

Towards a Universal Text Classifier: Transfer Learning using Encyclopedic Knowledge

Pu Wang

*Department of Computer Science
George Mason University
Fairfax, Virginia, USA
pwang7@cs.gmu.edu*

Carlotta Domeniconi

*Department of Computer Science
George Mason University
Fairfax, Virginia, USA
carlotta@cs.gmu.edu*

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. In this work, we propose a *universal text classifier*, which does not require any labeled 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 describing the classes of interest. Background knowledge is modeled using encyclopedic knowledge, namely Wikipedia. The universal text classifier can also be used to perform document retrieval. In our experiments with real data we test the feasibility of our approach for both the classification and retrieval tasks.

Keywords—Transfer learning; Text classifiers; Wikipedia

I. INTRODUCTION

Text classification is an essential task to enable the organization and usage of very large numbers of documents. 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. 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 just based on the specific words mentioned therein, but also by leveraging

background knowledge on the subject. Our approach simulates 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 [1]. 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.

Background knowledge is modeled using encyclopedic knowledge, namely Wikipedia. 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. 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. 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 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.

II. RELATED WORK

Transfer Learning. Much recent work has examined embedding domain knowledge into learning, including methods that use labels or relevance judgment on features [2], [3], [4], [5], [6]. These methods convert labeled features into labeled instances, and apply a standard learning algorithm.

Raina et al. [7] built a term covariance matrix using an auxiliary problem to measure the co-occurrence between terms, and applied it to the target learning task. The authors in [8] proposed self-taught learning, which uses labeled and unlabeled data to aid the target task.

Do et al. [9] modeled the text classification problem using a linear function with different parameters, and a meta-learning method was introduced to learn how to tune them. Dai et al. [10] modified the Naive Bayes classifier, and in [11] the authors altered the Boosting algorithm to handle a cross-domain classification tasks.

The authors in [12] used co-clustering [13] to perform cross-domain text classification (CoCC algorithm). Common words between the documents to be classified and the auxiliary ones were used to bridge the gap between the two domains. In [14], this idea was extended by making the latent semantic relationship between the two domains explicit with the use of Wikipedia.

Text Classification using Wikipedia. Gabrilovich et al. [15], [16] proposed a method to integrate text classification with Wikipedia. Their approach only leverages similarity between text fragments and Wikipedia articles, ignoring the abundant structural information within Wikipedia.

In [17], the authors constructed an informative thesaurus from Wikipedia, which explicitly derives synonymy, polysemy, hyponymy, and associative relations between concepts. This thesaurus was used to embed semantic information in documents. The UTC approach avoids this time consuming step by applying probabilistic topic modeling directly on Wikipedia’s articles.

Recently, Chang et al. [1] proposed a data-less classification approach to classify documents without training data based on explicit semantic analysis (ESA) [18]. It finds Wikipedia concepts related to given terms descriptive of the categories of interest. 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.

Topic Models. Blei et al. [19] introduced Latent Dirichlet Allocation (LDA), a probabilistic and unsupervised generative model that can be used to estimate multinomial data [19], [20]. In [20], an hybrid approach was proposed to combine 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 can improve the results for information retrieval.

III. 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 integrating 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: $p(\mathcal{W} | \vec{\alpha}, \vec{\beta}) = \prod_{m=1}^M p(\vec{w}_m | \vec{\alpha}, \vec{\beta})$. The estimation of LDA parameters by maximization of the likelihood of the whole data collection is intractable. The solution to this problem is to use approximate estimation methods such as Variational Methods [19], Expectation-propagation [21], or Gibbs sampling [22], [23]. We provide here the fundamental steps of Gibbs sampling. 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: $p(z_t = k | \vec{z}_{-t}, \vec{w}) = \frac{n_{k,-t}^{(t)} + \beta_t}{[\sum_{v=1}^V n_k^{(v)} + \beta_v] - 1} \frac{n_{m,-t}^{(k)} + \alpha_k}{[\sum_{j=1}^K n_m^{(j)} + \alpha_j] - 1}$, where $n_{k,-t}^{(t)}$ is the number of times word t is assigned to topic k , except the current assignment; $\sum_{v=1}^V n_k^{(v)} - 1$ is the number of words assigned to topic k , except the current assignment; $n_{m,-t}^{(k)}$ is the number of words in document m assigned to topic k , except the current assignment; and $\sum_{j=1}^K n_m^{(j)} - 1$ 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_k s ($k = 1, \dots, K$) are the same, and similarly for all β_v s ($v = 1, \dots, V$).

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 Equation 1.

IV. UNIVERSAL TEXT CLASSIFIER

Discriminant Words for Categories. 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 assume that such list is given as input to the UTC. Table I gives an example of how we can generate discriminant words to characterize the categories for the 20 Newsgroups data. 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.

Related Wikipedia Articles. 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 [17]. 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 at 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.

