

Integrating Compound Terms in Bayesian Text Classification

Jing Bai, Jian-Yun Nie, Guihong Cao
Département d'Informatique et de Recherche Opérationnelle,
Université de Montréal
C.P. 6128, succursale Centre-ville
Montréal, Québec, H3C 3J7 Canada
{baijing, nie, caogui}@iro.umontreal.ca

Text classification usually assumed a word-based document representation. In this paper, we propose a new approach to integrate compound terms in Bayesian text classification. Compound terms are used as complementary features to single words. An acute problem is to consider their dependence with the component words. In this paper, we propose to use smoothing techniques to combine both compound term and word representations. Experiments have been conducted on two corpora. Our results show that this approach can slightly but steadily improve the classification performance on both test corpora.

1. Introduction

For text classification, a number of methods have been proposed, such as Naïve Bayes (NB), K-Nearest Neighbor and Support Vector Machines (SVM). Although, SVM has shown to be one of the best methods in previous studies, in practice, NB is still widely used for its high efficiency in training [6, 12]. Until now, most studies on text classification have been carried out using classical keywords (single words) as features, and they are assumed to be independent. It is known that words are not precise descriptors of document contents, and compound terms should also be used [1, 4, 11]. For example, a document about “computer architecture” can be much better represented by the compound term “computer architecture” than two separate single words “computer” and “architecture”. The aim of this paper is to propose a reasonable and tractable way to integrate compound terms into text classification.

When compound terms are used to represent document contents, they cannot be used alone, because (1) only part of the document contents can be represented by compound terms and the other part is still represented by single words; (2) not all compound terms can be detected in a text as they can be written in variant forms and fail to be recognized by an automatic process [1]. Therefore, it is obvious that compound terms can only be used as complementary descriptors to single words, instead of replacing the latter. With the presence of both compound terms and single words as descriptors of document contents, an acute problem is to consider the relationship between them, or a way to combine them in the classification process. This problem is important because we cannot assume independence between a compound term such as “computer architecture” and its component words, as in NB

method. The method we will use to combine compound terms and their components is smoothing, which is inspired from the language modeling approach in information retrieval (IR) [14, 17]. This approach turns out to be simple but effective. Our experiments conducted on two corpora show that our approach can bring slight but consistent improvements on classification accuracy in both cases.

2. Bayesian classification

2.1 Naïve Bayes Classifier

Given a document d and a set of predefined classes $\{\dots c_i, \dots\}$, a Naïve Bayes classifier computes the posterior probability $P(c_i | d)$, and assigns the document to the class with the highest probability value(s). That is, the best class $C(d)$ is determined by:

$$C(d) = \arg \max_{c_i} P(c_i | d) = \arg \max_{c_i} P(c_i) P(d | c_i) \quad (1)$$

In NB, it is assumed that words are conditionally independent given a class, i.e., for a document $d = d_1, \dots, d_m$:

$$C(d) = \arg \max_{c_i} P(c_i) \prod_{j=1}^m P(d_j | c_i) = \arg \max_{c_i} \left[\log P(c_i) + \sum_{j=1}^m \log P(d_j | c_i) \right] \quad (2)$$

2.2 Augmented Naïve Bayes

The independence assumption between words in a document is a major weakness of NB. One possible way to relax it is to incorporate some dependencies between words into the Bayesian structure. For example, Friedman and Goldszmidt [8] assume that dependencies among words form a tree structure. Then a Tree-Augmented Naïve Bayes (TAN) is used to characterize the relationships between words. Figure 1 shows an example of such a structure, in which the solid lines denotes dependencies between words. Then the Bayesian classification problem is stated as follows:

$$C(d) = \arg \max_{c_i} P(c_i) \prod_{d_j \in d} P(d_j | c_i, d_k) P(d_k | c_i)$$

in which $P(d_j | c_i, d_k)$ encodes a possible dependency between d_j and d_k according to the tree structure. As the tree dependence structure does not contain all the possible dependencies among words, one has to make a selection. In [8], the links between words are selected by the Minimal Description Length (MDL) principle. A similar approach has been used in probabilistic IR model [15]. In a similar vein (although not in classification context), Gao et al. [9] propose a statistical method to recognize the possible

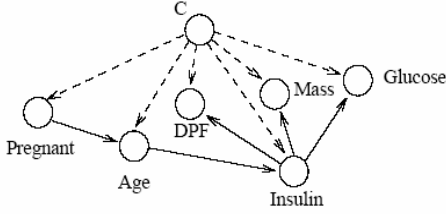


Figure 1. A TAN model learned for the data set “pima”. The dashed lines are those edges required by the naïve Bayesian classifier. The solid lines are correlation edges between features (from [8]).

dependencies between words in a sentence. Although these approaches allow us to relax the independence assumption to some degree, we observe that the basic representation is still based on single words. On the other hand, many NLP methods and tools have been developed to recognize more meaningful compound terms [11]. Such compound terms are complementary to the dependency determined in [8]. For example, while we can recognize a relationship between “Age” and “Pregnant” in a TAN structure (Fig. 1), if the concept in a document is indeed “golden age”, this concept may not necessarily have the same dependency with “Pregnant” as its component “age” does. This example shows that compound terms are diagonal to the TAN approach, and can be used to refine the latter. In this study, we will not investigate the combination of compound terms in a way similar to TAN. Instead, we will consider a representation with explicit compound terms, and investigate the role of compound terms in text classification, together with single words.

