ISSN:0975-9646

Frequency and Compactness for Text Categorization

Sumayya Hasan Osmani, T.Naresh Kumar
Department of Computer Science & Information Technology, Jyothishmathi Institute of Tech &Sciences *JNTUH, Hyderabad, AP, INDIA*

***Abstract*—Text categorization is the task of assigning predefined categories to natural language text. With the widely used "bag-of-word" representation, previous researches usually assign a word with values that express whether this word appears in the document concerned or how frequently this word appears. Although these values are useful for text categorization, we also use naval values assigned to a word are called distributional features, which include the compactness of the appearance of a word, and the position of the first appearance of the word, but experiments show that the first position of the appeared is not enough to categorized the text because in some documents *last appeared* word can be more important than the first appeared. different features are combined using ensemble learning technique. Further analysis shows that the distributional features are especially useful when documents are long and the writing style is casual.**

Index Terms—*Text categorization, machine learning, distributional feature,*

INTRODUCTION

In the last 10 years, content-based document management tasks have gained a prominent status in the information system field, due to the increased availability of documents in digital form and the ensuring need to access them in flexible ways [30]. Among such tasks, Text Categorization assigns predefined categories to natural language text according to its content. Text categorization has attracted more and more attention from researchers due to its wide applicability. Since this task can be naturally modeled as a supervised learning problem, many classifiers widely used in the Machine Learning (ML) community have been applied, such as Naı¨ve Bayes, Decision Tree, Neural Network, k Nearest Neighbor (kNN), Support Vector Machine (SVM), and AdaBoost. Recently, some excellent results have been obtained by SVM and AdaBoost . While a wide range of classifiers have been used, virtually all of them were based on the same text representation, "bag of words," where a document is represented as a set of words appearing in this document. Values assigned to each word usually express whether the word appears in a document or how frequently this word appears. These values are indeed useful for text categorization. However, are these values enough? Considering the following example, "Here you are" and "You are here" are two sentences corresponding to the same vector using the frequency-related values, but their meanings are totally different. Although this is a somewhat extreme example, it clearly illustrates that besides the appearance and the frequency of appearances of a word, the distribution of a word is also important. Therefore, this paper attempts to design some distributional features to measure the characteristics of a word's distribution in a document. Note that the word "feature" in "distributional features" indicates the value assigned to a word, which is somewhat different from its usual meaning, i.e., the element used to characterize a document. The first consideration is the compactness of the appearances of a word. Here, the compactness measures whether the appearances of a word concentrate in a specific part of a document or spread over the whole document. In the former situation, the word is considered as compact, while in the latter situation, the word is considered as less compact. This consideration is motivated by the following facts. A document usually contains several parts. If the appearances of a word are less compact, the word is more likely to appear in different parts and more likely to be related to the theme of the document. For example, consider Document A (NEWID = 2,367) and Document B (NEWID =7154) in Reuters-21578. Document A talks about the debate on whether to expand the 0/92 program or to just limit this program on wheat. Obviously, this document belongs to the category "wheat." Document B talks about the US Agriculture Department's proposal on tighter federal standards about insect infections in grain shipments, and this document belongs to the category "grain" but not to the category "wheat." Let us consider the importance of the word "wheat" in both documents. Since the content of Document A is more closely related to wheat than Document B, the importance of the word "wheat" should be higher in Document A than in Document B. However, the frequency of this word is almost the same in both documents. Therefore, the frequency is not enough to distinguish this difference of importance. Here, the compactness of the appearances of a word could provide a different view. In Document A, since the document mostly discusses the 0/9 program on wheat, the word "wheat" appears in different parts of this document. In2 Document B, since the document mainly discusses the contents of the new standard on grain shipment and just one part of the new standard refers to wheat, the word "wheat" only appears in one paragraph of this document.

