Department of Computer Science
George Mason University
Technical Report Series

4400 University Drive MS#4A5
Fairfax, VA 22030-4444 USA
http://cs.gmu.edu/
703-993-1530

Towards a Universal Text Classifier: Transfer Learning using Encyclopedic Knowledge

Pu Wang pwang7@gmu.edu
Carlotta Domeniconi carlotta@cs.gmu.edu
Jian Hu jianh@microsoft.com

Technical Report GMU-CS-TR-2009-6

Abstract

Document classification is a key task for many text mining applications. However, traditional text classification requires labeled data to construct reliable and accurate classifiers. Unfortunately, labeled data are seldom available, and often too expensive to obtain. In this work, we propose a *universal text classifier*, which does not require any labeled training document. Our approach simulates the capability of people to classify documents based on background knowledge. As such, we build a classifier that can effectively group documents based on their content, under the guidance of few words, which we call *discriminant words*, describing the classes of interest. Background knowledge is modeled using encyclopedic knowledge, namely Wikipedia. Wikipedia's articles related to the specific problem domain at hand are selected, and used during the learning process for predicting labels of test documents. The universal text classifier can also be used to perform document retrieval, in which the pool of test documents may or may not be relevant to the topics of interest for the user. In our experiments with real data we test the feasibility of our approach for both the classification and retrieval tasks. The results demonstrate the advantage of incorporating background knowledge through Wikipedia, and the effectiveness of modeling such knowledge via probabilistic topic modeling. The accuracy achieved by the universal text classifier is comparable to that of a supervised learning technique for transfer learning.

1 Introduction

Text classification is an essential task to enable the organization and usage of very large numbers of documents; and the need for it has become particularly acute with the exponential growth of the World Wide Web. However, traditional text classification requires labeled data to construct reliable and accurate classifiers. The need for labeled data greatly limits the applicability of classification approaches, since labeled data are seldom available, and often too expensive to obtain, especially in evolving scenarios. Recently, the paradigm of transfer learning has been introduced to enable effective learning strategies when labeled auxiliary data exist for a different but related domain to the target task. Although transfer learning can leverage training data which are drawn from a different distribution than testing data, it still suffers from the need of labeled data.

It is interesting to observe that when people classify documents, they seldom need training data. This is because people have common sense and background knowledge. Given few words or phrases characterizing the categories of interest, people are capable of skimming through a document, grasp its content through the appearance of some of the given words, or semantically related ones, and thus determine the class the document belongs to. If the given words or phrases characterize well the semantics of the categories, the manual classification can be done effectively.

Thus, people classify documents not only based on the specific words mentioned therein, but also by leveraging their background knowledge on the subject. In designing our approach, we would like to simulate the capability of people to classify documents based on background

knowledge. As such, our aim is to build a classifier that can effectively group documents based on their content, under the guidance of few words, which we call *discriminant words*, describing the classes of interest [2]. Typically, people make use of a limited number of words or phrases; likewise, our method operates under the same conditions, and only few words per category are used. For example, if a category of interest is "recreation", the words characterizing it could be "recreation", "motorcycles", "sport", and "hockey".

Background knowledge is modeled using encyclopedic knowledge, namely Wikipedia. Wikipedia's articles related to the specific problem domain at hand are selected, and used during the learning process for predicting labels of test documents. In a manual classification process, if the background knowledge of people grouping documents is not sufficient to achieve accurate results, people may resort to experts. Likewise, if Wikipedia cannot provide enough background information, users may provide auxiliary documents. Auxiliary documents may contain useful expertise to aid the classification process. For instance, suppose the test documents to be classified and submitted by the user concern a technical topic, such as "transfer learning", which may be far beyond Wikipedia's domain. The user could then provide auxiliary documents about "machine learning" to overcome the lack of background coverage. Our approach does not require labels for the auxiliary documents. We call our method *Universal Text Classifier*, or UTC, to emphasize the fact that it does not require any labeled training data.

The universal text classifier we propose can also be used to perform document retrieval, in which the pool of test documents may or may not be relevant to the topics of interest for the user. Given a user-defined query, the UTC is capable of ranking each test document according to its relevance to the topic specified by the query. As for classification, an enriched topic model representation is derived by means of background knowledge, defined in our case through Wikipedia.

The resulting universal text classifier has three important characteristics: (1) through the use of background knowledge and probabilistic topic modeling, it leverages a representation which is content-based and not merely a "bag-of-words"; (2) it does not require any labeled training data; and (3) it can handle the "open set problem", i.e., it can classify test sets which contain classes of documents not relevant for the problem at hand.

The rest of the paper is organized as follows. Section 2 discusses related work, and Section 3 provides the background on probabilistic topic modeling. Section 4 describes the universal text classifier in details. Section 5 introduces the cross-domain classification approaches used in our experiments for comparison. Section 6 describes the experimental settings and discusses the results for both classification and document retrieval. Finally, Section 7 discusses conclusions and future work.

2 Related Work

We discuss related work on transfer learning, text classification using Wikipedia, and topic models for classification and information retrieval.

2.1 Transfer Learning

Raina et al. [18] built a term covariance matrix using the auxiliary problem, to measure the co-occurrence between terms. They then applied the term covariance to the target learning task. For example, if the covariance between terms "moon" and "rocket" is high, and "moon" usually appears in documents of a certain category, it is inferred that "rocket" also supports the same category, even without observing this directly in the training data. The authors call their method Informative Priors.

In [17], Raina et al. proposed self-taught learning, which uses unlabeled data to aid the target task. Unlabeled data may be drawn from a different distribution than the labeled training data. High level features are extracted from unlabeled data (images). The resulting representation is then used with labeled data for classification. In contrast to this approach, our universal text classifier does not require labeled data.