3. Compound terms as features

It is known that compound terms are often more meaningful items to represent document contents than single words. Many studies have been devoted to the recognition of compound terms or noun phrases in texts and their utilization in IR [1, 4, 7, 11]. In this study, we will not develop a new term extraction tool¹. We assume that the terms this tool recognizes are the only compound terms. The key question we deal with now is, when both compound terms and single words are used to represent document contents, how should they be related? In other words, how can we deal with the dependence between a compound term and its components?

Our approach then consists of two steps:

(1) For each sentence in a document, we first segment it into a sequence of non-overlapping segments. For example, the following sentence can be segmented into 5 segments:

This program provides extensive business management skills.

s_1 s_2 s_3 s_4 s_5

¹ <http://www.nstein.com>

In particular, the segment s_5 contains two overlapping compound terms: “business management” and “management skill”.

(2) We then assume that the segments are independent, but within a segment, words/terms are related.

Assuming independence between segments, for a document d containing independent segments s_j we have:

$$P(d | c_i) = \prod_j P(s_j | c_i)$$

and
$$C(d) = \arg \max_{c_i} \left[\log P(c_i) + \sum_{j=1}^m \log P(s_j | c_i) \right] \quad (3)$$

The problem now is to calculate $P(s_j | c_i)$ of a compound segment, such as “business management skill”.

3.1 Smoothing Compounds with Single Words

A compound segment such as “computer architecture” can be represented at least in two ways: as a compound term or as its components, which may be smaller compound terms or single words. The utilization of both representations has been the object of a large number of studies in IR [1, 4, 7, 11]. When dependency between compound and simple terms is considered, complexity is a major concern. In order to make the model more tractable, we propose to consider two parallel representations of a concept: one by compound term, and another by single words. We consider the following smoothing relationship between them: we want to represent the concept as precise as possible by compound terms, but some of the occurrences of this concept may also be in the form of one or both component words. Therefore, we also admit the representation by single words as an acceptable but less precise alternative representation of the concept. This is to say that the compound term representation can be smoothed with single words, as illustrated in the following diagram where * means a smoothing relationship.

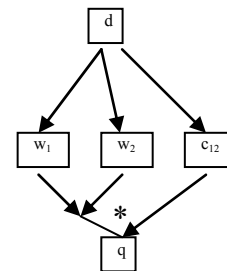


Figure 2. Diagram of the proposed approach

This smoothing can be compared to the traditional smoothing method [17]. Suppose we recognize “computer architecture” as a compound term. The smoothed estimation of $P(\text{“computer architecture”} | c_i)$ for the segment “computer architecture” may take the following form:

$$\begin{aligned} & P(\text{“computer architecture”} | c_i) \\ &= P_{ML}(\text{computer} | c_i) * P(\text{architecture} | \text{computer}, c_i) \end{aligned}$$

The second term can be smoothed with a unigram model as in traditional language modeling, i.e.,

$$\begin{aligned} & P(\text{architecture}|\text{computer},c_i) \\ &= \lambda P_{ML}(\text{architecture}|\text{computer},c_i) + (1-\lambda)P_{ML}(\text{architecture}|c_i) \\ &= \lambda \frac{P_{ML}(\text{architecture},\text{computer}|c_i)}{P_{ML}(\text{computer}|c_i)} + (1-\lambda)P_{ML}(\text{architecture}|c_i) \end{aligned}$$

where in the first part, $(\text{computer},\text{architecture})$ is considered together as a compound term, that we will denote as $\text{computer_architecture}$. Therefore,

$$\begin{aligned} & P(\text{"computer architecture"}|c_i) \\ &= \lambda P_{ML}(\text{computer_architecture}|c_i) \\ &+ (1-\lambda)P_{ML}(\text{computer}|c_i) * P_{ML}(\text{architecture}|c_i) \end{aligned}$$

This smoothing can be applied to the whole segment. Although it seems to work theoretically, we will encounter a serious practical problem: as a compound term becomes long, the first part of the above formula will be several magnitudes higher than the second part. Then the whole classification process will be dominated by the compound terms if they occur in a document to be classified. In order to solve this problem, we will use the idea of discriminative function [3, 5].

