J.T. Yao, V.V. Raghvan, G.Y. Wang (Eds.): WSS'04, pp. 28-36, 2004.

Web-supported Matching and Classification of Business Opportunities

Jing Bai [1], François Paradis [1,2], Jian-Yun Nie [1]

1. *DIRO*
Université de Montréal
C.P. 6128, succursale Centre-ville
Montréal, Québec, H3C 3J7, Canada

2. *Nstein Technologies*
75, Queen Street, Suite 4400
Montréal, Québec, H3C 2N6, Canada

{baijing, paradifr, nie}@iro.umontreal.ca

Abstract

More and more business opportunities are published on the Web; however, it is difficult to collect and process them automatically. This paper describes a tool and techniques to help users discovering relevant business opportunities, in particular, calls for tenders. The tool includes spidering, information extraction, classification, and a search interface. Our focus in this paper is on classification, which aims to organize calls for tenders into classes, so as to facilitate user's browsing. We describe a new approach to classification of business opportunities on the Web using language modeling (LM) approach. This utilization is strongly inspired by the recent success of LM in IR experiments. However, few attempts have been made to use LM for text classification so far. Our goal is to investigate whether LM can bring improvement to text classification. Our experiments are conducted on two corpora: Reuters containing newswire articles and FedBizOpps (FBO) containing calls for tenders (CFTs) published on the Web. The experimental results show that LM-based classification can significantly improve the classification performance on both test corpora, compared with the traditional Naïve Bayes (NB) classifier. In particular, it seems to have stronger impact on FBO than on Reuters. This result shows that LM can greatly improve classification on the Web.

1. Introduction

Finding and selecting business opportunities is a crucial activity for businesses, yet they often lack the resources or expertise to commit to this problem. To ease this task, many electronic tendering sites are now available. They usually follow either a centralizing approach, where information is received directly from the contracting authorities (for example, in the case of TED [1]), or an aggregation approach, where documents are collected from other sites (for example, SourceCan [2]). Although the centralizing approach allows to control the contents and richness of the information, it is difficult to apply to some domains where there is no recognized authority, and is often limited to one geographic area. Furthermore, additional information which might exist on the Web is ignored. On the other hand, with the aggregation approach it is difficult to extract and categorize relevant information, since documents do not follow a common form or model, and their contents can vary widely.

Business-related documents, in particular Calls for Tenders (CFTs), are typically classified according to an industry standard, for example, NAICS (North American Industry Classification System) or CPV (Common Procurement Vocabulary, for the European Union). Some CFTs are manually classified with these codes, whereas some others are not. A classification algorithm is a natural addition to organize and search and CFT into a browsable directory. It can also provide multi-code classification for conversion between standards or different versions of standards. However, automated classification is difficult on CFTs, especially when they are taken from the Web, where their contents can vary a lot and there can be a large number of unseen terms.

In this paper, we propose to improve the classification of CFTs using a language modeling approach. A language model (LM) refers to a set of probability estimates on a training corpus. It also uses smoothing to deal with the obtained-zero probability problem of unseen words in the corpus. In a classification context, LM is used to estimate the probability of a word within a class. We propose to use these estimates within the Naïve Bayes (NB) method.

The paper will be organized as follows. In Section 2, we will briefly describe the MBOI project. In Section 3, we describe our approach to text classification using language models. Section 4 presents the experimental design and results on the

[1] `http://ted.publications.eu.int/`

[2] `https://www.sourcecan.com/`

Reuters-21578 and FBO data sets respectively. Finally, Section 5 gives some conclusions.

2. The MBOI Project

The MBOI project (Matching Business Opportunities on the Internet) deals with the discovery of business opportunities on the Internet. In the first phase of the project we have implemented a tool to aid a user in this process. It includes spidering, information extraction, classification, and a search interface.

The information relevant to business opportunities comes from various types of documents: press releases, solicitation notices, awards, quarterly reports, etc. We are not so much interested in modeling these documents, however, but rather in extracting and organizing information that will help finding CFTs: not only information within the CFT, but also related to contracting authorities, prior clients, etc. This information is crucial for business decisions. For this reason, we will refer to the documents as evidence, from which the information can be inferred.

Figure 1 shows the information inference process. At the core of the model is the CFT synthesis, which combines evidence from various sites. For example, if two sites contain a French and English version of the same CFT, the synthesis will include relevant attributes (title and description) in both languages. Other characteristics such as submission and execution dates, classification codes, submission procedure, etc. will also be inferred from the call for tenders notices. Amendments can replace or add to some or all of the elements of the synthesis.