Thus, the compactness of the appearances of the word "wheat" is lower in Document A than in Document B, which well expresses the importance of this word. The second consideration is the position of the first appearance of a word. This consideration is based on an intuition that the author naturally mentions the important contents in the earlier parts of a document. Therefore, if a word first appears in the earlier parts of a document, this word is more likely to be important. Let us consider Document A (NEWID =3,981) and Document B (NEWID = 4,679) in Reuters-21578. Document A belongs to the category "grain" and talks about the heavy rain in Argentine grain area. Document B belongs to the category "cotton" and discusses that China is trying to increase cotton output. Obviously, the word "grain" should be more important in Document A than in Document B. Unfortunately, the frequency of the word "grain" is even lower in Document A than in Document B. Now, let us consider the position of the first appearance of the word "grain." In Document A, it first appears in the title. It is not strange,

since this document mainly talks about Argentine grain area. In Document B, the word "grain" first appears at the end of the document. It is not strange either. Since the theme of this document is about increasing cotton output, the suggestion that the production of cotton be coordinated with other crops such as grain is indirectly related to this theme, so the author naturally mentioned this suggestion at the end of the document. Obviously, the position of the first appearance of a word could express the importance of this word to some extent. But not in all cases where the word in document is not in first position this experiment does not work

Modeling a Word's Distribution:

In this paper, a word's distribution is modeled by two steps: first, a document is divided into several parts; then, the distribution of a word is modeled as an array where each element records the number of appearances of this word in the corresponding part. The length of this array is the total number of the parts.

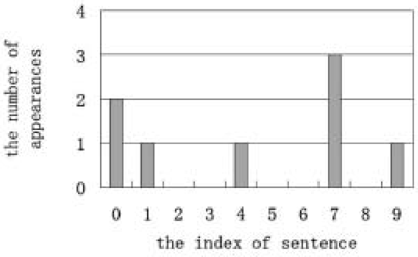

Fig. 1. The distribution of "corn."

Fig:1

For the above model, how to define a part becomes a basic problem. According to Callan, there are three types of passages used in information retrieval.2 Kim and Kim discussed the advantages and disadvantages of these three types of passages.

The *discourse passage* is based on logic components of documents such as sentences and paragraphs. The discourse passage is intuitive, but it has two problems: the length of passages is inconsistent, and sometimes, no passage decoration is provided for documents.

The *semantic passage* is partitioned according to contents. This type of passage is more accurate, since each passage corresponds to a topic or subtopic, but its performance is heavily influenced by the effect of the partition algorithm. The window passage is simply a sequence of words. The window passage is simple to implement, but it may break a sentence, and the length of window is hard to choose. Considering efficiency, the semantic passage is not used in the following experiments.

The *discourse passage* and window passages with different sizes are explored, respectively. Note that the window passage used in this paper is no overlapped. Now, an example is given. For a document d with 10 sentences, the distribution of the word "corn" is depicted in graph then, the distributional array for "corn" is [2, 1, 0, 0, 1, 0, 0, 3, 0, 1].

SVM and kNN are two classifiers that achieved the best performance in a previous comparative study. Thus, in this section, all experiments are based on these two classifiers.

3.1 support vector machine (SVM)

A concept in computer science for a set of related supervise learning methods that analyze data and recognize patterns, used for classification and regression analysis. The standard SVM takes a set of input data and predicts, for each given input, which of two possible classes the input is a member of, which makes the SVM a non-probabilistic binary linear classifier Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on

3.2 ***k*-nearest neighbor algorithm (*k*-NN)**

Is a method for classifying objects based on closest training examples in the feature space. *k*-NN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. The *k*-nearest neighbor algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its *k* nearest neighbors (*k* is a positive integer, typically small). If $k = 1$, then the object is simply assigned to the class of its nearest neighbor.

THE CONTRIBUTION OF THIS PAPER IS THE FOLLOWING:

Distributional features for text categorization are designed. Using these features can help improve the performance, while requiring only a little additional cost.

