고품질 바이그램을 이용한 문서 범주화 성능 향상

이 찬 도[†] · 체이드멩 탄[††] · 유안팡 왕[†††]

요 약

This paper presents an efficient text categorization algorithm that generates high quality bigrams by using the information gain metric, combined with various frequency thresholds. The bigrams, along with unigrams, are then given as features to a Naïve Bayes classifier. The experimental results suggest that the bigrams, while small in number, can substantially contribute to improving text categorization. Upon close examination of the results, we conclude that the algorithm is most successful in correctly classifying more positive documents, but may cause more negative documents to be classified incorrectly.

Improving Text Categorization with High Quality Bigrams

Chan-Do Lee[†] · Chade-Meng Tan[††] · Yuan-Fang Wang[†††]

ABSTRACT

본 논문은 정보이익을 사용하여 고품질 바이그램을 생성하는 효율적 문서 범주화 알고리즘을 제안한다. 실험 결과 유니그램에 적은 수의 바이그램을 추가해서 나이브 베이즈 분류기에 적용했을 때 문서 범주화 성공률은 상당히 향상되었다. 결과 분석은 제안한 알고리즘이 양의 문서를 분류하는데 더 우수하다는 것을 제시한다.

키워드 : 문서 범주화(Text Categorization), 문서 분류(Text Classification), 기계 학습(Machine Learning)

1. Introduction

At present, text categorization techniques are predominantly keyword-based. Many researchers in the field have used different classifiers, but most of them treat a document as a *bag of words*. They identify terms with all the words occurring in the document, and perform categorizations based mainly on the presence or absence of these keywords.

It is intuitively apparent that *phrases* make better features than single words. In many cases, a phrase describes the concept better than its component words. In other cases, the concept is described only by a phrase, not by its component words. For example, when classifying computer science documents the words "computer" and "science" are good descriptors, but the phrase "computer science" is an even better concept descriptor. In the cases of "artificial intelligence" and "neural nets", the phrases are much better features than their component words, since each of those words individually also describes concepts outside the computer science fields. More formally, phrases improve the performance of text categorization by disambiguating the indexing terms. Each of the words in a phrase provides a context that limits the meaning of the other words in the phrase. For example, the words "set" and "theory" are both highly ambiguous, yet the phrase "set theory" is much more specific.

However, in a number of experiments [1, 5], it has been found that the use of more sophisticated representations than single words, i.e., *phrases*, actually causes text categorization performance to degrade. Despite of these discouraging results, investigations of using phrases have been actively pursued [3, 6, 8, 9]. This paper presents our attempt to improve categorization performance by automatically extracting and using phrases, especially bigrams.

2. The Use Of Phrases In Text Categorization

Lewis [5] examined extensively the use of phrases in text

※ The first author was supported by Daejeon University for his sabbatical year at the Department of Computer Science, University of California, Santa Barbara, where this work was performed. The second and the third authors were supported in part by NSF grant IIS-9908441.
† 종신회원 : 대전대학교 컴퓨터정보통신공학부 교수
†† 비 회 원 : UCSB 대학원 컴퓨터과학과
††† 비 회 원 : UCSB 컴퓨터과학과 교수
논문접수 : 2001년 10월 10일, 심사완료 : 2002년 6월 11일

categorization and showed that phrases give *worse* performance than single words. The degradation in performance was due to that high dimensionality, low frequency, and high degree of synonymy using phrases as features outweigh the advantages phrases have in lowering ambiguity. Several efforts have been made to circumvent the possible problems posed by using phrases. Some research results showed that the addition of *n-grams* (sequences of words of length *n*) to the BOW representation indeed improved performance. However, sequences of length $n > 3$ were shown to be not useful and may decrease the performance.

Mladenić and Grobelnik [6] generated new features based on word sequences of different length up to 5, selected according to term frequency. They showed that using word sequences of length up to 3 instead of using only single words improved the categorization performance.

Fürnkranz [3] came to a similar conclusion. He used term frequency and document frequency as criteria in generating features. His experimental results indicated that word sequences of length 2 or 3 were most useful.

