CLASSIFYING STRUCTURED WEB SOURCES USING AGGRESSIVE FEATURE SELECTION

Hieu Quang Le, Stefan Conrad
Institute of Computer Science, Heinrich-Heine-Universität Düsseldorf, D-40225 Düsseldorf, Germany
lqhieu@cs.uni-duesseldorf.de, conrad@cs.uni-duesseldorf.de

Keywords: Deep Web, Classification, Database, Feature selection, Gaussian processes.

Abstract: This paper studies the problem of classifying structured data sources on the Web. While prior works use all features, once extracted from search interfaces, we further refine the feature set. In our research, each search interface is treated simply as a bag-of-words. We choose a subset of words, which is suited to classify web sources, by our feature selection methods with new metrics and a novel simple ranking scheme. Using aggressive feature selection approach, together with a Gaussian process classifier, we obtained high classification performance in an evaluation over real web data.

1 INTRODUCTION

There are a large number of websites that store information in form of structured data with attribute-value pairs (Chang et al., 2004), forming an important part of the huge Deep Web (Bergman, 2001). These structured web sources provide search interfaces so that users can query their databases. On the one hand, integrated access over multiple sources is needed. For examples, a user may want to compare prices of a book in different online shops; or s/he may buy air tickets and book a room in a hotel online while preparing for a trip. On the other hand, there have been researches on data integration of a relative small number of heterogeneous sources (e.g., Chawathe et al.,1994; Levy et al., 1996), as well as on large-scale search over multiple text databases (e.g., Callan et al., 1999; Ipeirotis et al., 2001). Consequently, projects aiming at providing integrated data access to a large number of structured web sources, such as MetaQuerier (Chang et al., 2005) and WISE (He et al., 2005), have been born. Building and maintaining such large-scale accessing services involves a number of tasks: source finding and categorization, schema mapping and query translation, data extraction and integration, etc. The categorization task is an integral part of these projects, as sources that have been collected must be grouped according to similarity before other tasks, such as schema mapping (He and Chang, 2003) or query interface integration (Wu et al., 2004), can be performed. In this paper, we address this important problem of web source categorization.

The search interfaces of structured web sources, which serve as the "entrances" to underlying databases, are used as the main source for the categorization task. He et al. (2004) argued that the form labels of search interfaces (e.g., 'Title', 'ISBN(s)', ... in an Amazon's search form) are the right "representatives" for structured sources, and used only them. Subsequently, in addition to form labels as the most important feature, Lu et al. (2006) identified and utilized other features such as form values (e.g., 'hardcover', 'paperback') and other regular text terms. In these two works, features inputed to their clustering algorithms must be extracted from HTML pages by another technique (see Lu et al., 2006). In contrast, Barbosa et al. (2007) argued that such an extraction task is hard to automate, so they used all the text (bag-of-words) of a search interface, which is partitioned into text of the form and text of the page, together with backlinks pointing to the interface.

A common thing in the prior works is that features, once extracted, are all used without any further selection. However, it is not difficult to see that in a search interface, words that help in distinguishing categories (e.g., 'author', 'textbooks') mingle with many more other words. Indiscriminative or noisy terms (e.g., ‘sort by’, ‘state’) also occur inside forms, as observed in (Lu et al., 2006). Thus, this paper investigates on how to identify features suitable for categorizing structured web sources, i.e., the feature selection (FS) problem.

Our classification approach employs a filtering FS method in text categorization (Sebastiani, 2002), together with a Gaussian process classifier (Rasmussen and Williams, 2006). In our research, we treated each complete search interface simply as a bag-of-words. To choose a suitable subset of words, we conducted experiments with various FS techniques, such as χ^2 (CHI), Information Gain (IG) (Yang and Pedersen, 1997), Bi-normal separation (BNS) (Forman, 2003), as well with our methods. [1] By using aggressive feature selection with χ^2, IG or our methods, we obtained the high classification performance with the subset selected, which is significantly higher than the performance obtained when using the much larger set of all words. This result not only shows that our classification approach has its own strength, but is also a convincing evidence that extracted features should be further selected. This is our first contribution.

Our second contribution is that we propose a new feature selection method. As pointed out in (Barbosa et al., 2007), it is prone to make clustering mistakes among domains with overlapping vocabulary (e.g., Movies and Musics), and for domains with a highly heterogeneous vocabulary. Our FS methods, with new metrics and a novel ranking scheme, aim at tackling the issues. In the mentioned experiments with different FS techniques, we obtained the best performance with the new methods.

