Classifying Biomedical Citations without Labeled Training Examples

Xiaoli Li, Rohit Joshi, Sreeram Ramachandaran, Tze-Yun Leong
School of Computing,
National University of Singapore
lixl@comp.nus.edu.sg

Abstract

In this paper we introduce a novel technique for classifying text citations without labeled training examples. We first utilize the search results of a general search engine as original training data. We then proposed a mutually reinforcing learning algorithm (MRL) to mine the classification knowledge and to "clean" the training data. With the help of a set of established domain-specific ontological terms or keywords, the MRL mining step derives the relevant classification knowledge. The MRL cleaning step then builds a Naive Bayes classifier based on the mined classification knowledge and tries to clean the training set. The MRL algorithm is iteratively applied until a clean training set is obtained. We show the effectiveness of the proposed technique in the classification of biomedical citations from a large medical literature database.

1. Introduction

In traditional text classification, a classifier usually is built using labeled training documents of every class. To build a text classifier, the user first collects a set of training examples, which are labeled with pre-defined classes (labeling is often done manually). A classification algorithm is then applied to the training data to build a classifier. This approach for building classifiers is called *supervised learning*/*classification* because the training examples/documents all have pre-labeled classes.

The main problem with the traditional classification technique is that it needs a large number of labeled training examples in order to build an accurate classifier [21]. Manual labeling is very labor intensive and time consuming. We will discuss all the related work in Section 2.

To deal with the problem of labeling a large training set in classification, recently several techniques are designed. [21, 6] proposed a technique using a small set of labeled data of every class and a large unlabeled set for classifier building. It was shown that the unlabeled data does help classification. [10, 17, 16] also proposed techniques to learn from only positive and unlabeled sets (without labeled negative examples). These research efforts all aim to reduce the burden of manual labeling.

In this paper, we explore a novel technique to build classifiers without the labeled training examples. The ability to build classifiers without the labeled training data is particularly useful if one needs to do classification for different topics. For example, a doctor needs to track the new development of a few diseases simultaneously, e.g. colorectal cancer, SARS, bird flu, etc. Furthermore, given a particular disease, he/she would like to classify documents into predefined categories: diagnostic procedures, risk factors, screening methods, and treatment therapies. Following traditional classification, for each disease (topic), labeling of training examples for every category is needed. Obviously, techniques that can provide the accurate classification without manual labelling any document will be preferred.

However, to build an accurate classifier without labeled training examples is not a trivial task because the supervised learning techniques can not be used directly due to lack of labeled training examples.

This paper proposes a novel technique to build a robust classifier without labeled training examples. The main idea of the proposed technique is as follows: given a particular user's query (e.g., colorectal cancer), our approach first constructs an original training set for each category by utilizing the results from a search engine (such as Google). This set of returned pages by the search engine acts as the initial set of labeled training documents. With the help of a set of established domain-specific ontological concepts, our proposed Mutually Reinforcing Learning (*MRL*) algorithm then tries to derive classification knowledge from original training set. *MRL* then builds a Naive Bayes classifier based on the mined classification knowledge and tries to clean the training set. The *MRL* algorithm is iteratively applied until a clean training set is obtained. Finally, an accurate classifier will be built to classify any future document or test set.

The reason that this technique works is because our mining step in *MRL* can obtain the discriminative semantic concepts (we call them *knowledge phrases*) for each category from the original training set. Our cleaning step in *MRL* thus builds a classifier based on the discriminative concepts and revises the label of training set. When the *MRL* algorithm is applied again in the next iteration, we can get better discriminative concepts and consequently a more accurate *NB* classifier will be built. We believe that the quality of features used in classification has profound effects on the performance of classifier. We argue that classification techniques based on the semantic concepts of a domain can produce better classifiers than those based only on the words or the keywords. Our results show that the proposed technique is highly effective. We believe this is a promising method for text classification.

The rest of the paper is organized as follows. In Section 2, we discuss the related work. Section 3 describes our proposed technique. Section 4 shows the application of our technique to classify the citations from the biomedical domain. Experimental results are presented in Section 5. The paper concludes in Section 6.

2. Related work

Text classification (text categorization) has been studied extensively in information retrieval and machine learning. Existing techniques can be grouped into two main groups: supervised learning, semi-supervised learning. The proposed technique is related to but significantly different from all these existing approaches. We discuss and compare these approaches with our proposed technique below.

In supervised learning/classification, a set of labeled training documents of every class is used by a learning algorithm to build a classifier. Existing text classification techniques includes the Rocchio algorithm [23], the naive Bayesian method [15, 18], K-nearest neighbour [26], and support vector machines (SVMs) [24, 13]. As we discussed in the introduction section, these techniques require manual labelling of the training set, which is labour intensive and time consuming. Our proposed technique is different from this classic supervised learning, as the proposed technique does not require the human experts to label any training documents.

