Mult}(\theta_d)$

      ii. Draw a word w_i from category $c_i, w_i \sim \text{Mult}(\phi_{c_i})$

The joint distribution of the model (hidden and observed variables) is:

$$P(\phi_{1:C}, \theta_{1:D}, z_{1:D}, w_{1:D}|\alpha, \beta, \lambda_{1:C}) = \prod_{j=1}^{C} P(\phi_j|\lambda_j \times \beta) \prod_{d=1}^{D} P(\theta_d|\alpha) \left(\prod_{n=1}^{N} P(z_{d,n}|\theta_d) P(w_{d,n}|\phi_{1:C}, z_{d,n}) \right) \quad (7.2)$$

Since the task is to tag documents with Wikipedia categories, the latent topics in our model that are associated to each document are Wikipedia categories, and a few most important ones are then used as document's tags. Unlike LDA, we add an additional dependency link to the topic-word distribution ϕ through the matrix λ of size $C \times V$ that we use to encode word prior knowledge.

7.4.2 Building Word-Category Prior Matrix λ

The first step in creating the λ matrix is to prepare the DBpedia category network. Wikipedia has a massive categorization system, which is loosely organized in a hierarchical manner. It contains over $940,000$ categories where relationships between the categories are represented using `SKOS`[2] vocabulary in the DBpedia ontology. The relation between a Wikipedia article and a category is defined by the `subject` property of the Dublin Core[3] vocabulary (prefixed by `dcterms:`). Moreover, a category's parent and child categories are extracted by querying for the properties `skos:broader` and `skos:broaderOf`, respectively.

[2] `http://www.w3.org/2004/02/skos/`
[3] `http://dublincore.org/`

Each category in Wikipedia has a collection of articles placed within it. These articles provide labeled data from which we can infer category-word distributions in sOntoLDA. We use the aforementioned properties and extract these articles to represent each Wikipedia category. Thus, we create a vector of representative terms λ_c for each category c by merging the term vectors of articles defined under c. We assign a tf-idf weight $\delta_w^{(c)}$ to each term w based on its significance to the category as follows:

$$\delta_w^{(c)} = tf_w \times log(\frac{C}{cf_w}) \tag{7.3}$$

where tf_w is the number of occurrences of word w in category c; C is the total number of categories in Wikipedia and cf_w is the number of categories that have this word. Therefore, for each category c:

$$\lambda_c = \left[\delta_{w_1}^{(c)}, \delta_{w_2}^{(c)}, \ldots, \delta_{w_V}^{(c)}\right]^T \tag{7.4}$$

where V is the size of the vocabulary and $\sum_{i=1}^{V} \delta_{w_i}^{(c)} = 1$. Using λ_c as the c'th column, we construct the $V \times C$ word-category matrix λ. This matrix encodes the prior knowledge about the words probabilities in various categories and incorporates this domain knowledge into the topic model. For example, suppose the word "RDF" has a very high weight in the category "Semantic Web". Thus, this word has a much higher probability to be drawn from the "Semantic Web" category word distribution in Eq. 7.8, which indicates that documents having the word "RDF" are more likely related to the topic "Semantic Web".

We also encode the hierarchical structure of the categories into the word-

category matrix λ as it augments the amount of information associated to each category and increases the generality of the categories (topics) assigned to documents. In order to do that, we enhance the vector of representative terms for each category of interest with the set of term-mapping vectors associated to the descendent categories in the hierarchy of categories, under the category of interest. For example, we add the term-vectors that are associated to "Knowledge representation" and "Machine learning" categories to the "Artificial intelligence" category, as they both are sub-categories of "Artificial intelligence" in the Wikipedia hierarchical category network. This includes all of the sub-categories in the hierarchy down to a specific level ℓ. Based on our initial experiments, we empirically restrict the hierarchy height to $\ell = 3$. The reasons for this restriction are: (1) going down deeper and adding more sub-categories makes the λ matrix larger and accordingly, computing the sOntoLDA parameters computationally more expensive, and (2) although increasing the sub-categories' information enhances the quantity of information related to the main category, it also augments the amount of noise. By noise, we mean a subset of sub-categories that becomes very particular and contains information that is specifically related to the sub-categories, but not related to the main category. For instance, "Speech synthesis software" is a sub-category of the "Health" category if $\ell = 6$, but this category primarily includes articles and sub-categories that are more related to "Technology" category.

7.4.3 Parameter Estimation using Gibbs Sampling

Since the posterior inference of sOntoLDA is intractable, we need to find an algorithm for estimating this posterior inference. A variety of algorithms have been used to estimate the parameters of topic models, such as variational EM [17] and Gibbs sampling [38]. In our sOntoLDA topic model presented in this paper, we use the collapsed Gibbs sampling procedure. Collapsed Gibbs sampling [38] is a Markov Chain Monte Carlo (MCMC) algorithm, which constructs a Markov chain over the latent variables in the model and converges to the posterior distribution after a number of iterations. In our case, we aim to construct a Markov chain that converges to the posterior distribution over $\mathbf{c}$ conditioned on the observed words $\mathbf{w}$, word-category prior matrix $\boldsymbol{\lambda}$ and hyperparameters α and β.

We derive the posterior inference from Eq. 7.2 as follows:

$$P(\mathbf{c}|\mathbf{w}, \boldsymbol{\lambda}, \alpha, \beta) = \frac{P(\mathbf{c}, \mathbf{w}|\boldsymbol{\lambda}, \alpha, \beta)}{P(\mathbf{w}|\boldsymbol{\lambda}, \alpha, \beta)} \propto P(\mathbf{c}, \mathbf{w}|\boldsymbol{\lambda}, \alpha, \beta) \propto P(\mathbf{c})P(\mathbf{w}|\boldsymbol{\lambda}, \mathbf{c}) \tag{7.5}$$

where

$$P(\mathbf{c}) = \left(\frac{\Gamma(C\alpha)}{\Gamma(\alpha)^C}\right)^D \prod_{d=1}^{D} \frac{\prod_{c=1}^{C} \Gamma(n_c^{(d)} + \alpha)}{\Gamma(\sum_{c'}(n_{c'}^{(d)} + \alpha))} \tag{7.6}$$

$$P(\mathbf{w}|\boldsymbol{\lambda}, \mathbf{c}) = \left(\frac{\Gamma(\sum_{w=1}^{V} \lambda_w \beta)}{\prod_{w=1}^{V} \Gamma(\lambda_w \beta)}\right)^C \prod_{c=1}^{C} \frac{\prod_{w=1}^{V} \Gamma(n_w^{(c)} + \lambda_{wc}\beta)}{\Gamma(\sum_{w'}(n_{w'}^{(c)} + \lambda_{w'c}\beta))} \tag{7.7}$$

$$P(c_i = c|w_i = w, \mathbf{c}_{-i}, \mathbf{w}_{-i}, \lambda, \alpha, \beta) \propto \frac{n^{(d)}_{c,-i} + \alpha_c}{\sum_{c'} (n^{(d)}_{c',-i} + \alpha_{c'})} \times \frac{n^{(c)}_{w,-i} + \lambda_{wc}\beta_w}{\sum_{w'} (n^{(c)}_{w',-i} + \lambda_{w'c}\beta_{w'})} \quad (7.8)$$

where $n^{(c)}_w$ is the number of times the word w is assigned to the concept c. $n^{(d)}_c$ denotes the number of times the concept c is associated with the document d. The subscript $-i$ indicates that the contribution of the current word w_i being sampled is disregarded. Instead of using symmetric estimation of the parameters α, we use the moment matching methods [83] to approximate these parameters.

After Gibbs sampling, we can use the sampled categories to estimate the probability of a category, given a document θ_{cd} and the probability of a word, given a category ϕ_{wc}:

$$\theta_{cd} = \frac{n^{(d)}_c + \alpha_c}{\sum_{c'} (n^{(d)}_{c'} + \alpha_{c'})} \quad \phi_{wc} = \frac{n^{(c)}_w + \lambda_{wc}\beta_w}{\sum_{w'} (n^{(c)}_{w'} + \lambda_{w'c}\beta_{w'})} \quad (7.9)$$

7.5 Experiments

In order to test sOntoLDA, we performed two different types of experiments. In the first experiment, we focused on how well the proposed method is able to predict the categories of a collection of the Wikipedia articles. Here, we were able to compare the quality of the sOntoLDA-generated tags (categories) to those assigned by the Wikipedia's human curators. In the second experiment, we assigned Wikipedia categories to a corpus of Reuters news articles and investigated the relevance the

top-k topics assigned to the documents, as compared to the pre-assigned categories of the Reuters documents.

