Semantic Tagging Using Topic Models Exploiting Wikipedia Category Network

Mehdi Allahyari
Computer Science Department
University of Georgia, Athens, GA, USA
Email: mehdi@uga.edu

Krys Kochut
Computer Science Department
University of Georgia, Athens, GA, USA
Email: kochut@cs.uga.edu

***Abstract*—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.**

***Keywords*—Statistical learning, topic modeling, document topic tagging, DBpedia ontology**

I. Introduction

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. 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 [1] 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 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) [2] 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 I shows the top words of four topics learned from a collection of news articles; the topics have been labeled by a human "*Space Science*", "*Financial Market*", "*Politics*", and "*Sports*", respectively.

TABLE I. Examples of two topics with their labels.

Space Science	Financial Market	Politics	Sports
spance	stock	united	league
nasa	profile	states	city
station	research	washington	goal
launch	buzz	obama	team
earch	quote	president	win
mission	bank	iran	champions
satellite	banks	defence	points
flight	financial	military	club
rocket	european	nuclear	game
mars	fund	security	home

We consider the Wikipedia categories as the ***topics*** in the probabilistic model. 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 pre-assigned categories, which is used as a training set. A classifier is trained based on the training set and then is used to predict the categories of previously unseen documents. In the work presented here, we intend to 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 "knowledge", "semantic" and "metadata" have likely higher weights (see section IV-B) under the "Knowledge engineering" category and, similarly, the words "finance", "investment", and "corporate" are more related to the "Business" category.

We should point out that there exist several knowledge bases such as DBpedia [1] (constructed based on the content of Wikipedia [3]), YAGO [4], and Freebase [5] 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,

[6] uses linguistic techniques to address annotation of Web resources. [7], [8] utilize various natural language processing and information extraction techniques and [9] employs regular expression patterns for semantic tagging. 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 used within the document.

Some other related works include [10] where they use article titles and categories of Wikipedia to identify document topics. In their method, they first find all the related Wikipedia articles to a document by matching their titles with the words of the document. Then, they select categories assigned to these articles and rank them, and finally choose the categories with the highest weights as the topics of the document. [11] 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 [12], 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 [13], 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 [12], [13] 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 concept-word distribution that is later used to tag other documents, such as Web pages, news articles, and other online documents.

II. 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 [14] and text classification [15]. 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 content, using a variety of techniques, such as Natural Language Processing (NLP), information extraction, and probabilistic methods. For example, [7] uses a conditional random fields (CRF) approach for semantic annotation. [8] introduces a system called SemTag to perform semantic tagging utilizing NLP techniques. In more recent works, [6] uses linguistic patterns and learning methods to discover entities in text and associate them to classes of an ontology. In [9], authors employ regular expression patterns for semantic annotation of documents.

Wikipedia's category network has previously been used for document topic identification. In [10], the authors propose a method that uses Wikipedia article titles as well as the category network to identify topics of documents. [11] 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, [12] proposes a concept-topic model (CTM) 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 [13], the authors describe a hierarchical concept-topic model (HCTM) that extends the CTM in [12] to integrate the hierarchical relations between the concepts. Our work presented in this paper is along the same line of research as [12] and [13] where we combine the prior knowledge of ontology concepts with probabilistic topic models. However, our approach 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.

In the next section, we present a brief overview of Latent Dirichlet Allocation (LDA), the state-of-art probabilistic topic modeling technique, and in the following section describe our proposed sOntoLDA model and compare it with the standard LDA.

III. Latent Dirichlet Allocation (LDA)

The latent Dirichlet allocation (LDA) [2] is a generative probabilistic model which has been extensively used for dis-

covering 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 1(a) 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:

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

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.

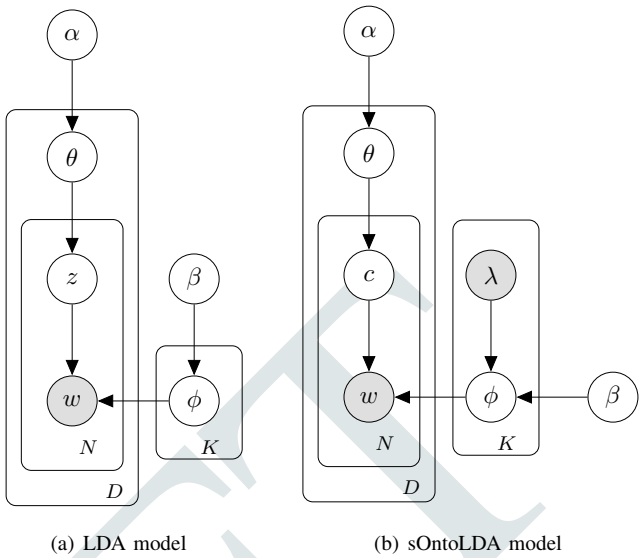
IV. Semantic Tagging using Ontology-based Topic Models

In this section, we formally introduce our model and describe how we 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.