Due to the problem of manual labelling, partially-supervised learning or semi-supervised learning techniques are proposed, which includes two main paradigms: (1) learning with a small set of labeled examples and a large set of unlabeled examples; and (2) learning with only positive and unlabeled examples (no negative examples). (Nigam et al., 2000) shows that learning can be done in the first scenario. They demonstrated that the unlabeled data helps classification. If only the small labeled document set is available, the classifier built is often poor due to insufficient information. However, with the help of a large unlabeled set, the classification accuracy improves. Since Nigam et al.[21], a number of other researchers have studied this problem [21,5, 7, 11, 12, 20, 22, 28]. Another related work in this area is co-training [6], which uses different feature subsets of the data to iteratively produce more labeled training examples. These are different from our work, as we do not use labeled data.

In learning with only positive and unlabeled examples, some theoretical studies have been done in [9, 14, 19, 17]. Liu et al [17] also proposes a practical algorithm to solve the problem. The method is based on a spy technique and the EM algorithm [8, 21]. [27] proposes a technique based on SVM for Web page classification. [10] proposes a related technique based on naïve Bayesian classification. [16] reports a technique called Roc-SVM. In this technique, reliable negative documents are extracted by using the information retrieval technique Rocchio. Then SVM classifier with a classifier selection criterion is designed to catch a good classifier from iterations of SVM. Our proposed technique is different as it does not use any labeled training examples. Instead, it explores a different approach for building text classifiers. The proposed technique first utilizes the search results of a general search engine as original training data. Using a set of established domain-specific ontological terms or keywords, our proposed MRL is iteratively applied to derive the classification knowledge and to clean the training data. In the end, an accurate classifier will be built using the purer training set.

In a related effort in the biomedical domain, W. John Wilbur [25] exploited boosting Naïve Bayesian Learning to build REBASE (a restriction enzyme database) by classifying the citations from MEDLINE. But this is the supervised learning technique since it needs to prepare the training examples for both positive and negative classes.

3. The proposed technique

Our proposed technique is first to construct original training set using a search engine. Then Mutually Reinforcing Learning (*MRL*) algorithm, which contains the mining step and the cleaning step, is applied iteratively. The mining step basically derives classification knowledge from the noisy training set with the help of a set of established domain-specific ontological terms or keywords. The cleaning step built a Naïve Bayes classifier based on mining classification knowledge to clean the training set.

3.1. Construct the original training set

Without manual labeling training set, we query a search engine (i.e. Google) to construct the original training set. For a set of predefined classes, $C = \{C_1, C_2, \dots, C_{|C|}\}$, we generate search query $Q_1, Q_2, \dots, Q_{|C|}$ by combining the user's query and category descriptive words (provided by user). For example "colorectal cancer" +"screening" are combined to give results for the class "colorectal cancer and screening methods". The set of returned pages by the search engine acts as the initial set of labeled training documents: $\{T_1, T_2, \dots, T_{|C|}\}$. A search engine typically considers many factors in its ranking algorithm, e.g., *word count-weight*, *hyperlink information*, *type-weight* (title, anchor, URL, font size, etc), and *type-prox-weight* (how close multi-words occur in every type). So the top search results from search engine like Google are not too noisy. The top rank returned pages are generally related to the user's query for the following reasons: 1. usually query words occur many times; 2. the query words occur in important HTML tags or big font size; 3. multi-words in query are close.

Once we have constructed the original training set using a search engine, we will derive (mine) those semantic phrases with discriminating power for each category. Since mining process is done in the noisy training set, a filtering and extending strategy is designed in order to find the discriminative phrases with high recall and precision.

3.2. Mining step: mining classification knowledge from noisy training set

The target of this step is to derive the semantic concepts that have discriminative power to support the classification. We want to automatically extract some characteristic phrases for each class, which we call *knowledge phrases* of the class. We then build a classifier based on these *knowledge phrases*. Different kinds of features can be extracted as the *knowledge phrases* from the original training set, for example, keywords, concepts, and semantic types etc. Those *knowledge phrases* that have definite meanings and higher discriminating power will be extracted out as important keywords for each class. We believe that classification techniques based on the *knowledge phrases* can produce better classifiers than those based only on the words or the keywords.

In order to get the *knowledge phrases*, a pre-processing step is needed to label the semantic information of original training examples. The semantic information can be obtained by searching some lexical reference systems, for example, Wordnet. Wordnet provides the rich semantic information such as synonyms and hypernyms. For a particular domain, there also exist some established domain-specific ontological terms or concepts available. For example, for a phrase in biomedical domain, Unified Medical Language System (UMLS) ontology provides its semantic information such as the mapping concept word, synonyms (meta-candidates) and semantic types. Semantic types are more generic concepts and correspond to hypernyms in Wordnet. In this study, we applied our proposed technique into the biomedical domain, so we use UMLS as our main ontology. In the following sections, we use the semantic types of UMLS as the representation for the more generic concepts, i.e. hypernyms. We will give detail description of UMLS in section 4.

