Cross-domain Text Classification using Wikipedia

Pu Wang, Carlotta Domeniconi, and Jian Hu

***Abstract*—Traditional approaches to document classification requires labeled data in order to construct reliable and accurate classifiers. Unfortunately, labeled data are seldom available, and often too expensive to obtain, especially for large domains and fast evolving scenarios. Given a learning task for which training data are not available, abundant labeled data may exist for a different but related domain. One would like to use the related labeled data as auxiliary information to accomplish the classification task in the target domain. Recently, the paradigm of transfer learning has been introduced to enable effective learning strategies when auxiliary data obey a different probability distribution.**

A co-clustering based classification algorithm has been previously proposed to tackle cross-domain text classification. In this work, we extend the idea underlying this approach by making the latent semantic relationship between the two domains explicit. This goal is achieved with the use of Wikipedia. As a result, the pathway that allows to propagate labels between the two domains not only captures common words, but also semantic concepts based on the content of documents. We empirically demonstrate the efficacy of our semantic-based approach to cross-domain classification using a variety of real data.

***Index Terms*—Text Classification, Wikipedia, Kernel methods, Transfer learning.**

I. Introduction

Document classification is a key task for many text mining applications. For example, the Internet is a vast repository of disparate information growing at an exponential rate. Efficient and effective document retrieval and classication systems are required to turn the massive amount of data into useful information, and eventually into knowledge. Unfortunately, traditional approaches to classification requires labeled data in order to construct reliable and accurate classifiers. Labeled data are seldom available, and often too expensive to obtain, especially for large domains and fast evolving scenarios. On the other hand, given a learning task for which training data are not available, abundant labeled data may exist for a different but related domain. One would like to use the related labeled data as auxiliary information to accomplish the classification task in the target domain. Traditional machine learning approaches cannot be applied directly, as they assume that training and testing data are drawn from the same underlying distribution. Recently, the paradigm of transfer learning has been introduced to enable effective learning strategies when auxiliary data obey a different probability distribution.

A co-clustering based classification algorithm has been proposed to tackle cross-domain text classification [17]. Let D_i be the collection of labeled auxiliary documents, called *in-domain* documents, and D_o be the set of (*out-of-domain*) documents to be classified (for which no labels are available). D_i and D_o may be drawn from different distributions. Nevertheless, since the two domains are related, e.g., baseball vs. hockey, effectively the conditional probability of a class label given a word is similar in the two domains. The method leverages the shared dictionary across the in-domain and the out-of-domain documents to propagate the label information from D_i to D_o. This is achieved by means of a two-step co-clustering procedure [17]. Specifically, it is assumed that class labels for D_i and D_o are drawn from the same set of class labels (for example, one class label may be "sport"; the documents in D_i are about baseball, and those in D_o are about hockey). Two co-clustering steps are carried out: one finds groups of documents and words for the out-of domain documents, and the other discovers groups of labels and words. In both cases, the set of words considered is the union of the terms appearing in D_i and D_o.

Thus, the words shared across the two domains allow the propagation of the class structure from the in-domain to the out-of-domain. Intuitively, if a word cluster $\hat{w}$ usually appears in class c in D_i, then, if a document $d \in D_o$ contains the same word clusters $\hat{w}$, it is likely that d belongs to class c as well.

The co-clustering approach in [17] (called CoCC) leverages the common words of D_i and D_o to bridge the gap between the two domains. The method is based on the "Bag of Words" (BOW) representation of documents, where each document is modeled as a vector with a dimension for each term of the dictionary containing all the words that appear in the corpus. In this work, we extend the idea underlying the CoCC algorithm by making the latent semantic relationship between the two domains explicit. This goal is achieved with the use of Wikipedia. By embedding background knowledge constructed from Wikipedia, we generate an enriched representation of documents, which is capable of keeping multi-word concepts unbroken, capturing the semantic closeness of synonyms, and performing word sense disambiguation for polysemous terms. By combining such enriched representation with the CoCC algorithm, we can perform cross-domain classification 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 (e.g., synonyms are used to express similar concepts), our technique is able to bridge the gap by embedding semantic information in the extended representation of documents. As such, improved classification accuracy is expected, as also demonstrated in our experimental results.

In our previous work [31], a thesaurus was derived from Wikipedia, which explicitly defines synonymy, hyponymy and associative relations between concepts. Using the thesaurus

Pu Wang and Carlotta Domeniconi are with the Department of Computer Science, George Mason University, Fairfax, VA, 22030 USA e-mail: pwang7@gmu.edu, carlotta@cs.gmu.edu.

Jian Hu is with Microsoft Research Asia, 49 Zhichun Road, Beijing 100080, P.R. China email: jianh@microsoft.com

constructed from Wikipedia, semantic information was embedded within the document representation, and the authors proved via experimentation that improved classification accuracy can be achieved [30]. In this work, we leverage these techniques to develop a semantic-based cross-domain classification approach.

The rest of the paper is organized as follows. In Section II, we discuss related work. In Section III, the background on co-clustering and the CoCC algorithm is covered. Section IV describes the structure of Wikipedia, and how we build a thesaurus from Wikipedia [31]. Section V presents the methodology to embed semantics into document representation, and Section VI describes our overall approach to cross-domain classification. In Section VII experiments are presented, and Section VIII provides conclusions and ideas for future work.

II. Related Work

In this section, we review background work in the areas of transfer learning, and text classification using encyclopedic knowledge.

A. Transfer learning

Cross-domain classification is related to transfer learning, where the knowledge acquired to accomplish a given task is used to tackle another learning task. In [28], the authors built a term covariance matrix using the auxiliary problem, to measure the co-occurrence between terms. The resulting term covariance is then applied to the target learning task. For instance, 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 [21], the authors model the text classification problem using 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 (i.e., a good parameter function) for new tasks, replacing hours of hand-tweaking.

In [19], Dai et al. 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 represent the distribution distance between the 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 [18], 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.

B. Text classification using encyclopedic knowledge

Research has been done to exploit ontologies for content-based categorization of large corpora of documents. In particular, WordNet has been widely used. Siolas et al. [13] build a semantic kernel based on WordNet. Their approach can be viewed as an extension of the ordinary Euclidean metric. Jing et al. [10] define a term similarity matrix using WordNet to improve text clustering. Their approach only uses synonyms and hyponyms. It fails to handle polysemy, and breaks multi-word concepts into single terms. Hotho et al. [9] integrate WordNet knowledge into text clustering, and investigate word sense disambiguation strategies and feature weighting schema by considering the hyponymy relations derived from WordNet. Their experimental evaluation shows some improvement compared with the best baseline results. However, considering the restricted coverage of WordNet, the effect of word sense disambiguation is quite limited. The authors in [5], [14] successfully integrate the WordNet resource for document classification. They show improved classification results with respect to the Rocchio and Widrow-Hoff algorithms. Their approach, though, does not utilize hypernyms and associate terms (as we do with Wikipedia). Although [4] utilized WordNet synsets as features for document representation and subsequent clustering, the authors did not perform word sense disambiguation, and found that WordNet synsets actually decreased clustering performance.

Gabrilovich et al. [7], [8] propose a method to integrate text classification with Wikipedia. They first build an auxiliary text classifier that can match documents with the most relevant articles of Wikipedia, and then augment the BOW representation with new features which are the concepts (mainly the titles) represented by the relevant Wikipedia articles. 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 feature generation procedure acts similarly to a retrieval process: it receives a text fragment (such as words, a sentence, a paragraph, or the whole document) as input, and then maps it to the most relevant Wikipedia articles. This method, however, only leverages text similarity between text fragments and Wikipedia articles, ignoring the abundant structural information within Wikipedia, e.g. internal links. The titles of the retrieved Wikipedia articles are treated as new features to enrich the representation of documents [7], [8]. The authors claim that their feature generation method implicitly performs words sense disambiguation: polysemous words within the context of a text fragment are mapped to the concepts which correspond to the sense shared by other context words. However, the processing effort 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. Furthermore, implicit word sense disambiguation processing is not as effective as explicit disambiguation, as we perform in our approach.

