Blog Classification: Adding Linguistic Knowledge to Improve the K-NN Algorithm

Ines Bayoudh [1,2], Nicolas Bechet [2] and Mathieu Roche [2]

[1] INSAT

Université du 7 Novembre à Carthage

Centre Urbain Nord, Tunis

Tunisie

Ines.Bayoudh@lirmm

[2] LIRMM

UMR 5506, CNRS

Université Montpellier 2, France

Nicolas.Bechet @lirmm.fr, Mathieu.Roche@lirmm.fr

ABSTRACT: Blogs are interactive and regularly updated websites which can be seen as diaries. These websites are composed by articles based on distinct topics. Thus, it is necessary to develop Information Retrieval approaches for this new web knowledge. The first important step of this process is the categorization of the articles. The paper above compares several methods using linguistic knowledge with k-NN algorithm for automatic categorization of weblogs articles.

KEYWORDS: blog, categorization, linguistic knowledge, K-NN

1. Introduction

The work presented in this paper has been made with the collaboration of PaperBlog Company. This company hosts a website that proposes blog indexing, taken from partner websites. Blogs are similar to websites composed by articles chronologically or ante chronologically ranked. Each article is written like a log book which can be commented. This new type of websites, illustrating the concepts of Web 2.0, became very popular these last years due to its easiness of publication and its interactivity. However, blogs can be written in various ways of expression which constitute the main problem for information searching.

The main purpose of the Paperblog Company is to answer to the question: How to find an article of a specific theme from blogs? Thus, blog articles are evaluated according to their relevance and then associated to a category such as culture, computers, unusual, etc. This approach helps to retrieve information of a specific theme contained in blogs. The purpose of our work is to find a method which can automatically classify articles which is currently, manually done.

For this task, we chose to implement a classic algorithm of text classification: the K-Nearest Neighbor (K-NN). This classifier will be first implemented in a standard form then will be associated to different approaches by using Part-Of-Speech (POS) knowledge. Thus, we will be able to evaluate different data representations in order to determinate the most suitable one. We used a 3.4 Mb corpus of 2520 articles, written in French and composed by more than 400 000 words. This corpus is divided into 5 classes: food, talent, people, cooking, and market.

The following section introduces the state of the art of text classification and the K-NN algorithm, while the 3rd one will describe the grammatically-based approaches. Finally, the 4th section will describe the approach based on weighting of Tf-Idf matrices and will analyze the obtained results.

2. The state of the art of text classification

Our paper is based on a supervised approach with the automatic classification of blog articles in defined classes. We worked on manually classified articles provided by PaperBlog Company thus it is necessary to automate the categorization process. The purpose of this procedure is gathering articles which have the same thematic.

The learning process consists in realizing an automatic classifier which considers the characteristics of preordered examples. This classifier allows to add new articles and to find out their belonging category. The second part of our article presents two methods currently used in our categorization process.

– **The Support Vector Machine (SVM)**

Many methods use the SVM concept on multi-classes problem. However, they need several stages and everyone creates a new binary classification. The order of class processing has an influence on classification results. It was shown that SVM method needs more learning time (Joachim (1998)) than Naive Bayes or K-NN (described in the following section). The SVMs are more accurate when applied on text classification (Lewis et al. (2004)). A detailed description of SVM is introduced by (Burges (1998)).

– **Naive Bayes classifier**

These classifiers, based on the Bayes theorem, are defined as follows:

$$[1]\ P(h|D) = \frac{P(D|h) \times P(h)}{P(D)}$$

- P (h|D) = probability of h hypotesis given D (post probability),
- P (h) =H probability that h is independently verified of D (ex-ante probability),
- P (D) = Probability of observing D data regardless of h,
- P (D|h) = Probability of observing D knowing that H is verified.