3.2.1. Association rule mining

Association rule mining algorithm was proposed by Agrawal [4]. Given a dataset D which is set of transaction T, an association rule is of the form: **X**→**Y** (**X** implies **Y**), where **X** and **Y** are mutually exclusive sets of items. An association rule **X**→**Y** presents the pattern when **X** occurs, **Y** also occurs with certain probability. The rule's statistical significance is measured by *support degree*, and the rule's strength by *confident degree*. The *support degree* s% of the rule is defined as the percentage of transaction Ts in D contains both **X** and **Y**; the *confident degree* c% is the ratio of the *support degree* of itemset $\mathbf{X} \cup \mathbf{Y}$ to the *support degree* of the itemset **X**.

The mining algorithm tries to find all the rules that satisfy the user-specified *minimum support* (*minsup*) and *minimum confidence* (*minconf*).

In our case, we want to find those *knowledge phrases* that have discriminating power to indicate which class a text citation may belong to. So, **X** is a phrase from the Mining object set M = {keywords, concepts, semantic types} and $Y \in C = \{C_1, C_2, \dots, C_{|C|}\}$. The problem is how to set the *minsup* and *minconf* in noisy environment.

Since mining is done in noisy training data, some good *knowledge phrases* cannot be derived if we restrict the *minsup and minconf* to higher values. For example, suppose phrases **X** is a *knowledge phrase* of class C_i . If some documents of class C_i are regarded as another class C_j (noisy training set), then the *confident degree* of the rule **X**→C_i is probably less than a expected value. So we set the lower *confident degree* value in order not to miss some true *knowledge phrases*. We set lower value for *minconf*, i.e. *minconf*=60%. In this setting, we can get the rules with high *recall*. We set *minsup* as $\sum_{w} freq(w)/|V|$, which is average word frequency of all the words in training set (*w* is a word and *V* is the vocabulary of training set).

We define the basic *candidate rules set* CR= {**X**→**Y** | **X**→**Y** *.conf*> *minconf* & **X**→**Y**.*sup*> $\sum_{Z} freq(Z)/|V|$ }.

The rules in CR will be further filtered using other semantic information in order to get the rules with high *precision*.

3.2.2. Heuristic strategy: filtering and extending

Filtering concepts

Obviously, it is possible that there are still some undesirable rules in CR. Our heuristic filtering strategy will filter some rules with less *semantic support*.

For any candidate rule $(\mathbf{X} \rightarrow C_i) \in$ CR, the phrase **X** in CR should have some *semantic support* concepts within a class C_i. If a concept **X** is a phrase, then its *semantic support* concepts are synonyms or its *similar concepts* with same semantic types.

If any concept in CR with less *semantic support* concepts within corresponding class, then it is considered as an occasional case and is filtered out. In detail, for all the phrases in CR, we first search the synonyms and semantic types and store them into a set CS. Then we begin to filter. If X is a phrase, then we compute *semantic support* by checking how many times its synonyms and semantic types occurred in CS. Similarly, if **X** is a semantic type, we compute its *semantic support* by counting the number of times the concepts in CS has **X** as their semantic type. A concept **X** is filtered if its *semantic support* is less than a predefined threshold. After this filtering step, we can get high precision rules. In the end, we construct the *knowledge phrases* set K_{Ci} for each class C_i: $K_{Ci} = \{ X \mid X \rightarrow C_i \in CR \}$.

```
1 Loop for all the term t in the CR
2    CS = CS ⋃ { t.semantic types or t.synonyms};
3 Loop for all the term t in the CR
4    If t is the phrase,
5       Search its synonyms and semantic types in CS;
6       Loop for each synonym and semantic type of t
7       If (t.semantic types or t. synonyms) ∈ CS
8           t.sup++;
9    Else // t is a semantic type
10      Loop all the concept c in CS;
11      If  c.semantic types = t
12          t.sup++;
13   If t.sup < $\delta$
14      CR= CR –{t}
15 Loop for all the rules in CR
16 $K_{Ci} = \{ X \mid X \rightarrow C_i \in CR \}$
```

Figure 1 filter the undesired rules from CR

Figure 1 gives the detail algorithm to filter the undesired rules from CR within class C_i. Step1–Step2 constructs a set CS which contains all the semantic types and synonyms. Step3-step14 checks each term *t* in the CR and deletes the occasional concepts from CR. Step4-step8 computes the *semantic support* when *t* is a phrase. It searches its synonyms and semantic types and checks their frequency in CS. Similarly, step10-step12 computes the *semantic support* of semantic type *t* by counting the number of concept in CS, whose semantic type is *t*. Step13-14 deletes those phrases *t* whose *semantic support* is less than a predefined δ (we set δ =2). Step15-16 constructs the *knowledge phrases* set for each class C_i .