In [16], Banerjee et al. tackled the daily classification

task (DCT) [22] by importing Wikipedia knowledge into documents. The method is quite straightforward: using Lucene (http://lucene.apache.org) to index all Wikipedia articles, each document is used as a query to retrieve the top 100 matching Wikipedia articles. The corresponding titles become new features. This technique is prone to bring a lot noise into documents. Similarly to [22], documents are further enriched by combining the results of the previous n daily classifiers with new testing data. By doing so, the authors claim that the combined classifier is at least no worse than the previous n classifiers. However, this method is based on the assumption that a category may be comprised of a union of (potentially undiscovered) subclasses or themes, and the class distribution of these subclasses may shift over time.

Milne et al. [25] build a professional, domain-specific thesaurus of agriculture from Wikipedia. Such thesaurus takes little advantage of the rich relations within Wikipedia articles. On the contrary, our approach relies on a general thesaurus, which supports the processing of documents concerning a variety of topics. We investigate a methodology that makes use of such thesaurus, to enable the integration of the rich semantic information of Wikipedia into a kernel.

III. Co-clustering

Clustering aims at organizing data in groups so that objects similar to each other are placed in the same group, or cluster. Co-clustering exploits the duality between objects and features, and simultaneously performs clustering along both dimensions. For example, for text mining applications, co-clustering discovers groups of documents and groups of words, thus leveraging the interplay between documents and words when defining similar documents.

The authors in [20] model the data contingency table as a joint probability distribution between two discrete random variables, and define an information-theoretic co-clustering algorithm that maps rows and columns to row-clusters and column-clusters, respectively. Optimality is defined in terms of mutual information between the clustered random variables. 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.

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

$$I(X;Y) = \sum_x \sum_y p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \quad (1)$$

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) [20]. The smaller the loss, the higher the quality of the co-clustering.

A. Co-clustering based Classification Algorithm (CoCC)

The authors in [17] use co-clustering to perform cross-domain text classification. Since our approach is based on their technique, we summarize here the CoCC algorithm [17].

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 [20], 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 [17], as illustrated in Figure 1 (the initialization step is discussed later). 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 [20], the quality of the co-clustering $(\hat{\mathcal{D}}_o, \hat{\mathcal{W}})$ is measured by the loss in mutual information

$$I(\mathcal{D}_o; \mathcal{W}) - I(\hat{\mathcal{D}}_o; \hat{\mathcal{W}}) \quad (2)$$

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

$$I(\mathcal{C}; \mathcal{W}) - I(\mathcal{C}; \hat{\mathcal{W}}) \quad (3)$$

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 (2) and (3), the objective of co-clustering based classification becomes:

$$\min_{\hat{\mathcal{D}}_o, \hat{\mathcal{W}}} \{I(\mathcal{D}_o; \mathcal{W}) - I(\hat{\mathcal{D}}_o; \hat{\mathcal{W}}) + \lambda(I(\mathcal{C}; \mathcal{W}) - I(\mathcal{C}; \hat{\mathcal{W}}))\} \quad (4)$$

where λ is a trade-off parameter that balances the effect of the two clustering procedures. Equation (4) 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.

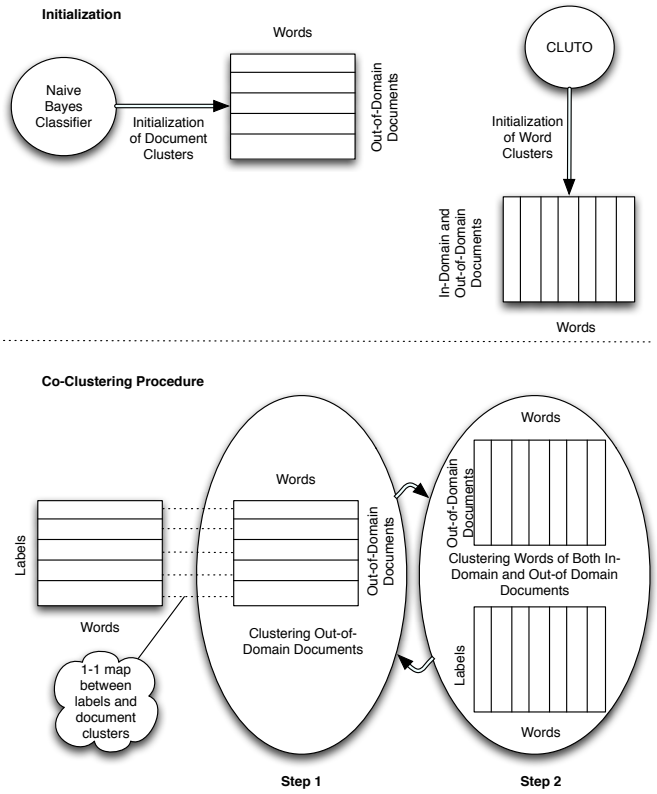

Fig. 1. Co-Clustering for Cross-domain Text Classification

To solve the optimization problem (4), the authors in [17] introduce an iterative procedure aimed at minimizing the divergence between distributions before and after clustering. To see this, lets first consider some definitions. $f(\mathcal{D}_o; \mathcal{W})$ represents the joint probability distribution of $\mathcal{D}_o$ and $\mathcal{W}$. $\hat{f}(\mathcal{D}_o; \mathcal{W})$ represents the joint probability distribution of $\mathcal{D}_o$ and $\mathcal{W}$ under co-clustering $(\hat{\mathcal{D}}_o; \hat{\mathcal{W}})$. Similarly, $g(\mathcal{C}, \mathcal{W})$ denotes the joint probability distribution of $\mathcal{C}$ and $\mathcal{W}$, and $\hat{g}(\mathcal{C}, \mathcal{W})$ denotes the joint probability distribution of $\mathcal{C}$ and $\mathcal{W}$ under the word clustering $\hat{\mathcal{W}}$. The marginal and conditional probability distributions can also be defined. In particular:

$$\hat{f}(d|\hat{w}) = \hat{f}(d|\hat{d})\hat{f}(\hat{d}|\hat{w}) = p(d|\hat{d})p(\hat{d}|\hat{w}) \quad (5)$$

$$\hat{f}(w|\hat{d}) = \hat{f}(w|\hat{w})\hat{f}(\hat{w}|\hat{d}) = p(w|\hat{w})p(\hat{w}|\hat{d}) \quad (6)$$

In [17], the following results are proven.

Lemma 1: For a fixed co-clustering $(\hat{\mathcal{D}}_o; \hat{\mathcal{W}})$, we can write the loss in mutual information as:

$$\begin{aligned} &I(\mathcal{D}_o; \mathcal{W}) - I(\hat{\mathcal{D}}_o; \hat{\mathcal{W}}) + \lambda I(\mathcal{C}; \mathcal{W}) - I(\mathcal{C}; \hat{\mathcal{W}}) \quad (7) \\ &= D(f(\mathcal{D}_o; \mathcal{W})||\hat{f}(\mathcal{D}_o; \mathcal{W})) + \lambda D(g(\mathcal{C}, \mathcal{W})||\hat{g}(\mathcal{C}, \mathcal{W})) \end{aligned}$$

where $D(\cdot||\cdot)$ is the KL-divergence defined as

$$D(p(x)||q(x)) = \sum_x p(x) \log \frac{p(x)}{q(x)}$$

Lemma 2:

$$\begin{aligned} &D(f(\mathcal{D}_o, \mathcal{W})||\hat{f}(\mathcal{D}_o, \mathcal{W})) \\ &= \sum_{\hat{d} \in \hat{\mathcal{D}}_o} \sum_{d \in \hat{d}} f(d) D(f(\mathcal{W}|d)||\hat{f}(\mathcal{W}|\hat{d})) \quad (8) \end{aligned}$$

$$\begin{aligned} &D(f(\mathcal{D}_o, \mathcal{W})||\hat{f}(\mathcal{D}_o, \mathcal{W})) \\ &= \sum_{\hat{w} \in \hat{\mathcal{W}}} \sum_{w \in \hat{w}} f(w) D(f(\mathcal{D}_o|w)||\hat{f}(\mathcal{D}_o|\hat{w})) \quad (9) \end{aligned}$$

Lemma 3:

$$\begin{aligned} &D(g(\mathcal{C}, \mathcal{W})||\hat{g}(\mathcal{C}, \mathcal{W})) \\ &= \sum_{\hat{w} \in \hat{\mathcal{W}}} \sum_{w \in \hat{w}} g(w) D(g((\mathcal{C}|w)||\hat{g}(\mathcal{C}|\hat{w}))) \quad (10) \end{aligned}$$

Lemma 1 states that to solve the optimization problem (4), we can minimize the KL-divergence between f and $\hat{f}$, and the KL-divergence between g and $\hat{g}$. Lemma 2 tells us that the minimization of $D(f(\mathcal{W}|d)||\hat{f}(\mathcal{W}|\hat{d}))$ for a single document d can reduce the value of the objective function of Equation (8). The same conclusion can be derived for the minimization of $D(f(\mathcal{D}_o|w)||\hat{f}(\mathcal{D}_o|\hat{w}))$ for a single word w. Similar conclusions can be derived from Lemma 3. Based on Lemmas 2 and 3, the approach described in Algorithm 1 computes a co-clustering $(\mathcal{C}_{\mathcal{D}_o}, \mathcal{C}_{\mathcal{W}})$ that corresponds to a local minimum of the objective function given in Lemma 1 [17] .