In the following section, we review related work. In Section 3, we describe the classification process. Then come its details: the feature selection method in Section 4, and the choices of weighting scheme and of covariance function in Section 5. Section 6 presents our experimental result and discussion. We conclude with a summary and future work in Section 7.

2 RELATED WORK

We relate our work to other categorization problems, to the researches on the same topic of structured web source (SWS) categorization and on feature selection.

First, SWS categorization is related to text database classification (e.g., Callan et al., 1999; Ipeirotis et al., 2001) as they work with the Deep Web’s sources. A text database normally stores documents of multiple categories. Its interfaces contain simple (e.g., single-label) forms and little information about stored documents. Therefore, to have the representative summary of a text database, query submission techniques that send queries to sources and analyze returned results are needed. In contrast, a structured database stores data objects of a single domain. Its search interfaces provide much information, such as form labels and values (describing exported schema), advertised products (i.e., data items), hence can be used to categorize the database. Furthermore, it is necessary that the domain of a structured source, which contains complex and multi-label search forms, is known before a query submission technique can be applied. Since utilizing search interfaces, SWS categorization is also related to web page categorization (e.g., Chakrabarti et al., 1998; Zamir and Etzioni, 1998) which uses terms and links, and to text categorization (e.g., Joachims, 1998), which uses only terms. However, the goal of SWS categorization is not to classify a search page (i.e, an HTML page) itself but the database, to which the page’s search form serves as an “entrance”. As a result, discriminative features extracted from or related to search forms are most important to the task.

Second, as mentioned in Section 1, the works of He et al. (2004), Lu et al. (2006) and Barbosa et al. (2007), together with this paper, are on the same topic of SWS categorization. While the prior works show that it is feasible to determine the domain of a source by using discriminative features extracted from source’s search interfaces, we take a step further. We refine the set of features once extracted, and use the refined set in order to increase classification performance. In addition, the prior studies employed clustering (i.e., unsupervised learning) algorithms, while we use a classifying (i.e., supervised learning) technique. One reason is that we group web databases so that other integrating tasks can be performed (see Section 1). There is no online requirement as, for example, in search engines where web documents may need to be clustered dynamically within a few seconds in response to a user’s need (Zamir and Etzioni, 1998). The emphasis is thus on accuracy. The other reason is that our goal is to categorize a large number of sources. It is appropriate that we build an initial domain hierarchy, either manually or by clustering together with manually checking, from a small number of sources; then classifying the rest so as to make use of sample data better through a learning process.

Third, the problem of feature selection in text categorization has been intensively studied, e.g., in

[1]We use the phrase of “FS metric” to indicate a scoring formula, and “FS method/technique” to indicate a scoring formula with a ranking scheme.

(Yang and Pedersen, 1997; Mladenic, 1998; Soucy and Mineau, 2001; Forman, 2003; Gabrilovich and Markovitch, 2004). These works, as well as our methods, use the filtering approach, in which terms are scored by a metric, then the highest ranked terms are selected (Sebastiani, 2002). There are a number of FS metrics, such as χ^2, IG and BNS, and each has its own rationale. (A comprehensive list can be found in (Sebastiani, 2002; Forman, 2003).) As discussed later, our metrics are designed to find terms that help in distinguishing closely related domains. In terms of the ranking technique, the standard scheme sorts terms regardless of categories (Sebastiani, 2002), whereas our approach ranks terms with respect to the category that they represent best.

3 CLASSIFICATION PROCESS

In our research, each complete search interface is treated simply as a text document, i.e., a bag-of-words extracted from its HTML content. Similar to other studies (He et al., 2004; Lu et al., 2006; Barbosa et al., 2007), we assume that one web database, together with its search interfaces, belongs to one category. Since a document representing a web source belongs to one category, and there are multiple categories, our problem is equivalent to a single-label multi-class text categorization problem (Sebastiani, 2002). [2] We choose the Gaussian Processes, a non-parametric, kernel-based and supervised learning method, as our classifier because it works in high dimensional feature spaces and has a sound probabilistic foundation (Rasmussen and Williams, 2006).

Formally, the process of categorizing documents using a Gaussian process classifier (GPC) is described as follows: Let $C = \{c_i\}_1^n$ be the set of n categories; $D = \{d_j\}_1^m$ be the training set of m text documents, in which each document d_j is labeled with a predetermined "correct" category c_i; V be the set of all terms (i.e., processed words) appearing in D. Let d_* denote a "new" document not in D that is to be classified into some category c_i in C.