Extending concepts

The *similar concepts* are those that share with same semantic types. A *similar concept group* provides a concept cluster with similar meaning. Given a particular category, if several concepts from a *similar concept group* occurred in a same class C_i (equal to or larger than 3), we will add entire *similar concept group* into *knowledge phrases*. By adding the similar concepts, we extend classification knowledge. In other words, some *knowledge phrases* that cannot be derived from the noisy training set will be appended (refer to a example in section 5).

3.3. Mining step: mining classification knowledge from noisy training set

The section will discuss how to clean the original training set. The basic idea is that we build a classifier using the noisy training set with mining classification knowledge. Then we classify the training examples and revise the labels of the training set according to the classification results. After we obtain the classification knowledge from the mining step, we can use the *knowledge phrases* for classifier building since they have strong discriminating power to accurately predicate the class of a text citation. Hence, compared with standard Naïve Bayes classifier, a NB classifier with *knowledge phrases* is more accurate.

There are several machine learning techniques available for classifier building. However, not all of them are suitable for our purpose. Since learning is done in noisy environment, compared with Naïve Bayes technique, some classification techniques, such as SVM, KNN, Rocchio, are more sensitive to the noise in training set. As a result, they are not applicable to our problem. Naïve Bayes classification technique, on the other hand, is a probability-based method. It is not too sensitive to noise. So we choose it to build our final classifier. Next, we will introduce the standard NB and later we will show how to add *knowledge phrases* in Naïve Bayes framework for our purpose.

The naïve Bayesian classifier (NB) is an effective text classification method [18, 15]. The basic idea of NB is to use the joint probabilities of words and classes to estimate the probabilities of classes given a document.

Like most classification techniques, NB builds a classifier using a set of labeled training examples *D*. Each example document is considered an ordered list of words. We use $w_{d_i,k}$ to denote the word in position *k* of document d_i, where each word is from the vocabulary $V = \{w_1, w_2, \ldots, w_{|v|}\}$. The vocabulary is the set of all words we consider for classification. We also have a set of pre-defined classes, $C = \{c_1, c_2, \ldots, c_n\}$. In order to perform classification, we need to compute the posterior probability $P(c_j|d_i)$, where c_j is a class and d_i is a document. Based on the Bayesian probability and the multinomial model, we have

$$\mathrm{P}(c_j) = \frac{\sum_{i=1}^{|D|} \mathrm{P}(c_j \mid d_i)}{|D|} \qquad (1)$$

and with Laplacian smoothing,

$$\mathrm{P}(w_t \mid c_j) = \frac{1 + \sum_{i=1}^{|D|} N(w_t, d_i)\mathrm{P}(c_j \mid d_i)}{|V| + \sum_{s=1}^{|V|}\sum_{i=1}^{|D|} N(w_s, d_i)\mathrm{P}(c_j \mid d_i)} \qquad (2)$$

where $N(w_t,d_i)$ is the count of the number of times the word w_t occurs in document d_i and $P(c_j|d_i) \in \{0,1\}$ depending on the class label of the document.

Finally, assuming that the probabilities of words are independent given the class, we obtain the NB classifier:

$$\mathrm{P}(c_j \mid d_i) = \frac{\mathrm{P}(c_j)\prod_{k=1}^{|d_i|} \mathrm{P}(w_{d_i,k} \mid c_j)}{\sum_{r=1}^{|C|} \mathrm{P}(c_r)\prod_{k=1}^{|d_i|} \mathrm{P}(w_{d_i,k} \mid c_r)} \qquad (3)$$

Next, we will introduce how to use *knowledge phrases* mined to boost the classifier building. We modify the standard NB classifier in two ways.

1. Add *knowledge phrases* { K_{Ci} }, i = 1, 2, …, |C| into vocabulary set *V* and computed the condition probability $P(w_t \mid c_j)$ for all $w_t \in K_{Ci}$ (w_t is *knowledge phrases* of class C_i).

V= *V*$\cup$ { K_{Ci} }, i = 1, 2, …, |C|, for $w_t \in K_{Ci}$, we modify the computation of the conditional probability $P(w_t \mid c_j)$ in equation (2) in the following way:

$$\left\{ \begin{array}{l} \textbf{If } j = i, \\ \quad \text{replace } \sum_{i=1}^{|D|} N(w_t, d_i) P(c_j \mid d_i) \text{ with } \sum_{i=1}^{|D|} N(w_t, d_i) \,; \\ \textbf{Else} \\ \quad \text{We set } \sum_{i=1}^{|D|} N(w_t, d_i) P(c_j \mid d_i) = 0; \end{array} \right.$$

w_t is one of *knowledge phrases* of class C_i and it has discriminating power to distinguish C_i from other categories, so we no longer use its word distribution information among the classes to estimate the conditional probability $P(w_t \mid c_i)$. Instead, in equation (2), if $j = i$, we use the total word frequency of w_t in all classes, replace word frequency only in C_i . In other words, we think w_t occurred in other classes C_j ($j \neq i$) just because the training set is noisy. Correspondingly if $j \neq i$, we set $\sum_{i=1}^{|D|} N(w_t, d_i) P(c_j \mid d_i)$ =0 since w_t should only occur in class C_i .