Algorithm 1 The Co-clustering based Classification Algorithm (CoCC) [17]

1: **Input**: in-domain data D_i (labeled); out-of-domain data D_o (unlabeled); a set $\mathcal{C}$ of all class labels; a set $\mathcal{W}$ of all the word features; initial co-clustering $(\mathcal{C}^{(0)}_{\mathcal{D}_o}, \mathcal{C}^{(0)}_{\mathcal{W}})$; the number of iterations T.
2: Initialize the joint distributions f, $\hat{f}$, g and $\hat{g}$
3: **for** $t \leftarrow 1, 3, 5, ..., 2T+1$ **do**
4: Compute the document clusters:

$$\mathcal{C}^{(t)}_{\mathcal{D}_o}(d) = \operatorname{argmin}_{\hat{d}} D(f(\mathcal{W}|d)||\hat{f}^{(t-1)}(\mathcal{W}|\hat{d})) \quad (11)$$

5: Update the probability distribution $\hat{f}^{(t)}$ based on $\mathcal{C}^{(t)}_{\mathcal{D}_o}$, $\mathcal{C}^{(t-1)}_{\mathcal{W}}$. $\mathcal{C}^{(t)}_{\mathcal{W}} = \mathcal{C}^{(t-1)}_{\mathcal{W}}$ and $\hat{g}^{(t)} = \hat{g}^{(t-1)}$.
6: Compute the word clusters:

$$\begin{aligned} \mathcal{C}^{(t+1)}_{\mathcal{W}}(d) = \operatorname{argmin}_{\hat{w}} &f(w) D(f(\mathcal{D}_o|w)||\hat{f}(\mathcal{D}_o|\hat{w})) \\ &+\lambda g(w) D(g((\mathcal{C}|w)||\hat{g}(\mathcal{C}|\hat{w}))) \quad (12) \end{aligned}$$

7: Update the probability distribution $\hat{g}^{(t+1)}$ based on $\mathcal{C}^{(t+1)}_{\mathcal{W}}$. $\mathcal{C}^{(t+1)}_{\mathcal{D}_o} = \mathcal{C}^{(t)}_{\mathcal{D}_o}$ and $\hat{f}^{(t+1)} = \hat{f}^{(t)}$.
8: **end for**
9: **Output**: The partition functions $\mathcal{C}^{(T)}_{\mathcal{D}_o}$ and $\mathcal{C}^{(T)}_{\mathcal{W}}$

The CoCC algorithm requires an initial co-clustering $(\mathcal{C}^{(0)}_{\mathcal{D}_o}, \mathcal{C}^{(0)}_{\mathcal{W}})$ in input. As depicted in Figure 1, in [17] a Naive Bayes classifier is used to initialize the out-of-domain documents into clusters. The initial word clusters are generated using the CLUTO software [23] with default parameters. Once the co-clustering $(\mathcal{C}_{\mathcal{D}_o}, \mathcal{C}_{\mathcal{W}})$ is computed by Algorithm 1,

the class of each document $d \in D_o$ is identified using the following [17]:

$$c = \arg\min_{c \in \mathcal{C}} D(\hat{g}(\mathcal{W}|c)||\hat{f}(\mathcal{W}|\hat{d}))$$

IV. Wikipedia as a Thesaurus

In the following sections, we present the methodology based on Wikipedia to embed semantics into document representation, and our overall approach to cross-domain classification. We start with a description of the fundamental features of the thesaurus built from Wikipedia [31].

Wikipedia (started in 2001) is today the largest encyclopedia in the world. Each article in Wikipedia describes a topic (or concept), and it has a short title, which is a well-formed phrase like a term in a conventional thesaurus [25]. Each article belongs to at least one category, and hyperlinks between articles capture their semantic relations, as defined in the international standard for thesauri [9]. Specifically, the represented semantic relations are: equivalence (*synonymy*), hierarchical (*hyponymy*), and associative.

Wikipedia contains only one article for any given concept (called *preferred term*). *Redirect* hyperlinks exist to group equivalent concepts with the preferred one. Figure 2 shows an example of a redirect link between the synonyms "puma" and "cougar". Besides synomyms, redirect links handle capitalizations, spelling variations, abbreviations, colloquialisms, and scientific terms. For example, "United States" is an entry with a large number of redirect pages: acronyms (U.S.A., U.S., USA, US); Spanish translations (Los Estados, Unidos, Estados Unidos); common mispellings (Untied States); and synonyms (Yankee land) [2].

Disambiguation pages are provided for an ambiguous (or polysemous) concept. A disambiguation page lists all possible meanings associated with the corresponding concept, where each meaning is discussed in an article. For example, the disambiguation page of the term "puma" lists 22 associated concepts, including animals, cars, and a sportswear brand.

Each article (or concept) in Wikipedia belongs to at least one category, and categories are nested in a hierarchical organization. Figure 2 shows a fragment of such structure. The resulting hierarchy is a directed acyclic graph, where multiple categorization schemes co-exist [25].

Associative hyperlinks exist between articles. Some are one-way links, others are two-way. They capture different degrees of relatedness. For example, a two-way link exists between the concepts "puma" and "cougar", and a one-way link connects "cougar" to "South America". While the first link captures a close relationship between the terms, the second one represents a much weaker relation. (Note that one-way links establishing strong connections also exist, e.g., from "Data Mining" to "Machine Learning".) Thus, meaningful measures need to be considered to properly rank associative links between articles. Three such measures have been introduced in [15]: *Content-based*, *Out-link category-based*, and *Distance-based*. We briefly describe them here. In Section V-B we use them to define the proximity between associative concepts.

The content-based measure is based on the bag-of-words representation of Wikipedia articles. Each article is modeled as a *tf-idf* vector; the associative relation between two articles is then measured by computing the cosine similarity between the corresponding vectors. Clearly, this measure (denoted as S_{BOW}) has the same limitations of the BOW approach.