Do et al. [7] modeled the text classification problem with a linear function, which takes the document vector representation as input, and provides in output the predicted label. Under this setting, different text classifiers differ only on the parameters of the linear function. A meta-learning method is introduced to learn how to tune the parameters. The technique uses data from a variety of related classification tasks to obtain a good classifier (a good parameter function) for new tasks, replacing hours of hand-tweaking.

Dai et al. [5] modified the Naive Bayes classifier to handle a cross-domain classification task. The technique first estimates the model based on the distribution of the training data. Then, an EM algorithm is designed under the distribution of the test data. KL-divergence measures are used to compute the distribution distance between training and test data. An empirical fitting function based on KL-divergence is used to estimate the trade-off parameters of the EM algorithm.

In [3], Dai et al. altered the Boosting algorithm to address cross-domain classification problems. Their basic idea is to select useful instances from auxiliary data with a different distribution, and use them as additional training data for predicting the labels of test data. However,

in order to identify the most helpful additional training instances, the approach relies on the existence of some labeled testing data, which in practice may not be available.

The authors in [4] use co-clustering [6] to perform cross-domain text classification (CoCC algorithm). The key idea of this approach is leveraging the common words between the documents to be classified and the auxiliary ones to bridge the gap between the two domains. In [20], we extended the idea underlying the CoCC algorithm by making the latent semantic relationship between the two domains explicit with the use of Wikipedia. Since we compare the UTC with this approach in our experiments, we further discuss the CoCC algorithm in Section 5.

2.2 Text Classification using Wikipedia

In [9, 8], Gabrilovich et al. proposed a method to integrate text classification with Wikipedia. They first built an auxiliary text classifier that can match documents with the most relevant articles of Wikipedia. Then, the relevant Wikipedia articles' titles retrieved by the former step are treated as new features to enrich document representation. They perform feature generation using a multi-resolution approach: features are generated for each document at the level of individual words, sentences, paragraphs, and finally the entire document. This method, however, only leverages similarity between text fragments and Wikipedia articles, ignoring the abundant structural information within Wikipedia, e.g. internal links. The processing effort of this method is very high, since each document needs to be scanned many times. Furthermore, the feature generation procedure inevitably brings a lot of noise, because a specific text fragment contained in an article may not be relevant for its discrimination.

In [19], we constructed an informative thesaurus from Wikipedia, which explicitly derives synonymy, polysemy, hyponymy, and associative relations between concepts. We leveraged the thesaurus derived from Wikipedia to embed semantic information in documents' representation, and therefore achieved improved classification accuracy based on documents' content. However, this method is time-consuming, due to the size of Wikipedia, and mining concept relations is an error-prone process. The UTC approach avoids this step by applying probabilistic topic modeling directly on Wikipedia's articles.

Recently, Chang et al. [2] proposed a data-less classification approach to classify documents without training data. Their method is based on explicit semantic analysis (ESA) [10], and finds Wikipedia concepts related to given terms descriptive of the categories of interest. Such prior knowledge is then used to train a Naive Bayes classifier. Our work is related to this approach. While Chang's method, though, computes the degree of relatedness between Wikipedia's concepts, our method directly uses Wikipedia articles to learn topic models, which can automatically handle synonyms and perform disambiguation.

2.3 Topic Models

Blei et al. [1] first introduced Latent Dirichlet Allocation (LDA), which is a probabilistic generative model that can be used to estimate multinomial observations in an unsupervised fashion. LDA has been applied for dimensionality reduction in text categorization [1], and for ad hoc information retrieval [21]. In [21], Wei et al. proposed an hybrid approach that combines LDA with clustering for ad-hoc retrieval. LDA is assigned a small weight (the authors claim that LDA can hurt the performance otherwise); thus, its contribution to retrieval is very limited. On the other hand, our method shows that LDA is capable of improving the results for information retrieval.

3 Preliminary

3.1 Latent Dirichlet Allocation

LDA is a generative graphical model. It can be used to model and discover underlying topic structures in any kind of discrete data, of which text is a typical example. Its process can be interpreted as follows. A document $\vec{w}_m = \{w_{m,n}\}_{n=1}^{N_m}$ is generated by first selecting a distribution over topics $\vec{\theta}_m$ from a Dirichlet distribution $Dir(\vec{\alpha})$, which determines the topic assignment for words in that document. Then, the topic assignment for each word placeholder $[m,n]$ is performed by sampling a particular topic $z_{m,n}$ from a multinomial distribution $Mult(\vec{\theta}_m)$. Finally, a particular word $w_{m,n}$ is generated for the word placeholder $[m,n]$ by sampling from a multinomial distribution $Mult(\vec{\phi}_{z_{m,n}})$.

The joint distribution of all known and hidden variables, given the Dirichlet parameters, can be written as follows:

$$p(\vec{w}_m, \vec{z}_m, \vec{\theta}_m, \Phi | \vec{\alpha}, \vec{\beta}) = p(\Phi|\vec{\beta}) \prod_{n=1}^{N_m} p(w_{m,n}|\vec{\phi}_{z_{m,n}}) p(z_{m,n}|\vec{\theta}_m) p(\vec{\theta}_m|\vec{\alpha})$$

The likelihood of a document $\vec{w}_m$ is obtained by inte-

grating over $\vec{\theta}_m$ and Φ, and summing over $\vec{z}_m$ as follows:

$$p(\vec{w}_m|\vec{\alpha},\vec{\beta}) = \int\int p(\vec{\theta}_m|\vec{\alpha})p(\Phi|\vec{\beta})\cdot\prod_{n=1}^{N_m} p(w_{m,n}|\vec{\theta}_m,\Phi)d\Phi d\vec{\theta}_m$$

Finally, the likelihood of the whole data collection $\mathcal{W} = \{\vec{w}_m\}_{m=1}^{M}$ is given by the product of the likelihoods of all documents:

$$p(\mathcal{W}|\vec{\alpha},\vec{\beta}) = \prod_{m=1}^{M} p(\vec{w}_m|\vec{\alpha},\vec{\beta}) \quad (1)$$

The estimation of LDA parameters by maximization of the likelihood of the whole data collection in Equation (1) is intractable. The solution to this problem is to use approximate estimation methods such as Variational Methods [1], Expectation-propagation [14], or Gibbs sampling [11]. Gibbs sampling is a special case of Markov-chain Monte Carlo (MCMC), and often yields relatively simple algorithms for approximate inference in high-dimensional models such as LDA.