3.2 Using Discriminative Function

The idea of linear discriminant function is to combine a set of feature functions linearly. In our case, we define two types of feature function: one for compound terms and the other for its components. Our feature function is simply the log probability. Therefore,:

$$\text{score}(s_j, c_i) = \lambda \log P_c(s_j | c_i) + (1-\lambda) \log P_e(s_j | c_i)$$

where $P_c(s_j | c_i)$ is the probability of s_j as a compound term in class c_i , and $P_e(s_j | c_i)$ is the probability of s_j as being composed by its components. We can further set another parameter α in the following formula to balance the importance of the class probability:

$$C(d) = \arg \max_{c_i} \left[\alpha \log P(c_i) + \sum_{j=1}^m \text{score}(s_j, c_i) \right]$$

We will see that with a value of α different from 1, the results are slightly better.

It is also to be noted that it is not reasonable to determine a single λ for all types of decomposition. We assume that the same type of decomposition shares the same λ . In practice, only a few decompositions other than 2 to 1+1 will be detected. So we can be contented with just λ_{2-11} which corresponds to the decomposition of a 2-word term into 2 single words.

All the above parameters are estimated by using Simulated Annealing algorithm. The objective function is the micro-averaging F1 of the classification of some training documents. In order to avoid over-fitting, we

employ 10-fold Cross-Validation technique in determining these parameters.

4. Experiments

4.1 Setting

In order to compare with the previous results, our experiments have been conducted on the benchmark corpus of Reuters-21578², containing Reuter’s newswire articles. We chose the ModApte split of Reuters-21578 data set [11]. The second collection contains very different documents – Calls For Tenders (CFT) on the FedBizOpps website³, which are in the period from September 2000 to October 2003. All the CFTs published on this site are manually classified using NAICS codes. We use the third level NAICS codes in our experiments.

Table 1. Test corpora

	Training Docs	Test Docs	Categories
Reuters	7 769	3 019	90
FBO	15 132	6 627	86

For the purpose of comparison with previous works, we evaluate the performance of classification in terms of standard recall, precision and F_1 measure. We report both macro-averaging and micro-averaging of F_1 .

4.2 Results

In our previous experiments [2], we found that, for the class model $P(d_j|c_i)$ in formula 2, when we use some other smoothing techniques than Laplace (or add-one), the classification performance is higher. In particular, absolute discounting turned out to be a good choice.

Table 2 shows the classification effectiveness with different methods: “NB” is the classical NB approach with Laplace smoothing; “NB with AD” is the NB in which Absolute discount smoothing is used; “BCT” is our Bayesian classification approach which incorporates compound terms. The percentages are the relative changes with respect to “NB” (+) and “NB with AD” (++) respectively.

As we can see, on Reuters-21578 corpus, using Absolute discount smoothing in NB leads to some improvement on micro- F_1 , and a very large improvement on macro- F_1 . When we add compound terms, the performance is further improved. The parameter λ_{2-11} shown in the table is tuned by Simulated Annealing. Although the increase scale is relatively small, it is worth noting that the addition of compound terms consistently increases the classification effectiveness with any non-zero value of λs . Table 3 shows the performance with different values of λ_{2-11} . All the micro- F_1 values are higher than “NB with AD”.

We can observe slight improvements when we use the parameter α : when α is tuned to a different value than 1, we can obtain further improvements. One may also notice the

² <http://www.daviddlewis.com/resources/testcollections/reuters21578/>

³ <http://cbd.cos.com/>

Table 2. Classification performance on Reuters corpus

	miR	miP	miF ₁	maF ₁	Error
NB	0.6990	0.8668	0.7739	0.1838	0.0056
NB with AD ($\delta=0.83$)	0.7118	0.8827	0.7881 (+1.8%)	0.4839 (+163.3%)	0.0053
BCT $\lambda_{2-11}=0.95, \alpha=1.0$	0.7238	0.8977	0.8014 (++1.7%)	0.4921 (++1.7%)	0.0049
BCT $\lambda_{2-11}=0.95, \alpha=4.5$	0.7286	0.9036	0.8067 (++2.4%)	0.4823 (--0.3%)	0.0048

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

miF₁: micro-averaging F₁ maF₁: macro-averaging F₁

decrease in macro-F₁ while micro-F₁ increases in the last row of Table 2. This is not surprising because the objective function we used in Simulated Annealing is micro-F₁.

Table 3. Performance with different values of λ_{2-11}

λ_{2-11}	1	0.95	0.9	0.8	0.7	0.6
miF ₁	0.8005	0.8014	0.8008	0.8005	0.7999	0.7967
λ_{2-11}	0.5	0.4	0.3	0.2	0.1	0
miF ₁	0.7970	0.7970	0.7946	0.7928	0.7923	0.7881

On the second test collection FBO, the performances of our Bayesian classification approach are shown in Table 4. We can observe very similar but larger impact by integrating compound terms in classification.