1. Use a FS method to choose from V a subset of N terms $V_{FS} = \{t_k\}_1^N$.
2. Represent each document d_j in D as a N-dimension vector $v_j = (w_{j1}, \ldots, w_{jN})$, in which w_{jk} is the weight of a term t_k in the document d_j according to some weighting scheme.
3. Choose a GPC covariance function, then train the Gaussian process classifier with the m vectors $v_1, \ldots, v_m$ as inputs and their predetermined categories as target outputs.
4. After the learning process completed, the Gaussian process classifier is ready to classify the vectorized representation of the document d_*.

In the following section, we will describe step 1, then in Section 5 step 2 and 3.

[2]Note that, our approach can be easily adapted to a multi-label case (i.e., an interface is to be classified into several categories) by transforming the problem into independent problems of binary classification between a category c_i and its complement $\overline{c_i}$ (Sebastiani, 2002).

4 FEATURE SELECTION

The feature selection problem in text categorization is to identify words that are suited to categorize documents. This section describes our methods that consist of: (1) new metrics, (2) a novel ranking scheme.

4.1 Feature Selection Metrics

We first define new metrics, then give their intuitive explanation and comparison with χ^2 metric.

Metric Definition. Let $\{c_i\}_1^n$ be the set of n $(n \geq 2)$ categories; $P(t|c_i)$ be the conditional probability that a random document in a category c_i contains a term t; $P(c_i|t)$ be the conditional probability that a random document containing a term t belongs to a category c_i. Let c_1 and c_2 denote the two categories that have the first and second highest values of all probabilities $P(c_i|t)$ for $i = 1, \ldots, n$. The score of a term t is given in one of the two formulas below:

Top-two-category separation (T2CS):

$$S_1(t) = \left[P(t|c_1) - P(t|c_2)\right] \cdot \left[P(c_1|t) - P(c_2|t)\right]$$

Top-two-category separation - χ^2 (T2CS-CHI):

$$\begin{aligned} S_2(t) = \quad & \left[P(t|c_1) \cdot P(\bar{t}|c_2) - P(t|c_2) \cdot P(\bar{t}|c_1)\right] \cdot \\ & \cdot \left[P(c_1|t) \cdot P(c_2|\bar{t}) - P(c_2|t) \cdot P(c_1|\bar{t})\right] \end{aligned}$$

Metric Explanation. We now explain the intuition behind T2CS metric through an example with three categories Airfares, Books and Musics, denoted as Af, Bk and Ms respectively. When a document contains the word 'music', together with other words, such as 'title', 'elvis' and 'surround', we can normally determine quickly that its category is either Musics or Books, but not Airfares. We observe that Musics and Books are the two categories to which a random document containing 'music' most likely belongs, and of which vocabularies usually have a large overlap. Since the main categorization difficulty is in deciding between them, we are interested in how much 'music' helps to classify documents into Musics or Books. We formalize it by selecting the "top" two categories that have the first and second highest values of all probabilities $P(Category|music)$.

Next, we formulate the scoring function S_1 of T2CS metric. When a document contains the word 'music', we tend to think that its category is likely to be Musics. In that view, we take $\big[P(Ms|music) - P(Bk|music)\big]$ to measure how much 'music' *helps* if the document in fact belongs to Musics, as well as how much it *misleads* us if the document in fact belongs to Books instead. When all documents are taken into consideration, the more frequently 'music' appears in Musics (i.e., the higher probability $P(music|Ms)$) the better, but the more frequently 'music' appears in Books (i.e., the higher probability $P(music|Bk)$) the less it helps (i.e., the more it misleads us). Hence, we evaluate the score of 'music' in differentiating all documents of Musics from all documents of Books, as follows:

$$P(music|Ms) \cdot \big[P(Ms|music) - P(Bk|music)\big] - \\ -P(music|Bk) \cdot \big[P(Ms|music) - P(Bk|music)\big],$$

which is the same as the formula $S_1(music)$. Note that since the sum of $P(Af|music)$, $P(Bk|music)$ and $P(Ms|music)$ is 1, the role of Airfares is indirectly taken into account, though it does not explicitly appear in the scoring formula.

Finally, let us consider all the example words 'music', 'elvis', 'surround', 'title' again. Statistics on the dataset we used shows that 'music' appears very frequently in Musics, relatively frequently in Books; 'elvis' relatively frequently and only in Musics; 'surround' rarely and only in Musics; 'title' very frequently in Books as well in Musics. Under our scoring scheme, 'surround' and 'title' have low ranks and will usually be discarded. In other metrics like IG or χ^2, similar situation happens to 'surround', but not to 'title'. The other metrics score 'title' high, as it may distinguish Books (and Musics) from Airfares. Meanwhile, we score 'title' low as its usage may increase the mis-classification between Books and Musics. Note further that in text categorization, contrary to a widely held belief in information retrieval, common terms are informative, and scored in favor over rare terms (Yang and Pedersen, 1997); and we will discuss about it more in Section 6.3.