Schütze *et al.* [9] used single words and two-word phrases that were chosen by term frequency and showed that a reduced feature space was both practical and beneficial for document routing.

Schapire *et al.* [8] used words and phrases in text filtering. They also used term frequency as a criterion to choose which phrases to select.

Our approach is different in many aspects from the above-mentioned studies. First, we use bigrams *in addition to*, not *in place of*, unigrams. Bigrams are two-word phrases that occur adjacently. Second, we are highly selective of the bigrams we use to avoid high dimensionality. Our algorithm finds bigrams whose number does not exceed 2% of the number of unigrams. Finally, we use the information gain (*infogain*) measure in addition to term frequency and document frequency for feature selection. This means that the bigrams that we select are likely to be good discriminators and less likely to be noisy.

3. Algorithm

Our algorithm first finds the list of unigrams that appear in a significant number of documents, and then uses them as seeds. All the training documents are then scanned and we gather all bigrams where at least one of its component words is a seed. We then select only the bigrams, among those extracted, with high occurrences and infogain. (Figure 1) shows the pseudo-code of our algorithm.

```
1. Find S = { set of words that occur in at least df_seed *
             number of documents }
2. Set B = { }.
3. For each document in the training set
4. {
5.    Preprocess document by removing all function words.
6.    For each pair of adjacent words ( w1, w2 )
7.       If ( w1 ∈ S or w2 ∈ S ) add bigram "w1 + w2" to B.
8. }
9. For each b in B
10. {
11.    For each category c
12.     If ( number of b < df_thresh * number of documents in c )
13.       OR ( number of b < tf_thresh in all documents )
14.          remove b from B.
15.    If ( b is not removed and infogain of b < ig_thresh )
          remove b from B.
16. }
17. Output B.
```

(Figure 1) Bigram extraction algorithm

We performed some pilot experiments and the number 0.01 seemed to be good for *df_seed*, 0.005 for *df_thresh*, and 3 for *tf_thresh*. For *ig_thresh*, we set it to the infogain of the single word at position *igat_unigram* in the list of unigrams sorted by decreasing infogain. *igat_unigram* was set at approximately 1 percent of all unigrams. In our experiments, *igat_unigram* was set to 300 (about 1% of the 30,000+ unique unigrams).

4. Experiments

4.1 Procedure

The Reuters-21578 corpus was used in the experiment. Reuters-21578 is a manually classified collection of Reuters newswire articles appeared in 1987. Each story was assigned one or more indexing labels from a fixed list. The labels were broken into categories such as TOPIC and PLACES, but most researches consider only the 135 TOPIC labels for classification. This collection contains 21,578 documents with 30,765 unique words. Fully one half of the topic labels are represented by fewer than 10 documents (15 categories have no documents at all and 53 categories have 1~9 documents), and the 10 largest categories account for 75% of the total number of positive classifications in the corpus. Ninety categories have at least one training example and at least one test instance.

For this experiment, we used the "Modified Apté Split" (ModApte) that has 9603 training examples and 3299 testing examples, with 8676 unused. Like many other studies [2, 4, 7] we used top 10 most populous categories for experimen-

tation. <Table 1> shows the list of the top 10 categories in the ModApte split used in the current study.

<Table 1> Top 10 categories of the Reuters-21578 corpus in the ModApte Split

Category	Number of Training Docs	Number of Test Docs
earn	2877	1087
acq	1650	719
money-fx	538	179
grain	433	149
crude	389	189
trade	369	117[1]
interest	347	131
ship	197	89
wheat	212	71
corn	181[2]	56

All documents were preprocessed to retain only the bodies of each document by discarding headers and the likes. In addition, all numbers and punctuation were removed and all words set to lower case. Finally, all stopwords were removed using a standard stopwords list.

The categorization experiment was treated as a series of sub-tasks, each performing a binary categorization on the chosen category. For each category, we classified whether a document was in the category or not in the category. The experiments were run for categorization using only unigrams, and also for categorization using bigrams as well as unigrams.