The use of Gibbs sampling for estimating LDA parameters was first introduced in [11], and a comprehensive description of this method can be found in [12]. Here, we just provide the fundamental steps. Let $\vec{w}$ and $\vec{z}$ be the vectors of all words and topics of the whole data collection, respectively. The topic assignment for a particular word depends on the current topic assignments of all the other word positions. More specifically, the topic assignment of a particular word t is sampled from the following multinomial distribution:

$$p(z_i = k|\vec{z}_{\neg i},\vec{w}) = \frac{n^{(t)}_{k,\neg i}+\beta_t}{[\sum_{v=1}^{V} n^{(v)}_k+\beta_v]-1}\,\frac{n^{(k)}_{m,\neg i}+\alpha_k}{[\sum_{j=1}^{K} n^{(j)}_m+\alpha_j]-1} \quad (2)$$

where $n^{(t)}_{k,\neg i}$ is the number of times word t is assigned to topic k, except the current assignment; $\sum_{v=1}^{V} n^{(v)}_k - 1$ is the number of words assigned to topic k, except the current assignment; $n^{(k)}_{m,\neg i}$ is the number of words in document m assigned to topic k, except the current assignment; and $\sum_{j=1}^{K} n^{(j)}_m - 1$ is the total number of words in document m, except the current word t. Usually, the Dirichlet parameters $\vec{\alpha}$ and $\vec{\beta}$ are symmetric, that is, all α_ks ($k = 1,\ldots,K$) are the same, and similarly for all β_vs ($v = 1,\ldots,V$).

At completion of the Gibbs sampling procedure, the topic-document distribution $\theta_{m,k}$ and the topic-word distribution $\phi_{k,t}$ are computed as shown in Equations (4) and (3), respectively.

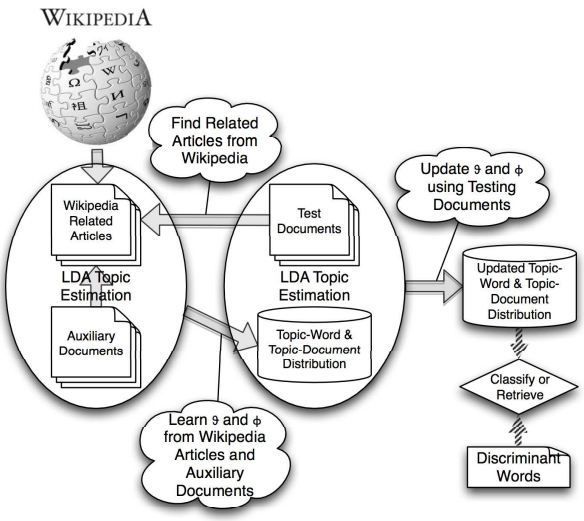

Figure 1: UTC: The overall procedure

4 Universal Text Classifier

In this Section, we introduce the proposed Universal Text Classifier (UTC). Figure 1 depicts the overall approach. In the following, we describe each step in details.

4.1 Discriminant Words for Categories

We would like the UTC to simulate the capability of people to classify documents based on background knowledge. Given few words or phrases characterizing the categories of interest, people are capable of skimming through a document, grasp its content through the appearance of some of the given words, or semantically related ones, and thus determine the class the document belongs to. If the given words or phrases characterize well the semantics of the categories, the manual classification can be done effectively. Similarly, our aim is to build a classifier that can effectively group documents based on their content, under the guidance of few words (called *discriminant words*) describing the classes of interest [2]. Typically, people make use of a limited number of words or phrases; thus, UTC operates under the same conditions, and only few words per category are used. Background knowledge is modeled using encyclopedic knowledge, namely Wikipedia.

How do we identify meaningful words or phrases for the categories of interest? Given an application domain, it is relatively easy for the user to provide a short list of keywords describing the topics he/she is interested in. Thus, we can assume that such list of discriminant words is given as input to the UTC. Table 1 gives an example of how we can generate discriminant words to character-

Table 1: Labels and corresponding Discriminant Words for 20 Newsgroups data

Label	*Discriminant Words*
rec	recreation, sport, baseball, hockey
talk	politics, mideast, religion, misc
sci	science, cryptography, crypt, electronics

ize the categories for the 20 Newsgroups data set (used in our experiments). For instance, the words *recreation*, *sport*, *baseball*, and *hockey* describe the category (or label) *rec*. They are derived from the 20 Newsgroups hierarchies *rec.sports.baseball* and *rec.sport.hockey*. The discriminant words for the label *sci* (science) are generated from *sci.crypt* and *sci.electronics*, where *crypt* is an abbreviation for cryptography. Since the meaning of "crypt" might be obscure to many, and not commonly used in science articles, the term cryptography is also used to describe the science class.

4.2 Related Wikipedia Articles

As mentioned above, we leverage Wikipedia to provide background knowledge to the UTC. In a manual classification process, if the background knowledge of people grouping documents is not sufficient to achieve accurate results, they may resort to experts. Likewise, if Wikipedia cannot provide enough background knowledge to the UTC, users may feed the UTC with auxiliary documents. Auxiliary documents may contain useful expertise knowledge to aid the classification process. For instance, suppose the unlabeled documents to be classified and submitted by the user concern a technical topic, such as "transfer learning", which may be far beyond Wikipedia's domain. The user could then provide the UTC with auxiliary documents about "machine learning" to overcome the lack of background coverage. It is important to emphasize that the UTC does not require labels for the auxiliary documents.