2. Emphasize the *knowledge phrases* when we classify a document. In other words, we give high weights to *knowledge phrases*. We use the equation (4) to replace the equation (3) to classify any document

$$P(c_j \mid d_i) = \frac{P(c_j) \prod_{k=1}^{|d_i|} P(w_{d_i,k} \mid c_j) * \mu(w_{d_i,k})}{\sum_{r=1}^{|C|} P(c_r) \prod_{k=1}^{|d_i|} P(w_{d_i,k} \mid c_r) * \mu(w_{d_i,k})} \qquad (4)$$

Here $\mu(w_{d_i,k})$ is the weight we assign to word $w_{d_i,k}$. If $w_{d_i,k}$ is one of *knowledge phrases*, then we give it a high weight. In effect, we build two NB classifiers. Classifier 1 NB1 is a normal classifier, which used the original training set. Classifier 2 NB2 is a classifier based on *knowledge phrases*. The final classifier NB is more depend on the classifier NB2 since we give the high weight to NB, i.e.

$$P(c_j, d_i)_{NB} = \mu_1 * P(c_j, d_i)_{NB1} + (1 - \mu_1) * P(c_j, d_i)_{NB2}$$

Where we set μ_1 =0.1 in our experiment, which can make use of the discriminating power of *knowledge phrases*.

3.4. Mutually Reinforcing Learning Algorithm

Below, we present the MRL algorithm through combining the two main components: the mining step and the cleaning step. The mining step identified the classification knowledge from the noisy training set; cleaning step built a Naïve Bayes classifier based on the mining classification knowledge. The NB classifier built can be used to clean the training set through classification because knowledge words aid in assigning the correct label to each citation in the training set. This will result in a purer training set. Furthermore, if the mining step of MRL is done with the purer training set, we will get better *knowledge phrases*, which will in-turn build an accurate classifier in the cleaning step. Suppose C = {C1, C2, … , C|C|}, and the corresponding training set T = { T1, T2, … , T|C|}, Figure 2 gives the MRL algorithm.

```
1. Loop for each document $d \in$ T
     2. Assigned semantic information to $d$ using domain-specific ontological terms;
3. Loop if the labels of documents in training set T change
     4. Perform mining step to get rule set CR for each $C_i$;
     5. Filtering rules use Figure 1 algorithm;
     6. Extending the knowledge phrases for each $C_i$;
     7. Build final classifier NB using the mining knowledge phrases;
     8. Classify the training set using NB;
     9. Revised the label of training set of T according to NB's classification results;
```

Figure 2 Mutually Reinforcing Learning (*MRL*) algorithm

4. Applications of MRL to biomedical citations

This section introduces some background and vocabulary of an case study in applying our MRL technique to classify the biomedical citations in a large medical literature database, i.e., MEDLINE. MEDLINE is a premier bibliographic database in biomedical domain containing over 12 million citations.

Each citation in MEDLINE consists of title, abstract and keyword terms called MeSH terms, and some other information. MEDLINE citations are indexed by MeSH terms which manifest the topics and the relevant contexts for these articles. These terms are manually assigned by the trained individuals.

MeSH Terms Ontology

MeSH Terms ontology consists of all the MeSH terms used in the MEDLINE. Mesh terms can be further divided into two parts: the Medical Subject Heading (MHs) and Subheadings (SHs). MHs are the preferred descriptors for subjects; SHs, also called MeSH qualifiers, are used to express a certain aspect of a MH. In general, indexers assign the most specific MHs available from Mesh Vocabulary in order to bring out the main focus of the citation. For each MH, SHs are chose as the topical subheadings from the allowable qualifier (AQ) list for that heading MH. Figure 3 shows an example of *screening* citations of **MEDLINE**. In this example, “Adenoma/pathology” means that “*Adenoma*” is a MH while “*pathology*” is a SH.

Both MHs and SHs describe the subject content of a citation. These Mesh terms contains valuable category information to aid in building classifier. Both MHs and SHs are our mining objects.

N Engl J Med. 2003 Dec 4;349(23):2191-200. Epub 2003 Dec 01.
Title: Computed tomographic virtual colonoscopy to screen for colorectal neoplasia in asymptomatic adults.
Abstract: BACKGROUND: We evaluated the performance characteristics of computed tomographic (CT) virtual colonoscopy for the detection of colorectal neoplasia in an average-risk screening population. METHODS: A total of 1233 asymptomatic adults (mean age, 57.8 years) underwent same-day virtual and optical colonoscopy.
………………………
MeSH Terms:
Adenoma/pathology
Adenoma/radiography*
Aged
Colonic Polyps/pathology
Colonic Polyps/radiography*
Colonography, Computed Tomographic*
Colonoscopy
Colorectal Neoplasms/pathology
Colorectal Neoplasms/radiography*
Comparative Study
……………………….