Comparing T2CS-CHI to χ^2. T2CS-CHI is a variant of T2CS metric. Besides the presence of a term t in a document (i.e., $P(c_i|t)$) is into account as in T2CS, T2CS-CHI also makes use of the information on the non-presence of t (i.e., $P(c_i|\bar{t})$). (Note that since the sum of $P(t|c_i)$ and $P(\bar{t}|c_i)$ is 1, $\big[P(t|c_1) - P(t|c_2)\big]$ is equal to $\big[P(t|c_1) \cdot P(\bar{t}|c_2) - P(t|c_2) \cdot P(\bar{t}|c_1)\big]$.) T2CS-CHI metric has its name from the fact that in the case of binary categorization (i.e., $n = 2$), it can be proved to be equivalent to χ^2 metric defined in (Sebastiani, 2002) as follows:

$$\chi^2(t) = |V| \cdot \frac{\big[P(t,c_i) \cdot P(\bar{t},\overline{c_i}) - P(t,\overline{c_i}) \cdot P(\bar{t},c_i)\big]^2}{P(t) \cdot P(\bar{t}) \cdot P(c_i) \cdot P(\overline{c_i})}$$

where $|V|$ denotes the total number of terms; $P(t,c_i)$ the probability that a random document contains a term t and belongs to a category c_i; $\overline{c_i}$ the complement of a category c_i. In the binary case, T2CS-CHI differs from χ^2 only in the number $|V|$, which is the same for all scores. However, in the case of multi-class categorization, T2CS-CHI and χ^2 are different from each other in the way of selecting categories. That is, T2CS-CHI metric utilizes only the "top" two categories. Meanwhile, in the common approach, χ^2, as well as other metrics such as BNS, are calculated from a category and its complement. Hence, for each term, there is a score for every category; and the final score of the term is the maximum or average value (Yang and Pedersen, 1997). In Section 6.4, we will present the results of experiments comparing FS metrics, as well as the standard and a new ranking schema described in the following.

4.2 Feature Selection Procedure

This subsection discusses how our FS methods preprocess words, then rank and select terms.

In the preprocessing step, we do not apply a stemming algorithm to reduce words with the same stem to a common form. The reason is that, for example, word 'book' appears frequently in both Airfares and Books domains, while word 'book*s*' is often found in Books but not in Airfares. Thus, these two words should be seen as two different discriminative features instead of being merged by a stemming algorithm. In addition, terms that appear less than some small K times in every category are to be eliminated. This technique is to remove noisy words.

In the next steps, while the standard scheme of the filtering approach ranks terms by their scores and selects them top down regardless of categories (Sebastiani, 2002), we approach differently. In the investigation of the standard ranking scheme with a metric (e.g., IG, χ^2), we put each term t of the top N terms selected into the category c_i, to which a document containing the term t most likely belongs (i.e., $P(c_i|t)$ is the highest), or in other words, into the category c_i which the term t represents best. We observed that some categories get many more terms assigned than other categories (e.g., Jobs or Automobiles may contain around 10 times higher than Books). This imbalance may make classification more error prone, since there may be not enough discriminative features to accept or reject whether a document belongs to a category with the small number of terms. The category with the small number, in turn, usually has a highly heterogeneous vocabulary. Hence, we propose a new simple ranking scheme aiming at balancing the number of terms representing each category. It is the steps 1 and 2 of the FS procedure described as follows:

1. Compute scores of terms by a metric (our metrics or the others), and assign each term t to the category c_i with maximum $P(c_i|t)$.
2. Sort terms in each category by their scores in descending order; then re-rank all terms together first by the relative ranks in their categories in ascending order and second by their scores in descending order.
3. Select top N ranked terms from the set of all terms, where N is a parameter determined through a cross-validation procedure.

5 WEIGHTING SCHEME AND GPC COVARIANCE FUNCTION

We chose a non-weighted feature scheme for document vectors in the step 2, and a dot product GPC covariance function in the step 3 of the classification process (see Section 3), for we obtained the best performance with these modest settings.