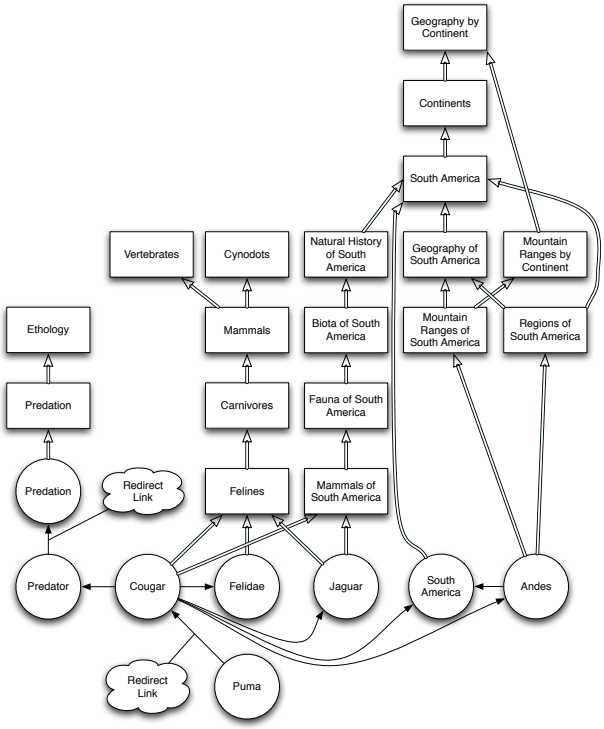

Fig. 2. A fragment of Wikipedia's taxonomy

The out-link category-based measure compares the out-link categories of two associative articles. The out-link categories of a given article are the categories to which out-link articles from the original one belong. Figure 3 shows (a fraction of) the out-link categories of the associative concepts "Data Mining", "Machine Learning", and "Computer Network". The concepts "Data Mining" and "Machine Learning" share 22 out-link categories; "Data Mining" and "Computer Network" share 10; "Machine Learning" and "Computer Network" share again the same 10 categories. The larger the number of shared categories, the stronger the associative relation between the articles. To capture this notion of similarity, articles are represented as vectors of out-link categories, where each component corresponds to a category, and the value of the i-th component is the number of out-link articles which belong to the i-th category. The cosine similarity is then computed between the resulting vectors, and denoted as S_{OLC}. The computation of S_{OLC} for the concepts illustrated in Figure 3 gives the following values, which indeed reflect the actual semantic of the corresponding terms: $S_{OLC}(\textit{Data Mining}, \textit{Machine Learning}) = 0.656$, $S_{OLC}(\textit{Data Mining}, \textit{Computer Network}) = 0.213$, $S_{OLC}(\textit{Machine Learning}, \textit{Computer Network}) = 0.157$.

The third measure is a distance measure (rather then a similarity measure like the first two). The distance between two articles is measured as the length of the shortest path connecting the two categories they belong to, in the acyclic graph of the category taxonomy. The distance measure is

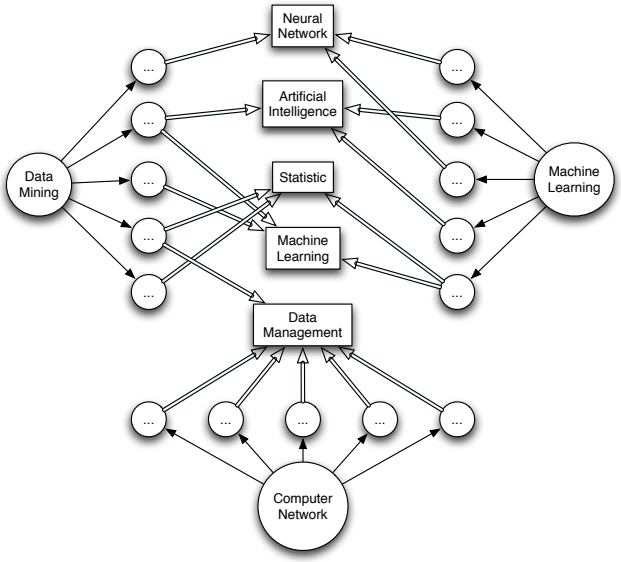

Fig. 3. Out-link categories of the concepts "Machine Learning", "Data Mining", and "Computer Network"

normalized by taking into account the depth of the taxonomy. It is denoted as D_{cat}.

A linear combination of the three measures allows to quantify the overall strength of an associative relation between concepts:

$$S_{overall} = \lambda_1 S_{BOW} + \lambda_2 S_{OLC} + (1-\lambda_1-\lambda_2)(1-D_{cat}) \quad (13)$$

where $\lambda_1, \lambda_2 \in (0,1)$ are parameters to weigh the individual measures. Equation (13) allows to rank all the associative articles linked to any given concept.

V. Concept-based Kernels

As mentioned before, the "Bag of Words" (BOW) approach breaks multi-word expressions, maps synonymous words into different components, and treats polysemous as one single component. Here, we overcome the shortages of the BOW approach by embedding background knowledge into a semantic kernel, which is then used to enrich the representation of documents.

In the following, we first describe how to enrich text documents with semantic kernels, and then illustrate our technique for building semantic kernels using background knowledge constructed from Wikipedia.

A. Kernel Methods for Text

The BOW model (also called Vector Space Model, or VSM) [29] of a document d is defined as follows:

$$\phi : d \mapsto \phi(d) = (tf(t_1,d), tf(t_2,d), \ldots, tf(t_D,d)) \in \mathcal{R}^D$$

where $tf(t_i,d)$ is the frequency of term t_i in document d, and D is the size of the dictionary.

The basic idea of kernel methods is to embed the data in a suitable feature space, such that solving the problem (e.g., classification or clustering) in the new space is easier (e.g., linear). A kernel represents the similarity between two objects (e.g., documents or terms), defined as dot-product in this new vector space. The kernel trick [12] allows to keep the mapping implicit. In other words, it is only required to know the inner products between the images of the data items in the original space. Therefore, defining a suitable kernel means finding a good representation of the data objects.

TABLE I
Example of document term vectors

	Puma	Cougar	Feline	...
d_1	2	0	0	...
d_2	0	1	0	...

In text classification, semantically similar documents should be mapped to nearby positions in feature space. In order to address the omission of semantic content of the words in VSM, a transformation of the document vector of the type $\tilde{\phi}(d) = \phi(d)S$ is required, where S is a semantic matrix. Different choices of the matrix S lead to different variants of VSM. Using this transformation, the corresponding vector space kernel takes the form

$$\begin{aligned} \tilde{k}(d_1,d_2) &= \phi(d_1)SS^\top\phi(d_2)^\top \\ &= \tilde{\phi}(d_1)\tilde{\phi}(d_2)^\top \end{aligned} \quad (14)$$

Thus, the inner product between two documents d_1 and d_2 in feature space can be computed efficiently directly from the original data items using a kernel function.

The semantic matrix S can be created as a composition of embeddings, which add refinements to the semantics of the representation. Therefore, S can be defined as:

$$S = RP \quad (15)$$

where R is a diagonal matrix containing the term weightings or relevance, and P is a *proximity matrix* defining the semantic similarities between the different terms of the corpus. One simple way of defining the term weighting matrix R is to use the inverse document frequency (*idf*).

P has non-zero off diagonal entries, $P_{ij} > 0$, when the term i is semantically related to the term j. Embedding P in the vector space kernel corresponds to representing a document as a less sparse vector, $\phi(d)P$, which has non-zero entries for all terms that are semantically similar to those present in document d. There are different methods for obtaining P [32], [1]. Here, we leverage the external knowledge provided by Wikipedia.

Given the thesaurus built from Wikipedia, it is straightforward to build a proximity (or similarity) matrix P. Here is a simple example. Suppose the corpus contains one document d_1 that talks about pumas (the animal). A second document d_2 discusses the life of cougars. d_1 contains instances of the word "puma", but no occurrences of "cougar". Vice versa, d_2 containts the word "cougar", but "puma" does not appear in d_2. Fragments of the BOW representations of d_1 and d_2 are given in Table I, where the feature values are term frequencies. The two vectors may not share any features (e.g., neither document contains the word "feline"). Table II shows a fragment of

TABLE II
EXAMPLE OF A PROXIMITY MATRIX

...	Puma	Cougar	Feline	...
Puma	1	1	0.4	...
Cougar	1	1	0.4	...
Feline	0.4	0.4	1	...
...				...

TABLE III
EXAMPLE OF "ENRICHED" TERM VECTORS

	Puma	Cougar	Feline	...
d_1'	2	2	0.8	...
d_2'	1	1	0.4	...

a proximity matrix computed from the thesaurus based on Wikipedia. The similarity between "puma" and "cougar" is one since the two terms are synonyms. The similarity between "puma" and "feline" (or "cougar" and "feline") is 0.4, as computed according to equation (13). Table III illustrates the updated term vectors of documents d_1 and d_2, obtained by multipling the original term vectors (Table I) with the proximity matrix of Table II. The new vectors are less sparse, with non-zero entries not only for terms included in the original document, but also for terms semantically related to those present in the document. This enriched representation brings documents which are semantically related closer to each other, and therefore it facilitates the categorization of documents based on their content. We now discuss the enrichment steps in detail.

B. Semantic Kernels derived from Wikipedia

The thesaurus derived from Wikipedia provides a list of concepts. For each document in a given corpus, we search for the Wikipedia concepts mentioned in the document. Such concepts are called *candidate concepts* for the corresponding document. When searching for candidate concepts, we adopt an exact matching strategy, by which only the concepts that explicitly appear in a document become the candidate concepts. (If an m-gram concept is contained in an n-gram concept (with $n > m$), only the last one becomes a candidate concept.) We then construct a vector representation of a document, which contains two parts: terms and candidate concepts. For example, consider the text fragment "Machine Learning, Statistical Learning, and Data Mining are related subjects". Table IV shows the traditional BOW term vector for this text fragment (after stemming), where feature values correspond to term frequencies. Table V shows the new vector representation, where boldface entries are candidate concepts, and non-boldface entries correspond to terms.