Wikipedia is an enormous knowledge base consisting of millions of articles (over $5,000,000$ in the English language section, as of this writing) and nearly a million of categories ($940,000$). Using the full set of articles and categories (category network) included in Wikipedia is computationally very expensive. Thus, we selected a subset of categories and their associated articles that were relevant to our datasets. We created a *topic graph* from Wikipedia hierarchical category graph for each of the main categories, including *Business*, *Applied Sciences*, and *Health*. For each category's sub-graph, we restricted the levels of hierarchy to three and removed the Wikipedia administrative and maintenance categories. The *final topic graph*, which we used as the prior knowledge, was the union of these three topic graphs. For each category in the final topic graph, we retrieved all of the associated articles that had at least 200 words. The final topic graph included $C = 1,353$ categories, the vocabulary of size $V = 99,665$ (excluding punctuation, stopwords, numbers, and words occurring fewer than 5 times in the corpus) and $|A| = 30,300$ articles. From the final topic graph, we constructed the λ matrix of size 1353×99665.

7.5.1 Tagging Wikipedia Articles

For this experiment, we first extracted the Wikipedia categories and sub-categories (1,353 of them) from the three main categories, including Business, Applied Sciences, and Health. We then randomly selected 5 articles from each category

and constructed an initial corpus of $|D_{initial}| = 6,765$ articles. Then, we divided the corpus into a training set (80%) and a test set (20%) and retrieved the corresponding Wikipedia articles. The final sizes of the training and test sets were $|D_{train}| = 3,142$ and $|D_{test}| = 725$ documents, respectively. We used the training dataset to estimate the parameters of the sOntLDA topic model. We assumed the symmetric Dirichlet prior and set $\alpha = 50/K$ and $\beta = 0.01$, respectively. We ran the Gibbs sampling algorithm for 500 iterations and computed the posterior inference after the last sampling iteration.

After estimating the parameters of sOntoLDA, we ran the model on the previously unseen documents of the test set and assigned the top-k categories to each document, using Eq. 7.9. Then, we evaluated how many of the official (i.e., curator-assigned) categories of each document have been assigned by sOntoLDA, which we called the *"exact match"*.

In order to quantitatively measure the quality of the assigned categories (tags) we adopted the *Precision@k* and *Mean Average Precision (MAP)* measures, which have been widely used in the area of information retrieval [72]. We also utilized the *Coverage* metric described in [48]. *Precision@k* is the percentage of correctly identified categories among top-k categories in the test documents. In other words, this metric assesses ***how many relevant/irrelevant categories are retrieved at top-k ranks*** and is defined as follows:

$$Precision@k = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{CI@k}{k} \tag{7.10}$$

Table 7.2: Precision, Coverage and MAP values of "exact match" for Wikipedia Dataset.

Top-k	**Precision**	**Coverage**	**MAP**
1	0.479	0.479	0.479
2	0.509	0.584	0.494
3	0.559	0.645	0.516
4	0.586	0.68	0.533
5	0.605	0.698	0.548
10	0.648	0.744	0.592
15	0.678	0.775	0.617
20	0.702	0.799	0.636
25	0.71	0.804	0.65
30	0.719	0.811	0.661

where $|Q|$ is the number of test documents and $CI@k$ is the number of official (Wikipedia-assigned) categories retrieved among the top-k categories. Note that if k is smaller than the number of official categories, we presume that there are only k official categories.

The measurement values are represented in Table 7.2, and Figure 7.3 illustrates the corresponding plot of the evaluation results of *"exact match"* for the Wikipedia dataset. It shows that on average 65% of the official categories have been retrieved among the *top-10* categories, and the percentage of the *Precision* grows to 72% as we increase the number of the top-k categories assigned to documents to $k = 30$. It should be noted that the top categories are *immediate official categories* assigned to each document in the Wikipedia's hierarchical category network.

The other metric that we used in our evaluation was the *Mean Average Pre-*

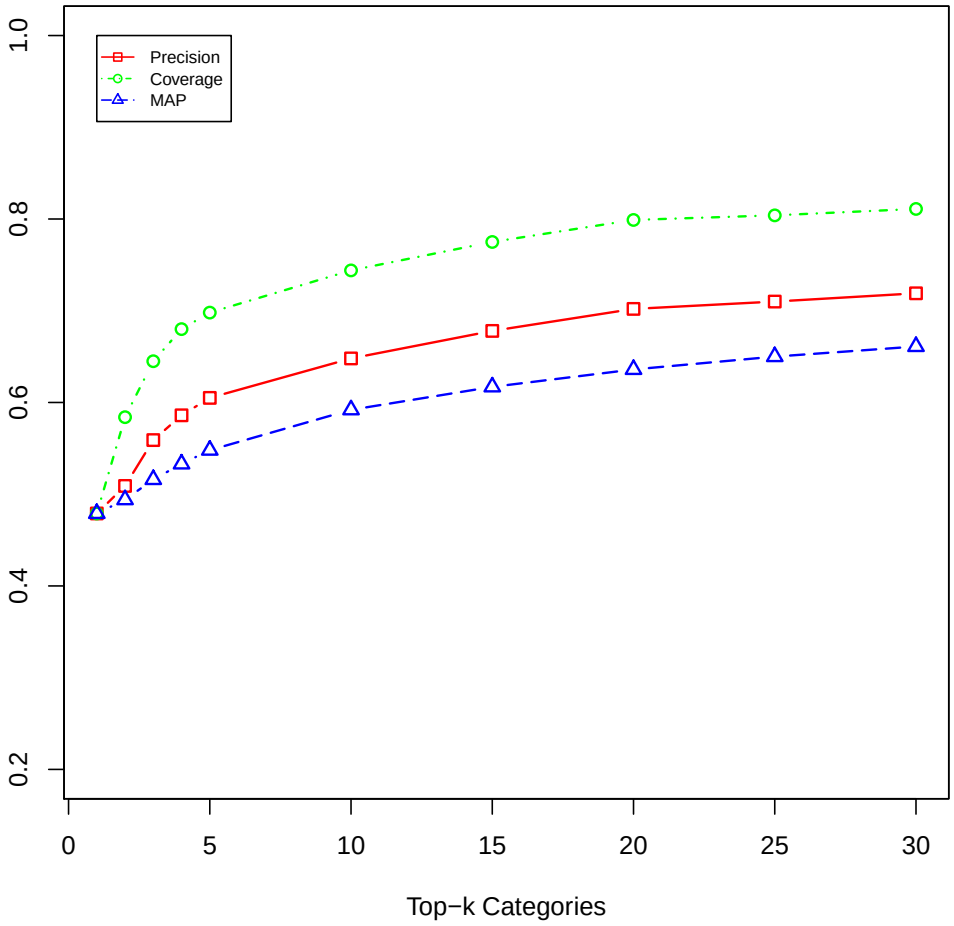


Figure 7.3: Precision, Coverage and MAP of EXACT MATCH for Wikipedia Dataset.

cision (MAP), which measures ***how well the retrieved relevant categories are ranked at top-k***. It is formally defined as follows:

$$MAP(Q) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} Precision(R_{jk}) \tag{7.11}$$

where $|Q|$ is the number of test documents, m_j is the number of relevant categories for document j. $Precision(R_{jk})$ is the *Precision@k* of document j. The higher the *MAP*, the more relevant are the *top-k categories* ranked.

Similarly to *Precision*, we calculated the *MAP* for different numbers of top-k categories, ranging from 1 to 30. As shown in Figure 7.3, the *MAP* at *top-10* categories is 59% and increases to 66% for $k = 30$.

Coverage is the proportion of the documents for which the method has found at least one Hit and is defined as follows:

$$Coverage@k = \frac{\#\text{documents with at least on Hit at rank} \leq k}{\#\text{documents}} \tag{7.12}$$

As illustrated in Figure 7.3, we can see that our proposed method recognized *at least* one official category for 55% of the examined documents within the first *top-5* categories and it grows to over 66% for $k = 30$.