The standard performance measures for text categorization are *recall* and *precision*. Recall is the percentage of total documents for the given topic that are correctly classified, while precision is the percentage of predicted documents for the given topic that are correctly classified. Each level of recall is associated with a level of precision. We can plot a graph that shows precision given different levels of recall. Such a graph is called *Precision-Recall graph*. In general, the higher the recall, the lower the precision, and vice versa. The point at which recall equals precision is the break-even point (*BEP*), which is often used as a single summarizing measure for comparing results. There are instances where a real BEP does not exist. Other useful measures for evaluating the effectiveness of classifiers are the F measures. Among many variants of them, *F1 measure* is used in this study, which is defined as :

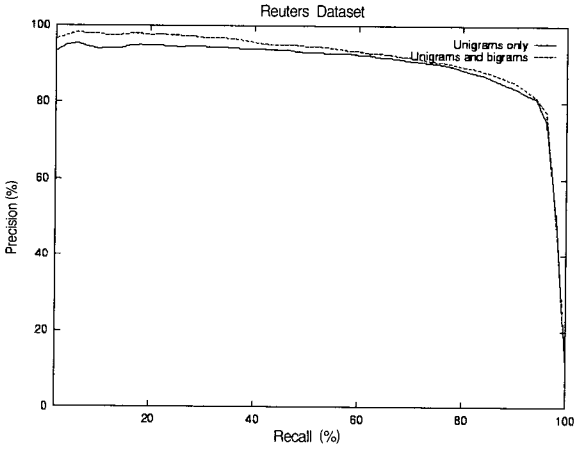
$$F1 = 2rp/(r+p),$$

where r denotes recall and p, precision.

4.2 Results

The algorithm was very successful at extracting bigrams that accurately describe some concepts, such as "united + states" and "crude + oil". Even though the number of bigrams was small relative to that of unigrams (there was only an average of 381 bigrams and about 30,000 unigrams), the former was represented better in terms of infogain : about 40 out of the top 100 features were bigrams. We also observed that the bigrams improved the overall quality of the feature set. Without bigrams, the average infogain of the features over all the categories was 1.37326-04. When the bigrams were added, it increased by 35.7% to 1.86369e-04.

As (Figure 2) shows, the overall performance improved when both unigrams and bigrams were used.



(Figure 2) Precision-Recall graph

<Table 2> summarizes the results for the entire corpus. BEP increased in all but one category, with the highest at 18.7%. However, the performance as measured by F1 was mixed. While the largest improvement remained at 3.4%, six out of ten categories showed a *drop*. The possible reasons of the degradation will be examined in the next section.

5. Discussion

As shown in <Table 2>, some categories showed a decrease in the F1 measure when bigrams were added. The major cause of it was that some bigrams over-emphasized concepts that were common to documents in both positive and negative categories. Here is an example. In the *acq* category of Reuters-21578, the addition of bigrams caused re-

1) Some literature shows the number as 118. The document 19918 lists "trade" twice in the <TOPICS> section.

2) Some shows as 182. The document 5467 lists "corn" twice as topics.

〈Table 2〉 Performance improvements

w/o means "unigrams only," and *w/i* denotes "bigrams added."

Category	BEP (w/o)	BEP (w/i)	Improvement (%)	F1 Measure (w/o)	F1 Measure (w/i)	Improvement (%)
acq	0.9617	0.9590	-0.3	0.958	0.953	-0.5
corn	0.5841	0.6727	15.2	0.480	0.474	-1.3
crude	0.8285	0.8812	6.4	0.845	0.831	-1.7
earn	0.9692	0.9702	0.1	0.944	0.968	2.5
grain	0.7759	0.8629	11.2	0.701	0.676	-3.6
interest	0.6844	0.7225	5.6	0.697	0.709	1.7
money-fx	0.6741	0.7521	11.6	0.703	0.717	2.0
ship	0.7934	0.8380	5.6	0.814	0.806	-1.0
trade	0.6383	0.7575	18.7	0.535	0.553	3.4
wheat	0.6434	0.7553	17.4	0.597	0.576	-3.5
Overall (micro-average)	0.8578	0.8657	0.9	0.838	0.841	0.4
Average (macro-average)	0.7553	0.8171	8.2	0.727	0.727	0.0

call to increase from 0.974 to 0.976 but caused precision to drop from 0.943 to 0.931. That drop caused the F1 measure to go down. The immediate reason for the drop in the precision was the increased number of false positives. (Figure 3) shows an example of one such case.