We observe that, for each document, if a word only appears in candidate concepts, it won't be chosen as a term feature any longer. For example, in the text fragment given above, the word "learning" only appears in the candidate concepts "Machine Learning" and "Statistical Learning". Therefore, it doesn't appear as a term in Table V. On the other hand, according to the traditional BOW approach, after stemming, the term "learn" becomes an entry of the term vector (Table IV).

TABLE IV
TRADITIONAL BOW TERM VECTOR

Entry	*tf*
machine	1
learn	2
statistic	1
data	1
mine	1
relate	1
subject	1

TABLE V
VECTOR OF CANDIDATE CONCEPTS AND TERMS

Entry	*tf*
machine learning	1
statistical learning	1
data mining	1
relate	1
subject	1

Furthermore, as illustrated in Table V, we keep each candidate concept as it is, without performing stemming or splitting multi-word expressions, since multi-word candidate concepts carry meanings that cannot be captured by the individual terms.

When generating the concept-based vector representation of documents, special care needs to be given to polysemous concepts, i.e., concepts that have multiple meanings. It is necessary to perform word sense disambiguation to find the specific meaning of ambiguos concepts within the corresponding document. For instance, the concept "puma" is an ambiguous one. If "puma" is mentioned in a document, its actual meaning in the document should be identified, i.e., whether it refers to a kind of animal, or to a sportswear brand, or to something else. In Section V-C we explain how we address this issue.

Once the candidate concepts have been identified, we use the Wikipedia thesaurus to select synonyms, hyponyms, and associative concepts of the candidate ones. The vector associated to a document d is then enriched to include such related concepts: $\phi(d) = (<$*terms*$>,<$*candidate concepts*$>,<$*related concepts*$>)$. The value of each component corresponds to a *tf-idf* value. The feature value associated to a related concept (which does not appear explicitely in any document of the corpus) is the *tf-idf* value of the corresponding candidate concept in the document. Note that this definition of $\phi(d)$ already embeds the matrix R as defined in equation (15).

We can now define a proximity matrix P for each pair of concepts (candidate and related). The matrix P is represented in Table VI. For mathematical convenience, we also include the terms in P. P is a symmetrical matrix whose elements are defined as follows. For any two terms t_i and t_j, $P_{ij} = 0$ if $i \neq j$; $P_{ij} = 1$ if $i = j$. For any term t_i and any concept c_j, $P_{ij} = 0$. For any two concepts c_i and c_j:

$$P_{ij} = \begin{cases} 1 & \text{if } c_i \text{ and } c_j \text{ are synonyms;} \\ \mu^{-depth} & \text{if } c_i \text{ and } c_j \text{ are hyponyms;} \\ S_{overall} & \text{if } c_i \text{ and } c_j \text{ are associative concepts;} \\ 0 & \text{otherwise.} \end{cases}$$

TABLE VI
PROXIMITY MATRIX

	Terms				Concepts			
Terms	1	0	$\cdots$	0	0	0	$\cdots$	0
	0	1	$\cdots$	0	0	0	$\cdots$	0
	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
	0	0	$\cdots$	1	0	0	$\cdots$	0
Concepts	0	0	$\cdots$	0	1	a	$\cdots$	b
	0	0	$\cdots$	0	a	1	$\cdots$	c
	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
	0	0	$\cdots$	0	b	c	$\cdots$	1

TABLE VII
COSINE SIMILARITY BETWEEN THE REUTERS DOCUMENT #9 AND THE WIKIPEDIA'S ARTICLES CORRESPONDING TO THE DIFFERENT MEANINGS OF THE TERM "STOCK"

Meanings of "Stock"	*Similarity with Reuters #9*
Stock (finance)	**0.2037**
Stock (food)	0.1977
Stock (cards)	0.1531
Stocks (plants)	0.1382
Stock (firearm)	0.0686
Livestock	0.0411
Inventory	0.0343

$S_{overall}$ is computed according to equation (13). *depth* represents the distance between the corresponding categories of two hyponym concepts in the category structure of Wikipedia. For example, suppose c_i belongs to category A and c_j to category B. If A is a direct subcategory of B, then $depth = 1$. If A is a direct subcategory of C, and C is a direct subcategory of B, then $depth = 2$. μ is a back-off factor, which regulates how fast the proximity between two concepts decreases as their category distance increases. (In our experiments, we set $\mu = 2$.)

By composing the vector $\phi(d)$ with the proximity matrix P, we obtain our extended vector space model for document d: $\tilde{\phi}(d) = \phi(d)P$. $\tilde{\phi}(d)$ is a less sparse vector with non-zero entries for all concepts that are semantically similar to those present in d. The strength of the value associated with a related concept depends on the number and frequency of occurrence of candidate concepts with a close meaning. An example of this effect can be observed in Table III. Let us assume that the concept "feline" is a related concept (i.e., did not appear originally in any of the given documents). "feline" appears in document d'_1 with strength 0.8, since the original document d_1 contains two occurrences of the synonym concept "puma" (see Table I), while it appears in d'_2 with a smaller strength (0.4), since the original document d_2 contains only one occurrence of the synonym concept "cougar" (see Table I). The overall process, from building the thesaurus from Wikipedia, to constructing the proximity matrix and enriching documents with concepts, is depicted in Figure 4.

C. Disambiguation of Concept Senses

If a candidate concept is polysemous, i.e. it has multiple meanings, it is necessary to perform word sense disambiguation to find its most proper meaning in the context where

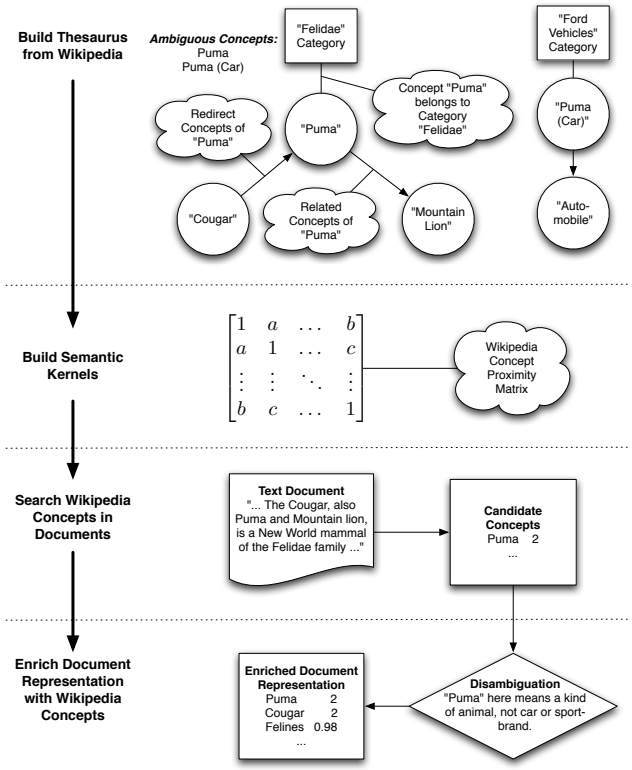

Fig. 4. The process that derives semantic kernels from Wikipedia

it appears, prior to calculating its proximity to other related concepts. We utilize text similarity to do explicit word sense disambiguation. This method computes document similarity by measuring the overlapping of terms. For instance, the Reuters-21578 document #9 [3] talks about stock splits, and the concept "stock" in Wikipedia refers to several different meanings, as listed in Table VII. The correct meaning of a polysemous concept is determined by comparing the cosine similarities between the *tf-idf* term vector of the text document (where the concept appears), and each of Wikipedia's articles (corresponding *tf-idf* vectors) describing the different meanings of the polysemous concept. The larger the cosine similarity between two *tf-idf* term vectors is, the higher the similarity between the two corresponding text documents. Thus, the meaning described by the article with the largest cosine similarity is considered to be the most appropriate one. From Table VII, the Wikipedia article describing "stock" (finance) has the largest similarity with the Reuters document #9, and this is indeed confirmed to be the case by manual examination of the document (document #9 belongs to the Reuters category "earn").