The above results are for the *"exact match"* and are based on the constraint that only the official categories must be among the top categories. However, in Wikipedia, the categories are hierarchically related via *"sub-category"* relation. In other words, the structure of the Wikipedia categorization systems and the

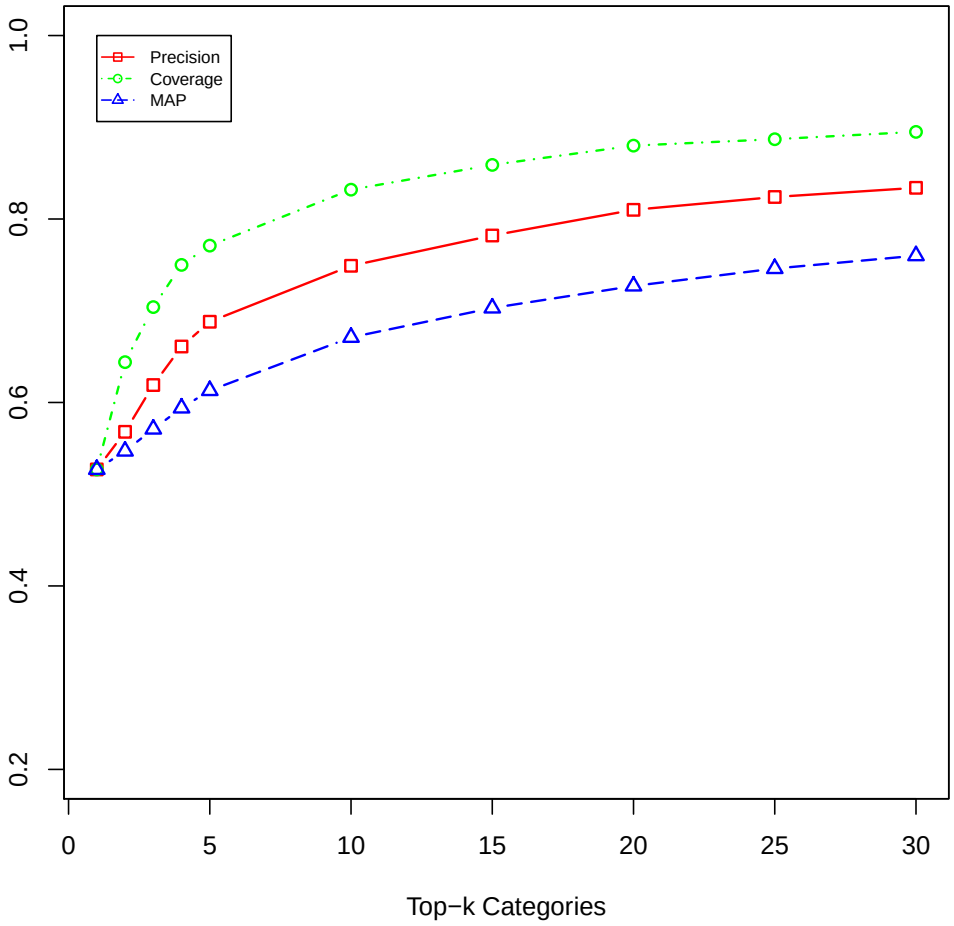


Figure 7.4: Precision, Coverage and MAP of HIERARCHICAL MATCH for Wikipedia Dataset.

Table 7.3: Precision, Coverage and MAP values of "Hierarchical match" for Wikipedia Dataset.

Top-k	Precision	Coverage	MAP
1	0.527	0.527	0.527
2	0.568	0.644	0.547
3	0.619	0.704	0.571
4	0.661	0.75	0.594
5	0.688	0.771	0.613
10	0.749	0.832	0.671
15	0.782	0.859	0.703
20	0.81	0.88	0.727
25	0.824	0.887	0.746
30	0.834	0.895	0.76

relationships between the categories are represented by SKOS properties, including `skos:broader` and `skos:broaderOf`. Moreover, there are thousands of very fine-grained, specific categories created and assigned to Wikipedia articles. These highly specific categories may not be of high interest to users or not be quite informative and meaningful. For example, the Wikipedia article "Semantic Web" contains several categories, including "Internet ages". This category is very specialized and only assigned to two articles. But, one of its super-categories, "World wide web" is more general and informative, which makes it more likely to be interesting to users and a better choice for tagging documents. As another example, the article "Tim Berners-Lee" involves 31 categories, including "Fellows of the British Computer Society". This category is very particular and possibly not suitable enough for tagging, as opposed to "Information technology", which is one of

its ancestor categories and a better choice for tagging the article. Although the results for the *"exact match"* indicate that sOntoLDA works really well, it would be a better approach to also consider the super-categories of official categories as suitable tags for documents.

If we *relax* the constraint of only considering the official, exact categories, and take into account also their super-categories, which we call *"Hierarchical match"*, *Precision*, *Coverage* and *MAP* improve approximately $5 - 12\%$, $5 - 10\%$ and $5-10\%$, respectively. The values of these measurements are presented in Table 7.3. Figure 7.4 shows the results when *Hierarchical match* is taken into consideration, which indicates the significant enhancement in the performance results.

7.5.2 Example of Tagging a Wikipedia Article

As an example, Table 7.4 shows the top five categories that our sOntoLDA model assigned as tags to the article "Tooth brushing". In Wikipedia, only a single official category "Oral hygiene" is assigned to this article, which our method has identified as the top category with the highest probability. The only official category has received roughly four times the probability of the second category, and except for the "Chiropractic treatment techniques" category, which might not be very relevant, the other categories are strongly related to the main category "Oral hygiene" and, correspondingly, to the more general category of "Health" by *"super-category"* relationship (`skos:broader`). Figure 7.5 shows some of the relationships between the top four categories and the article using the Wikipedia hierarchical network. The thickness of the ellipse encapsulating a category node is proportional

to the probability of the category given the article.

Table 7.4: Top 5 categories selected for the article "Tooth brushing".

Article Title: Tooth brushing	
Category	**Probability**
Oral hygiene	0.1533
Dentistry	0.0478
Self care	0.0403
Personal hygiene products	0.0302
Chiropractic treatment techniques	0.0227

7.5.3 Tagging Evaluation

To evaluate our method on a real-world document set, we selected a corpus of $D = 2,914$ documents from the Reuters' news articles divided (by Reuters editors) into three main categories: Business, Science and Health. The reason we chose our corpus from these categories was that our prior knowledge was created out of the corresponding Wikipedia categories. We can also map the categories of the documents in this text corpus to their corresponding Wikipedia categories. It should be noted that the number of categories tagged to each document in the Reuters corpus was at least 1 and at most 3. Therefore, in order to be able to directly evaluate the performance of top-k Wikipedia categories assigned to these documents via our method, we employed the *"Hierarchical match"* method used for the Wikipedia dataset. For each main category, not only the corresponding Wikipedia category but also all the descendent categories resulting from the subgraph of that main category were considered as the correct topics of the document.

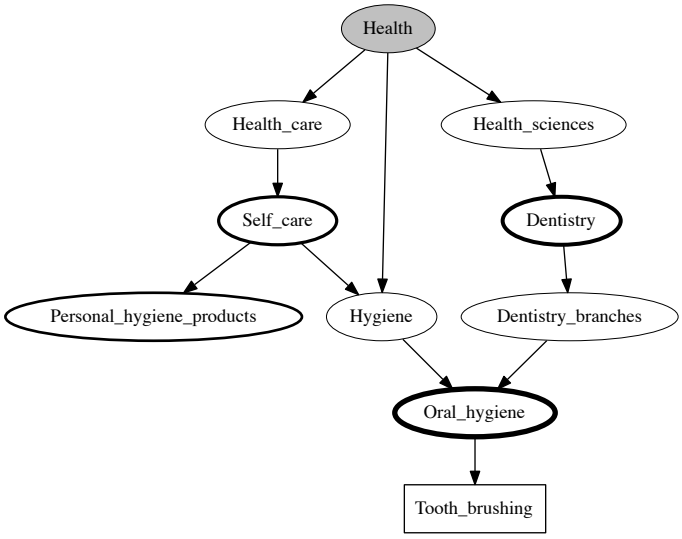


Figure 7.5: Relations between the top 4 Wikipedia categories assigned to the article *"Tooth brushing"*.

For example, if one of the top-k categories of a test document was "Scientific phenomena", this document was classified under the "Science" category, because "Scientific phenomena" is a descendant of the "Science" category in the Wikipedia's hierarchical category network. Similarly to the first experiment, we pre-processed the dataset by removing the punctuation, stopwords, numbers, and words appearing fewer that five times in the corpus. We need to note that any word found in D, which was not defined in the matrix λ was considered to be out-of-vocabulary and removed from D.