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 :

$$\phi_{k,t} = \frac{n_k^{(t)} + \beta_t}{\sum_{v=1}^V n_k^{(v)} + \beta_v} \quad \theta_{m,k} = \frac{n_m^{(k)} + \alpha_k}{\sum_{j=1}^K n_m^{(j)} + \alpha_j} \quad (1)$$

M is the total number of related Wikipedia articles (and auxiliary documents if given); K is the number of topics; V is vocabulary size; $\bar{\alpha}$ and $\bar{\beta}$ are the Dirichlet priors for θ and ϕ ; α_k is the k^{th} component of the vector $\bar{\alpha}$; β_t is the t^{th} component of the vector $\bar{\beta}$; $n_k^{(t)}$ 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; $\phi_{k,t}$ is the probability of the t^{th} term given the k^{th} topic; $\theta_{m,k}$ is 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 [23], [24]. 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 relearned using the unlabeled documents:

$$\phi'_{k,t} = \frac{n_k^{(t)} + \underline{n}_k^{(t)} + \beta_t}{\sum_{v=1}^V n_k^{(v)} + \underline{n}_k^{(v)} + \beta_v} \quad \theta'_{\underline{m},k} = \frac{n_m^{(k)} + \alpha_k}{\sum_{j=1}^K n_m^{(j)} + \alpha_j} \quad (2)$$

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; $n_m^{(k)}$ is the number of words in the $\underline{m}^{\text{th}}$ 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.

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} :

$$\lambda_{\underline{m},t} = \sum_{k=1}^K \phi'_{k,t} \cdot \theta'_{\underline{m},k} \quad (3)$$

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:

$$t^* = \arg \max_t \lambda_{\underline{m}, t} \quad (4)$$

and c_{t^*} is the class label assigned to document \underline{m} .

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: $\lambda_{\underline{m}, p} = \prod_{i=1}^{N_p} \lambda_{\underline{m}, t_{p,i}}$, where $t_{p,i}$ is the i^{th} word in phrase p , and N_p is the number of words in phrase p .

V. EXPERIMENTAL RESULTS

Data Sets. We evaluated the classification performance using the 20 Newsgroups data [25], and the retrieval performance of the UTC using the LA Times documents from TREC 6. Following [14], we split the 20 Newsgroups data into auxiliary and test documents so that the resulting subsets are drawn from related but different domains. Table I illustrates the splitting for all categories used.

The LA Times data set from TREC 6 contains daily 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. 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 II 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 Eqs. (3) and (4).

Evaluation. For the classification task, we report the accuracy. For the retrieval task, we report precision and recall within the top 20, 50 and 100 retrieved documents, denoted as $P@N$ and $R@N$, respectively.

Implementation Details. We use the March 12, 2008 version of Wikipedia XML dump file. We use WikiPrep¹ to extract all articles from the Wikipedia XML dump file. 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 [23], 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

Table II
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

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, both are considered, and therefore both corresponding Wikipedia articles are retrieved.

We compare the UTC with three other methods, NB-EM [2], semantic kernel [17] and CoCC [14]. For the NB-EM method, we use the same representative terms for each class as used for the UTC. For the semantic kernel approach and the CoCC algorithm, we use the auxiliary documents as training data, and predict on the test documents.

As preprocessing, stop words and rare words (with document frequency less than three) are removed. Stemming is performed using the Porter algorithm [26].

Results. For the classification task, we test the performance of the UTC under four conditions: (1) only test documents are available (denoted as “w/o Wiki&Aux”); (2) Wikipedia related articles are available (“w/ Wiki”); (3) auxiliary documents are available (“w/ Aux”); and (4) both Wikipedia related articles and auxiliary documents are available (“w/ Wiki&Aux”).

Table III shows the classification accuracy of the UTC, along with the accuracy obtained with the NB-EM method, the semantic kernel approach with Wikipedia (“S.-K. w/ Wiki”) and the CoCC algorithm with Wikipedia (“CoCC w/ Wiki”). Both semantic kernel and CoCC use Wikipedia to achieve an enriched representation of documents. As for the UTC, for all the ten classification problems, the use of Wikipedia or auxiliary data improves accuracy; when used in combination (w/ Wiki&Aux), further improvement is observed. Comparing the UTC with the NB-EM, semantic kernel and CoCC, for all the ten classification problems, NB-EM is worse than semantic kernel, and semantic kernel is worse than UTC w/o Wiki&Aux; for all the binary and three class classification problems, UTC w/ Wiki&Aux and CoCC w/ Wiki achieved similar results, while on the four

¹<http://www.cs.technion.ac.il/~gabr/resources/code/wikiprep>

Table I
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
		sci	sci.crypt sci.electronics	sci.space sci.med	science space medicine
4 Categories	sci vs rec vs talk vs comp	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
		sci	sci.crypt sci.electronics	sci.space sci.med	science space medicine

class classification problem, CoCC w/Wiki still performs significantly better. By taking into consideration the fact that the UTC does not use any labeled data, and semantic kernel and CoCC are supervised learning algorithms, the results obtained by UTC are very promising.

For the retrieval task, we perform two experiments: retrieval based only on test documents (“w/o Wiki”), and retrieval using Wikipedia (“w/ Wiki”). Table IV gives precision and recall 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 demonstrate the advantage of incorporating background knowledge through Wikipedia, and the effectiveness of modeling such knowledge via topic modeling.