How to use the distributional features is answered. Combining traditional term frequency

with the distributional features results in improved performance. But using first appearance of the word is not a always preferred.

The factors affecting the performance of the distributional features are discussed. The benefit of the distributional features is closely related to the length of documents in a corpus and the writing style of documents.

Extracting Distributional Features

Given a word's distribution, this section concentrated on implementing the two intuitively proposed distributional features.

For the compactness of the appearances of a word, three implementations are shown as follows (note that under the word distribution model mentioned above, the position of a word's appearance is just the index of the corresponding part):

$ComPact_{PartNum}$. The number of parts where a word appears can be used to measure the concept of compactness. This is a natural implementation of the idea proposed in the introduction part. As what is mentioned, a word is less compact if it appears in different parts of a document.

ComPact_{FLDist}. The distance between a word's first and last appearance is used to measure the compactness. It is motivated by the fact that, for a less compact word, the distance between the first mention and the last mention should be long. A slightly extreme example is the word that the author first mentions at the beginning of the document and then mentions again at the end of the document.

ComPact_{PosVar.} The variance of the positions of all appearances is used to measure the compactness. It is a natural implementation of the idea of compactness using the language of statistics. The mean position of all appearances is first calculated, and then, the mean distance between the position of each appearance and the mean position is calculated as the position variance.

For the position of the first appearance, this feature can be extracted directly from the proposed word distribution model.

Suppose in a document *d* containing n sentences, the distributional array of the word *t* is array (t, d) = [c_0,c_1, . . ., c_{n-1}] Then, the compactness *(ComPact)* of the appearances of the word *t* and the position of the first appearance *(FirstApp)* of the word *t* are defined, respectively, as follows:

$$FirstApp(t,d) = \min_{i \in \{0 \dots n-1\}} c_i > 0 ? i : n, \qquad (1)$$

$$ComPact_{PartNum}(t,d) = \sum_{i=0}^{n-1} c_i > 0 ? 1 : 0, \qquad (2)$$

$$\begin{aligned} LastApp(t,d) &= \max_{i \in \{0 \dots n-1\}} c_i > 0 ? i : -1, \\ ComPact_{FLDist}(t,d) &= LastApp(t,d) - FirstApp(t,d), \end{aligned} \qquad (3)$$

$$\begin{aligned} count(t,d) &= \sum_{i=0}^{n-1} c_i, \\ centroid(t,d) &= \frac{\sum_{i=0}^{n-1} c_i \times i}{count(t,d)}, \\ ComPact_{PosVar}(t,d) &= \frac{\sum_{i=0}^{n-1} c_i \times |i - centroid(t,d)|}{count(t,d)}. \end{aligned} \qquad (4)$$

Here, *exp = a?b : c* means that if condition *a* is satisfied, the value of expression *exp* is *b;* otherwise, the value is *c*. The example in Fig. 1 is used again to illustrate how to calculate the distributional features:

$$\begin{aligned} & FirstApp(corn,d) \\ & \quad = min\{0,1,10,10,4,10,10,7,10,9\} = 0, \\ & ComPact_{PartNum}(corn,d) \\ & \quad = 1+1+0+0+1+0+0+1+0+1 = 5, \\ & LastApp(corn,d) \\ & \quad = max\{0,1,-1,-1,4,-1,-1,7,-1,9\} = 9, \\ & ComPact_{FLDist}(corn,d) \\ & \quad = 9 - 0 = 9, \\ & count(corn,d) \\ & \quad = 2+1+1+3+1 = 8, \\ & centroid(corn,d) \\ & \quad = (2 \times 0 + 1 \times 1 + 1 \times 4 + 3 \times 7 + 1 \times 9)/8 = 4.375, \\ & ComPact_{PosVar}(corn,d) \\ & \quad = (2 \times 4.375 + 1 \times 3.375 + 1 \times 0.375 + 3 \times 2.625 \\ & \qquad + 1 \times 4.625)/8 = 3.125. \end{aligned}$$