In this experiment, we did not train sOntoLDA on a training set and run it

on a test set but directly estimated the sOntoLDA parameters using the entire corpus and evaluated its performance utilizing the same metrics, described in the previous section. The results are shown in Figure 7.6 and the measurement values are presented in Table 7.5. The results indicate that our sOntoLDA topic model performs very well on various types of documents. An important difference can be seen for the *Precision@1* which is 62% for the Reuters corpus while it achieves 53% on Wikipedia collection. Similarly as shown in Figure 7.6, the *Mean Average Precision (MAP)* is 62% at *top-1*, which explains that the top categories are very relevant. Regarding *Coverage*, we can see that our method finds at least a relevant category (tag) for 96% of documents among the *top-5* categories, which is 77% for Wikipedia dataset. For most documents in this dataset $k = 1$ which explains why *Precision* and *Coverage* lines nearly overlap. This experiment demonstrates a superior coverage over the entire document collection and a much greater ability to identify broader categories (topics). The prior knowledge about the words probabilities in diverse categories encoded in the λ matrix leads to better document modeling and semantic tagging, which demonstrates the power of the prior knowledge.

7.5.4 Examples of Topics and Word Distributions

In this section, we present some examples of the topics from both datasets and their probability distributions over the vocabulary. Note that, as mentioned in previous sections, topics of the model are the same as Wikipedia categories. Thus, we essentially find the distributions of Wikipedia categories over the words by

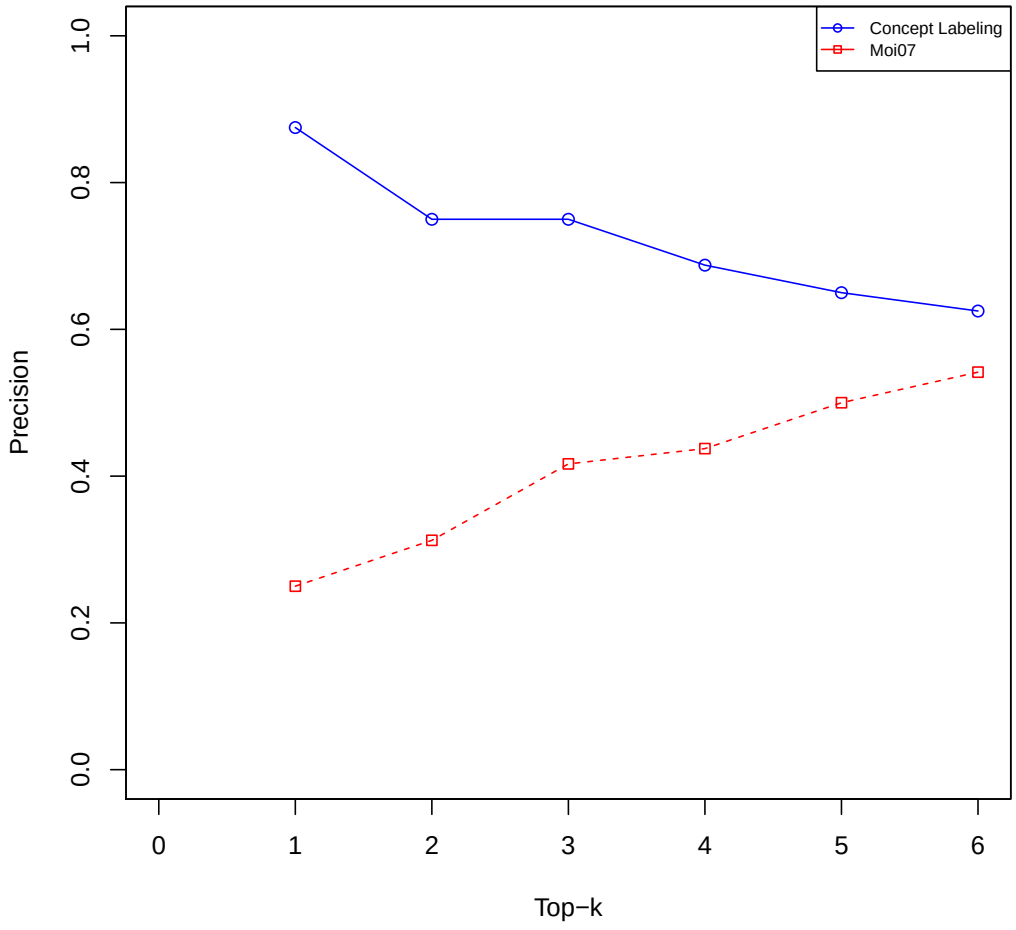


Figure 7.6: Precision, Coverage and MAP for Reuters Dataset.

Table 7.5: Precision, Coverage and MAP values of "Hierarchical match" for Reuters Dataset.

Top-k	**Precision**	**Coverage**	**MAP**
1	0.617	0.617	0.617
2	0.795	0.808	0.706
3	0.897	0.899	0.77
4	0.935	0.935	0.811
5	0.964	0.964	0.842
10	0.995	0.995	0.915
15	0.999	0.999	0.942
20	1	1	0.957
25	1	1	0.965
30	1	1	0.971

learning the topics of sOntoLDA.

Table 7.6 describes examples of five topics as learned by the sOntoLDA model from Wikipedia dataset. Each topic is extracted from a sample at the 500th iteration of the Gibbs sampler. The total number of topics in the model was equalized to the number of categories in the Wikipedia hierarchical ontology, $K = 1,353$. Each topic is represented by top 10 words most likely to be generated conditioned on the topic. The first row of the table shows the titles Wikipedia categories (topics).

Considering the title of each topic and topical words, we can see that our topic model has qualitatively produced coherent results. For each topic, we italicized and marked in red the incorrect topical words (although this is a subjective task and we do not expect everybody to accept it, but we relied on two human judges).

Table 7.6: An example of top-10 words for 5 categories (topics) in Wikipedia Dataset

Taxation	**Bankruptcy**	**Space Medicine**	**Pharmacology**	**Sports Business**
rate	security	solar	patients	games
pay	bankruptcy	lunar	treatment	teams
paid	secured	disturbances	effects	sport
taxes	creditors	astronauts	efficacy	club
dividend	payment	mmhg	pharmacists	promotion
levied	debts	spaceflight	therapeutic	clubs
taxpayer	bankrupt	weightless	compounding	championship
irs	priority	*garn*	pharmacological	fans
made	estate	skylab	antidepressants	season
years	creditor	astronautical	*due*	fan

Table 7.7 illustrates similar types of results for 5 selected topics from Reuters dataset, where again incorrect topical words are shown in italics and marked red. However, since the λ matrix is constructed based on the vocabulary of the Wikipedia category network and many words of the Reuters dataset vocabulary may not occur in λ and consequently be discarded, it is more likely (as shown in Table 7.7) that top words of topics include more general and incorrect words.

Table 7.7: An example of top-10 words for 5 categories (topics) in Reuters Dataset

Taxation	**Bankruptcy**	**Space Medicine**	**Pharmacology**	**Sports Business**
taxes	claims	risk	patients	million
asset	settlement	acute	patient	screens
corporations	property	*ros*	large	premier
stamp	pay	disturbances	number	fan
taxation	loan	crewmembers	*years*	fans
roads	equity	*including*	*longer*	*bing*
levies	judgment	made	make	giants
based	*petition*	early	made	attendance
make	transactions	years	cognizant	*years*
large	liabilities	*include*	*eventually*	*longer*

7.6 Classifying the Tagged Documents

We evaluated our method on a Wikipedia dataset as well as a collection of documents from Reuters news articles. In case of the Wikipedia dataset, one or more categories from the Wikipedia category network have been assigned to each article by Wikipedia editors. These categories can later be used to evaluate our proposed method and measure the performance straightforwardly. However, there are no such Wikipedia categories pre-assigned to articles in the Reuters dataset, which makes the performance analysis of our method more difficult. Hence, for further assessment of the proposed method, we have set up another experiment. For this experiment, we considered the same corpus as the one used in section 7.5.3. In this experiment, we first created a gold standard by classifying the corpus of documents into their predefined categories based solely on their content (we used three

standard text classification methods). Subsequently, we generated semantic tags for all the documents using our method. Now we treated documents as composed of tags only (i.e, a document was regarded as a bag of tags). We classified the tag-represented documents again and investigated how well semantic tags described the categories of the documents by comparing the results with the classification results based on the full content of the documents (i.e., the gold standard).