SOUTHEAST BANCORP ACTS ON BRAZILIAN DEBT

WASHINGTON, April 8 - Following the lead of other major banks, Southeast Banking Corp told the Securities and Exchange Commission it would place 54.2 mln dlrs of medium- and long-term Brazilian debt on non-accrual or cash status.

Based on current interest rates, it estimated in a filing that the move will reduce net income by about 800,000 dlrs in the first quarter and 3.2 mln dlrs for all of 1987. The company also said it did not believe the Brazilian debt situation would have a "material adverse" effect on it.

It also said it would issue 1,080,000 common shares in connection with its acquisition of Popular Bancshares Corp.

(Figure 3) Case of a false positive caused by addition of bigrams

Even though this document was originally classified as "NOT acq," it was changed to *acq*, since it contained the bigrams which were more common in the *acq* category than the "NOT acq" category, such as "common + shares," "exchange + commission," and "securities + exchange." The false positives occurred because concepts that were common to many documents in both positive and negative category were over-emphasized. The problem might be solved if our algorithm could find the right bigrams to reinforce the negative category. For example, if we had found bigrams such as "public + offering" and "repay + debt" to reinforce the "Not *acq*" category, then documents such as the one in (Figure 3) might not be wrongly classified.

It is immediately apparent that some categories benefited much more from the addition of bigrams than others. Why was it, then, that some categories did not do significantly better with bigrams? The reason is that our algorithm is good at increasing recall but not as good at increasing precision. This is the way it is expected to work, as the algorithm uses bigrams to reinforce existing unigrams and most of the bigrams found by our algorithm are from the positive category. Hence, they work better on the positive documents than on the negative ones. In other words, our algorithm is better at increasing correct positives than at reducing false positives. Hence, it works best in cases where recall is originally low because, in such cases, our algorithm can increase the performance by increasing correct positives. Precision-Recall graph (Figure 2) shows that when recall was low, our algorithm improved precision, but as recall rate went up the difference became smaller, leaving no room for improvement. This made the algorithm's performance dependent on precision. In fact, the only categories that exhibited F1 improvements were the ones that showed increases in precision, as shown in <Table 3>.

As demonstrated in the preceding paragraph, the strength of our algorithm is its ability to increase the number of positive documents classified correctly, but its weakness is that it may cause more negative documents to be classified incorrectly. The most likely reason is that our algorithm favors bigrams from the positive category. Indeed, we found that of all the bigrams found in our experiments, less than 5% came from the negative category. This happened because

〈Table 3〉 Recall and Precision for each category

Category	w/o			w/i			F1 Improvement (%)
	Recall	Precision	F1	Recall	Precision	F1	
acq	0.974	0.943	0.958	0.976	0.931	0.953	-0.5
corn	0.857	0.333	0.480	0.893	0.323	0.474	-1.3
crude	0.984	0.741	0.845	0.989	0.716	0.831	-1.7
earn	0.983	0.909	0.944	0.972	0.963	0.968	2.5
grain	0.960	0.552	0.701	0.953	0.524	0.676	-3.6
interest	0.924	0.560	0.697	0.947	0.566	0.709	1.7
money-fx	0.978	0.549	0.703	0.983	0.564	0.717	2.0
ship	0.888	0.752	0.814	0.888	0.738	0.806	-1.0
trade	0.872	0.386	0.535	0.940	0.391	0.553	3.4
wheat	0.930	0.440	0.597	0.930	0.418	0.576	-3.5
Overall (micro-average)	0.964	0.742	0.838	0.966	0.745	0.841	0.4
Average (macro-average)	0.935	0.617	0.727	0.948	0.616	0.727	0.0

we used the same criteria for finding bigrams in both categories, but the size of the positive category tended to be much smaller than that of negative category.