A. 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 category network



(a) LDA model (b) sOntoLDA model

Fig. 1. Graphical representation of different models

in the form a λ matrix, as illustrated in Figure 1(b). The generative process is given as follows:

1) For each Wikipedia category $c \in \{1, 2, \ldots, K\}$,
   (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 $c_i, w_i \sim \text{Mult}(\phi_{c_i})$

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.

B. 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 hierarchical manner. It contains over $940,000$ categories where relationships between the categories are represented using **`SKOS`**[1] vocabulary in the DBpedia ontology. The relation between a Wikipedia article and a category is defined by the **`subject`** property of the Dublin Core[2] 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.

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

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

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:

$$\delta_w^{(c)} = tf_w \times log(\frac{|C|}{cf_w}) \tag{2}$$

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:

$$\lambda_c = \left[\delta_{w_1}^{(c)}, \delta_{w_2}^{(c)}, \ldots, \delta_{w_{|V|}}^{(c)}\right]^T \tag{3}$$

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. 4, 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 with 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.

C. Inference 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 [2] and Gibbs sampling [16]. In our sOntoLDA topic model presented in this paper, we use the collapsed Gibbs sampling procedure. Collapsed Gibbs sampling [16] 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 $\boldsymbol{c}$ conditioned on the observed words $\boldsymbol{w}$ and hyperparameters α and β.

We derive the posterior inference as follows:

$$\begin{aligned} P(\boldsymbol{c}|\boldsymbol{w}, \lambda, \alpha, \beta) &= \frac{P(\boldsymbol{c}, \boldsymbol{w}|\lambda, \alpha, \beta)}{P(\boldsymbol{w}|\lambda, \alpha, \beta)} \\ &\propto P(\boldsymbol{c}, \boldsymbol{w}|\lambda, \alpha, \beta) \propto P(\boldsymbol{c})P(\boldsymbol{w}|\lambda, \boldsymbol{c}) \\ P(c_i = c|w_i = w, \boldsymbol{c}_{-i}, \boldsymbol{w}_{-i}, \lambda, \alpha, \beta) &\propto \\ \frac{n_{c,-i}^{(d)} + \alpha_c}{\sum_{c'}(n_{c',-i}^{(d)} + \alpha_{c'})} &\times \frac{n_{w,-i}^{(c)} + \lambda_{wc}\beta_w}{\sum_{w'}(n_{w',-i}^{(c)} + \lambda_{w'c}\beta_{w'})} \end{aligned} \tag{4}$$

where $n_w^{(c)}$ is the number of times the word w is assigned to the concept c. $n_c^{(d)}$ 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 [17] 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}:

$$\theta_{cd} = \frac{n_c^{(d)} + \alpha_c}{\sum_{c'}(n_{c'}^{(d)} + \alpha_{c'})} \quad \phi_{wc} = \frac{n_w^{(c)} + \lambda_{wc}\beta_w}{\sum_{w'}(n_{w'}^{(c)} + \lambda_{w'c}\beta_{w'})} \tag{5}$$

V. 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 $5,000,000$ 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

TABLE II. 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

words. The final topic graph included $K = 1,353$ 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.

A. Tagging Wikipedia Articles

For this experiment, we first extracted the Wikipedia categories from the three main categories, including BUSINESS, APPLIED SCIENCES, and HEALTH. We 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 was $|D_{train}| = 3,142$ 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. 5. 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 [18]. We also utilized the *Coverage* metric described in [19]. *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:

$$Precision@k = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{CI@k}{k} \tag{6}$$

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 II, and Figure 2 illustrates the corresponding plot of the evaluation

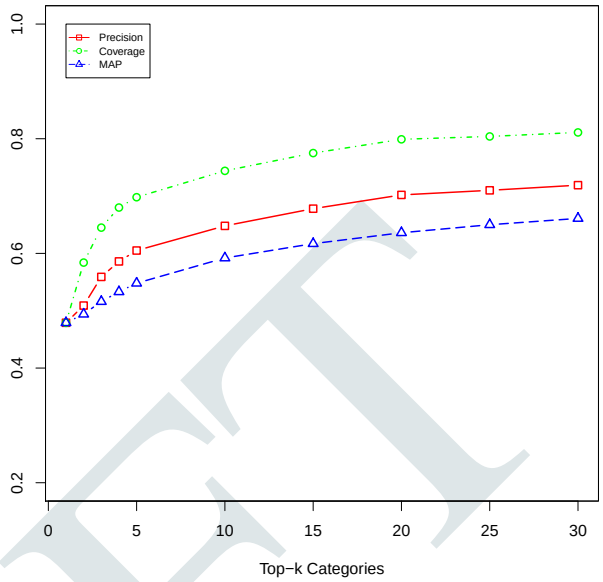


Fig. 2. Precision, Coverage and MAP of EXACT MATCH for Wikipedia Dataset.