VI. CONCLUSION

We proposed a classifier that can effectively group documents based on their content under the guidance of few words describing the class of interest. In our future work, we

will explore automated mechanisms to generate discriminant words to describe the categories of interest.

REFERENCES

- [1] M. Chang, L. Ratinov, D. Roth, and V. Srikumar, “Importance of semantic representation: Dataless classification,” in *AAAI*, 2008.
- [2] B. Liu, X. Li, W. S. Lee, and P. S. Yu, “Text classification by labeling words,” in *In AAAI-2004*, 2004, pp. 425–430.
- [3] R. E. Schapire, M. Rochery, M. G. Rahim, and N. Gupta, “Incorporating prior knowledge into boosting,” in *ICML*, 2002, pp. 538–545.
- [4] X. Wu and R. Srihari, “Incorporating prior knowledge with weighted margin support vector machines,” in *KDD*, 2004, pp. 326–333.
- [5] A. Dayanik, D. D. Lewis, D. Madigan, V. Menkov, and A. Genkin, “Constructing informative prior distributions from domain knowledge in text classification,” in *SIGIR*, 2006, pp. 493–500.

Table III
CLASSIFICATION ACCURACY

Data set	w/o Wiki&Aux	w/ Wiki	w/ Aux	w/ Wiki&Aux	NB-EM	S.-K. w/ Wiki	CoCC w/ Wiki
comp vs sci	0.938	0.962	0.970	0.977	0.916	0.923	0.987
rec vs talk	0.959	0.971	0.978	0.985	0.941	0.957	0.998
rec vs sci	0.930	0.951	0.955	0.963	0.909	0.924	0.984
sci vs talk	0.947	0.965	0.969	0.978	0.925	0.940	0.988
comp vs rec	0.929	0.964	0.971	0.980	0.918	0.921	0.993
comp vs talk	0.962	0.973	0.978	0.983	0.947	0.956	0.995
rec vs sci vs comp	0.874	0.885	0.896	0.900	0.848	0.852	0.904
rec vs talk vs sci	0.878	0.927	0.936	0.945	0.853	0.866	0.979
sci vs talk vs comp	0.869	0.883	0.898	0.907	0.836	0.858	0.912
sci vs rec vs talk vs comp	0.621	0.630	0.637	0.640	0.547	0.617	0.713
Average	0.890	0.911	0.918	0.925	0.864	0.881	0.945

Table IV
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
AVERAGE	0.345	0.021	0.376	0.031	0.398	0.065	0.545	0.028	0.582	0.049	0.588	0.101

- [6] G. Druck, G. Mann, and A. McCallum, "Learning from labeled features using generalized expectation criteria," in *SIGIR*, 2008, pp. 595–602.
- [7] R. Raina, A. Y. Ng, and D. Koller, "Constructing informative priors using transfer learning," in *ICML*, 2006.
- [8] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: transfer learning from unlabeled data," in *ICML*, 2007, pp. 759–766.
- [9] C. Do and A. Y. Ng, "Transfer learning for text classification," in *NIPS*, 2005.
- [10] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, "Transferring naive bayes classifiers for text classification," in *AAAI*, 2007.
- [11] —, "Boosting for transfer learning," in *ICML*, 2007.
- [12] —, "Co-clustering based classification for out-of-domain documents," in *KDD*, 2007.
- [13] I. S. Dhillon, S. Mallela, and D. S. Modha, "Information-theoretic co-clustering," in *KDD*, 2003.
- [14] P. Wang, C. Domeniconi, and J. Hu, "Using wikipedia for co-clustering based cross-domain text classification," in *ICDM*, 2008.
- [15] E. Gabrilovich and S. Markovitch, "Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge," in *AAAI*, 2006.
- [16] —, "Feature generation for text categorization using world knowledge," in *IJCAI*, 2005.
- [17] P. Wang and C. Domeniconi, "Building semantic kernels for text classification using wikipedia," in *KDD*, 2008, pp. 713–721.
- [18] E. Gabrilovich and S. Markovitch, "Computing semantic relatedness using wikipedia-based explicit semantic analysis," in *IJCAI*, 2007.
- [19] D. Blei, A. Ng, and M. Jordan, "Latent dirichlet allocation," in *Journal of machine Learning Research* 3, 2003, pp. 993–1022.
- [20] X. Wei and W. B. Croft, "Lda-based document models for ad-hoc retrieval," in *SIGIR*, 2006, pp. 178–185.
- [21] T. Minka and J. Lafferty, "Expectation-propagation for the generative aspect model," in *UAI*, 2002, pp. 352–359.
- [22] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *National Academy of Sciences*, vol. 101, pp. 5228–5235, April 2004.
- [23] G. Heinrich, "Parameter estimation for text analysis," in *Technical Report, University of Leipzig, Germany*, 2008.
- [24] 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*, 2008, pp. 91–100.
- [25] K. Lang, "Newsweeder: Learning to filter netnews," in *ICML*, 1995.
- [26] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980.