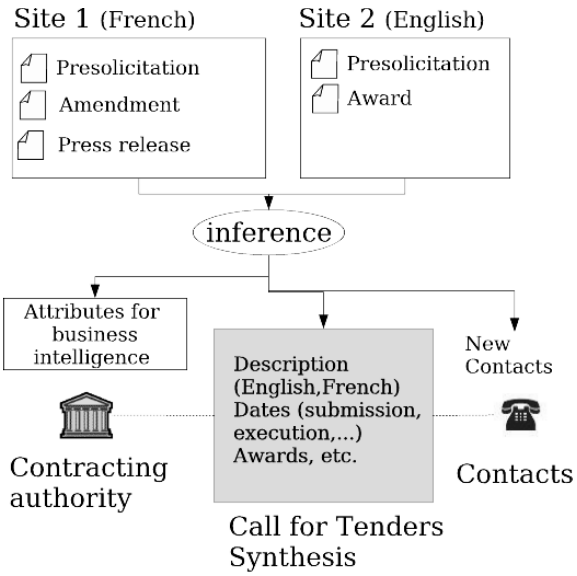


Figure 1: Information inference

Other information can add to the existing knowledge about contracting authorities and their contacts. These could later be used for business intelligence.

Since information can be extracted from several documents, there must be a strategy for the combination of evidence. Even for official documents such as call for tenders, there can be more than one version, published on the same site, or on several sites. Pairing these documents can be difficult if editors create their own solicitation numbers, sometimes without explicit reference to the contracting authority. We thus define a confidence measure on the inferred information. This confidence measures the validity of inference rules. It can also reflect the confidence of the source of the information: for example a contracting authority publishing its own documents can be deemed more trustworthy then an aggregator site.

Figure 2 shows a simplified example of a presolicitation notice and its amendment, regarding a contract for the office supplies of the Saskatchewan government. Both documents were fetched from the Merx site. From these documents, the system infers a synthesis with extracted information such as: publication and closing dates, title (both French and English), contact, etc. It also classifies the CFT: in this case, to NAICS code "418210" ("Stationery and Office Supplies Wholesaler-Distributors"). The synthesis is stored in an XML format inspired by xCBL (Common Business Language) and UBL (Universal Business Language) [5].

Presolicitation (on Merx):

Reference Number: CFAB4

Source ID: PV.MN.SA.213412

Published: 2003/10/08

Closing: 2003/10/28 02:00PM

Organisation Name: Saskatchewan Government

Title (English): Office Supplies

Title (French): Fournitures de Bureau

Description: The Government of Saskatchewan invites tenders to provide office supplies to its offices in Regina. The supplier is expected to start delivery on December 5, 2003, and enter an agreement of at least 2 years.

Contact: Bernie Juneau, (306) 321-1542

Amendment (on Merx):

Reference Number: CFAB4

Description: The start delivery date has been revised to January 5, 2004.

Figure 2: A call for tenders

Figure 3 shows the MBOI system architecture. There are two main processes: indexing, i.e., creating an index with the information inferred from the Web documents, and querying/browsing, which is the search interface for the user.

The first step of indexing is to collect documents from Web sites. We use a robot that can connect with a username and password (for sites with restricted access), look for URL patterns, fill out forms, and follow links of a given form. The next step is the inference of information, which includes information extraction and classification. Finally, an index is created and organized by fields of information (i.e., corresponding to elements in the CFT synthesis).

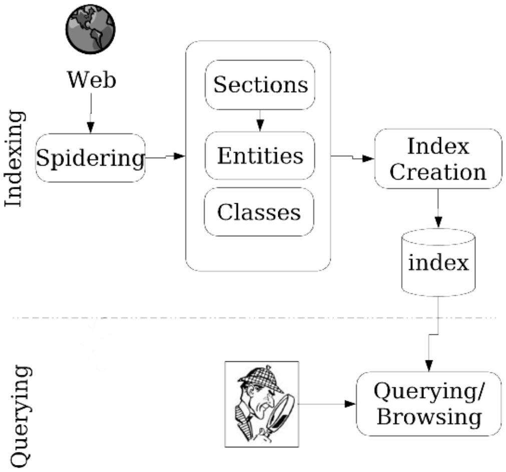
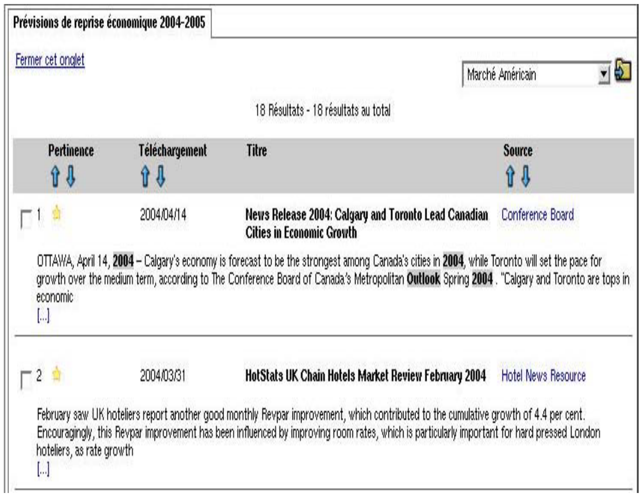