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

$$MAP(Q) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} Precision(R_{jk}) \tag{7}$$

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 2, 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:

$$Coverage@k = \frac{\text{\#documents with at least on Hit at rank} \leq k}{\text{\#documents}} \tag{8}$$

As illustrated in Figure 2, 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 $k = 30$.

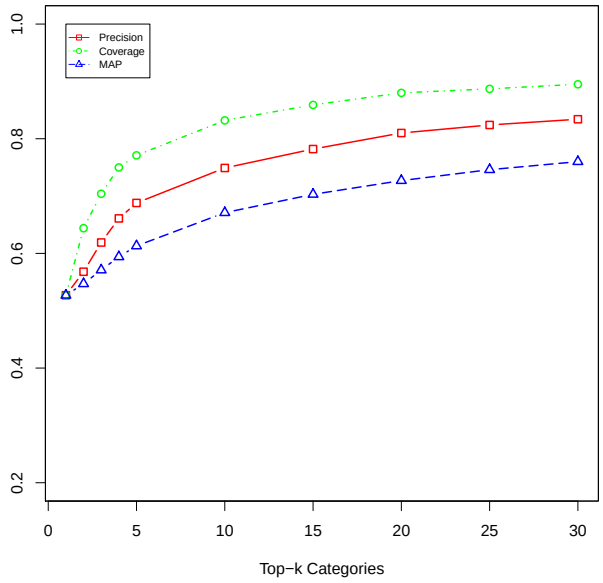


Fig. 3. Precision, Coverage and MAP of HIERARCHICAL MATCH for Wikipedia Dataset.

TABLE III. 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

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 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 an 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 ancestors 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.

TABLE IV. 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

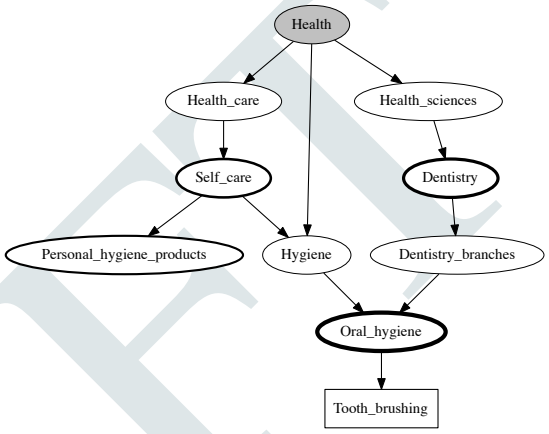


Fig. 4. Relations between the top 4 Wikipedia categories assigned to the article *"Tooth brushing"*.

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 III. Figure 3 shows the results when *Hierarchical match* is taken into consideration.

B. Example of Tagging a Wikipedia Article

As an example, Table IV 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 4 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.

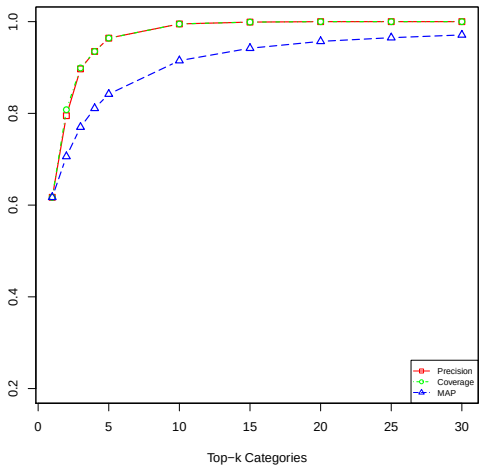


Fig. 5. Precision, Coverage and MAP for Reuters Dataset.

TABLE V. 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

C. 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 mapped 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 sub-graph of that main category were considered as the correct topics of the document. 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.