Figure3 One example of MEDLINE citations

Unified Medical Language System (UMLS) Ontology
The Unified Medical Language System (UMLS) is a compilation of more than 60 controlled vocabularies in the biomedical domain. The UMLS is structured around three separate components: Metathesaurus, SPECIALIST Lexicon and Semantic Network. For our purpose, we only need the *UMLS Metathesaurus.* It provides a representation of biomedical knowledge consisting of concepts (more than 800,000 concepts) classified by semantic type and both hierarchical and non-hierarchical relationships among the concepts.

UMLS also provides a parser to segment the phrases and output the semantic type of mapping words for any given citation. Figure 4 is an analysis results of the phrase “virtual colonoscopy”. From the analysis results, we know that phrase “virtual colonoscopy” has 5 candidates (called Meta Candidates) that are related to it. Meta Mapping phrase is the best among the candidates (note the phrase and the meta mapping concept can be different). Semantic type information displayed is **(Colonography, Computed Tomographic) [Diagnostic Procedure]**

We can get the similar information from UMLS for any word or MeSH terms. For example, for the Mesh term “*Radiography*”, its semantic type is **(Diagnostic radiologic examination) [Diagnostic Procedure]**.

Phrase: "virtual colonoscopy"
Meta Candidates (5)
1000 Virtual Colonoscopy **(Colonography, Computed Tomographic) [Diagnostic Procedure]**
861 Colonoscopy **[Diagnostic Procedure,Therapeutic or Preventive Procedure]**
789 Colonoscope (Colonoscopes) **[Medical Device]**
789 Virtue (Virtues) **[Idea or Concept]**
761 Coloscopes **[Medical Device]**
Meta Mapping (1000)
1000 Virtual Colonoscopy (Colonography, Computed Tomographic) [Diagnostic Procedure]

Figure 4 UMLS analysis results of phrase “virtual colonoscopy”

The mapping phrases and the semantic types are also considered as potential mining objects as they have discriminative power to support the classifier building.

5. Experimental Results

Now we evaluate the proposed technique on the biomedical citations. We classify two kind of diseases “colorectal cancer” and “SARS” into 4 classes: diagnosis”, “risk factor”, “screening”, and “treatment”. Below, we first present the detail results for the disease “colorectal cancer”.

Construct the original training set: we query search engine Google to construct the original training set. The queries we generated are “colorectal cancer diagnosis”, “colorectal cancer risk factor”, “colorectal cancer screening”, and “colorectal cancer treatment”. We restrict Google only to search from **MEDLINE**. The set of returned pages by Google acts as the initial set of labeled training documents. For each class, we fetch 1000 documents and after simple filtering (such as filter those pages that are found in more than 1 category or that does not have any abstract), then we got the training set for each category.

Association rule mining: We use UMLS tools to label the semantic concepts of each citation. After mining, we get the CR set and list a few rules found in our dataset :
[Diagnostic Procedure] → diagnosis; conf 0.66;
Colonoscopies →diagnosis; conf 0.99;
Diet → risk factors; conf 0.89;
Color index →screening; *conf* 0.92;
(Chemotherapy-Oncological Procedure) [Therapeutic or Preventive Procedure] → treatment; *conf* 0.91;
(Enzyme Inhibitor Drugs) [Pharmacologic Substance] →treatment; *conf* 0.91;
Population → diagnosis; *conf* 0.88;

Filtering Rules: Some rules are still not desirable. For example, a rule Population → diagnosis is not a correct one. The phrase “Population” should not act as a discriminating word of diagnosis category as this rule is just an occasional case. We filter out the rules by using the algorithm in Figure 1 if they do not have the required *semantic support.*

Extending concepts: Some similar concept groups are also added. For example, for the screening category, "*Faecal occult blood*", "*blood*", "*Screening*", "*Blood vessel*" were in the original CR, so cluster "*Faecal occult blood screen*", "*Faecal*" , "*Occult*" , "*Faeces bloodstained*","*Bloods*" , "*Screening*" , "*Blood vessel*" , "*Occult blood screen*" "*Blood stain*" , "*Vascular*", "*Faecal occult blood*" etc are added as *knowledge phrase*. After filtering step and extending step, we store *knowledge phrases* into corresponding set K_{Ci} for each category C_i.

Knowledge phrases: Appendix lists a part of the *knowledge phrases* that we get in the last iteration of MRL algorithm. We found that the more iteration the algorithm MRL runs, the better concepts we get.

Test set: In order to evaluate the performance of classifier, we manually label 500 citations from the **MEDLINE**. Note that this labeling is needed only for the evaluation, but not in the implementation of the proposed technique. The documents constituting the test set are the most recently published articles (published in 2003). It is interesting to know the effects when we use the "old" training set to classify the new published citations.