Thus, before performing classification, all Wikipedia articles, and corresponding titles, are collected. Each article (title) is considered as a single concept [19]. The user submits a set of test documents to be classified, and may submit an additional set of auxiliary documents. The UTC identifies, by exact match, all Wikipedia concepts (titles) that contain al least one discriminant word, and all Wikipedia concepts mentioned in the test documents, and in the auxiliary documents if given. Concepts which appear rarely (in less than five documents in our experiments) are discarded to reduce noise. The selected Wikipedia articles provide an enriched representation of the categories of interest discussed in the test documents. They embed the collective knowledge of Wikipedia relevant to the specific problem domain at hand.

4.3 Learning Topics

After the identification of related Wikipedia concepts, the UTC applies LDA to learn a model of the topics discussed in the corresponding Wikipedia articles. If auxiliary documents are provided, topics are learned from both Wikipedia articles and auxiliary documents. The learned topics represent priors, i.e., distributions modeled from the available knowledge. In particular, LDA learns two distributions, the topic-word distribution ϕ, and the topic-document distribution θ:

$$\phi_{k,t} = \frac{n_k^{(t)} + \beta_t}{\sum_{v=1}^{V} n_k^{(v)} + \beta_v} \tag{3}$$

$$\theta_{m,k} = \frac{n_m^{(k)} + \alpha_k}{\sum_{j=1}^{K} n_m^{(j)} + \alpha_j} \tag{4}$$

M is the total number of related Wikipedia articles (and auxiliary documents if given); K is the number of topics; V is vocabulary size; $\vec{\alpha}$ and $\vec{\beta}$ are the Dirichlet priors for θ and ϕ; α_k is the k^{th} component of the vector $\vec{\alpha}$; β_t is the t^{th} component of the vector $\vec{\beta}$; $n_k^{(t)}$ is the number of times the t^{th} word is assigned to the k^{th} topic; $n_m^{(k)}$ is the number of words in the m^{th} document assigned to the k^{th} topic; $\phi_{k,t}$ represents the probability of the t^{th} term given the k^{th} topic; $\theta_{m,k}$ denotes the probability of the k^{th} topic given the m^{th} document.

The prior distributions $\phi_{k,t}$ and $\theta_{m,k}$ are then updated into posteriors using the unlabeled documents provided by the user [12, 15]. Specifically, the topic-word distribution $\phi_{k,t}$ is updated into a new $\phi'_{k,t}$, and a new topic-document distribution $\theta'_{\underline{m},k}$ is learned from scratch using the unlabeled documents:

$$\phi'_{k,t} = \frac{n_k^{(t)} + \underline{n}_k^{(t)} + \beta_t}{\sum_{v=1}^{V} n_k^{(v)} + \underline{n}_k^{(v)} + \beta_v} \tag{5}$$

$$\theta'_{\underline{m},k} = \frac{n_{\underline{m}}^{(k)} + \alpha_k}{\sum_{j=1}^{K} n_{\underline{m}}^{(j)} + \alpha_j} \tag{6}$$

where $\underline{M}$ is the total number of unlabeled documents; $\underline{n}_k^{(t)}$ is the number of times the t^{th} word is assigned to the k^{th} topic within the $\underline{M}$ documents; $n_{\underline{m}}^{(k)}$ is the number of words in the $\underline{m}^{th}$ document assigned to the k^{th} topic.

Why UTC performs probabilistic topic modeling? Lets think about how people classify documents. Given a category of interest, e.g. "sport", if an article talks about soccer or basketball, the reader can immediately assign the document to the "sport" category since the fact that soccer and basketball are sports is common knowledge. Similarly, if a document contains the term "NBA", the reader can infer that the article is about basketball, and

therefore sport. In other words, people classify documents not only based on the specific words mentioned therein, but also by leveraging their background knowledge on the subject. Topic modeling aims at simulating such human capability, by providing UTC with an enriched representation of the categories of interest, thus allowing background knowledge to play a role in the classification and prediction process. Table 2 gives examples of learned topics for the 20 Newsgroups data set. For each topic, the 30 words corresponding to the highest probabilities are listed, sorted in descending order. It is apparent from the word distributions that topic modeling successfully allows the characterization of the semantics for the target categories. Such topic model representation is used by the classification algorithm, as described in the following Section.

4.4 Classification Algorithm

LDA provides the topic-word distribution ϕ' and the topic-document distribution θ'. Given a discriminant word t associated to a category, and given an unlabeled document $\underline{m}$, we can compute the probability of t given document $\underline{m}$:

$$\lambda_{\underline{m},t} = \sum_{k=1}^{K} \phi'_{k,t} \cdot \theta'_{\underline{m},t} \tag{7}$$

When classifying a document $\underline{m}$, the probability $\lambda_{\underline{m},t}$ is computed for each discriminant word t associated to each label (or category). Let c_t represent the class label corresponding to word t. The label assigned to document $\underline{m}$ is the one that corresponds to the word with the largest probability value:

$$t^* = \arg\max_t \lambda_{\underline{m},t} \tag{8}$$

and c_{t^*} is the class label assigned to document $\underline{m}$. In our experiments, we assume that categories do not share discriminant words, so that c_{t^*} is uniquely defined for each document. However, this is not required in general, since, if appropriate, we can assign multiple labels to a given document.

For information retrieval, in our experiments we use phrases in addition to discriminant words to characterize classes of interest (phrases and discriminant words are derived from a user-defined query). The probability of a phrase p given a document $\underline{m}$, is computed as follows:

$$\lambda_{\underline{m},p} = \prod_{i=1}^{N_p} \lambda_{\underline{m},t_{p,i}} \tag{9}$$

where $t_{p,i}$ is the i^{th} word in phrase p, and N_p is the number of words in phrase p.