This theorem supposes that the solutions can be found from probability distributions contained in the hypothesis and data. In case of texts classification, a Naive Bayesian classifier helps to determine the class of a specified document assuming that the documents are independent. This hypothesis of independence does not reflect the reality hence the name Naive. The class of a new object is determined after combining the predictions of all hypotheses by weighting them by their ex-ante probabilities. For a group of classes C and a set of attributes A, the value of c naive Bayesian classification is defined as follows:

$$[2]\ c = argmax\ P(c_j)_{c_j \in C} \prod_{a_t \in A} P(a_t | c_t)$$

This classification has been less efficient for text classification then the other methods (Weiss et al. (2005)). Nevertheless, it remains efficient when applied on incomplete data and can be used in many areas (legal, medical, economic, etc.).

These two methods are commonly employed in classification of texts containing a comparison (with opinion's comments) such as (Chen and al. (2006)).

There are other approaches for text categorization such as Decision Trees or DTree (Quinlan (1986)) or C4.5 (Quinlan (1993)). These trees determine rules (or terms) to separate and to classify texts according to their common attributes. We can also mention Artificial Neural Networks (NNet) which simulate the functioning of human neurons (Mcculloch et Pitts (1943)). The main inconvenience of this approach is the growth of calculating time with the size of learning corpus.

Finally, we introduce the K Nearest Neighbors (K-NN) which has been applied in our work. In fact, this method is very simple, quick to implement and provides satisfactory results (Yang (1999)). In addition, this method is still robust in case of incomplete data, which is quite common for blog articles. This approach will be detailed in the following section.

– **The K-NN algorithm**

The principle of K-NN algorithm (Cover et Hart (on 1967)) is to measure the similarity between a new document and all the documents already ordered. These documents can be considered as a learning dataset even if there is no learning phase in the K-NN algorithm.

This algorithm means constituting a vector space in which each document is represented by a vector of words. The dimension of a vector is the number of words it contains. Each element of this vector is constituted by the number of

words occurrences came from the learning set. The classified documents are decreasingly ordered so that the first document is the one with the highest score of similarity with the document to classify. Then, they are ordered according to the value of k, this made a classification of k closest documents. The measure of similarity usually used is the calculation of the cosine of the angle formed by both vectors of documents. The cosine between two vectors A and B is the scalar product of vectors A and B divided by the product of the norm of A and B. Having identified the k nearest neighbors, we have to define a methodology to assign a class to the new document. The second phase calculates the number of documents belonging to every category among the k closest one.

Let us take for example a document d to classify among four classes, C1, C2, C3 and C4. Let us define k = 6 and consider the following classification of dnew with the set of learning documents D containing documents di:

Table 1. Example of text classification using K-NN

Documents	Documents class
d1	C2
d2	C2
d3	C4
d4	C4
d5	C1
d6	C4

By using our approach, we would attribute the class C4 in dnew. Indeed, the class C4 is the one who possesses most documents among the k nearest neighbors (three documents).

In our experiments, we used two parameters:

- The threshold of class that fixes a minimal number of terms that must belong to a class so that a new document is assigned to this class,
- The threshold of similarity below which the new document will not be anymore allowed among the k nearest neighbors.

3. The used approaches

We propose in this paper, approaches establishing new representations of original corpus by using grammatical knowledge. To obtain such knowledge, we use a Part-Of-Speech Tagger.

3.1. The Part-Of-Speech TreeTagger

We chose the TreeTagger (Schmid (1995)), which allows texts labeling in several languages such us French. The step of TreeTagger is based on a set of trigrams, constituted by three consecutive Part-Of-Speech labels. For example, TreeTagger proposes the following results for the sentence: *The authors added linguistic information*

The	DT	the
authors	NNS	author
added	VVD	add
linguistic	JJ	linguistic
information	NN	information

The first column corresponds to the words of the sentence; the second one informs on the word category and the last one gives the lemmatized form. We propose to use these different information on various approaches presented in the following section.