Experiment measures: We use accuracy as the evaluation measure of the system. Accuracy is adequate because it reflects the average effect of every category (averages the performance of every class). Accuracy can be defined as:

$$Acc = \sum_{i=1}^{|C|} TP(C_i) / |\{d \mid d \in T\}|$$

where $TP(C_i)$ is the true positive number of the category C_i and $|\{d \mid d \in T\}|$ is the total number of test set.

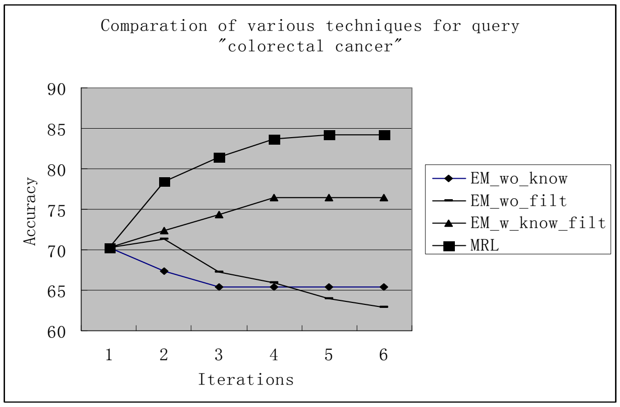

Figure 5 Comparison of various techniques for query "colorectal caner"

We compare our proposed technique with the Expected Maximization (EM) technique [3, 21] algorithm. In order to evaluate the separate contributions of *knowledge phrases* and filtering step, we include the results of several techniques: EM without *knowledge phrases* (EM_wo_know), EM with *knowledge phrases* but without filtering (EM_wo_filt) and EM with both *knowledge phrases* and filtering (EM_w_know_filt). Note all the three EM based techniques, compared with MRL, do not have the cleaning step to revise the label of the training set.

Figure 5 gives us the accuracy results for "colorectal cancer" of each iteration using three EM-based techniques and *MRL*. Here the baseline NB classifier gets 70.3%. The accuracy of EM_wo_know decreases with the iterations of EM. In other words, without the help of *knowledge phrases*, EM can not improve the NB's results. The accuracy of EM_wo_filt increases first but then decreases. So filtering is a very important step and directly using concepts mined will hurt the performance of a classifier. With the help of *knowledge phrases* (with filtering), EM_w_know_filt gets the 76.4%, 6.1% higher than NB's results. Our proposed MRL technique MRL achieves the accuracy of 84.2%, which improves the NB and EM_w_know_filt 13.9% and 7.8% respectively.

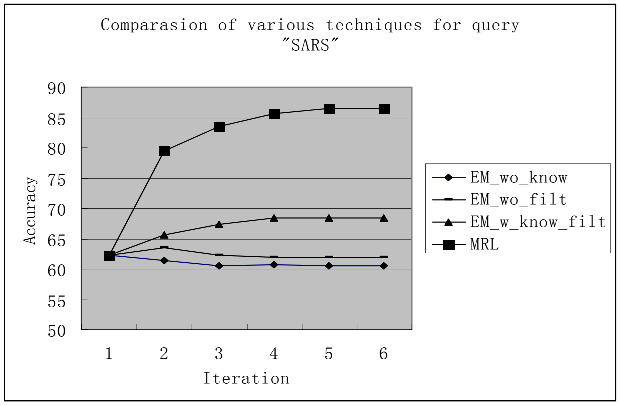

Figure 6 Comparison of various techniques for query "SARS"

The second experiment we have done is to classify "SARS" documents. Figure 6 shows the accuracy results of various techniques. Both EM_wo_know and EM_wo_filt can not improve the NB's results. The accuracy of EM_w_know_filt is 6.1% higher than the NB's result. MRL get the best results 86.5%, 18.1% higher than EM_w_know_filt.

From the figure 5 and 6, we can conclude that qualified *knowledge phrases* does help learning algorithm EM and MRL to build an accurate classifier. Moreover, the cleaning step of *MRL* makes it perform significantly better than the EM algorithm.

6. Conclusion

In this paper, we propose a new approach to build a classifier to classify citations in the **MEDLINE** database without the labeled training dataset. Traditional text classifiers are built using a set of labeled training documents. Labeling is typically done manually, which is a time consuming process. This paper proposed a novel approach. In this approach, we utilize the search results from a general search engine as the original training data. With the help of a set of established domain-specific ontological terms or keywords, a mutually reinforcing learning algorithm is applied iteratively to extract the classification knowledge and cleaning the training data. A Naive Bayes classifier is built based on the cleaned training data and classification knowledge. Experimental results show this is very promising approach for text classification.

7. References

[1] **MEDLINE**, http://www.ncbi.nlm.nih.gov/PubMed/

[2] **UMLS**, http://umlsks.nlm.nih.gov/

[3] **Google**, http://www.google.com

[4] Agrawal R., Srikant R., "Fast Algorithms for Mining Association Rules", VLDB-1994.

[5] Basu, S., Banerjee, A., & Mooney, R. (2002) "Semi-supervised clustering by seeding." *ICML-02*.