5 Comparison Method: CoCC

In this Section we briefly introduce a co-clustering based cross-domain classification algorithm (CoCC). We compare UTC with CoCC in our experiments.

Co-clustering [6] exploits the duality between objects and features, and simultaneously performs clustering along both dimensions. Formally, let X and Y be two discrete random variables that take values in the sets $\{x_1,...,x_m\}$ and $\{y_1,...,y_n\}$, respectively, and let $p(X,Y)$ be their joint probability distribution. The goal is to simultaneously cluster X into k disjoint clusters, and Y into l disjoint clusters. Let $\{\hat{x}_1,\hat{x}_2,...,\hat{x}_k\}$ be the k clusters of X, and $\{\hat{y}_1,\hat{y}_2,...,\hat{y}_l\}$ the l clusters of Y. Then, the objective becomes finding mappings C_X and C_Y such that $C_X : \{x_1,...,x_m\} \rightarrow \{\hat{x}_1,\hat{x}_2,...,\hat{x}_k\}$, $C_Y : \{y_1,...,y_n\} \rightarrow \{\hat{y}_1,\hat{y}_2,...,\hat{y}_l\}$. The tuple (C_X,C_Y) represents a co-clustering.

The amount of information a random variable X that can reveal about a random variable Y (and vice versa) is measured by using the mutual information $I(X;Y)$, which is defined as follows:

$$I(X;Y) = \sum_x \sum_y p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \tag{10}$$

The quality of a co-clustering is measured by the loss in mutual information $I(X;Y) - I(\hat{X};\hat{Y})$ (subject to the constraints on the number of clusters k and l) [6]. The smaller the loss, the higher the quality of the co-clustering.

The authors in [4] use co-clustering to perform cross-domain text classification. Let D_i and D_o be the set of in-domain and out-of-domain data, respectively. Data in D_i are labeled, and $\mathcal{C}$ represents the set of class labels. The labels of D_o (unknown) are also drawn from $\mathcal{C}$. Let $\mathcal{W}$ be the dictionary of all the words in D_i and D_o. The goal of co-clustering D_o is to simultaneously cluster the documents D_o into $|\mathcal{C}|$ clusters, and the words $\mathcal{W}$ into k clusters. Let $\hat{\mathcal{D}}_o = \{\hat{d}_1,\hat{d}_2,...,\hat{d}_{|\mathcal{C}|}\}$ be the $|\mathcal{C}|$ clusters of D_o, and $\hat{\mathcal{W}} = \{\hat{w}_1,\hat{w}_2,...,\hat{w}_k\}$ the k clusters of $\mathcal{W}$. Following the notation in [6], the objective of co-clustering D_o is to find mappings C_{D_o} and $C_{\mathcal{W}}$ such that

$$C_{D_o} : \{d_1,...,d_m\} \rightarrow \{\hat{d}_1,\hat{d}_2,...,\hat{d}_{|\mathcal{C}|}\}$$
$$C_{\mathcal{W}} : \{w_1,...,w_n\} \rightarrow \{\hat{w}_1,\hat{w}_2,...,\hat{w}_k\}$$

where $|D_o| = m$ and $|\mathcal{W}| = n$. The tuple $(C_{D_o},C_{\mathcal{W}})$, or $(\hat{\mathcal{D}}_o,\hat{\mathcal{W}})$, represents a co-clustering of D_o.

To compute $(\hat{\mathcal{D}}_o,\hat{\mathcal{W}})$, a two step procedure is introduced in [4]. Step 1 clusters the out-of-domain documents into $|\mathcal{C}|$ document clusters according to the word clusters $\hat{\mathcal{W}}$. Step 2 groups the words into k clusters, according to class labels and out-of-domain document clusters simultaneously. The second step allows the propagation of class information from D_i to D_o, by leveraging

word clusters. Word clusters, in fact, carry class information, namely the probability of a class given a word cluster. This process allows to fulfill the classification of out-of-domain documents.

As in [6], the quality of the co-clustering $(\hat{\mathcal{D}}_o, \hat{\mathcal{W}})$ is measured by the loss in mutual information $I(\mathcal{D}_o; \mathcal{W}) - I(\hat{\mathcal{D}}_o \hat{\mathcal{W}})$ Thus, co-clustering aims at minimizing the loss in mutual information between documents and words, before and after the clustering process. Similarly, the quality of word clustering is measured by $I(\mathcal{C}; \mathcal{W}) - I(\mathcal{C}\hat{\mathcal{W}})$ where the goal is to minimize the loss in mutual information between class labels $\mathcal{C}$ and words $\mathcal{W}$, before and after the clustering process. By combining these two measures, the objective of co-clustering based classification becomes:

$$\min_{\hat{\mathcal{D}}_o, \hat{\mathcal{W}}} \{I(\mathcal{D}_o; \mathcal{W}) - I(\hat{\mathcal{D}}_o; \hat{\mathcal{W}}) + \lambda (I(\mathcal{C}; \mathcal{W}) - I(\mathcal{C}; \hat{\mathcal{W}}))\} \tag{11}$$

where λ is a trade-off parameter that balances the effect of the two clustering procedures. Equation (11) enables the classification of out-of-domain documents via co-clustering, where word clusters provide a walkway for labels to migrate from the in-domain to the out-of-domain documents.

To solve the optimization problem (11), the authors in [4] introduce an iterative procedure aimed at minimizing the divergence between the two distributions before and after clustering. The key idea of this approach is leveraging the common words of D_i and D_o to bridge the gap between the two domains.

In our previous work [20], we extended the idea underlying the CoCC algorithm by making the latent semantic relationship between the two domains explicit. This goal was achieved with the use of Wikipedia. By embedding background knowledge constructed from Wikipedia, an enriched representation of documents was generated. Such representation keeps multi-word concepts unbroken, captures the semantic closeness of synonyms, and performs word sense disambiguation for polysemous terms. By combining such enriched representation with the CoCC algorithm, cross-domain classification was performed based on a *semantic bridge* between the two related domains. That is, the resulting pathway that allows to propagate labels from D_i to D_o not only captures common words, but also semantic concepts based on the content of documents. As a consequence, even if the two corpora share few words, the technique is able to bridge the gap by embedding semantic information in the extended representation of documents. As such, improved classification accuracy can be achieved [20].