TABLE VI. AN EXAMPLE OF TOP-10 WORDS FOR 5 CATEGORIES (TOPICS) IN WIKIPEDIA DATASET

Finance	Biophysics	Nutrition	Psychoanalysis	Pollution
stock	dna	disease	relationship	air
investors	biological	acid	behaviour	carbon
traded	biology	nutrient	emotional	bacteria
interest	chemical	intake	attachment	pollutants
stocks	sample	vitamins	secure	epa
discounted	membrane	diets	desire	paints
payments	lipid	fructose	reparenting	contaminant
liquidity	biophysics	*bayer*	strange	plastic
lender	strands	*long*	caregiver	carcinogens
made	hemoglobin	years	behaviours	radionuclides

TABLE VII. AN EXAMPLE OF TOP-10 WORDS FOR 5 CATEGORIES (TOPICS) IN REUTERS DATASET

Finance	Biophysics	Nutrition	Psychoanalysis	Pollution
bank	dna	calories	children	fish
stock	proteins	drinks	child	carbon
assets	acids	diets	anxious	organic
bonds	particles	*bayer*	esteem	epa
investment	major	ldl	longer	oceans
banking	*end*	metabolic	*make*	concentrations
liquidity	*bmis*	alcoholic	*due*	estrogen
credit	*make*	nutritional	long	*called*
traders	number	vitamins	*including*	*due*
default	*called*	*years*	*asked*	*making*

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, mentioned in the previous section. The results are shown in Figure 5 and the measurement values are presented in Table V. 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 5, 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.

D. 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 learning the topics of sOntoLDA.

Table VI 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 VII 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 the λ and consequently be discarded, it is more likely (as shown in Table VII) that top words of topics include more general and incorrect words.

VI. 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. Experimental 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 an interesting future work. It also would be interesting to investigate the usage of this model for the text classification task. [20] 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 [21].

References

[1] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, "Dbpedia-a crystallization point for the web of data," *Web Semantics: science, services and agents on the world wide web*, vol. 7, no. 3, pp. 154–165, 2009.

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

[3] Z. S. Syed, T. Finin, and A. Joshi, "Wikipedia as an ontology for describing documents." in *ICWSM*, 2008.

[4] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum, "Yago2: A spatially and temporally enhanced knowledge base from wikipedia," in *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*. AAAI Press, 2013, pp. 3161–3165.

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

[6] D. Sánchez, D. Isern, and M. Millan, "Content annotation for the semantic web: an automatic web-based approach," *Knowledge and Information Systems*, vol. 27, no. 3, pp. 393–418, 2011.

[7] J. Tang, M. Hong, J. Li, and B. Liang, "Tree-structured conditional random fields for semantic annotation," in *The Semantic Web-ISWC 2006*. Springer, 2006, pp. 640–653.

[8] S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, A. Tomkins, J. 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*. ACM, 2003, pp. 178–186.

[9] M. Laclavik, M. Šeleng, M. Ciglan, and L. Hluchỳ, "Ontea: Platform for pattern based automated semantic annotation," *Computing and Informatics*, vol. 28, no. 4, pp. 555–579, 2012.

[10] P. Schonhofen, "Identifying document topics using the wikipedia category network," in *Web Intelligence, 2006. WI 2006. IEEE/WIC/ACM International Conference on*. IEEE, 2006, pp. 456–462.

[11] M. M. Hassan, F. Karray, and M. S. Kamel, "Automatic document topic identification using wikipedia hierarchical ontology," in *Information Science, Signal Processing and their Applications (ISSPA), 2012 11th International Conference on*. IEEE, 2012, pp. 237–242.

[12] C. Chemudugunta, A. Holloway, P. Smyth, and M. Steyvers, "Modeling documents by combining semantic concepts with unsupervised statistical learning," in *The Semantic Web-ISWC 2008*. Springer, 2008, pp. 229–244.

[13] C. Chemudugunta, P. Smyth, and M. Steyvers, "Combining concept hierarchies and statistical topic models," in *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 2008, pp. 1469–1470.

[14] B. Shapira, N. Ofek, and V. Makarenkov, "Exploiting wikipedia for information retrieval tasks," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2015, pp. 1137–1140.

[15] P. Wang, J. Hu, H.-J. Zeng, and Z. Chen, "Using wikipedia knowledge to improve text classification," *Knowledge and Information Systems*, vol. 19, no. 3, pp. 265–281, 2009.

[16] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proceedings of the National academy of Sciences of the United States of America*, vol. 101, no. Suppl 1, pp. 5228–5235, 2004.

[17] T. Minka, "Estimating a dirichlet distribution," 2000.

[18] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*. Cambridge university press Cambridge, 2008, vol. 1.

[19] I. Hulpus, C. Hayes, M. Karnstedt, and D. Greene, "Unsupervised graph-based topic labelling using dbpedia," in *Proceedings of the sixth ACM international conference on Web search and data mining*. ACM, 2013, pp. 465–474.

[20] M. Allahyari, K. J. Kochut, and M. Janik, "Ontology-based text classification into dynamically defined topics," in *Semantic Computing (ICSC), 2014 IEEE International Conference on*. IEEE, 2014, pp. 273–278.

[21] P. Kapanipathi, P. Jain, C. Venkataramani, and A. Sheth, "User interests identification on twitter using a hierarchical knowledge base," in *The Semantic Web: Trends and Challenges*. Springer, 2014, pp. 99–113.