Formally, let $V_{FS} = \{t_k\}_1^N$ be the set of N selected terms. In the *non-weighted feature scheme*, a document d_j is represented as a N-dimension vector of weights $v_j = (w_{j1}, \ldots, w_{jN})$, where:

$$w_{jk} = \begin{cases} 1 & \text{if } t_k \text{ is in } d_j \\ 0 & \text{otherwise.} \end{cases}$$

We observe that the distinctive words of a domain (e.g., ‘ISBN’, ‘paperback’ and ‘hardcover’ for Books domain) often appear only once in a search interface as its form labels or values. This observation helps to explain why the non-weighted feature scheme is suitable for the categorization task at hand. Furthermore, in our approach, the weight of a term is not only determined by a weighting scheme but also through training a Gaussian process classifier.

For the Gaussian process classifier we use, its *dot product covariance function* between two document vectors v_l and v_m has the form:

$$\sum_{k=1}^{N} (w_{lk} \cdot w_{mk}).$$

Table 1: Dataset of 431 web sources in 8 domains.

Domain	*# of sources*	*Domain*	*# of sources*
Airfares	43	Hotels	34
Automobiles	80	Jobs	50
Books	66	Movies	71
CarRentals	21	Musics	66

Our choice of a linear kernel (i.e., a dot product covariance function) is similar to other studies in text categorization (e.g., Joachims, 1998; Gabrilovich and Markovitch, 2004) that used a Support Vector Machine classifier with a linear kernel.

6 EXPERIMENTS

6.1 Dataset and GPC Implementation

We used *the TEL-8 dataset of the UIUC Web integration repository* (UIUC, 2003), which contains the search interfaces of 447 structured web sources classified into 8 domains. After converting HTML pages to text documents and manually checking, we kept 431 sources. The other sources were not usable because the offline contents of search pages required an online update while the pages themselves no longer exist on the Web. Table 1 describes the dataset on which we conducted experiments. In the FS preprocessing step, we ignored words that appear less than twice in every category (i.e., $K = 2$) so as to remove noisy words. For the problem at hand, web sources are to be discovered automatically by a crawler (see, e.g., Barbosa and Freire, 2005). Due to the dynamic nature of the Web, we assume that there is no prior information in regard to the category of a new web source to be classified. [3] Therefore, in the implementation of FS metrics, we assigned the value $(1/n)$ to the probability $P(c_i)$ that a random document belongs to a category c_i, where n is the total number of categories.

For the GPC implementation, we used *the fbm software* (Neal, 1997). We utilized a jitter of 0.1 for improving the conditioning of the matrix computations, and the standard hybrid Monte Carlo updating procedure for the sampling phase. These and other simulation setups we used are fairly standard as described in the software’s documentation.

[3]In the TEL-8 dataset used by us and in (He et al., 2004; Barbosa et al., 2007), the ratio between the source number of Books domain and that of Jobs domain is 66 : 50 (1.32), while in another dataset used in (Lu et al., 2006) the ratio is 85 : 20 (4.25). This substantial difference in the ratio of sources is additional support for our assumption.

6.2 Performance Measures and Feature Selection Methods

To evaluate classification performance for all categories, we use overall precision p, recall r and F-measure F_1 defined in (Lu et al., 2006) as follows:

$$p = \sum_{i=1}^{n} (p_i \cdot \frac{m_i}{m}), \quad r = \sum_{i=1}^{n} (r_i \cdot \frac{m_i}{m}), \quad F_1 = \frac{2 \cdot p \cdot r}{p + r},$$

where p_i is the precision of a category c_i (i.e., the ratio of the number of sources correctly classified over the number of sources that are assigned into c_i); r_i is the recall of a category c_i (i.e., the ratio of the number of sources correctly classified over the the number of sources that should have been classified into c_i); m_i is the number of sources in a category c_i; m is the total number of sources and n is the total number of categories. For single-labeled datasets, the overall recall, as well as widely used micro-precision, micro-recall and micro-F_1 (Sebastiani, 2002), can be proved to be equal to accuracy, which is conventionally used in text categorization researches. Hence, we chose accuracy as the single measure used in our graphs comparing FS methods. All performance values reported were obtained using 2-fold cross-validation scheme.

In addition to our two metrics, we carried out experiments with other reportedly most effective metrics: χ^2 (CHI), the multi-class version of Information Gain (IG), Document Frequency (DF) (Yang and Pedersen, 1997), the binary version of Information Gain (IG2) (Gabrilovich and Markovitch, 2004), Bi-normal separation (BNS) (Forman, 2003), Odds Ratio (OR) (Mladenic, 1998). For metrics with one value per category (χ^2, IG2, BNS, OR), we used the maximum value as the score, for it performs better than the average value across metrics, classifiers, and text collections (Rogati and Yang, 2002). We tested each selected metric with both the standard and new ranking schema. In the next two subsections, we will discuss T2CS, T2CS-CHI, χ^2, IG and DF metrics, while omit the others for their results are either similar to or lower than that of χ^2 or IG.