Figure 3. System architecture

The front-end to the system allows the user to search for CFTs by topic, date, class code, etc. or with an all-fields free text query. It also includes functionalities for browsing the class hierarchy, save the results in topic folders, etc. Figure 4 shows an example of results for a query about economic recovery. This is a saved query, i.e., one that has been defined by the user and is executed on a routine basis. This function is useful for a user who checks for a particular type of business opportunities on a daily basis.



Prévisions de reprise économique 2004-2005

Fermer cet onglet

Marché Américain

18 Résultats - 18 résultats au total

Pertinence	Téléchargement	Titre	Source
1	2004/04/14	**News Release 2004: Calgary and Toronto Lead Canadian Cities in Economic Growth**	Conference Board

OTTAWA, April 14, **2004** – Calgary's economy is forecast to be the strongest among Canada's cities in **2004**, while Toronto will set the pace for growth over the medium term, according to The Conference Board of Canada's Metropolitan **Outlook** Spring **2004** . "Calgary and Toronto are tops in economic
[...]

Pertinence	Téléchargement	Titre	Source
2	2004/03/31	**HotStats UK Chain Hotels Market Review February 2004**	Hotel News Resource

February saw UK hoteliers report another good monthly Revpar improvement, which contributed to the cumulative growth of 4.4 per cent. Encouragingly, this Revpar improvement has been influenced by improving room rates, which is particularly important for hard pressed London hoteliers, as rate growth
[...]

Figure 4. Querying in MBOI

The indexing and retrieval processes used in MBOI use the classical IR approaches of vector space model, with some enhancements to deal with structures of CFTs (e.g., section, title, etc.). We will not describe these processes in detail. Instead, we will concentrate on the classification process of CFTs in which we use a new method based on the statistical language modeling approach.

3. Using Languahe Models for Text Classification

Language models have been successfully applied in many application areas such as speech recognition and statistical NLP. Recently, a number of studies have confirmed that language model is also an effective and attractive approach for information retrieval (IR) [6, 11]. It not only provides an elegant theoretical framework to IR, but also results in effectiveness comparable to the best state-of-the-art systems. This success has triggered a great interest in IR community, and LM has since been used to other IR-related tasks, such as topic detection and tracking [7]. However, until now, few attempts have been made to use language models for text classification although there is a strong relationship between IR and classification.

Text classification aims to assign text documents into one or more predefined classes based on their contents. Many machine learning techniques have been applied to automatic text classification, such as Naïve Bayes (NB), K-Nearest Neighbor and Support Vector Machines (SVM).

Indeed, classification shares several common processings with IR. It is then possible that LM can also bring significant improvement to classification. Our goal in using language models to classification is to investigate whether language models can also improve the performance of classification. In particular, we will first integrate NB with language models, because we can observe a strong similarity between them.

3.1 Naïve Bayes Classifier

Let us first describe the principle of Naïve Bayes classifier.

Given a document d and a set of predefined classes $\{\ldots c_i, \ldots\}$, a Naïve Bayes classifier first computes the posterior probability that the document belongs to each particular class c_i, i.e., $P(c_i \mid d)$, and then assigns the document to the class(es) with the highest probability value(s). The posterior probability is computed by applying the Bayes rule:

$$P(c_i \mid d) = \frac{P(d \mid c_i)P(c_i)}{P(d)} \quad (1)$$

The denominator $P(d)$ in formula (1) is independent from classes; therefore, it can be ignored for the purpose of class ranking. Thus:

$$P(c_i \mid d) \propto P(d \mid c_i)P(c_i) \quad (2)$$

In Naïve Bayes, it is further assumed that words are independent given a class, i.e., for a document $d = d_1,\ldots,d_m$:

$$P(d \mid c_i) = \prod_{j=1}^{m} P(d_j \mid c_i)$$

Formula (2) can then be simply expressed as follows:

$$P(c_i \mid d) \propto \prod_{j=1}^{m} P(d_j \mid c_i)P(c_i) \quad (3)$$

In formula (3), $P(c_i)$ can be estimated by the percentage of the training examples belonging to class c_i:

$$P(c_i) = \frac{N_i}{N}$$

where N_i is the number of training documents in class c_i, and N is the total number of training documents respectively. $P(d_j \mid c_i)$ is usually determined by:

$$P(d_j \mid c_i) = \frac{1 + count(d_j, c_i)}{|V| + |c_i|}$$

where $count(d_j, c_i)$ is the number of times that term d_j occurs within the training documents of class c_i, $|V|$ is the total number of terms in vocabulary, and $|c_i|$ is the number of terms in class c_i. This estimation uses the Laplace (or add-one) smoothing to solve the zero-probability problem.

3.2 Language Modeling Approach in IR

Language modeling has been applied successfully in information retrieval [6, 11, 12] and several related applications such as topic detection and tracking [7]. Given a document d and a query q, the basic principle of this approach is to compute the conditional probability $P(d \mid q)$ as follows:

$$P(d \mid q) = \frac{P(q \mid d)P(d)}{P(q)} \propto P(q \mid d)P(d)$$

If we assume $P(d)$ to be a constant, then the ranking of a document d for a query q is determined by $P(q \mid d)$. The calculation of this value is performed as follows: We first construct a statistical language model $P(. \mid d)$ for the document d, called document model. Then $P(q \mid d)$ is estimated as the probability that the query can be generated from the document model. This probability is often calculated by making the assumption that words are independent (in a unigram model) in a similar way to Naïve Bayes. This means that for a query $q = q_1, \ldots, q_n$, we have:

$$P(q \mid d) = \prod_{j=1}^{n} P(w_j \mid d)$$

In previous studies, it turns out that smoothing is a very important process in building a language model [11]. The effectiveness of a language modeling approach is strongly dependent on the way that the document language model is smoothed. The primary goal of smoothing is to assign a non-zero probability to the unseen words and to improve the maximum likelihood estimation. However, in IR applications, smoothing also allows us to consider the global distribution of terms in the whole collection, i.e., the IDF factor used in IR [11].

Several smoothing methods such as Dirichlet, Absolute discount, etc., have been applied in language models. In Zhai and Lafferty [11], it has been found that the retrieval effectiveness is generally sensitive to the smoothing parameters. In our experiments on classification, we also observed similar effects.

3.3 Using Language Modeling Approach for Text Classification

If we compare Naïve Bayes with the general language modeling approach in IR, we can observe a remarkable similarity: the general probabilistic framework is the same, and both use smoothing to solve the zero-probability problem. The difference between them lies in the objects which a language model is constructed for and applied to. In IR, one builds a LM for a document and applies it to a query, whereas in NB classifier, one builds a LM for a class and applies it to a document. However, we also observe that in the implementation of NB, one usually is limited to the Laplace smoothing. Few attempts have been made in using more sophisticated smoothing methods.

As the experiments in IR showed, the effectiveness of language modeling strongly depends on the smoothing methods, and several smoothing methods have proven to be effective. Then a natural question is whether it is also beneficial in classification to use other sophisticated smoothing methods instead of the Laplace smoothing. In this paper, we will focus on this problem. As we will see later in our experiments, it will be clear that such a replacement can bring improvements to Naïve Bayes classifier. Another question we will examine is whether a LM classification approach will have similar impact on different types of documents.

3.3.1. Principle. The basic principle of our approach to text classification using language models is straightforward.

As in Naïve Bayes, the score of a class c_i for a given document d is estimated by formula (3). However, the estimation of $P(d_j \mid c_i)$ is different: It will be estimated from the language modeling perspective. First, we construct a language model for each class with several smoothing methods. Then $P(d_j \mid c_i)$ is the probability that the term d_j can be generated from this model. As smoothing turns out to be crucial in IR experiments, it is also necessary to carefully select the smoothing methods. In the next section, we will describe those that have been used in several IR experiments.

3.3.2. Smoothing Methods for Estimation. A number of smoothing methods have been developed in statistical natural language processing to estimate the probability of a word or an n-gram. As we mentioned earlier, the primary goal is to attribute a non-zero probability to the words or n-grams that are not seen in a set of training documents. Two basic ideas have been used in smoothing: 1) using a lower-order model to supplement a higher-order model; 2) modifying the frequency of word occurrences.