Table 4: Most frequent subjects from the LA Times data set (TREC 6)

Subject	Frequency
MILITARY CONFRONTATIONS	878
POLITICAL CANDIDATES	602
IRAQ – ARMED FORCES – KUWAIT	556
FOOTBALL PLAYERS	542
SUITS	523
BUSH, GEORGE	491
ACQUISITIONS	472
GOVERNMENT REGULATION	447
BASEBALL PLAYERS	378
ENVIRONMENT	373

6 Experiments

6.1 Data Sets

We evaluated the classification performance of the UTC using the 20 Newsgroups data set [13], and the retrieval performance of the UTC using the LA Times documents from TREC 6. For the 20 Newsgroups data set, following the settings in [20], we split the original data into auxiliary and testing documents so that the resulting subsets of documents are drawn from related but different domains. Table 3 illustrates the splitting for all categories of the 20 Newsgroups data used in our experiments, along with the discriminant words associated to each class label.

The LA Times data set from TREC 6 contains every day's news from 1989 to 1990. Every news article has a headline, a byline, a text, and a date. Some of the articles have a subject and a type. The total number of LA Times documents is 131,896. We used the news articles of year 1990 with non-empty subject, for a total of 24,056 articles. Each news article may have more than one subject, each separated by a semicolon. When performing information retrieval, we use the subjects associated to documents as queries. The subjects of LA times are not evenly distributed; thus, we chose the most frequent ones as queries. Table 4 shows the subjects we selected, and their frequencies (i.e., number of documents per subject). An article $\underline{m}$ is ranked according to $\lambda_{\underline{m},t^*}$, computed using Equations (7) and (8).

6.2 Evaluation

For the classification task, we report the accuracy, i.e. the percentage of documents correctly classified. For the retrieval task, we report precision and recall within the top 20, 50 and 100 retrieved documents, denoted as $Precision@N$ and $Recall@N$, respectively:

$$Precision@N = \frac{|\{\text{relevant documents}\} \cap \{\text{top } N \text{ retrieved documents}\}|}{N}$$

Table 2: Learned topics for 20 Newsgroups data

Topic Related to "rec.hockey"	team game season hockei plai nhl cup player goal playoff win score leagu fina time stanlei career coach star overal record draft round ic ranger divis wing flyer won trade
Topic Related to "rec.baseball"	game leagu team basebal season run plai player seri home major hit win time red world record yanke left field base manag sox pitch pitcher career won stadium nation align
Topic Related to "sci.electronics"	electron sci hp au carter nasa eagl gov boi batteri audio circuit ingr radio amp detector caltech output help phone sdd larc wisc oz input led dec tape bison chip
Topic Related to "sci.crypt"	kei secur encrypt sci comp org crypt clipper chip alt eff algorithm netcom privaci com messag de pgp escrow phone public nsa bit govern cryptographi secret system att access attack
Topic Related to "talk.misc"	misc talk religion appl alt sandvik baptist purdu com kent concret brian arizona netcom koresh xref ecn fbi conspiraci polit royalroad gun mormon fire waterloo uug organpip peopl pagan skeptic
Topic Related to "talk.mideast"	armenian turkish soc soviet cultur peopl russian polit talk armenia greek turk zuma uucp kill serdar mideast moscow argic genocid turkei org sera muslim russia histori didn don sdpa anatolia

Table 3: Splitting of 20 Newsgroups categories for classification

	Data Set	Label	Auxiliary Documents	Test Documents	Discriminant Words
2 Categories	comp vs sci	comp	comp.graphics comp.os.ms-windows.misc	comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x	computer ibm mac hardware windows
		sci	sci.crypt sci.electronics	sci.med sci.space	science medicine space
	rec vs talk	rec	rec.autos rec.motorcycles	rec.sport.baseball rec.sport.hockey	recreation sport baseball hockey
		talk	talk.politics.guns talk.politics.misc	talk.politics.mideast talk.religion.misc	politics mideast religion misc
	rec vs sci	rec	rec.autos rec.sport.baseball	rec.motorcycles rec.sport.hockey	recreation motorcycles sport hockey
		sci	sci.med sci.space	sci.crypt sci.electronics	science crypt cryptography electronics
	sci vs talk	sci	sci.electronics sci.med	sci.crypt sci.space	science crypt cryptography space
		talk	talk.politics.misc talk.religion.misc	talk.politics.guns talk.politics.mideast	politics guns mideast
	comp vs rec	rec	rec.autos rec.sport.baseball	rec.motorcycles rec.sport.hockey	recreation motorcycles sport hockey
		comp	comp.graphics comp.sys.ibm.pc.hardware comp.sys.mac.hardware	comp.os.ms-windows.misc comp.windows.x	computer microsoft windows
	comp vs talk	talk	talk.politics.guns talk.politics.misc	talk.politics.mideast talk.religion.misc	politics mideast religion misc
		comp	comp.graphics comp.sys.mac.hardware comp.windows.x	comp.os.ms-windows.misc comp.sys.ibm.pc.hardware	computer microsoft windows ibm hardware
3 Categories	rec vs sci vs comp	rec	rec.motorcycles rec.sport.hockey	rec.autos rec.sport.baseball	recreation auto sport baseball
		sci	sci.med sci.space	sci.crypt sci.electronics	science crypt cryptography electronics
		comp	comp.graphics comp.sys.ibm.pc.hardware comp.sys.mac.hardware	comp.os.ms-windows.misc comp.windows.x	computer microsoft os graphic
	rec vs talk vs sci	rec	rec.autos rec.motorcycles	rec.sport.baseball rec.sport.hockey	recreation sport baseball hockey
		talk	talk.politics.guns talk.politics.misc	talk.politics.mideast talk.religion.misc	politics mideast religion misc
		sci	sci.med sci.space	sci.crypt sci.electronics	science crypt cryptography electronics
	sci vs talk vs comp	sci	sci.crypt sci.electronics	sci.space sci.med	science space medicine
		talk	talk.politics.mideast talk.religion.misc	talk.politics.misc talk.politics.guns	politics guns misc
		comp	comp.graphics comp.sys.mac.hardware comp.windows.x	comp.os.ms-windows.misc comp.sys.ibm.pc.hardware	computer microsoft windows ibm pc
4 Categories	sci vs rec vs talk vs comp	sci	sci.crypt sci.electronics	sci.space sci.med	science space medicine
		rec	rec.autos rec.motorcycles	rec.sport.baseball rec.sport.hockey	recreation sport baseball hockey
		talk	talk.politics.mideast talk.religion.misc	talk.politics.misc talk.politics.guns	politics misc guns
		comp	comp.graphics comp.os.ms-windows.misc	comp.sys.mac.hardware comp.sys.ibm.pc.hardware comp.windows.x	computer ibm mac hardware windows