Table 2: Classifcation performance.

Method	*Precision*	*Recall*	*F-measure*
All terms after preprocessing	89.62 %	89.33 %	89.47 %
T2CS$_{nr}$	95.61 %	95.36 %	95.49 %
T2CS-CHI$_{nr}$	95.56 %	95.35 %	95.46 %
χ^2	94.95 %	94.65 %	94.80 %
IG	94.25 %	94.20 %	94.23 %

Table 3: Classifcation results.

	Af		*Am*		*Bk*		*Cr*		*Ht*		*Jb*		*Mv*		*Ms*	
	a	b	a	b	a	b	a	b	a	b	a	b	a	b	a	b
Af	39	39	0	0	0	0	2	3	2	1	0	0	0	0	0	0
Am	0	0	79	79	0	0	0	1	0	0	1	0	0	0	0	0
Bk	0	0	0	0	62	65	0	0	0	0	2	0	2	1	0	0
Cr	5	1	0	0	0	0	12	19	4	1	0	0	0	0	0	0
Ht	4	1	0	0	1	0	2	2	26	31	1	0	0	0	0	0
Jb	0	0	0	0	0	0	0	0	2	0	48	50	0	0	0	0
Mv	0	0	0	0	3	2	0	0	2	0	0	0	59	67	7	2
Ms	0	0	0	0	2	0	0	0	0	0	1	0	3	5	60	61

(a) All terms after preprocessing, (b) T2CS$_{nr}$.

6.3 The Effect of Feature Selection

Table 2 shows how aggressive feature selection effects classification performance. We report results at FS levels tunned by using only training sets, and the number of terms which is the average value of training sets. When no FS methods are applied, i.e., using all of around 4400 terms after preprocessing, the overall precision, recall and F-measure obtained are lower than 89.7%. When applying our methods with either T2CS or T2CS-CHI metrics together with the new ranking scheme (indicated as T2CS$_{nr}$ and T2CS-CHI$_{nr}$), the three measures' values are higher than 95.35%. Those of existing FS methods, χ^2 and IG, are also high and respectively higher than 94.65% and 94.2%. The automatically determined FS levels of all the four FS methods are not higher than 320 terms. Thus, these FS methods improve classification performance significantly while using much smaller subsets than the set of all terms after preprocessing.

Table 3 presents the results of two methods "All terms after preprocessing" (column "a") and T2CS$_{nr}$ (column "b"), where Af, Am, Bk, Cr, Ht, Jb, Mv and Ms are abbreviation for 8 domains Airfares, Automobiles, Books, CarRentals, Hotels, Jobs, Movies and Musics respectively. In Table 3, for example, value 1 in the cell row "Cr" column "Af-b" means that one web source, which in fact belongs to CarRentals, has been assigned into Airfares when using T2CS$_{nr}$ method. (A cell value is the sum of the results given by validation sets.) Table 3 illustrates two results: (1) in general, when using T2CS$_{nr}$ method, better performance is obtained for most domains except Automobiles, specifically much better for two groups of closely related domains {Airfares, CarRentals, Hotels} and {Books, Movies, Musics}; (2) in particular, for Books, a domain with a highly heterogeneous vocabulary, T2CS$_{nr}$ method noticeably reduces mistakes in classifying sources from Hotels, Movies and Musics to it, as well as from it to Jobs and Movies.

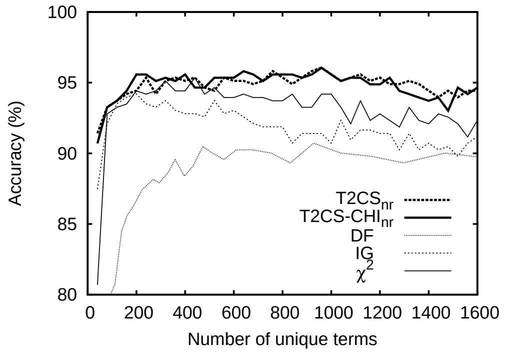
Consistently with prior studies in text categorization (Yang and Pedersen, 1997; Rogati and Yang, 2002), we find that rare words are not important. (This finding is in contrast to other works of SWS categorization, which weight rare words high (Lu et al., 2006; Barbosa et al., 2007)). As shown in Figure 1(a), DF method, which simply counts the number of documents containing the term in a whole collection, has an accuracy of 90.25% at 610 terms or an equivalent DF threshold of 16, and it maintains accuracy around this value through out higher FS levels. (We did not apply the technique of eliminating noisy words to this FS method.) As a result, a large number of "rare" terms can be removed without the loss of performance, as the total number of terms is around 16800.