Tagging web resources and online documents can significantly benefit many other information access tasks: (a) tagging facilitates future retrieval of documents; and (b) it can be used to categorize, summarize and share documents in an effective way. One of primary applications of tagging is that it enables us to automatically classify web pages into semantic categories, which consequently enhances searching and browsing on the Web. In this section, we demonstrate how semantic tags can be used to benefit the automatic classification of Web documents. For evaluation, we created a gold standard by classifying the documents via various classification algorithms taking only the text of the documents into account. It allows us to draw quantitative conclusions about how semantic tags, assigned by our proposed method, are beneficial in Web document classification. In addition, this evaluation also shows that our method generates and assigns appropriate semantic tags to documents.

For this evaluation, we selected the same corpus that was used in section 7.5.3. We define the *web document classification* task as follows:

1. Create a gold standard to compare against by utilizing the original text (bag of words B_w) of the documents.

2. Given a collection of documents represented by their top-k semantic tags (bag of tags B_t), classify the documents into their predefined categories using different classification algorithms.

3. Compare the outputs produced by tag-represented classification to the gold standard results using *Precision, Recall* and *F-Measure* evaluation metrics.

Following this setup, we obtained the results according to our evaluation metrics that reveal two interesting observations. First, the results show that our proposed method generates and assigns appropriate semantic tags to documents. Second, the results also demonstrate that using a bag of tags instead of a bag of words not only gives us comparable categorization results, but also significantly reduces the training and testing time. It is because representing the documents by bag of tags substantially decreases the size of the vocabulary, which consequently lowers the classification time.

7.6.1 Creating a Gold Standard

In order to derive the gold standard, we use the text of documents. Each document consists of a bag of words from a word vocabulary W. Since the documents are divided into three main categories (labels)−*Business*, *Science* and *Health*−each document $d \in \mathcal{D}$ has l_i labels where $1 \leq l_i \leq 3$. For example, a document that talks about the business aspects of a technology, belongs to both categories "Business" and "Science". Thus, to derive the gold standard, we have to choose a multi-label classification approach. For multi-label classification, there is a large

body of prior work, which has been well-explained in the literature (e.g. see [94, 104, 103]). Most approaches have employed some variation of "binary problem transformation" technique to alter the multi-label classification problem to a *set* of binary-classification problems, each of which can then be solved using a proper binary classifier. We employed a method in which L independent binary classifiers are trained, one classifier for each label. We considered decision trees, naïve Bayes and Support Vector Machine (SVM) as the binary classifiers in our evaluation.

7.6.2 Tag-represented Document Classification

The sOntoLDA topic model generates top-k tags for documents of the corpus. We first construct a tag vocabulary V consists of all the words extracted from the set of tags T assigned to the entire collection. We then, represent each document of a corpus as a bag of top-k semantic tags. In other words, we model each document d in the collection $\mathcal{D}$ using a bag of tags B_t, i.e., $d = \{w_1, w_2, \ldots, w_V\}$, where V is the size of the tag vocabulary. We run the same three supervised classification algorithms, decision trees, naïve Bayes and SVM, on the tag-represented document collection.

7.6.3 Evaluation Metrics

We chose to compare the classification results of tag-represented documents with the outputs of the model that classifies the corpus considering only the text of the documents using the *Precision, Recall* and *F-Measure* evaluation metrics [72].

For binary classification problems, precision, recall and F-Measure are defined

as:

$$Prec = \frac{TP}{TP + FP} \tag{7.13}$$

$$Rec = \frac{TP}{TP + FN} \tag{7.14}$$

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{7.15}$$

where TP is the number of true positives, FP is the number of false positives, and FN is the number of false negatives. In a multi-label classification problem, let TP_i, FP_i and FN_i be the number of true positive, false positive and false negative for label i, respectively. The precision and recall are then according to [106] defined as:

$$Prec = \frac{\sum_i TP_i}{\sum_i TP_i + \sum_i FP_i} \tag{7.16}$$

$$Rec = \frac{\sum_i TP_i}{\sum_i TP_i + \sum_i FN_i} \tag{7.17}$$

7.6.4 Evaluation Results

Table 7.8 presents the results of the multi-label classification using 10-fold cross validation. As can be seen, SVM produced the best results for both the original document collection as well as tag-represented documents. Moreover, we notice

Table 7.8: Multi-label classification Precision, Recall and F-Measure values of different algorithms.

	Original Corpus			**Tag-represented Corpus**		
	Precision	**Recall**	**F-Measure**	**Precision**	**Recall**	**F-Measure**
Decision Tree	0.881	0.903	0.892	0.831	0.836	0.833
Naïve Bayes	0.886	**0.942**	0.913	0.870	**0.892**	0.881
SVM	**0.921**	0.931	**0.926**	**0.885**	0.884	**0.884**

that the performance of the algorithm on the tag-represented corpus is comparable to the model that only takes the text of documents into account, and the difference is less than 4% for all the evaluation metrics. More interestingly, because the size of the tag vocabulary V is order of magnitude smaller than the word vocabulary W, the time of training and testing is significantly lower and classification is much faster. Hence, many text processing tasks can benefit from the semantic tags assigned to the documents. Table 7.9 shows the performance measures of each binary classifier individually for both the original corpus as well as tag-represented documents.

7.7 Conclusions

In this paper, we presented a probabilistic topic model, sOntoLDA, that integrates prior knowledge from the DBpedia hierarchical category network with statistical topic modeling into a single framework. We employed our model for semantic annotation of Web pages and online documents with Wikipedia categories. Ex-

Table 7.9: Precision, Recall and F-Measure values of different algorithms.

Business						
	Original Corpus			**Tag-represented Corpus**		
	Precision	**Recall**	**F-Measure**	**Precision**	**Recall**	**F-Measure**
Decision Tree	0.809	0.807	0.808	0.748	0.748	0.748
Naïve Bayes	**0.858**	0.838	**0.840**	**0.819**	**0.806**	**0.807**
SVM	0.839	**0.839**	0.839	0.803	0.803	0.803
Health						
	Original Corpus			**Ta-represented Corpus**		
	Precision	**Recall**	**F-Measure**	**Precision**	**Recall**	**F-Measure**
Decision Tree	0.949	0.95	0.949	0.886	0.886	0.886
Naïve Bayes	0.953	0.953	0.952	0.931	0.931	0.931
SVM	**0.969**	**0.969**	**0.969**	**0.937**	**0.937**	**0.937**
Science						
	Original Corpus			**Tag-represented Corpus**		
	Precision	**Recall**	**F-Measure**	**Precision**	**Recall**	**F-Measure**
Decision Tree	0.868	0.869	0.868	0.792	0.793	0.793
Naïve Bayes	0.903	0.902	0.902	0.858	0.850	0.852
SVM	**0.938**	**0.938**	**0.938**	**0.863**	**0.863**	**0.863**

perimental results demonstrate the effectiveness and robustness of the proposed method when applied on various domains of text collections. We observed that utilizing the prior knowledge about the words probabilities (tf-idf weights) obtained from the Wikipedia's hierarchical ontology encoded in the λ matrix can be successfully used for semantic tagging of documents, which is an important step towards Semantic Web.

There are many interesting future extensions to this work. We did not take into account the *hierarchical structure* of the Wikipedia categories directly in the topic model. Thus, exploring richer topic models that consider the hierarchical relations between the categories in the models would be interesting future work. It also would be interesting to investigate the usage of this model for the text classification task. [2] introduced an ontology-based text classification, which, in contrast to the traditional supervised text classification methods, did not need a training set. Since the topic models are naturally unsupervised techniques, exploring the possibilities of developing topic models, where topics of interest are defined based on ontological concepts included in DBpedia, Freebase, and other ontologies, would be a promising direction for the future work. Another direction of research is to explore more generative topic models that incorporate hierarchical knowledge bases in the models for personalization and recommendation tasks [52].

Chapter 8

Conclusions and Future Work

In this dissertation, we proposed two primary mechanisms to exploit domain knowledge from ontologies for a variety of text and data mining tasks.

First, we introduced an ontology-based text classification method in which the ontology itself is the classifier. Thus, we do not need to train the classifier. It also enables us to dynamically define (or change) the topics of interest without retraining the classifier.