In IR, both ideas have been used. On the first solution, it is common in IR to utilize the whole collection of documents to construct a background model. This model is considered as a lower-order model to the document model, although both models may be unigram models. This solution has been useful for relatively short documents. Although a class usually contains more than one document, thus longer than a single document, the same problem of imprecise estimation exists, especially for small classes. Therefore, one can use the same approach of smoothing to classification. The second solution is often used in combination with the first one (i.e., one simultaneously use the collection model and change the word counts), as we can see in the smoothing methods described below.

Two general formulations are used in smoothing: backoff and interpolation. Both smoothing methods can be expressed in the following general form [12]:

$$P(w \mid c_i) = \begin{cases} P_s(w \mid c_i) & \text{w is seen in } c_i \\ \alpha_{c_i} P_u(w \mid C) & \text{w is unseen in } c_i \end{cases}$$

That is, for a class c_i, one estimate is made for the words seen in the class, and another estimate is made for the unseen words. In the second case, the estimate for unseen words is based on the entire collection, i.e., the collection model. The effect of incorporating the collection model not only allows us solving the zero-probability problem, but also is a way to produce the same effect as the IDF factor commonly used in IR (as shown in [11]).

In our experiments, we tested the following specific smoothing methods. All of them use the collection model.

- Jelinek-Mercer (JM) smoothing:

$$P_{JM}(w \mid c_i) = (1-\lambda)P_{ml}(w \mid c_i) + \lambda P(w \mid C)$$

which linearly combines the maximum likelihood estimate $P_{ml}(w \mid c_i)$ of the class model with an estimate of the collection model.

- Dirichlet smoothing:

$$P_{Dir}(w \mid c_i) = \frac{c(w, c_i) + \mu P(w \mid C)}{\mid c_i \mid + \mu}$$

where $c(w, c_i)$ is the count of word w in c_i, $|c_i|$ is the size of c_i (i.e., the total word count of c_i) and μ is a pseudo-count.

- Absolute discount smoothing:

$$P_{AD}(w \mid c_i) = \frac{\max(c(w, c_i) - \delta, 0) + \delta \mid c_i \mid_u P(w \mid C)}{\mid c_i \mid}$$

in which the count of each word is reduced by a constant $\delta \in$ [0, 1], and the discounted probability mass is redistributed on the unseen words proportionally to their probability in the collection model. In the above equation, $|c_i|_u$ is the number of unique words in c_i.

- Two-Stage (TS) smoothing [12]:

$$P_{TS}(w \mid c_i) = (1-\lambda)\frac{c(w, c_i) + \mu P(w \mid C)}{\mid c_i \mid + \mu} + \lambda P(w \mid C)$$

This smoothing method combines Dirichlet smoothing with an interpolation smoothing.

In the previous experiments of IR, it turns out that Dirichlet and Two-stage smoothing methods provided very good effectiveness. In our experiments, we will test whether these smoothing methods, when applied to text classification, bring similar impact.

4. EXPERIMENTAL EVALUATION ON CLASSIFICATION

4.1 Corpora

In order to compare with the previous results, our experiments have been conducted on the benchmark corpus of Reuters-21578, containing Reuter's newswire articles. We chose the ModApte split of Reuters-21578 data set, which is commonly used for text classification research today [9]. There are 135 topic classes, but we used only those 90 for which there exists at least one document in both the training and test set. Then we obtained 7769 training documents and 3019 test documents. The number of training documents per class varies from 2877 to 1. The largest 10 classes contain 75% of the documents, and 33% classes have fewer than 10 training documents.

In our experiments of finding business opportunities on the Web, we created a collection of CFT documents by downloading the daily synopses from the FedBizOpps (FBO) website, which are in the period from September 2000 to October 2003. This resulted in 21945 documents, which were split 70% for training and 30% for testing in our experiments. Notice that all the CFTs published on this site are manually classified using NAICS codes. NAICS codes are organized hierarchically, where every digit of a six-digit code corresponds to a level of the hierarchy. In order to reduce the class space, we only consider the first three digits in our current study. Although class hierarchy is an aspect that makes the classification of CFTs different from the general classification problem with flat classes, we will postpone this problem to a later study. That is, our current study will consider the set of classes at the same level. After removing the classes that do not included at least one document in both training and test set, we obtained 86 classes, 15312 training documents and 6627 test documents. The largest 10 classes contain 72% of the documents, and 30% classes have fewer than 20 training documents. We can see that the FBO collection has quite similar a distribution to the Reuters collection.