Table 4. Classification performance on FBO corpus

	miR	miP	miF ₁	maF ₁	Error
NB	0.5144	0.5144	0.5144	0.1281	0.0113
NB with AD $\delta=0.1$	0.5716	0.5716	0.5716 (+11.1%)	0.3792 (+196.0%)	0.0100
BCT $\lambda_{2-11}=0.8, \alpha=1.0$	0.5849	0.5849	0.5849 (++2.3%)	0.3829 (++1.0%)	0.0097
BCT $\lambda_{2-11}=0.8, \alpha=8.0$	0.5912	0.5912	0.5912 (++3.4%)	0.3846 (++1.4%)	0.0095

When the values of λ s change, we observe similar impact as shown in Table 3. This second series of experiments further confirm that our method is not corpus-dependent. It can be used for classifying different types of documents.

Globally, our experiments show that incorporating compound terms in NB by means of interpolation smoothing consistently improves word-based NB method.

5. Discussions

Language modeling has been expensively used with success in IR in recent year. Despite of this, it has not been largely used in text classification, except in [13]. [13] uses smoothing to combine bigram and unigram class models. In our study, we use compound terms instead of bigrams.

On combining compound terms with single words, we showed that the idea of smoothing is naturally comparable to that used in language modeling. This leads to a tractable approach that is much less complex than the ones proposed by previous studies such as the one using TAN. In our experiments, it showed to be promising and resulted in consistent improvements in classification quality.

Our focus in this paper is only on the proper integration of compound terms with single words. We have not investigated the way in which compound terms are detected. It may be expected that the quality and the quantity of the extracted terms will have a great impact on the utilization of them. This problem is necessary to be studied.

The proposed approach is not restricted to text classification. It can also be used for IR. It would be interesting to investigate similar approaches to consider compound terms in language models for IR tasks.

References

- [1] A. Arampatzis, Th. P. van der Weide, C. H. A. Koster, and P. van Bommel (2000). Linguistically-motivated Information Retrieval. In *Allen Kent, editor, Encyclopedia of Library and Information Science*, volume 69, pp. 201-222. Marcel Dekker, Inc., New York, Basel.
- [2] J. Bai, F. Paradis and J. Y. Nie (2004). Web-supported Matching and Classification of Business Opportunities. *Workshop on Web-based Support Systems in Conjunction with WI*, pp. 28-36.
- [3] M. Collins and N. Duffy (2001). Convolution Kernels for Natural Language, *NLPS 14*.
- [4] W. B. Croft, H. R. Turtle, and D. D. Lewis (1991). The use of phrases and Structured Queries in Information Retrieval. *ACM-SIGIR91*, pp.32-45.
- [5] R. O. Duda, P. E. Hart, and D. G. Stork (2001). *Pattern Classification*. John Wiley & Sons, Inc.
- [6] S. T. Dumais, J. Platt, D. Heckerman and M. Sahami (1998). Inductive Learning Algorithms and Representations for Text Categorization. *ACM-CIKM98*, pp. 148-155.
- [7] J. L. Fagan (1987). Automatic Phrase Indexing for Document Retrieval: A Examination of Syntactic and Non-Syntactic Methods. PhD thesis, Department of Computer Science, Cornell University, Ithaca, New York 14853-7501.
- [8] N. Friedman, D. Geiger and M. Goldszmidt (1997). Bayesian Network Classifiers. *Machine Learning*, 29(2-3): 131-163.
- [9] J. Gao, J. Y. Nie, G. Wu and G. C (2004). Dependence Language Model for Information Retrieval. *SIGIR2004*, pp. 170-177.
- [10] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi (1983). Optimization by Simulated Annealing. *Science*, 220(4598):. 671-680.
- [11] D. D. Lewis and K. S. Jones (1996). Natural Language Processing for Information Retrieval. *Comm. of the ACM*, 39(1): 92-101..
- [12] A. McCallum and K. Nigam (1998). A Comparison of Event Models for Naïve Bayes Text Classification. *AAAI-98*.
- [13] F. Peng and D. Schuurmans (2003). Combining Naïve Bayes and N-gram Language Models for Text Classification. *25th European conf. on IR Research*, pp. 335-350.
- [14] J. Ponte and W. B. Croft (1998). A Language Modeling Approach to Information Retrieval. *SIGIR 98*, pp. 275-281.
- [15] C. J. Van Rijsbergen (1979). *Information Retrieval*, Butterworth-Heinemann, Newton, MA.
- [16] Y. Yang and X. Liu (1999). A Re-examination of Text Categorization Methods. *SIGIR99*, pp. 42-49.
- [17] Zhai and J. Lafferty (2001). A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval. *SIGIR01*, pp. 334-342.