$$Recall@N = \frac{|\{\text{relevant documents}\} \cap \{\text{top } N \text{ retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

6.3 Implementation Details

We use the March 12, 2008 version of Wikipedia XML dump file. We use WikiPrep[1] to extract all articles from the Wikipedia XML dump file. WikiPrep can extract a lot of information from a Wikipedia article, such as title, text, links to other articles, and the Wikipedia categories it belongs to. In our experiments we only use the title and text of a Wikipedia article, and we treat every title of an article as a Wikipedia concept.

We use the Java implementation of LDA [12], and set $\alpha = 0.5$ and $\beta = 0.01$. For all our experiments, we fix the number of topics K to 50. For classification, we run LDA on Wikipedia articles, and auxiliary documents if given, for 5000 iterations. To update the distributions using the test documents, we run LDA again for 5000 iterations. The number of test and auxiliary documents ranges from 2000 to 8000, and the number of related Wikipedia articles ranges from 2000 to 5000. Our analysis shows that 5000 iterations are sufficient to provide good results for these data sizes. For retrieval, the number of test documents is 24,056, and the number of related Wikipedia articles is more than 15,000. LDA is again run for 5000 iterations.

The UTC identifies, by exact match (case sensitive), all Wikipedia concepts (titles) mentioned in the test documents, and in the auxiliary documents if given. Concepts which appear rarely (in less than five documents in our experiments) are discarded to reduce noise. If a concept is contained in another one, e.g., the concept "University" is contained in the concept "University of California", both are considered, and therefore both corresponding Wikipedia articles are retrieved.

Before running LDA, stop words and rare words (those with document frequency less than three) are removed. Stemming is also performed using the Porter algorithm [16].

6.4 Experimental Results

6.4.1 Classification Results

For the classification task, we test the performance of the UTC under four conditions, based on the availability of Wikipedia articles and auxiliary documents: (1) only test documents are available (this case is denoted as "w/o Wiki&Aux"; (2) Wikipedia related articles are available (denoted as "w/ Wiki"); (3) auxiliary documents are available (denoted as "w/ Aux"); and (4) both Wikipedia related articles and auxiliary documents are available (denoted as "w/ Wiki&Aux"). It is expected that Wikipedia related articles and auxiliary documents improve the classification performance of the UTC.

[1] http://www.cs.technion.ac.il/~gabr/resources/code/wikiprep

Figure 2: Classification accuracy vs LDA iterations

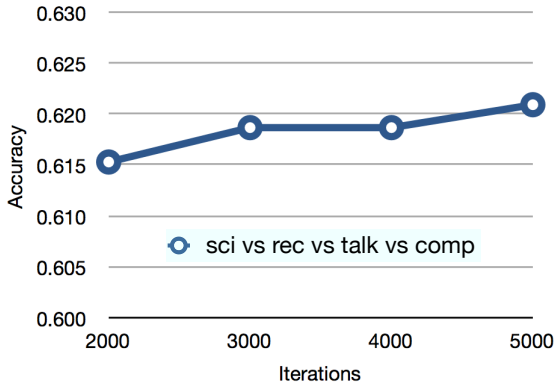

Table 5 shows the classification accuracy of the UTC, along with the accuracy obtained with the CoCC algorithm with Wikipedia (denoted as "CoCC w/Wiki"). CoCC w/Wiki uses Wikipedia to achieve an enriched representation of documents, and uses labeled auxiliary data [20]. For all ten classification problems, the use of Wikipedia or auxiliary data improves accuracy; when used in combination (w/ Wiki&Aux), further improvement is observed. For all the binary and three class classification problems, UTC w/ Wiki&Aux and CoCC w/Wiki achieved similar results. On the four class classification problem, CoCC w/Wiki still performs significantly better. By taking into consideration the fact that the UTC does not use any labeled auxiliary data, the results obtained with respect to CoCC are very promising.

Figure 2 plots the classification accuracy of UTC w/o Wiki&Aux as a function of the numbers of LDA iterations (after the first 2000 burn-in iterations) for the four-category problem "sci vs rec vs talk vs comp". We observe that the improvement in accuracy achieved between 3000 and 5000 iterations is quite small. This is an indication that 5000 iterations suffice for LDA to converge.