As mentioned above, document #9 discusses the stock split of a company, and belongs to the Reuters category "earn". The document contains several candidate concepts, such as "stock", "shareholder", and "board of directors". Table VIII gives an example of the corresponding related concepts identified by our method, and added to the vector representation of document #9 of the Reuters data set [30].

TABLE VIII
THE HYPONYM, ASSOCIATIVE, AND SYNONYM CONCEPTS INTRODUCED IN REUTERS DOCUMENT #9

Candidate Concepts	Hyponyms	Associative Concepts	Synonyms
Stock	Stock market Equity securities Corporate finance	House stock Bucket shop Treasury stock Stock exchange Market capitalization	Stock (finance)
Shareholder	Stock market	Board of directors Business organizations Corporation Fiduciary Stock	Shareholders
Board of directors	Business law Corporate governance Corporations law Management	Chief executive officer Shareholder Fiduciary Corporate governance Corporation	Boards of directors

VI. SEMANTIC-BASED CROSS-DOMAIN TEXT CLASSIFICATION

We apply the enriching procedure described in Section IV to all in-domain documents D_i and all out-of-domain documents D_o to perform cross-domain text classification. As a result, the representation of two related documents d_1 and d_2, such that $d_1 \in D_i$ and $d_2 \in D_o$, corresponds to two close vectors $\tilde{\phi}(d_1)$ and $\tilde{\phi}(d_2)$ in the extended vector space model. In other words, the extended vector space model applied to D_i and D_o has the effect of enriching the shared dictionary with concepts that encapsulate the content of documents. As such, related domains will have a shared pool of terms/concepts of increased size that has the effect of making explicit their semantic relationships.

We thus perform co-clustering based cross-domain classification by providing the CoCC algorithm (Algorithm 1) the extended vector space model of in-domain and out-of-domain documents. The set $\mathcal{W}$ now comprises the new dictionary, which includes terms and concepts (both candidate and related). We emphasize that concepts constitute individual features, without undergoing stemming, or splitting of multi-word expressions.

VII. EMPIRICAL EVALUATION

To evaluate the performance of our approach, we conducted several experiments using real data sets. We test scenarios for both binary and multiple category classification.

A. Processing Wikipedia XML data

The evaluation was performed using the Wikipedia XML Corpus [6]. The Wikipedia XML Corpus contains processed Wikipedia data parsed into an XML format. Each XML file corresponds to an article in Wikipedia, and maintains the original ID, title and content of the corresponding Wikipedia article. Furthermore, each XML file keeps track of the linked article ID, for every redirect link and hyperlink contained in the original Wikipedia article.

We do not include all concepts of Wikipedia in the thesaurus. Some concepts, such as "List of ISO standards", "1960s", and so on, do not contribute to the achievement of improved discrimination among documents. Thus, before building the thesaurus from Wikipedia, we remove concepts deemed not useful. To this end, we implement a few heuristics as explained below.

TABLE IX
NUMBER OF TERMS, CONCEPTS, AND LINKS AFTER FILTERING

Terms in Wikipedia XML corpus	**659,388**
Concept After Filtering	495,214
Redirected Concepts	413
Categories	113,484
Relations in Wikipedia XML corpus	**15,206,174**
Category to Subcategory	145,468
Category to Concept	1,447,347
Concept to Concept	13,613,359

First, all concepts of Wikipedia which belong to categories related to chronology, such as "Years", "Decades", and "Centuries", are removed. Second, we analyze the titles of Wikipedia articles to decide whether they correspond to useful concepts. In particular, we implement the following rules:

1) If the title of an article is a multi-word title, we check the capitalization of all the words other than prepositions, determiners, conjunctions, and negations. If all the words are capitalized, we keep the article.
2) If the title is one word title, and it occurs in the article more than three times [2], we keep the article.
3) Otherwise, the article is discarded.

After filtering Wikipedia concepts using these rules, we obtained about 500,000 concepts to be included in the thesaurus. Table IX provides a break down of the resulting number of elements (terms, concepts, and links) used to build the thesaurus, and therefore our semantic kernels. In particular, we note the limited number of redirected concepts (413). This is due to the fact that redirect links in Wikipedia often refers to the plural version of a concept, or to misspellings of a concept, and they are filtered out in the XML Corpus. Such variations of a concept, in fact, should not be added to the documents, as they would contribute only noise. For example, in Table VIII, the synonyms associated to the candidate concepts "Shareholder" and "Board of visitors" correspond to their plural versions. Thus, in practice they are not added to the documents.

B. Data Sets

We evaluated our approach using the 20 Newsgroups [11], and the SRAA [24] data sets. We split the original data in two corpora, corresponding to in-domain and out-of-domain documents. Different but related categories are selected for the two domains. Data sets across different classes are balanced.

20 Newsgroups. The 20 Newsgroups [11] data set is a popular collection of approximately 20,000 newsgroup documents, partitioned nearly evenly across 20 different newsgroups (about 1,000 per class).

We generated ten different data sets comprised of different combinations of categories. Each data set contains several top categories, which also define the class labels. Data are split into two domains based on their sub-categories. For example, one top category (i.e., class label) considered is "recreation"; the in-domain documents of this class talk about "autos" and "motorcycles", while the out-of-domain documents of the same class are concerned with "baseball" and "hockey" (they belong to the sub-category "sport"). This setting assures that documents in D_i and in D_o belong to different but related domains. Table X shows how categories were distributed for each data set generated from the 20 Newsgroups corpus. The setting of the six data sets for binary classification is the same as in [17].

SRAA. The SRAA [24] data set contains 73,218 articles from four discussion groups on simulated auto racing, simulated aviation, real autos, and real aviation. It is often used for binary classification, where the task can be defined as the separation of documents on "real" versus "simulated" topics, or as the separation of documents on "auto" vs. documents on "aviation". We generated two binary classification problems accordingly, as specified in Table XI.

TABLE X
SPLITTING OF 20 NEWSGROUPS CATEGORIES FOR CROSS-DOMAIN CLASSIFICATION

	Data Set	D_i	D_o
2 Categories	comp vs sci	comp.graphics comp.os.ms-windows.misc sci.crypt sci.electronics	comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x sci.med sci.space
	rec vs talk	rec.autos rec.motorcycles talk.politics.guns talk.politics.misc	rec.sport.baseball rec.sport.hockey talk.politics.mideast talk.religion.misc
	rec vs sci	rec.autos rec.sport.baseball sci.med sci.space	rec.motorcycles rec.sport.hockey sci.crypt sci.electronics
	sci vs talk	sci.electronics sci.med talk.politics.misc talk.religion.misc	sci.crypt sci.space talk.politics.guns talk.politics.mideast
	comp vs rec	rec.autos rec.sport.baseball comp.graphics comp.sys.ibm.pc.hardware comp.sys.mac.hardware	rec.motorcycles rec.sport.hockey comp.os.ms-windows.misc comp.windows.x
	comp vs talk	talk.politics.guns talk.politics.misc comp.graphics comp.sys.mac.hardware comp.windows.x	talk.politics.mideast talk.religion.misc comp.os.ms-windows.misc comp.sys.ibm.pc.hardware
3 Categories	rec vs sci vs comp	rec.motorcycles rec.sport.hockey sci.med sci.space comp.graphics comp.sys.ibm.pc.hardware comp.sys.mac.hardware	rec.autos rec.sport.baseball sci.crypt sci.electronics comp.os.ms-windows.misc comp.windows.x
	rec vs talk vs sci	rec.autos rec.motorcycles talk.politics.guns talk.politics.misc sci.med sci.space	rec.sport.baseball rec.sport.hockey talk.politics.mideast talk.religion.misc sci.crypt sci.electronics
	sci vs talk vs comp	sci.crypt sci.electronics talk.politics.mideast talk.religion.misc comp.graphics comp.sys.mac.hardware comp.windows.x	sci.space sci.med talk.politics.misc talk.politics.guns comp.os.ms-windows.misc comp.sys.ibm.pc.hardware
4 Categories	sci vs rec vs talk vs comp	sci.crypt sci.electronics rec.autos rec.motorcycles talk.politics.mideast talk.religion.misc comp.graphics comp.os.ms-windows.misc	sci.space sci.med rec.sport.baseball rec.sport.hockey talk.politics.misc talk.politics.guns comp.sys.mac.hardware comp.sys.ibm.pc.hardware comp.windows.x

TABLE XI
SPLITTING OF SRAA CATEGORIES FOR CROSS-DOMAIN CLASSIFICATION

Data Set	D_i	D_o
auto vs aviation	sim-auto & sim-aviation	real-auto & real-aviation
real vs simulated	real-aviation & sim-aviation	real-auto & sim-auto

C. Methods

In our experiments, we compare the classification results of the CoCC approach based on the BOW representation of documents, and of the CoCC approach based on the extended vector space model. We denote the first technique as CoCC *without enrichment*, and the second one as CoCC *with enrichment*.