Second, we proposed *knowledge-based* topic models that combine probabilistic topic models with prior knowledge from the ontologies such as DBpedia or Linked Open Data (LOD). Knowledge-based topic models allow the users to guide and impact the learned topics and direct the models towards the topics that are best aligned with user modeling goals.

8.1 Summary of Contributions

The major contributions of this dissertation is as follows:

1. **Ontology-based text classification method for dynamically defined topics.** In chapter 5, we proposed an ontology-based text classification technique for classifying documents into dynamically defined set of topics of interest. This method, which only needs a domain ontology and a set of user-defined topics known as contexts in the ontology, measures the semantic similarity of the thematic graph created from a text document and the ontology sub-graphs resulting from the projection of the defined contexts. Hence, the domain ontology becomes the classifier and unlike traditional supervised categorization methods, does not require a set of training documents. More importantly, our proposed approach allows dynamically changing the classification topics without retraining of the classifier.

2. **Ontology-based topic model for automatic topic labeling.** Chapter 6 introduced a knowledge-based topic model, OntoLDA, which integrates the ontology concepts with the LDA model for the task of automatic topic labeling. Unlike previous works in this area, which usually represent topics via groups of words selected from topics, OntoLDA automatically generates topic labels by considering ontology concepts rather than words alone. We also proposed a topic labeling method, based on the semantics of the concepts in the discovered topics, as well as ontological relationships existing among the concepts in the ontology. We applied our OntoLDA model on two datasets

to demonstrate how our model can be used for automatic topic labeling as well as linking text documents to ontology concepts and categories.

3. **Inference algorithm using collapsed Gibbs sampling for OntoLDA model.** We developed an inference algorithm using collapsed Gibbs sampling for OntoLDA topic model. The inference algorithm can be extended to ontologies containing tens of thousands of concepts by utilizing the association between the words and concepts of the ontology.

4. **A knowledge-based topic model for semantic tagging.** In chapter 7, we proposed a probabilistic topic model, sOntoLDA, that incorporates DBpedia knowledge into the topic model for tagging Web pages and online documents with topics discovered in them. We use the DBpedias hierarchical category network as our background knowledge, which includes the categories organized into a hierarchical structure and a set of articles from Wikipedia. We assign categories (topics) from Wikipedia to text documents for which there are no predefined or known categories. We learn the probability distribution of each category over the words using the statistical topic models taking into account the prior knowledge from Wikipedia about the words and their associated probabilities in various categories. We evaluated the effectiveness of our approach in terms of automatically assigning semantic tags to documents by conducting extensive experiments on two different datasets. Additionally, we performed an experiment to show how our method can benefit document classification task.

5. **Inference algorithm using collapsed Gibbs sampling for sOntoLDA model.** We developed an efficient inference algorithm using collapsed Gibbs sampling for sOntoLDA topic model. The computational complexity of this algorithm is the same as the standard LDA model.

8.2 Future Work

1. **Combining hierarchical relations between ontological concepts with the topic models.** In our OntoLDA model, we did not encode the hierarchical relations between the ontology concepts directly in the topic model. One potential future work is to model concept relations explicitly into the topic model, which makes the automatically generated labels more consistent to the meaning of the topics. Another similar interesting future extension to the sOntoLDA model is to take into account the hierarchical structure of the Wikipedia categories directly in the topic model.

2. **Document classification using knowledge-based topic models.** Since the topic models are naturally unsupervised techniques, we believe that exploring the possibilities of developing topic models, where topics of interest are defined based on ontological concepts included in DBpedia, Freebase, and other ontologies, would be a promising direction for the future work.

3. **Ontology-based topic models for discovering coherent topics.** An interesting future project is to develop entity-based topic models to effectively integrate an ontology with an entity topic model to improve the coherence

of the discovered topics. There are prior works in this direction that primarily use word-level domain knowledge in the model to enhance the topic coherence and ignore the rich information carried by entities (e.g., persons, location, organizations, etc.) associated with the documents. We believe that entities occurring in a document together with the relationships among them can determine the document's topics. Thus, utilizing this plentiful information is of great interest and can potentially improve the topic modeling and topic coherence. Furthermore, leveraging the information in individual documents including entities mentioned in the document text and joining it with the graph structure of the ontology by regularizing the topic model based on the entity network would be a promising direction for the future work.

4. **Semantic context-aware recommendation.** There are several potentially useful directions in which knowledge-based topic models can be extended. One interesting extension to explore is to develop probabilistic topic models that incorporate user interests, item representation and context information in a single framework in the Context-Aware Recommendation Systems (CARS). In this setting, contextual information is represented as a subset of the items feature space which is acquired from the knowledge available in the Linked Open Data (LOD). In this Semantic Context-aware Recommendation Model (SCRM), each user profile is represented as a multinomial distribution over a set of latent topics, while topics are distributions over items and item features. This probabilistic model would allow us to

both *infer* the semantic context and *model* this context in a systematic way. For a given user's profile u and context c, we then can compute the recommendation score for each item v as $p(v|c,u)$, rank the items based on these scores and select the *top-n* recommendations for the user.

Bibliography

[1] Mehdi Allahyari and Krys Kochut. Automatic topic labeling using ontology-based topic models. In *14th International Conference on Machine Learning and Applications (ICMLA), 2015*. IEEE, 2015.

[2] Mehdi Allahyari, Krys J Kochut, and Maciej Janik. Ontology-based text classification into dynamically defined topics. In *IEEE International Conference on Semantic Computing (ICSC), 2014*, pages 273–278. IEEE, 2014.

[3] David Andrzejewski, Xiaojin Zhu, and Mark Craven. Incorporating domain knowledge into topic modeling via dirichlet forest priors. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 25–32. ACM, 2009.

[4] David Andrzejewski, Xiaojin Zhu, Mark Craven, and Benjamin Recht. A framework for incorporating general domain knowledge into latent dirichlet allocation using first-order logic. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 1171, 2011.

[5] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cy-

ganiak, and Zachary Ives. *Dbpedia: A nucleus for a web of open data*. Springer, 2007.

[6] S. Banerjee and T. Pedersen. The design, implementation, and use of the Ngram Statistic Package. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, pages 370–381, Mexico City, February 2003.

[7] Evgeniy Bart, Ian Porteous, Pietro Perona, and Max Welling. Unsupervised learning of visual taxonomies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[8] Fabiano M Belém, Eder F Martins, Jussara M Almeida, and Marcos A Gonçalves. Personalized and object-centered tag recommendation methods for web 2.0 applications. *Information Processing & Management*, 50(4):524–553, 2014.

[9] Tim Berners-Lee, James Hendler, Ora Lassila, et al. The semantic web. *Scientific american*, 284(5):28–37, 2001.

[10] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data-the story so far. *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, pages 205–227, 2009.

[11] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. Dbpedia-a crystalliza-

tion point for the web of data. *Web Semantics: science, services and agents on the world wide web*, 7(3):154–165, 2009.

[12] David Blei and John Lafferty. Correlated topic models. *Advances in neural information processing systems*, 18:147, 2006.

[13] David M Blei and Michael I Jordan. Modeling annotated data. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 127–134. ACM, 2003.

[14] David M Blei and John D Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM, 2006.

[15] David M Blei and Jon D McAuliffe. Supervised topic models. In *NIPS*, volume 7, pages 121–128, 2007.

[16] David M Blei, Thomas L Griffiths, Michael I Jordan, and Joshua B Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *NIPS*, volume 16, 2003.

[17] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[18] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, 2008.

[19] Christopher Boston, Hui Fang, Sandra Carberry, Hao Wu, and Xitong Liu. Wikimantic: Toward effective disambiguation and expansion of queries. *Data & Knowledge Engineering*, 90:22–37, 2014.

[20] Jordan L Boyd-Graber, David M Blei, and Xiaojin Zhu. A topic model for word sense disambiguation. In *EMNLP-CoNLL*, pages 1024–1033. Citeseer, 2007.

[21] Li Cai, Guangyou Zhou, Kang Liu, and Jun Zhao. Large-scale question classification in cqa by leveraging wikipedia semantic knowledge. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1321–1330. ACM, 2011.

[22] Bob Carpenter. Integrating out multinomial parameters in latent dirichlet allocation and naive bayes for collapsed gibbs sampling. Technical report, Technical report, LingPipe, 2010.