6.4 Comparision of FS Methods

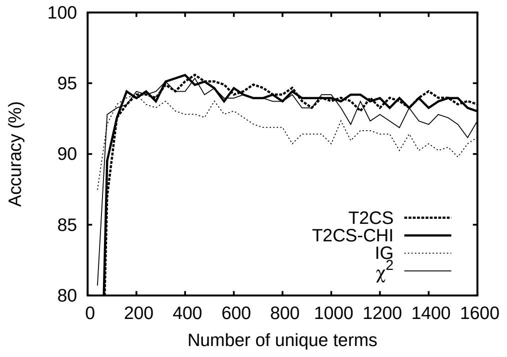
Figure 1(a) compares the accuracy of our FS methods, i.e., T2CS, T2CS-CHI metrics with the new ranking scheme (denoted as "T2CS$_{nr}$" and "T2CS-CHI$_{nr}$"), and that of existing FS methods, i.e., χ^2 and IG metrics with the standard ranking scheme (denoted as "χ^2" and "IG") at different FS levels. It can be observed that the new FS methods perform better than the existing methods not only at the optimal levels (i.e., an accuracy of little higher than 96% at 960 terms for our methods; 95.35% at 440 terms for χ^2; 94.2% at 200 terms for IG), but also on a wide range of FS levels. In addition, around the maximum point, our methods maintain steady high performance, while that of the existing methods decreases more quickly.

On the one hand, similar to prior studies in text categorization (Rogati and Yang, 2002; Gabrilovich and Markovitch, 2004), we observed that χ^2 and IG methods are the best performers of existing FS methods, and have the optimal FS ranges within the top 10% of all terms. Low-informative terms are considered redundant by these two methods, although they contain considerable information (Gabrilovich and Markovitch, 2004; Joachims, 1998). On the other hand, Figure 1(a) shows that while still giving high accuracy at relatively low FS levels (lower than 480 terms) like the other two methods, T2CS$_{nr}$ and T2CS-CHI$_{nr}$ methods move the optimal range to much higher FS levels (around 960 terms). This new trend suggests that comparing to the existing methods, our methods are able to sort out more low-informative but relevant terms, which help in improving performance as well keeping it steadily high.

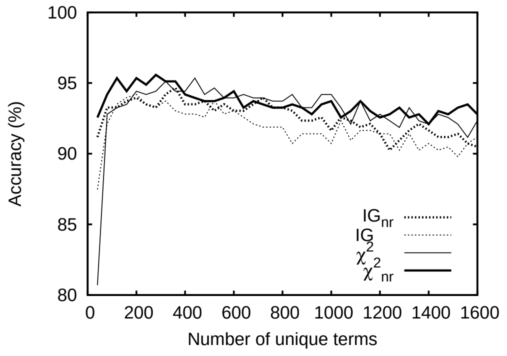
Figure 1(b) compares new metrics with existing metrics when using the standard ranking scheme. It shows that T2CS and T2CS-CHI metrics are slightly better than χ^2 metric, and clearly better than IG metric. Figure 1(c) compares the new and standard ranking schema when using existing metrics χ^2 and IG. It shows that the new ranking scheme is negligibly better than the standard scheme. Furthermore, two



(a) New metrics with new ranking (nr) vs. existing metrics with standard ranking.



(b) New metrics vs. existing metrics, both using standard ranking.



(c) New ranking (nr) vs. standard ranking, both using existing metrics.

Figure 1: Comparision of FS methods.

metrics T2CS-CHI and χ^2 differ only in the technique of selecting categories (see Section 4.1). Thus, it can be inferred that the approach of selecting the "top" two categories, together with the new ranking scheme, makes our FS methods noticeably better than the existing FS methods.

Lastly, it can be observed in Figures 1(a) and 1(b) that T2CS and T2CS-CHI metrics have similar performance regardless of the ranking schema. Meanwhile, these two metrics are different from each other only in $P(c_i|\bar{t})$ (see Section 4.1). Hence, the information that a term is absent from a document is not important when selecting features with our metrics.