[6] Blum, A., Mitchell, T. "Combining labeled and unlabeled data with co-training," *Proc. Of the 11th Conf. on Computational Learning Theory*, 1998.

[7] Bockhorst, J., and Craven, M. "Exploiting relations among concepts to acquire weakly labeled training data." *ICML-2002*.

[8] Dempster, A., Laird, N. M. & Rubin. D. "Maximum likelihood from incomplete data via the EM algorithm." *J. of the Royal Statistical Society*, B:39, 1-38, 1977.

[9] Denis, F. "PAC learning from positive statistical queries." *ALT- 1998*.

[10] Denis, F. Gilleron, R and Tommasi, M. Text classification from positive and unlabeled examples. *IPMU*-2002.

[11] Ghani, R. "Combining labeled and unlabeled data for multiclass text categorization." *ICML-2002*.

[12] Goldman, S. and Zhou, Y. "Enhancing supervised learning with unlabeled data." *ICML-2000*.

[13] Joachims, T. "Text categorization with support vector machines: Learning with many relevant features." *ECML-98*.

[14] Kearns, M. "Efficient noise-tolerant learning from statistical queries." *Journal of the ACM*, 45, pp. 983-1006, 1998.

[15] Lewis, D., Gale, W. "A sequential algorithm for training text classifiers." *SIGIR-1994*.

[16] Li X., Liu B. Learning to classify text using positive and unlabeled data. IJCAI-2003.

[17] Liu, B., Lee, W. S., Yu, P., and Li, X. "Partially supervised classification of text documents." *ICML-2002*.

[18] McCallum, A., Nigam, K. "A comparison of event models for naïve Bayes text classification." *AAAI-98 Workshop on Learning for Text Categorization.* 1998.

[19] Muggleton, S. "Learning from the positive data." *Machine Learning*, 2001.

[20] Muslea, I., Minton, S., and Knoblock, C. A. "Active + semi-supervised learning = robust multi-view learning." *ICML-2002*.

[21] Nigam, K., McCallum, A., Thrun, S. and & Mitchell, T. "Text classification from labeled and unlabeled documents using EM." *Machine Learning*, 39, 2000.

[22] Rakutti, B. Ferra, H. Kowalczyk, A. "Using unlabeled data for text classification through addition of cluster parameters." *ICML-2002*.

[23] Rocchio, J. "Relevant feedback in information retrieval." In G. Salton (ed.). *The smart retrieval system- experiments in automatic document processing*, Englewood Cliffs, NJ, 1971.

[24] Vapnik, V. *The nature of statistical learning theory*, 1995.

[25] Wilbur W. J., "Boosting Naïve Bayesian Learning on a large subset of MEDLINE", AMIA-2000.

[26] Yang, Y. and Liu, X. "A re-examination of text categorization methods." *SIGIR-99*.

[27] Yu, H., Han, J. & Chang, K. "PEBL: Positive example based learning for Web page classification using SVM." *KDD-2002*.

[28] Zelikovitz, S., & Hirsh, H. "Improving short-text classification using unlabeled background knowledge to assess document similarity." *ICML-2000.*

Appendix

Diagnosis: (Colonoscopy) [Diagnostic Procedure, Therapeutic or Preventive Procedure]; (Diagnostic) [Functional Concept]; (microsatellite instability diagnostic test) [Diagnosic Procedure]; Diagnosis <1>; [Diagnostic Procedure]; Colonoscopies; NOS ; Lower gastrointestinal tract examination;……

Risk factor: Alcohol; Color index; Diet; Dietary Fats; Drinking <2>; Insulin-Like Growth-Factor-Binding Proteins; Meat; [Food]; [Hazardous or Poisonous Substance];[Individual Behavior]; [Lipid, Food]; [Organic Chemical,Vitamin]; [Vitamin]; carcinogenic;……

Screening: (Color index level) [Laboratory or Test Result]; (Colonography, Computed Tomographic) [Diagnostic Procedure]; (Occult blood in stools) [Finding];(Screening for cancer) [Therapeutic or Preventive Procedure];(Screening for occult blood in feces) [Laboratory Procedure];(Screening procedure) [Diagnostic Procedure];(X-Ray Computed Tomography) [Diagnostic Procedure];(brief historical notes, excludes case histories) [Intellectual Product];(diagnostic imaging <1>) [Diagnostic Procedure]; (screening for colorectal cancer) [Therapeutic or Preventive Procedure]; ….

Treatment: (Chemotherapy-Oncologic Procedure) [Therapeutic or Preventive Procedure]; (Enzyme Inhibitor Drugs) [Pharmacologic Substance];(Operation on liver, NOS) [Therapeutic or Preventive Procedure]; (Pharmacotherapy) [Therapeutic or Preventive Procedure]; Surgical aspects) [Functional Concept]; Pemetrexed;; Irinotecan; Cancer Vaccines;Chemotherapy administration; [Virus]; therapy; drug therapy; ……