The CoCC algorithm uses a Naive Bayes classifier to initialize the out-of-domain documents into clusters. Thus, we also report the results of the Naive Bayes classifiers, with and without enrichment, respectively.

D. Implementation Details

Standard pre-processing was performed on the raw data. Specifically, all letters in the text were converted to lower case, stop words were eliminated, and stemming was performed using the Porter algorithm [27] (candidate and related concepts, though, are identified prior to stemming, and kept unstemmed). Words that appeared in less than three documents were eliminated from consideration. Term Frequency (TF) was used for feature weighting when training the Naive Bayes classifier, and for the co-clustering based classification (CoCC) algorithm.

To compute the enriched representation of documents, we need to set the parameters λ_1 and λ_2 in Equation (13). These parameters were tuned according to the methodology suggested in [31]. As a result, the values $\lambda_1 = 0.4$ and $\lambda_2 = 0.5$ were used in our experiments.

The co-clustering based classification algorithm requires the initialization of document clusters and word clusters. As mentioned earlier, here we follow the methodology adopted in [17], and compute the initial document clusters using a Naive Bayes classifier, and the initial word clusters using the CLUTO software [23] with default parameters. The Naive

Bayes classifier is trained using D_i. The trained classifier is then used to predict the labels of documents in D_o. In our implementation, we keep track of class labels associated to clusters by the Naive Bayes classifier, to compute the final labels of documents in D_o.

The following implementation issue is worth a mention here. We observe that some words may only appear in D_i (or D_o). For such a word, and for a document $d \in D_o$ ($d \in D_i$, respectively), the estimation of $p(w|d)$ is zero. Furthermore, if all words w in a word cluster $\hat{w}$ only appear in D_i, since the CoCC algorithm only clusters documents $d \in D_o$, the estimation of $p(\hat{w}|\hat{d})$ becomes zero as well.

According to Equation (6), if $p(\hat{w}|\hat{d}) = 0$, then $\hat{f}(w|\hat{d})$ will also be zero. As a consequence, $D(f(W|d)||\hat{f}(W|\hat{d})) = \sum_{w \in W} f(w|d) \log \frac{f(w|d)}{\hat{f}(w|\hat{d})}$ becomes unbounded. In order to avoid this, in Equation (11), when $\hat{f}(w|\hat{d}) = 0$, Laplacian smoothing [26] is applied to estimate the probabilities. We proceed similarly for the computation of $D(f(D_o|w)||\hat{f}(D_o|\hat{w}))$ and $D(g(C|w)||\hat{g}(C|\hat{w}))$ in Equation (12).

E. Results

Table XII presents the precision rates obtained with Naive Bayes and the CoCC algorithm, both with and without enrichment, for all data sets considered. The results of the CoCC algorithm corresponds to $\lambda = 0.25$, and 128 word clusters. The precision values are those obtained after the fifth iteration. In the following, we study the sensitivity of our approach with respect to the number of iterations, the value of λ, and the number of clusters.

From Table XII, we can see that the CoCC algorithm with enrichment provides the best precision values for all data sets. For each data set, the improvement offered by CoCC with enrichment with respect to the Naive Bayes classifier (with enrichment), and with respect to CoCC without enrichment is quite significant. These results clearly demonstrate the efficacy of a semantic-based approach to cross-domain classification.

As shown in Table XII, the most difficult problem appears to be the one with four categories, derived from the 20 Newsgroups data set: rec vs talk vs sci vs comp. A closer look to the precision rates obtained for each category reveals that almost all documents of classes "recreation" and "talk" in D_o are correctly classified. The misclassification error is mostly due to the fact that the top categories "science" and "computers" are closely related to each other (in particular, the sub-category "electronics" of "science" may share many words with the category "computers"). As a consequence, several "science" documents are classified as "computers" documents. Nevertheless, CoCC with enrichment achieves 71.3% accuracy, offering a 8.9% improvement with respect to CoCC without enrichment, and a 17.5% improvement with respect to Naive Bayes. It is interesting to observe that in all cases the Naive Bayes classifier itself largely benefits from the enrichment process.

The authors in [17] have proven the convergence of the CoCC algorithm. Here, we show the precision achieved by CoCC with enrichment as a function of the number of iterations for the four multi-category problems considered in our

TABLE XII
CROSS-DOMAIN CLASSIFICATION PRECISION RATES

Data Set	w/o enrichment		w/ enrichment	
	NB	CoCC	NB	CoCC
rec vs talk	0.824	0.921	0.853	0.998
rec vs sci	0.809	0.954	0.828	0.984
comp vs talk	0.927	0.978	0.934	0.995
comp vs sci	0.552	0.898	0.673	0.987
comp vs rec	0.817	0.915	0.825	0.993
sci vs talk	0.804	0.947	0.877	0.988
rec vs sci vs comp	0.584	0.822	0.635	0.904
rec vs talk vs sci	0.687	0.881	0.739	0.979
sci vs talk vs comp	0.695	0.836	0.775	0.912
rec vs talk vs sci vs comp	0.487	0.624	0.538	0.713
real vs simulation	0.753	0.851	0.826	0.977
auto vs aviation	0.824	0.959	0.933	0.992

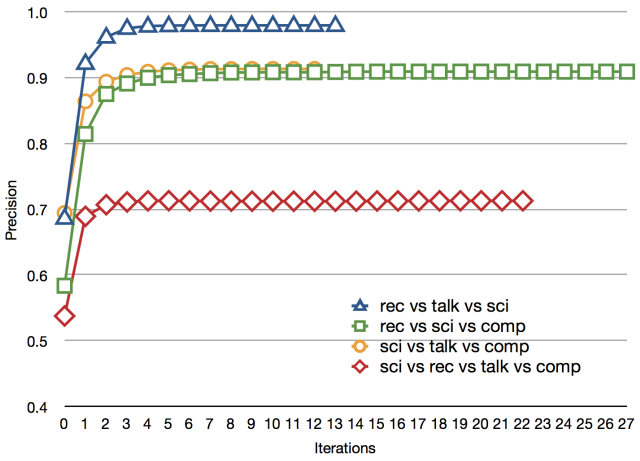

Fig. 5. CoCC with enrichment: Precision as a function of the number of iterations

experiments (see Figure 5). In each case, the algorithm reached convergence after a reasonable number of iterations (at most 27 iterations for the four data sets considered in Figure 5). The improvement in precision with respect to the initial clustering solution are confined within the first few iterations. During the subsequent iterations, the precision remains stable. We obtained a consistent result across all data sets. For this reason, in Table XII we provide the precision results obtained after the fifth iteration.

We also tested the sensitivity of CoCC with enrichment with respect to the λ parameter of Equation (4), and with respect to the number of clusters. We report the results obtained on the three category problem derived from the 20 Newsgroups data set: sci vs talk vs comp. Following the settings in [17], we used λ values in the range (0.03125, 8), with three different numbers of word clusters: 16, 64 and 128. Figure 6 shows the results. Overall, the precision values are quite stable. A reasonable range of values for λ is [0.25, 0.5].