6.4.2 Retrieval Results

For the retrieval task, we perform two experiments: retrieval based only on test documents ("w/o Wiki"), and retrieval using Wikipedia as well ("w/ Wiki"). Table 6 gives the precision and recall values for both settings. For all ten queries, and for all measures (@20, @50, and @100), the improvement due to the use of Wikipedia is substantial. These results clearly demon-

Table 5: Classification results

Data set	w/o Wiki&Aux	w/ Wiki	w/ Aux	w/ Wiki&Aux	CoCC w/ Wiki
comp vs sci	0.938	0.962	0.970	0.977	0.987
rec vs talk	0.959	0.971	0.978	0.985	0.998
rec vs sci	0.930	0.951	0.955	0.963	0.984
sci vs talk	0.947	0.965	0.969	0.978	0.988
comp vs rec	0.929	0.964	0.971	0.980	0.993
comp vs talk	0.962	0.973	0.978	0.983	0.995
rec vs sci vs comp	0.874	0.885	0.896	0.900	0.904
rec vs talk vs sci	0.878	0.927	0.936	0.945	0.979
sci vs talk vs comp	0.869	0.883	0.898	0.907	0.912
sci vs rec vs talk vs comp	0.621	0.630	0.637	0.640	0.713

strate the advantage of incorporating background knowledge through Wikipedia, and the effectiveness of modeling such knowledge via topic modeling.

7 Conclusion and Future work

We proposed a classifier that can effectively group documents based on their content under the guidance of few words describing the class of interest. The UTC uses background knowledge during the learning process for predicting labels of test documents. The UTC can also be used to perform document retrieval, in which the pool of test documents may or may not be relevant to the topics of interest. Experimental results demonstrate the feasibility of our approach, and the advantage of incorporating background knowledge through Wikipedia.

Learning topic models from documents is a time-consuming process. In our future work, we will consider alternative approaches to knowledge representation, and explore automated mechanisms to generate discriminant words to describe the categories of interest.

References

[1] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. In *Journal of machine Learning Research 3*, pages 993–1022, 2003.

[2] M. Chang, L. Ratinov, D. Roth, and V. Srikumar. Importance of semantic represenation: Dataless classification. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2008.

[3] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu. Boosting for transfer learning. In *International Conference on Machine Learning*, 2007.

[4] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu. Co-clustering based classification for out-of-domain documents. In *KDD*, 2007.

[5] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu. Transferring naive bayes classifiers for text classification. In *AAAI Conference on Artificial Intelligence (AAAI-2007)*, 2007.

[6] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *International Conference on Knowledge Discovery and Data Mining (KDD-2003)*, 2003.

[7] C. Do and A. Y. Ng. Transfer learning for text classification. In *Annual Conference on Neural Information Processing Systems (NIPS-2005)*, 2005.

[8] E. Gabrilovich and S. Markovitch. Feature generation for text categorization using world knowledge. In *International Joint Conference on Artificial Intelligence (IJCAI-2005)*, 2005.

[9] E. Gabrilovich and S. Markovitch. Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. In *AAAI Conference on Artificial Intelligence (AAAI-2006)*, 2006.

[10] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *International Joint Conference on Artificial Intelligence (IJCAI-2007)*, 2007.

[11] T. L. Griffiths and M. Steyvers. Finding scientific topics. *National Academy of Sciences*, 101:5228–5235, April 2004.

[12] G. Heinrich. Parameter estimation for text analysis. In *Technical Report*, 2008.

[13] K. Lang. Newsweeder: Learning to filter netnews. In *International Conference on Machine Learning*, 1995.

[14] T. Minka and J. Lafferty. Expectation-propagation for the generative aspect model. In *In Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*, pages 352–359, 2002.

[15] X. H. Phan, M. L. Nguyen, and S. Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *WWW*, pages 91–100, 2008.

Table 6: Retrieval results

Subject	w/o Wiki						w/ Wiki					
	P@20	R@20	P@50	R@50	P@100	R@100	P@20	R@20	P@50	R@50	P@100	R@100
MILITARY CONFRONTATIONS	0.850	0.019	0.820	0.046	0.820	0.093	0.950	0.021	0.960	0.054	0.950	0.108
POLITICAL CANDIDATES	0.350	0.011	0.340	0.028	0.340	0.056	0.600	0.019	0.700	0.057	0.710	0.117
IRAQ – ARMED FORCES – KUWAIT	0.500	0.017	0.640	0.057	0.640	0.114	0.750	0.026	0.780	0.070	0.790	0.141
FOOTBALL PLAYERS	0.350	0.114	0.400	0.032	0.450	0.073	0.350	0.114	0.460	0.037	0.520	0.085
SUITS	0.150	0.003	0.100	0.005	0.190	0.019	0.400	0.008	0.400	0.020	0.370	0.037
BUSH, GEORGE	0.250	0.010	0.260	0.026	0.250	0.050	0.550	0.022	0.480	0.048	0.510	0.103
ACQUISITIONS	0.100	0.004	0.120	0.012	0.180	0.037	0.550	0.023	0.520	0.054	0.460	0.096
GOVERNMENT REGULATION	0.100	0.004	0.060	0.006	0.070	0.015	0.350	0.015	0.360	0.040	0.390	0.086
BASEBALL PLAYERS	0.500	0.019	0.680	0.067	0.800	0.158	0.600	0.023	0.720	0.071	0.800	0.158
ENVIRONMENT	0.300	0.012	0.340	0.035	0.240	0.050	0.350	0.014	0.440	0.046	0.380	0.080

[16] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[17] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-taught learning: transfer learning from unlabeled data. In *International Conference on Machine Learning*, pages 759–766, 2007.

[18] R. Raina, A. Y. Ng, and D. Koller. Constructing informative priors using transfer learning. In *International Conference on Machine Learning*, 2006.

[19] P. Wang and C. Domeniconi. Building semantic kernels for text classification using wikipedia. In *KDD*, pages 713–721, 2008.

[20] P. Wang, C. Domeniconi, and J. Hu. Using wikipedia for co-clustering based cross-domain text classifcation. In *Proceedings of the IEEE International Conference on Data Mining*, 2008.

[21] X. Wei and W. B. Croft. Lda-based document models for ad-hoc retrieval. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 178–185, 2006.