4.2 Performance Measure

For the purpose of comparison with previous works, we evaluate the performance of classification in terms of standard recall, precision and F_1 measure. For evaluating average performance across classes, we used macro-averaging and micro-averaging. Macro-averaging scores are the averages of the scores of each class calculated separately. Micro-averaging scores are the scores calculated by mixing together the documents across all the classes. Macro-averaging gives an equal weight to every class regardless how rare or how common a class is. On the other hand, micro-averaging gives an equal weight to every document, thus putting more emphasis on larger classes. In [9], it is claimed that micro-averaging can better reflect the real classification performance than macro-averaging. Therefore, our observations will be made mainly on micro-averaging F_1.

4.3 Naïve Bayes Classifier

To provide the comparable results of classification on Reuters-21578 corpus, we used the multinomial mixture model of Naïve Bayes classifier of the Rainbow package, developed by McCallum [3].

In NB classifier, feature selection is important. The effect of feature selection is to remove meaningless features (words) so that classification can be determined according to meaningful features. Several feature selection methods are commonly used: information gain (IG), chi-square, mutual information, etc. Information gain has shown to produce good results in [9]. The information gain of a word *w* is calculated as follows:

$$IG(w) = -\sum_{i=1}^{k} P(c_i)\log P(c_i) + \\ P(w)\sum_{i=1}^{k} P(c_i \mid w)\log P(c_i \mid w) + P(\overline{w})\sum_{i=1}^{k} P(c_i \mid \overline{w})\log P(c_i \mid \overline{w})$$

where $\overline{w}$ means the absence of the word *w*.

One can choose a fixed number of features according to their IG, or set up a threshold on IG to make the selection. The following table shows the classification results by NB without feature selection and with a selection of 2000 features according to IG. The number 2000 is suggested in [9].

Table 2 shows the classification results by NB without feature selection and with a selection of 12,000 features according to IG. The number 12,000 produced the best performance on FBO collection.

NB	miR	miP	miF$_1$	maF$_1$	Error
all features	0.6990	0.8668	0.7739	0.1838	0.00563
2K features	0.7145	0.8861	0.7911	0.3594	0.00520

miR: micro-averaging recall miP: micro-averaging precision

miF$_1$: micro-averaging F$_1$ maF$_1$: macro-averaging F$_1$

Table 1. Performance of NB on Reuters-21578 collection

NB	miR	miP	miF$_1$	maF$_1$	Error
all features	0.5144	0.5144	0.5144	0.1281	0.01129
12K features	0.5342	0.5342	0.5342	0.2572	0.01083

Table 2. Performance of NB on FBO collection

4.4 Language Modeling Approach

In the experiments using language models, we used the Lemur toolkit, which is designed and developed by Carnegie Mellon University and the University of Massachusetts [2]. The system allows us to train a language model for each class using a set of training documents, and to calculate the likelihood of a document according to each class model, i.e. $P(d \mid c_i)$. The final score of a class can then be computed according to formula (2).

4.4.1. Different Smoothing Methods. In our experiments, we used the four smoothing methods that are described earlier by varying the parameters. Table 3 shows the results by each method. No feature selection is made. The percentages in the table are the relative changes with respect to NB with no feature selection (Table 1).

Smoothing	miR	miP	miF$_1$	maF$_1$	Error
Jelinek-Mercer (λ=0.31)	0.7078	0.8778	0.7837 (+1.3%)	0.4659 (+153.5%)	0.00538
Dirichlet (μ=9500)	0.7051	0.8745	0.7807 (+0.9%)	0.3986 (+116.9%)	0.00546
Absolute (δ=0.83)	0.7118	0.8827	0.7881 (+1.8%)	0.4839 (+163.3%)	0.00527
Two-stage (λ=0.86,μ=6000)	0.7260	0.9003	0.8038 (+3.9%)	0.4214 (+129.3%)	0.00488

Table 3. Performance of LM on Reuters

As we can see, on Reuters-21578 corpus, the three first smoothing methods only lead to marginal improvements on micro-averaging F_1 over NB. On the other hand, Two-stage smoothing produces a larger improvement over NB.

The performances of different LMs on FBO collection are shown in Table 4.