Then, let us analyze the cost of extracting the term frequency and the distributional features. Suppose the size of the longest document in the corpus is *l*, the size of the vocabulary is *m*, the biggest number of parts that a document contains is *n*, and the number of documents in the corpus is *s*. Usually, a memory block with size *l* is required for loading a document, and an m * 1 array is required for recording the number of appearances of each word in the vocabulary. When the scan of a document is completed, the term frequency can be directly obtained from the above array. In order to extract the distributional features, an additional m * n array is needed, since for each word, an n *1 array is used to record the distribution of this word. When the scan of a document is completed, (1) - (4) are used to calculate the distributional features. No other additional cost is needed, compared with extracting the term frequency. Overall, the additional computational cost for extracting the distributional features is s *m * (Cost of (1)-(4)), and the additional storage cost is m * n. It is worth noting that the above additional computational cost is the worst case, since practically, the calculation is only required for words that appear at least once in a document. Actually, the number of such words in a document is significantly smaller than *m*. Generally, the additional computational and storage cost for extracting the distributional features is not big. The process of extracting the term frequency and the distributional features is illustrated in Fig. 2.

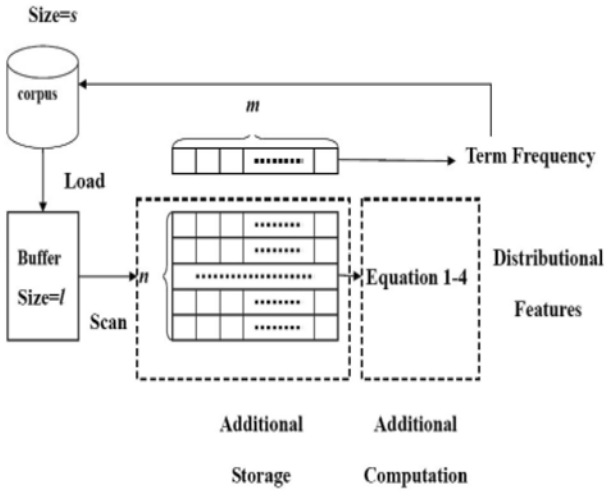

Fig: 2 The process of extracting the term frequency and distributional Features.

The extraction of the distributional features can be efficiently implemented using the inverted index constructed for the corpus. Many retrieval systems such as Lemur and Indri3 can support storing the positions of a word in a document in the index. Using such type of index, for a given word-document pair, we can obtain not only the frequencies of the word but also the positions where the word appears. With the position information and the length of the document, it is easy to construct the distribution of this word, and then, the distributional features can be computed.

CONCLUSION

Previous researches on text categorization usually use the appearance or the frequency of appearance to characterize a word. The distributional features encode a word's distribution from and the position of the first appearance and compactness of a word are used. Some aspects where word in document is last appeared this experiment fails these features are not enough for fully capturing the information contained in a document.

REFERENCES

[1]. L.D. BakerandA.K.McCallum,"Distributional Clustering ofWords for Text Classification," Proc. ACM SIGIR '98, pp. 96-103, 1998.

[2]. T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," Proc. 10th European Conf. Machine Learning (ECML '98), pp. 137-142, 1998.

[3]. Y. Ko, J. Park, and J. Seo, "Improving Text Categorization Using the Importance of Sentences," Information Processing and Management, vol. 40, no. 1, pp. 65-79, 2004

[4]. F. Li and Y. Yang, "A Loss Function Analysis for Classification Methods in Text Categorization," Proc. 20th Int'l Conf. Machine Learning (ICML '03), pp. 472-479, 2003.

[5]. S. Shankar and G. Karypis, "A Feature Weight Adjustment Algorithm for Document Classification," Proc. SIGKDD '00 Workshop Text Mining, 2000.