3.2. The experimental approaches

We suggest using combinations of words with the categories: Noun (N), Verb (V) and Adjective (A). This approach consists in reconstituting a corpus which contains only the words belonging to the defined combination. Let us take for example the combination V_N: such a corpus will contain only verbs and nouns. The used combinations are: N, V, A, N_V, N_A, V_A, and N_V_A. We also define respectively the methods F and L for the corpus with inflected forms and the corpus in lemmatized form[1].

The following section presents the experimental protocol and the results obtained with our various approaches.

4. Experiments

[1] Using the TreeTagger

4.1. Steps of the experimental protocol

For our experiments, we compared the performances of the algorithm of k-NN by using different methods. This evaluation includes several stages:

- Deletion of the Html tags and the stop words (generic words often coming back in the text as "thus", "someone", etc.) from the corpus.
- Application of one of the presented methods.
- Application of crossed validation by segmenting the data in five groups.
- Calculation of the rate of error.

The rate of error, which measures the rate of badly classified articles, is defined as following:

[3] $$\text{Rate of error} = \frac{\text{Number of badly classified articles}}{\text{Total number of articles}}$$

4.2 Normalization of the corpus

The normalization of our corpus was obtained by calculating the Tf-Idf (Term Frequency x Inverse Document Frequency) which is a statistical measure used to evaluate the importance of a word in a corpus. The Term frequency measures the importance of the term Ti within the particular document Dj. The Inverse document frequency measures the general importance of the term. The measure of Tf-Idf is defined as follows:

[4] $$Wij = Tfij \cdot \log_2 (N/n)$$

With:

- Wij = weight of the term Tj in the document Di,
- Tfij = frequency of the term Tj in the document Di,
- N = number of documents in the collection,
- n = number of documents where the term Tj appears at least once.

We used a value of 2 for the threshold of class and 0.2 for the threshold of similarity because these values were experimentally considered as the most suited to our works. Consequently, these measures imply that certain articles can be considered as not classified.

4.3 Results

First of all, we measured the contribution of normalization (Tf-Idf) and lemmatization on our corpus by using the approaches L (lemmatized form) and C (original corpus). The table 1 presents the rate of error obtained with the application of these approaches. It shows that the lemmatization of the corpus tends to degrade the results in terms of error rate. However, by applying Tf-Idf, this tendency is reversed with better results for the lemmatized from (method L), which obtained the lowest rate of error.

Table 2. Evaluation of the advantages of lemmatization and normalizing

Approach	Error rate
C	0,39
C and Tf-Idf	0,25
L	0,42
L and Tf-Idf	0,21

The table shows that the N-V method (nouns and verbs) gives good results by considering the application of Tf-Idf. However, it equals the method L. These experiments show that verbs and adjectives contain less useful information compared with nouns.

Table 3. Table of error rate obtained for different approaches

Approach	Error rate	
	without Tf-Idf	with Tf-Idf
L	0,42	0,21
N	0,33	0,27
V	0,58	0,47
A	0,51	0,44
N_V	0,27	0,21
N_A	0,36	0,27
V_A	0,34	0,29
N_V_A	0,36	0,27

According to the experiments that we made, we can conclude that certain grammatical combinations brought more information than others and can improve the process of blogs classification. We wanted to exploit this point by granting more importance for these words and affecting them a more important weight than for the others. This weighting consists in the multiplication of the Tf-Idf of the word, which has a certain category, by a factor of weight.

Table 4. Table estimating the influence of the weight of 2 on the Tf-Idf matrix of a lemmatized corpus

Noun	Verb	Adjective	Error rate
1	2	1	0.31
1	1	2	0.30
2	1	1	0.31
2	2	1	0.29
1	2	2	0.31
2	1	2	0.23

The tables 5 and 6 present the results obtained with two values of weight (2 and 3). For example Noun: 3, Verb: 3, and Adjective: 1 corresponds respectively to a multiplication by 3, 3 and 1, in the Tf-Idf matrix.