Smoothing	miR	miP	miF$_1$	maF$_1$	Error
Jelinek-Mercer (λ=0.05)	0.5603	0.5603	0.5603 (+8.9%)	0.3725 (+190.8%)	0.01023
Dirichlet (μ=500)	0.5262	0.5262	0.5262 (+2.3%)	0.3486 (+172.1%)	0.01102
Absolute (δ=0.05)	0.5748	0.5748	0.5748 (+11.7%)	0.3791 (+195.9%)	0.00989
Two-stage (λ=0.05,μ=0)	0.5603	0.5603	0.5603 (+8.9%)	0.3725 (+190.8%)	0.01023

Table 4. Performance of LM on FBO

If we compare the three first smoothing methods (with their best performances shown in Tables 3 and 4), we can see that, the Absolute smoothing produced better performances than the other two smoothing methods on both corpora. Dirichlet smoothing produced the least improvements. Two-stage smoothing produced the largest improvement on Reuters. However, the phenomenon on the FBO collection is not the same. In the case of Two-stage smoothing on FBO, the best performance is obtained when μ is set to 0, i.e., we indeed use the Jelinek-Mercer smoothing. The differences of the smoothing methods on the two collections show that FBO has different characteristic than newswire articles, and they may require different classification methods.

Globally, our experiments show that using language models may improve classification effectiveness over Naïve Bayes on both corpora. This is true especially for macro-averaging F_1 which is much higher than with NB. The improvements on micro-averaging F_1 are more evident on the FBO collection than on Reuters-21578.

In order to test statistical significance of the changes of performance, we use the macro t-test [9], which compares paired F_1 values obtained for each class. It turns out that all the improvements obtained on both corpora with the four smoothing methods are statistically significant, with p-values < 0.001[3].

The comparison of the improvements on macro- and micro-averaging F_1 suggests that language models can bring larger improvements to small classes than to large classes. A possible reason is that our smoothing methods also combine the collection probabilities, instead of only changing the frequencies of words as in NB (Laplace smoothing). By modifying the frequency of words in Laplace smoothing, all the unseen words, either meaningful or not, will be attributed an equal probability. However, the smoothing methods with the collection model attribute different probabilities to unseen words according to their global distribution in the collection. Therefore, the latter probabilities can better reflect the characteristics of the collection and of the language. In our experiments, the addition of the collection model seems to benefit greatly small classes which have less training data and for which a heavy smoothing is required.

Another advantage of using the collection to smooth the class model is that the meaningless features that do not allow us to distinguish different classes are now "neutralized" with the collection model, in such a way that their differences across classes are weakened. This is equivalent to feature selection in the other classification methods. As we will see in Section 4.4.2, it turns out that feature selection is not necessary with LM. This confirms that smoothing has the same effects as feature selection.

The absolute level of performances on FBO is lower than that of Reuters. This suggests that the classification of CFTs, or more globally, the classification of business opportunities on the Web, is a more difficult problem than that for newswire articles. The main difference between them is that a CFT usually contains a very short description of the goods or services (one or a few sentences), which is the object of the call. The insufficient description makes it difficult to obtain a thorough characterization of the goods or service. On the other hand, the remaining parts, which take an important portion of the CFT, describe unessential elements for classification, such as the conditions of submission, the deadline, etc. These latter are not directly related to the classification by domain (although they may be useful for other purposes). By using the classical term weighting methods based on term frequency (or inverted document frequency), it is difficult to filter out the non-important parts of a CFT. These particularities make the global performances of classification on CFTs lower than for newswire articles.

4.4.2. Feature Selection with Language Model. Feature selection has been very useful for NB classifier. Would it produce a similar effect on language models? In order to

[3] A p-value lower than 0.05 is considered to be statistically significant at the 0.95 significance level.

answer this question, we conducted a series of experiments using different numbers of features selected according to information gain. The following table shows the results of doing feature selection on the four smoothing methods shown in Table 3.