[23] Asli Celikyilmaz and Dilek Hakkani-Tür. Discovery of topically coherent sentences for extractive summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 491–499. Association for Computational Linguistics, 2011.

[24] Jonathan Chang and David M Blei. Relational topic models for document networks. In *International conference on artificial intelligence and statistics*, pages 81–88, 2009.

[25] Chaitanya Chemudugunta, America Holloway, Padhraic Smyth, and Mark Steyvers. Modeling documents by combining semantic concepts with unsupervised statistical learning. In *The Semantic Web-ISWC 2008*, pages 229–244. Springer, 2008.

[26] Chaitanya Chemudugunta, Padhraic Smyth, and Mark Steyvers. Combining concept hierarchies and statistical topic models. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 1469–1470. ACM, 2008.

[27] Jilin Chen, Jun Yan, Benyu Zhang, Qiang Yang, and Zheng Chen. Diverse topic phrase extraction through latent semantic analysis. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 834–838. IEEE, 2006.

[28] Zhiyuan Chen, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. Discovering coherent topics using general knowledge. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 209–218. ACM, 2013.

[29] Zhiyuan Chen, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. Exploiting domain knowledge in aspect extraction. In *EMNLP*, pages 1655–1667, 2013.

[30] Zhiyuan Chen, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. Leveraging multi-domain prior knowledge in topic

models. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 2071–2077. AAAI Press, 2013.

[31] Zhiyuan Chen, Arjun Mukherjee, and Bing Liu. Aspect extraction with automated prior knowledge learning. In *Proceedings of ACL*, pages 347–358, 2014.

[32] Hongbo Deng, Jiawei Han, Bo Zhao, Yintao Yu, and Cindy Xide Lin. Probabilistic topic models with biased propagation on heterogeneous information networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1271–1279. ACM, 2011.

[33] Stephen Dill, Nadav Eiron, David Gibson, Daniel Gruhl, R Guha, Anant Jhingran, Tapas Kanungo, Sridhar Rajagopalan, Andrew Tomkins, John A Tomlin, et al. Semtag and seeker: Bootstrapping the semantic web via automated semantic annotation. In *Proceedings of the 12th international conference on World Wide Web*, pages 178–186. ACM, 2003.

[34] Li Fei-Fei and Pietro Perona. A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 524–531. IEEE, 2005.

[35] Samah Fodeh, Bill Punch, and Pang-Ning Tan. On ontology-driven document clustering using core semantic features. *Knowledge and information systems*, 28(2):395–421, 2011.

[36] Evgeniy Gabrilovich and Shaul Markovitch. Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. In *AAAI*, volume 6, pages 1301–1306, 2006.

[37] Walter R Gilks, Sylvia Richardson, and David J Spiegelhalter. Introducing markov chain monte carlo. In *Markov chain Monte Carlo in practice*, pages 1–19. Springer, 1996.

[38] Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences of the United States of America*, 101 (Suppl 1):5228–5235, 2004.

[39] Thomas R Gruber. Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, 43 (5):907–928, 1995.

[40] Mostafa M Hassan, Fakhri Karray, and Mohamed S Kamel. Automatic document topic identification using wikipedia hierarchical ontology. In *Information Science, Signal Processing and their Applications (ISSPA), 2012 11th International Conference on*, pages 237–242. IEEE, 2012.

[41] Gregor Heinrich. Parameter estimation for text analysis. Technical report, Technical report, 2005.

[42] Swapnil Hingmire and Sutanu Chakraborti. Topic labeled text classification: a weakly supervised approach. In *Proceedings of the 37th international ACM*

SIGIR conference on Research & development in information retrieval, pages 385–394. ACM, 2014.

[43] Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, and Gerhard Weikum. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 3161–3165. AAAI Press, 2013.

[44] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

[45] Andreas Hotho, Alexander Maedche, and Steffen Staab. Ontology-based text document clustering. *KI*, 16(4):48–54, 2002.

[46] Xiaohua Hu, Xiaodan Zhang, Caimei Lu, Eun K Park, and Xiaohua Zhou. Exploiting wikipedia as external knowledge for document clustering. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 389–396. ACM, 2009.

[47] Yuening Hu, Jordan Boyd-Graber, Brianna Satinoff, and Alison Smith. Interactive topic modeling. *Machine Learning*, 95(3):423–469, 2014.

[48] Ioana Hulpus, Conor Hayes, Marcel Karnstedt, and Derek Greene. Unsupervised graph-based topic labelling using dbpedia. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 465–474. ACM, 2013.

[49] Jagadeesh Jagarlamudi, Hal Daumé III, and Raghavendra Udupa. Incorporating lexical priors into topic models. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 204–213. Association for Computational Linguistics, 2012.

[50] Maciej Janik and Krys Kochut. Training-less ontology-based text categorization. In *workshop on Exploiting Semantic Annotations in Information Retrieval (ESAIR)*, pages 3–17, 2008.

[51] Maciej Janik and Krys J Kochut. Wikipedia in action: Ontological knowledge in text categorization. In *Semantic Computing, 2008 IEEE International Conference on*, pages 268–275. IEEE, 2008.

[52] Pavan Kapanipathi, Prateek Jain, Chitra Venkataramani, and Amit Sheth. User interests identification on twitter using a hierarchical knowledge base. In *The Semantic Web: Trends and Challenges*, pages 99–113. Springer, 2014.

[53] Saurabh S Kataria, Krishnan S Kumar, Rajeev R Rastogi, Prithviraj Sen, and Srinivasan H Sengamedu. Entity disambiguation with hierarchical topic models. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1037–1045. ACM, 2011.

[54] Jon M Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

[55] Ralf Krestel and Peter Fankhauser. Personalized topic-based tag recommendation. *Neurocomputing*, 76(1):61–70, 2012.

[56] Ralf Krestel, Peter Fankhauser, and Wolfgang Nejdl. Latent dirichlet allocation for tag recommendation. In *Proceedings of the third ACM conference on Recommender systems*, pages 61–68. ACM, 2009.

[57] Michal Laclavik, Martin Šeleng, Marek Ciglan, and Ladislav Hluchỳ. Ontea: Platform for pattern based automated semantic annotation. *Computing and Informatics*, 28(4):555–579, 2012.

[58] Thomas K Landauer, Peter W Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284, 1998.

[59] Jey Han Lau, David Newman, Sarvnaz Karimi, and Timothy Baldwin. Best topic word selection for topic labelling. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 605–613. Association for Computational Linguistics, 2010.

[60] Jey Han Lau, Karl Grieser, David Newman, and Timothy Baldwin. Automatic labelling of topic models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1536–1545. Association for Computational Linguistics, 2011.

[61] Gregory F Lawler. *Introduction to stochastic processes*. CRC Press, 2006.

[62] Angeliki Lazaridou, Ivan Titov, and Caroline Sporleder. A bayesian model for joint unsupervised induction of sentiment, aspect and discourse representations. In *ACL (1)*, pages 1630–1639, 2013.

[63] Chenliang Li, Aixin Sun, and Anwitaman Datta. A generalized method for word sense disambiguation based on wikipedia. In *Advances in Information Retrieval*, pages 653–664. Springer, 2011.

[64] Chenliang Li, Aixin Sun, and Anwitaman Datta. Tsdw: Two-stage word sense disambiguation using wikipedia. *Journal of the American Society for Information Science and Technology*, 64(6):1203–1223, 2013.

[65] Jiwei Li, Claire Cardie, and Sujian Li. Topicspam: a topic-model based approach for spam detection. In *ACL (2)*, pages 217–221, 2013.

[66] Wei Li and Andrew McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *Proceedings of the 23rd international conference on Machine learning*, pages 577–584. ACM, 2006.

[67] Chenghua Lin and Yulan He. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 375–384. ACM, 2009.

[68] Erik Linstead, Paul Rigor, Sushil Bajracharya, Cristina Lopes, and Pierre Baldi. Mining eclipse developer contributions via author-topic models. In *Mining Software Repositories, 2007. ICSE Workshops MSR'07. Fourth International Workshop on*, pages 30–30. IEEE, 2007.

[69] Yan Liu, Alexandru Niculescu-Mizil, and Wojciech Gryc. Topic-link lda: joint models of topic and author community. In *proceedings of the 26th*

annual international conference on machine learning, pages 665–672. ACM, 2009.