6. Conclusions

In this paper, we proposed an efficient algorithm to enhance the performance of text categorization using bigrams, demonstrated that bigrams can enhance the performance of the classifier, and analyzed the experimental results in some detail to find out the strengths and weaknesses of the proposed algorithm. Since our focus has been on finding whether adding bigrams can improve the classification performance, our experiments were done only using a single classifier, that is, a Naïve Bayes classifier. It would be desirable to run more experiments using various classifiers as well to see if adding bigrams indeed improves classification performances. The corpora we used for the current study might have contributed to the poor performances in some categories. It would be also desirable to run the experiment on the other standard corpora as well. These are the research directions we will follow in the near future.

References

[1] Apté, C., Damerau, F., and Weiss, S., "Automated learning of decision rules for text categorization," *ACM Transactions on Information Systems*, 12(3), pp.233-251, 1994.

[2] Dumais, S., Platt, J., Heckman, D., and Sahami, M., "Inductive Learning Algorithms and Representations for Text Categorization," In Gardarin *et. al.* (Ed.), *Proceedings of CKIM-98, 7th ACM International Conference on Information and Knowledge Management*, New York : ACM Press, pp.148-155, 1998.

[3] Fürnkranz, J., *A Study Using n-gram features for Text Categorization*, Technical Report OEFAI-TR-98-30, Austrian Research Institute for Artificial Intelligence, Vienna, Austria, 1998.

[4] Joachims, T., "Text Categorization with Support Vector Machines : Learning with Many Relevant Features," In Nedellec, C. and Rouveiro, C. (Ed.), *Proceedings of ECML-98, 10th European Conference on Machine Learning*, Heidelberg : Springer Verlag, pp.137-142, 1998.

[5] Lewis, D., *Representation and Learning in Information Retrieval*, Technical Report UM-CS-1991-093, Department of Computer Science, University of Massachusetts, Amherst, MA, 1992.

[6] Mladenić, D. and Grobelnik, M., "Word sequences as features in text learning," In *Proceedings of the 17th Electrotechnical and Computer Science Conference (ERK-98)*, Ljubljana, Slovenia, pp.145-148, 1998.

[7] Nigam, K., McCallum, A., Thrun, S., and Mitchell, T., "Text Classification from Labeled and Unlabeled Documents using EM," *Machine Learning*, 39, pp.103-134, 2000.

[8] Schapire, R, Singer, Y., and Singhal, A., "Boosting and Rocchio Applied to Text Filtering," In Croft *et. al.* (Ed.), *Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, New York : ACM Press, pp.215-223, 1998.

[9] Schütze, H., Hull, D., and Pederson, J., "A Comparison of Classifiers and Document Representations for the Routing Problem," In Croft *et. al.* (Ed.), *Proceedings of SIGIR-95, 18th ACM International Conference on Research and Development in Information Retrieval*, New York : ACM Press, pp.229-237, 1995.

Chan-Do Lee

e-mail : cdlee@dju.ac.kr

1975년 Department of German Education, Seoul National University (B.A.)

1984년 Department of German, Arizona State University (M.A.)

1987년 Department of Computer Science, Indiana University (M.S.)

1991년 Department of Computer Science, Indiana University (Ph.D.)

1991년~1992년 Senior Researcher, Center for Artificial Intelligence Research, KAIST

2000년~2001년 Visiting scholar, Department of Computer Science, University of California, Santa Barbara

1992년 to present Associate Professor, Division of Computer and Communication Engineering, Daejeon University

Research interests : Artificial Intelligence (Natural Language Processing, Intelligent Agents, Text Categorization), Soft Computing, Hypertext fiction

Chade-Meng Tan

e-mail : meng@google.com

1995년 Department of Computer Engineering, Nanyang Technological University (B.S.)

2000년 Department of Computer Science, University of California, Santa Barbara (M.S.)

2000년 to present Software Engineer, Google, Inc.

Research interests : Artificial Intelligence, especially Text Categorization

Yuan-Fang Wang

e-mail : yfwang@cs.ucsb.edu

B.S., National Taiwan University

M.S., University of Texas at Austin

Ph.D., University of Texas at Austin

Professor, Department of Computer Science, University of California, Santa Barbara

Research interests : Digital Library, Computer Vision, Medical Image Processing, Robotics, Computer Graphics