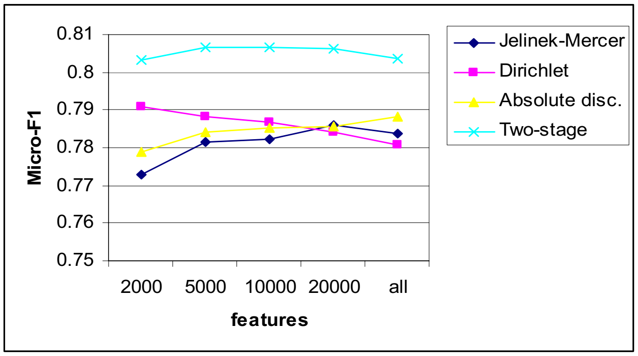


Figure 5. The effects of feature selection on Reuters

These results do not show significant performance improvement when we use feature selection, except for Dirichlet smoothing. On the contrary, for absolute smoothing and Jelinek-Mercer smoothing, the effect of feature selection is rather negative: We obtain lower performances if we select a subset of features. This conclusion seems contradictory to the results with NB, and counter-intuitive at the first glance. However, one can possibly explain this by the fact that, as the class model has been massively smoothed by the collection model, those non-discriminative features do not make a significant difference between documents with respect to a class. Therefore, the inclusion of such features in the calculation of the score does not hurt as much as in NB, which does not incorporate the collection model. This suggests that the consideration of the collection model in smoothing renders feature selection less necessary. Therefore, another important advantage of using LMs is that it can avoid the need for explicit feature selection.

5. Conclusion

We have described a tool to help the discovery of business opportunities on the Internet, and propose a new approach for the classification of such documents. The MBOI tool has been in use for a year and a half by our commercial partners, and deployed in several applications: as an aid for business opportunities watch for the St-Hyacinthe (Quebec) region, as a CFT search facility for the Canada's metal industry portal (NetMetal[4]), and as an "issue" or "thematic" watch for the Quebec travel industry. All have reported a significant improvement to their activities by using our system.

On classification, we used LM to enhance NB. In particular, the Laplace smoothing commonly used in NB is replaced by some other smoothing methods that integrate the collection model. Our experiments on Reuters-21578 and FBO collections have shown significant improvements over NB, especially on the macro-averaging F_1. On micro-averaging F_1, we also observed noticeable improvements, in particular, on FBO collection. This preliminary study did show that language models can contribute in improving text classification by NB.

Our comparison on two document collections show that language modeling approaches can be useful for the classification of both newswire articles and business opportunities on the Web, despite the differences between these documents. To further improve the classification performance of business opportunities, it will be necessary to study specific methods adapted to this type of data. In particular, we will have to deal with the problem with very short useful description in Calls for Tenders. We have noticed quite a bit of noise in the FBO documents in terms of irrelevant content, for example, pertaining to procedural instructions rather than the topic of the CFT. This is typical of Web documents, and therefore we think that it is quite encouraging that the improvement using LM was greater on FBO (a Web corpus) than on Reuters (a controlled test collection).

Our preliminary study is limited to the utilization of unigram models. We will investigate the integration of bigram language models for text classification in our future work. Other future works include: extending hierarchical classification, incorporating LMs into other classification algorithms, and using other types of features in classification (e.g., concepts, named entities as extracted using Nstein's tools).

ACKNOWLEDGMENT

This work has been carried out within a joint research project with Nstein technologies. We would like to thank Nstein and NSERC for their support.

REFERENCES

[1] S. T. Dumais, J. Platt, D. Heckerman and M. Sahami (1998). Inductive learning algorithms and representations for text categorization. *In Proceedings of ACM-CIKM98*, Nov. 1998, pp. 148-155.

[2] The lemur toolkit for language modeling and information retrieval. http://www-2.cs.cmu.edu/~lemur

[3] A. McCallum (1996). Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/~mccallum/bow

[4] A. McCallum and K. Nigam (1998). A comparison of event models for Naïve Bayes text classification. *In Proceedings of AAAI-98 Workshop*, AAAI Press.

[5] E. Dumbill. High hopes for the universal business language. *XML.com*, *O'Reilly*, November 7 2001.

[6] J. Ponte and W. B. Croft (1998). A language modeling approach to information retrieval. *In Proceedings of SIGIR 1998*. pp. 275-281.

[7] M. Spitters and W. Kraaij (2001), TNO at TDT2001: language model-based topic detection, *In Proceedings of Topic Detection and Tracking (TDT) Workshop 2001.*

[4] http://www.netmetal.net/

[8] Y. Yang (1999). An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*, Vol. 1, No. 1/2, pp. 67–88.

[9] Y. Yang and X. Liu (1999). A re-examination of text categorization methods. *In Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 42-49.

[10] Y. Yang (2001). A study on thresholding strategies for text categorization. *In Proceedings of SIGIR 2001*, pp 137-145.

[11] C. Zhai and J. Lafferty (2001). A study of smoothing methods for language models applied to ad hoc information retrieval. *In Proceedings of SIGIR 2001*, pp. 334-342.

[12] C. Zhai and J. Lafferty (2002). Two-stage language models for information retrieval. *In Proceeding of SIGIR 2002*, pp. 49-56.