7 CONCLUSIONS

In this paper, we study the problem of categorizing structured web sources by using their search interfaces. Our approach employs a filtering feature selection technique together with a Gaussian process classifier. In our research, we treated each search interface simply as a bag-of-words. We conducted experiments with our FS methods with new metrics and a novel simple ranking scheme, as well with existing FS methods. The experimental result indicates that: (1) feature selection techniques improve classification performance significantly; (2) our classification approach and the proposed FS methods are effective. Our research also points out that rare words are not important to the categorization task.

For future work, in terms of the classification method, one possible improvement to our research is to identify and use different feature types from a search interface. In terms of the feature selection technique, we plan to evaluate the effectiveness of the new methods in other text categorization problems.

REFERENCES

Barbosa, L. and Freire, J. (2005). Searching for hidden-web databases. In *WebDB'05*, pages 1–6.

Barbosa, L., Freire, J., and Silva, A. (2007). Organizing hidden-web databases by clustering visible web documents. In *ICDE'07*, pages 326–335.

Bergman, M. K. (2001). White paper - The Deep Web: Surfacing hidden value. Accessible at http://www.brightplanet.com.

Callan, J. P., Connell, M., and Du, A. (1999). Automatic discovery of language models for text databases. In *SIGMOD'99*, pages 479–490.

Chakrabarti, S., Dom, B., and Indyk, P. (1998). Enhanced hypertext categorization using hyperlinks. *ACM SIGMOD Record*, 27(2):307 – 318.

Chang, K. C.-C., He, B., Li, C., Patel, M., and Zhang, Z. (2004). Structured databases on the Web: Observations and implications. *SIGMOD Record*, 33(3):61–70.

Chang, K. C.-C., He, B., and Zhang, Z. (2005). Toward large scale integration: Building a MetaQuerier over databases on the web. In *CIDR'05*, pages 44–55.

Chawathe, S., Garcia-molina, H., Hammer, J., Irel, K., Papakonstantinou, Y., Ullman, J., and Widom, J. (1994). The Tsimmis project: Integration of heterogeneous information sources. *JIIS*, 8(2):7–18.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *JMLR*, 3:1289–1305.

Gabrilovich, E. and Markovitch, S. (2004). Text categorization with many redundant features: using aggressive feature selection to make SVMs competitive with C4.5. In *ICML'04*, pages 321–328.

He, B. and Chang, K. C.-C. (2003). Statistical schema matching across web query interfaces. In *SIGMOD'03*, pages 217–228.

He, B., Tao, T., and Chang, K. C.-C. (2004). Organizing structured web sources by query schemas: A clustering approach. In *CIKM'04*, pages 22–31.

He, H., Meng, W., Yu, C., and Wu, Z. (2005). WISE-Integrator: A system for extracting and integrating complex web search interfaces of the Deep Web. In *VLDB'05*, pages 1314–1317.

Ipeirotis, P. G., Gravano, L., and Sahami, M. (2001). Probe, count, and classify: categorizing hidden web databases. In *SIGMOD'01*, pages 67–78.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *ECML'98*, pages 137–142.

Levy, A. Y., Rajaraman, A., and Ordille, J. J. (1996). Querying heterogeneous information sources using source descriptions. In *VLDB'96*, pages 251–262.

Lu, Y., He, H., Peng, Q., Meng, W., and Yu, C. (2006). Clustering e-commerce search engines based on their search interface pages using WISE-Cluster. *DKE Journal*, 59(2):231–246.

Mladenic, D. (1998). Feature subset selection in text-learning. In *ECML'98*, pages 95–100.

Neal, R. M. (1997). Monte Carlo implementation of Gaussian process models for Bayesian regression and classification. Available from http://www.cs.toronto.edu/∼radford/.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. The MIT Press.

Rogati, M. and Yang, Y. (2002). High-performing feature selection for text classification. In *CIKM'02*.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM CSUR*, 34(1):1–47.

Soucy, P. and Mineau, G. W. (2001). A simple feature selection method for text classification. In *ICAI'01*.

UIUC (2003). The UIUC Web integration repository. Computer Science Dept., Uni. of Illinois at Urbana-Champaign. http://metaquerier.cs.uiuc.edu/repository.

Wu, W., Yu, C., Doan, A., and Meng, W. (2004). An interactive clustering-based approach to integrating source query interfaces on the Deep Web. In *SIGMOD'04*.

Yang, Y. and Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *ICML'97*, pages 412–420.

Zamir, O. and Etzioni, O. (1998). Web document clustering: a feasibility demonstration. In *SIGIR'98*, pages 46–54.