According to the rate of error, we can notice an improvement of the obtained results for all the grammatical combinations and with the weight of 3. These results confirm that the combination of nouns and verbs realizes a finer classification with a very weak rate of error (0.06).

These results show that it is important to take into account all grammatical information (nouns, verbs, but also adjectives) giving different weights to the types of words to improve the classification tasks.

Table 5. Table estimating the influence of the weight of 3 on the Tf-Idf matrix of a lemmatized corpus

Noun	Verb	Adjective	Error rate
1	3	1	0.10
1	1	3	0.29
3	1	1	0.11
3	3	1	0.06
1	3	3	0.10
3	1	3	0.09

5. Conclusion

In this article, we presented an automatic categorization of blogs articles of the PaperBlog Company. We have used the algorithm of k Nearest Neighbors than we have compared with different approaches using Part-Of-Speech information. These experiments showed the advantages within the application of normalization. Then an important weight was assigned to the words which have a specific Part-Of-Speech tag (in our experiments, Nouns and Verbs). This improves the results of the categorization task.

In our future work, we will apply a machine learning approach to calculate the optimal weight to assign to the types of words. Moreover, we will experiment our approach with other categorization algorithms.

Acknowledgement

We are grateful to the PaperBlog Company (http://www.paperblog.fr/) for providing access to the Blog data, and to Nicolas Verdier and Maxime Biais in particular for their participation in this work.

References

Bergo, A. (2001). Text categorization and prototypes. Technical report.
Borko, H. et M. Bernick (1963). Automatic document classification. J. ACM 10(2), 151–162.
Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2(2), 121–167.
Chen, C., F. Ibekwe-SanJuan, E. SanJuan, et C. Weaver (2006). Visual analysis of conflicting opinions. vast 0, 59–66.
Cormack, R. M. (1971). "A review of classification" (with discussion). the Royal Statistical Society 3, 321–367.
Cornuéjols, A. et L. Miclet (2002). "Apprentissage artificiel, Concepts et algorithme"s. Eyrolles.
Cover, T. et P. Hart (1967). Nearest neighbor pattern classification. Information Theory, IEEE Transactions on 13(1), 21–27.
Joachims, T. (1998). "Text categorization with support vector machines: learning with many relevant features". In Proc. 10th European Conference on Machine Learning ECML-98, pp. 137–142.
Johnson, S. C. (1967). "Hierarchical clustering schemes". Psychometrika 32, 241–254.
Lewis, D. D., Y. Yang, T. G. Rose, et F. Li (2004). "Rcv1 : A new benchmark collection for text categorization research". Journal of Machine Learning Research 5(Apr), 361–397.
Mcculloch, W. et W. Pitts (1943). "A logical calculus of the ideas immanent in nervous activity". Bulletin of Mathematical Biophysics 5, 115–133.
Moulinier, I., G. Raskinis, et J. Ganascia (1996). "Text categorization : a symbolic approach". In In Proceedings of the Fifth Annual Symposium on Document Analysis and Information Retrieval, pp. 87–99.
Quinlan, J. R. (1986). Induction of decision trees. Mach. Learn. 1(1), 81–106.
Quinlan, J. R. (1993). C4.5 : programs for machine learning. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc.
Schmid, H. (1995). "Improvements in part-of-speech tagging with an application to german". In Proceedings of the ACL SIGDAT-Workshop, Dublin.
Vapnik, V. (1995). The Nature of Statistical Learning Theory. Springer, N.Y.
Weiss, S. M., N. Indurkhya, T. Zhang, et F. Damerau (2005). Text Mining : Predictive Methods for Analyzing Unstructured Information. Springer.
Yang, Y. (1999). An Evaluation of Statistical Approaches to Text Categorization. Information Retrieval 1(1-2), 69–90.

Yang, Y. et X. Liu (1999). “A re-examination of text categorization methods”. In SIGIR ’99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, pp. 42–49. ACM Press.