[70] Qiming Luo, Enhong Chen, and Hui Xiong. A semantic term weighting scheme for text categorization. *Expert Systems with Applications*, 38(10): 12708–12716, 2011.

[71] Davide Magatti, Silvia Calegari, Davide Ciucci, and Fabio Stella. Automatic labeling of topics. In *Intelligent Systems Design and Applications, 2009. ISDA'09. Ninth International Conference on*, pages 1227–1232. IEEE, 2009.

[72] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[73] Xian-Ling Mao, Zhao-Yan Ming, Zheng-Jun Zha, Tat-Seng Chua, Hongfei Yan, and Xiaoming Li. Automatic labeling hierarchical topics. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 2383–2386. ACM, 2012.

[74] Qiaozhu Mei, Chao Liu, Hang Su, and ChengXiang Zhai. A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In *Proceedings of the 15th international conference on World Wide Web*, pages 533–542. ACM, 2006.

[75] Qiaozhu Mei, Xuehua Shen, and ChengXiang Zhai. Automatic labeling of multinomial topic models. In *Proceedings of the 13th ACM SIGKDD inter-*

national conference on Knowledge discovery and data mining, pages 490–499. ACM, 2007.

[76] Qiaozhu Mei, Deng Cai, Duo Zhang, and ChengXiang Zhai. Topic modeling with network regularization. In *Proceedings of the 17th international conference on World Wide Web*, pages 101–110. ACM, 2008.

[77] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into texts. Association for Computational Linguistics, 2004.

[78] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[79] David Mimno and Andrew McCallum. Expertise modeling for matching papers with reviewers. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 500–509. ACM, 2007.

[80] David Mimno and Andrew McCallum. Topic models conditioned on arbitrary features with dirichlet-multinomial regression. *arXiv preprint arXiv:1206.3278*, 2012.

[81] David Mimno, Wei Li, and Andrew McCallum. Mixtures of hierarchical topics with pachinko allocation. In *Proceedings of the 24th international conference on Machine learning*, pages 633–640. ACM, 2007.

[82] David Mimno, Hanna M Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. In

Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 262–272. Association for Computational Linguistics, 2011.

[83] Thomas Minka. Estimating a dirichlet distribution, 2000.

[84] Arjun Mukherjee and Bing Liu. Mining contentions from discussions and debates. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 841–849. ACM, 2012.

[85] Hilary Nesi. Bawe: an introduction to a new resource. *New trends in corpora and language learning*, pages 212–28, 2011.

[86] David Newman, Chaitanya Chemudugunta, and Padhraic Smyth. Statistical entity-topic models. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 680–686. ACM, 2006.

[87] David Newman, Sarvnaz Karimi, and Lawrence Cavedon. External evaluation of topic models. In *in Australasian Doc. Comp. Symp., 2009*. Citeseer, 2009.

[88] James Petterson, Wray Buntine, Shravan M Narayanamurthy, Tibério S Caetano, and Alex J Smola. Word features for latent dirichlet allocation. In *Advances in Neural Information Processing Systems*, pages 1921–1929, 2010.

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

corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 248–256. Association for Computational Linguistics, 2009.

[90] Petar Ristoski and Heiko Paulheim. Semantic web in data mining and knowledge discovery: A comprehensive survey. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2016.

[91] Christian P Robert and George Casella. *Monte Carlo statistical methods*, volume 319. Citeseer, 2004.

[92] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 487–494. AUAI Press, 2004.

[93] Michal Rosen-Zvi, Chaitanya Chemudugunta, Thomas Griffiths, Padhraic Smyth, and Mark Steyvers. Learning author-topic models from text corpora. *ACM Transactions on Information Systems (TOIS)*, 28(1):4, 2010.

[94] Timothy N Rubin, America Chambers, Padhraic Smyth, and Mark Steyvers. Statistical topic models for multi-label document classification. *Machine Learning*, 88(1-2):157–208, 2012.

[95] David Sánchez, David Isern, and Miquel Millan. Content annotation for the semantic web: an automatic web-based approach. *Knowledge and Information Systems*, 27(3):393–418, 2011.

[96] Peter Schönhofen. Identifying document topics using the wikipedia category network. *Web Intelligence and Agent Systems: An International Journal*, 7 (2):195–207, 2009.

[97] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.

[98] Bracha Shapira, Nir Ofek, and Victor Makarenkov. Exploiting wikipedia for information retrieval tasks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1137–1140. ACM, 2015.

[99] Masumi Shirakawa, Kotaro Nakayama, Takahiro Hara, and Shojiro Nishio. Concept vector extraction from wikipedia category network. In *Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication*, pages 71–79. ACM, 2009.

[100] Mark Steyvers, Padhraic Smyth, Michal Rosen-Zvi, and Thomas Griffiths. Probabilistic author-topic models for information discovery. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 306–315. ACM, 2004.

[101] Zareen Saba Syed, Tim Finin, and Anupam Joshi. Wikipedia as an ontology for describing documents. In *ICWSM*, 2008.

[102] Jie Tang, Mingcai Hong, Juanzi Li, and Bangyong Liang. Tree-structured

conditional random fields for semantic annotation. In *The Semantic Web-ISWC 2006*, pages 640–653. Springer, 2006.

[103] Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *Dept. of Informatics, Aristotle University of Thessaloniki, Greece*, 2006.

[104] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Mining multi-label data. In *Data mining and knowledge discovery handbook*, pages 667–685. Springer, 2009.

[105] Suppawong Tuarob, Line C Pouchard, and C Lee Giles. Automatic tag recommendation for metadata annotation using probabilistic topic modeling. In *Proceedings of the 13th ACM/IEEE-CS joint conference on Digital Libraries*, pages 239–248. ACM, 2013.

[106] Celine Vens, Jan Struyf, Leander Schietgat, Sašo Džeroski, and Hendrik Blockeel. Decision trees for hierarchical multi-label classification. *Machine Learning*, 73(2):185–214, 2008.

[107] Hanna M Wallach, David Minmo, and Andrew McCallum. Rethinking lda: Why priors matter. 2009.

[108] Chong Wang and David M Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 448–456. ACM, 2011.

[109] Chong Wang, David Blei, and Fei-Fei Li. Simultaneous image classification and annotation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1903–1910. IEEE, 2009.

[110] Chong Wang, David Blei, and David Heckerman. Continuous time dynamic topic models. *arXiv preprint arXiv:1206.3298*, 2012.

[111] Dingding Wang, Shenghuo Zhu, Tao Li, and Yihong Gong. Multi-document summarization using sentence-based topic models. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 297–300. Association for Computational Linguistics, 2009.

[112] Pu Wang, Jian Hu, Hua-Jun Zeng, and Zheng Chen. Using wikipedia knowledge to improve text classification. *Knowledge and Information Systems*, 19 (3):265–281, 2009.

[113] Xiaogang Wang and Eric Grimson. Spatial latent dirichlet allocation. In *Advances in neural information processing systems*, pages 1577–1584, 2008.

[114] Xuerui Wang and Andrew McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 424–433. ACM, 2006.

[115] Xuerui Wang, Andrew McCallum, and Xing Wei. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *Data*

Mining, 2007. ICDM 2007. Seventh IEEE International Conference on, pages 697–702. IEEE, 2007.

[116] Yang Wang, Payam Sabzmeydani, and Greg Mori. Semi-latent dirichlet allocation: A hierarchical model for human action recognition. In *Human Motion–Understanding, Modeling, Capture and Animation*, pages 240–254. Springer, 2007.

[117] Xing Wei and W Bruce Croft. Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185. ACM, 2006.

[118] I Witten and David Milne. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, AAAI Press, Chicago, USA*, pages 25–30, 2008.

[119] Ke Xu, Weiguo Yang, Guo-Ping Liu, and Hongbin Sun. Unsupervised satellite image classification using markov field topic model. *Geoscience and Remote Sensing Letters, IEEE*, 10(1):130–134, 2013.

[120] Yang Yang, Niran Chawla, Yizhou Sun, and Jiawei Hani. Predicting links in multi-relational and heterogeneous networks. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 755–764. IEEE, 2012.

[121] Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. Struc-

tured relation discovery using generative models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1456–1466. Association for Computational Linguistics, 2011.

[122] Liyang Yu. *A developer's guide to the semantic Web*. Springer, 2011.

[123] Erheng Zhong, Wei Fan, and Qiang Yang. User behavior learning and transfer in composite social networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 8(1):6, 2014.