The precision values as a function of different number of clusters are given in Figure 7. We tested different numbers of clusters between 2 and 512 for three different values of λ: 0.125, 0.25, and 1.0. As Figure 7 shows, the same trend was obtained for the three λ values. Precision increases significantly until a reasonable number of word clusters is achieved (too few word clusters do not allow discrimination

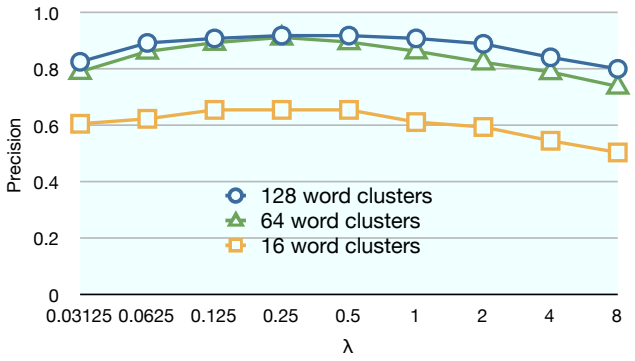

Fig. 6. CoCC with enrichment: Precision as a function of λ

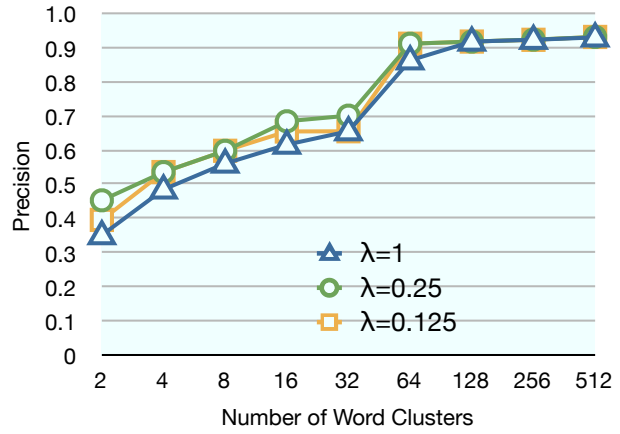

Fig. 7. CoCC with enrichment: Precision as a function of the number of word clusters

across classes). A value of 128 provided good results for all problems considered here (this finding is consistent with the analysis conducted in [17]).

VIII. Conclusions and Future Work

We extended the co-clustering approach to perform cross-domain classification by embedding background knowledge constructed from Wikipedia. In particular, we combine the CoCC algorithm with an enriched representation of documents, which allows to build a semantic bridge between related domains, and thus achieve high accuracy in cross-domain classification. The experimental results presented demonstrate the efficacy of a semantic-based approach to cross-domain classification.

The words shared between related domains play a key role to enable the migration of label information, and thus fulfill classification in the target domain. In our future work, we plan to explore alternate methodologies to leverage and organize the common language substrate of the given domains. We also plan to extend our approach to perform cross-language text classification, an interesting problem with difficult challenges.

Acknowledgment

This work was in part supported by NSF CAREER Award IIS-0447814.

References

[1] L. AlSumait and C. Domeniconi. Local Semantic Kernels for Text Document Clustering. In *Workshop on Text Mining, SIAM International Conference on Data Mining*, Minneapolis, MN, 2007. SIAM.

[2] R. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. In *Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy, 2006.

[3] Carnegie Group, Inc. and Reuters, Ltd. *Reuters-21578 text categorization test collection*, 1997.

[4] K. Dave, S. Lawrence, and D. M. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *International World Wide Web Conference*, Budapest, Hungary, 2003.

[5] M. de Buenega Rodriguez, J. M. Gomez-Hidalgo, and B. Diaz-Agudo. Using wordnet to complement training information in text categorization. In *International Conference on Recent Advances in Natural Language Processing*, 1997.

[6] L. Denoyer and P. Gallinari. The Wikipedia XML Corpus. *SIGIR Forum*, 2006.

[7] E. Gabrilovich and S. Markovitch. Feature generation for text categorization using world knowledge. In *International Joint Conference on Artificial Intelligence*, Edinburgh, Scotland, 2005.

[8] E. Gabrilovich and S. Markovitch. Overcoming the brittleness bottleneck using wikipedia: enhancing text categorization with encyclopedic knowledge. In *National Conference on Artificial Intelligence (AAAI)*, Boston, Massachusetts, 2006.

[9] A. Hotho, S. Staab, and G. Stumme. Wordnet improves text document clustering. In *Semantic Web Workshop, SIGIR Conference*, Toronto, Canada, 2003. ACM.

[10] L. Jing, L. Zhou, M. K. Ng, and J. Z. Huang. Ontology-based distance measure for text clustering. In *Workshop on Text Mining, SIAM International Conference on Data Mining*, Bethesda, MD, 2006. SIAM.

[11] K. Lang. Newsweeder: Learning to filter netnews. In *International Conference on Machine Learning*, Tahoe City, California, 1995. Morgan Kaufmann.

[12] J. Shawe-Taylor and N. Cristianini. *Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[13] G. Siolas and F. d'Alché Buc. Support vector machines based on a semantic kernel for text categorization. In *International Joint Conference on Neural Networks (IJCNN'00)*, pages 205–209, Como, Italy, 2000. IEEE.

[14] L. A. Urena-Lopez, M. Buenaga, and J. M. Gomez. Integrating linguistic resources in TC through WSD. *Computers and the Humanities*, 35:215–230, 2001.

[15] P. Wang, J. Hu, H.-J. Zeng, L. Chen, and Z. Chen. Improving text classification by using encyclopedia knowledge. In *International Conference on Data Mining*, pages 332–341, Omaha, NE, 2007. IEEE.

[16] S. Banerjee. Boosting inductive transfer for text classification using wikipedia. In *International Conference on Machine Learning and Applications (ICMLA-2007)*, 2007.

[17] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu. Co-clustering based classification for out-of-domain documents. In *International Conference on Knowledge Discovery and Data Mining (KDD-2007)*, 2007.

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

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

[20] 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.

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

[22] G. Forman. Tackling concept drift by temporal inductive transfer. In *Annual ACM Conference on Research and Development in Information Retrieval (SIGIR-2006)*, 2006.

[23] G. Karypis. Cluto software for clustering high-dimensional datasets.

[24] A. K. McCallum Simulated/real/aviation/auto usenet data.

[25] D. Milne, O. Medelyan, and I. H. Witten. Mining domain-specific thesauri from wikipedia: A case study. In *International Conference on Web Intelligence*, 2006.

[26] T. M. Mitchell. Machine learning. In *McGraw Hill*, 1997.
[27] M. F. Porter An algorithm for suffix stripping *Program*, 14(3): 130–137, 1980.
[28] R. Raina, A. Y. Ng, and D. Koller. Constructing informative priors using transfer learning. In *International Conference on Machine Learning (ICML-2006)*, 2006.
[29] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.
[30] P. Wang and C. Domeniconi. Building semantic kernels for text classification using wikipedia. In *Submit to International Conference on Knowledge Discovery and Data Mining (KDD-2008)*, 2008.
[31] P. Wang, J. Hu, H.-J. Zeng, L. Chen, and Z. Chen. Improving text classification by using encyclopedia knowledge. In *International Conference on Data Mining (ICDM-2007)*, 2007.
[32] S. K. M. Wong, W. Ziarko, and P. C. N. Wong. Generalized vector space model in information retrieval. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, 1985.

Pu Wang received a B.E. degree from Beihang University, Beijing, China, in 2004, and an M.S. degree from Peking University, Beijing, China, in 2007. From 2005 to 2007, he was an intern in the Machine Learning Group at Microsoft Research Asia, Beijing, China. He is currently a Ph.D. student in the Department of Computer Science at George Mason University, USA. His research interests focus on machine learning and data mining.

Carlotta Domeniconi received the Laurea degree in computer science from the University of Milano, Milan, Italy, in 1992, the M.S. degree in information and communication technologies from the International Institute for Advanced Scientic Studies, Salerno, Italy, in 1997, and the Ph.D. degree in computer science from the University of California, Riverside, in 2002. She is currently an Associate Professor in the Department of Computer Science, George Mason University, Fairfax, VA. Her research interests include machine learning, pattern recognition, data mining, and feature relevance estimation, with applications in text mining and bioinformatics. Dr. Domeniconi is a recipient of a 2004 Ralph E. Powe Junior Faculty Enhancement Award, and an NSF CAREER Award.

Jian Hu is currently an Assistant Researcher at Microsoft Research Asia, Beijing, China. He received Master's and Bachelor degrees from the Department of Computer Science and Technology at Shanghai Jiao Tong University, in 2006 and 2003 respectively. His current research interests include information retrieval, natural language processing, and Web usage data mining.