MEDLINE ABSTRACTS CLASSIFICATION
Average-based Discrimination for Noun Phrases Selection and Weighting Applied to Categorization of MEDLINE Abstracts

Fernando Ruiz-Rico, Jose Luis Vicedo and María-Consuelo Rubio-Sánchez
University of Alicante, Spain

Keywords: Text classification, MEDLINE abstracts.

Abstract: Many algorithms have come up in the last years to tackle automated text categorization. They have been exhaustively studied, leading to several variants and combinations not only in the particular procedures but also in the treatment of the input data. A widely used approach is representing documents as Bag-Of-Words (BOW) and weighting tokens with the TFIDF schema. Many researchers have thrown into precision and recall improvements and classification time reduction enriching BOW with stemming, n-grams, feature selection, noun phrases, metadata, weight normalization, etc. We contribute to this field with a novel combination of these techniques. For evaluation purposes, we provide comparisons to previous works with SVM against the simple BOW. The well known OHSUMED corpus is exploited and different sets of categories are selected, as previously done in the literature. The conclusion is that the proposed method can be successfully applied to existing binary classifiers such as SVM outperforming the mixture of BOW and TFIDF approaches.

1 INTRODUCTION

In order to arrange all data in MEDLINE database, each time a new document is added, it must be assigned to one or several MESH[1] terms. More than 100,000 citations are inserted every year, leading to a tedious task, hard to be completed. During the last decades, an important effort has been focused on developing systems to automate the categorization process. In this context, several statistical and machine learning techniques have been extensively studied. We can emphasize Rocchio's based approaches, Bayesian classifiers, Support Vector Machines, Decision Trees and k-Nearest Neighbours among others (Sebastiani, 1999; Aas and Eikvil, 1999; Yang and Liu, 1999). Most of them treat the classified items as feature vectors, where documents are transformed into vectors using the Bag-Of-Words (BOW) representation, where commonly each feature corresponds to a single word or token.

At a first sight, some problems may arise from using the simple BOW. First, a lot of linguistic information is lost, such as word sequence. Also different terms have different importance in a text, so we should think about how to quantify the relevance of a feature so that we have a valid indicative of the degree of the information represented. From an intuitive point of view, a simple consideration of phrases as features may increment the quality and quantity of information contained by feature vectors. For example, the expression “heart diseases” loses its meaning if both words are treated separately. Moreover, we can associate to each phrase sophisticated weights containing some statistical information such as the number of occurrences in a document, or within the whole training set or even how the phrase is distributed among different categories.

The paper is organized as follows. First, we have a look at previous efforts on the same matter by reviewing the literature and pointing out some relevant techniques for feature selection and weighting. Second, we try to remark the most important characteristics of our algorithm by explaining the intuitions that took us to carry out our experiments. Third, the details of the investigation are given by providing a full description of the algorithm and the evaluation procedure. Finally, several results and comparisons are presented and discussed.

2 RELATED WORK

The above observations have led numerous researchers to focus on enriching the BOW model for

[1] Medical Subject Headings. More information in *www.nlm.nih.gov/mesh/meshhome.html*

Ruiz-Rico F., Luis Vicedo J. and Rubio-Sánchez M. (2008).
MEDLINE ABSTRACTS CLASSIFICATION - Average-based Discrimination for Noun Phrases Selection and Weighting Applied to Categorization of MEDLINE Abstracts.
In *Proceedings of the First International Conference on Health Informatics*, pages 94-101

many years. Most of them have experimented with n-grams (n consequent words) (Scott and Matwin, 1999; Tan et al., 2002; Tesar et al., 2006) and others with itemsets (n words occurring together in a document) (Antonie and Zaane, 2002; Z. Yang and Zhanhuai, 2003; Tesar et al., 2006). In some cases, a significant increment in the performance was reported, but many times only marginal improvement or even a certain decrease was given.

This work proposes a new automatic feature selection and weighting schema. Some characteristics of this approach are based on ideas (noun phrases, meta-information, PoS tagging, stopwords, dimensionality reduction, etc.) that have been successfully tried out in the past (Basili et al., 2000; Granitzer, 2003; Moschitti and Basili, 2004). However, they have never been combined altogether in the way we propose.

For concept detection and isolation we use especial n-grams as features, also known as noun phrases (Scott and Matwin, 1999). The ones we propose are exclusively made of nouns that may or may not be preceded by other nouns or adjectives.

For selecting the relevant expressions of each category, a lot of approaches use only the TF (*term frequency*) and DF (*document frequency*) measures calculated over the whole corpus. This way, most valuable terms occur frequently and have a discriminative nature for their occurrence in only a few documents. We propose using this concept along with the use of TF and DF as individual category membership indicatives, since the greater they are for a particular category corpus, the more related category and term are. Additionally, we define and use *category frequency* (CF) as discriminator among categories. Each of these measures (TF, DF, CF, ...) isolates a set of relevant expressions for a category using its average as discrimination threshold (*average-based discrimination*). The final representative expressions for a category will be obtained by intersecting the sets of relevant expressions for each of the proposed measures.

Finally, relevant expressions are weighted according to a new schema that aggregates all these measures into a single weight.

Normalization over the feature weights has also been proved to be effective (Buckley, 1993). For example, it solves the problem of differences in document sizes: long documents usually use the same terms repeatedly and they have also numerous different terms, increasing the average contribution of their features towards the query document similarity in preference over shorter documents (Singhal et al., 1996).

Once each document in the corpus is represented as features, an algorithm must be provided to get the final classification. Although quite efficient category ranking approaches can be found in the literature (Ruiz-Rico et al., 2006), they are not suitable for MEDLINE abstracts indexation, where a document must be classified as relevant or not relevant to every particular MESH topic, by taking binary decisions over each of them. For this purpose, SVM is known to be a very accurate binary classifier. Since its complexity grows considerably with the number of features, it is often used together with some techniques for dimensionlality reduction.

3 SPECIFIC CONSIDERATIONS

This section highlights some concepts whose analysis is considered important before describing in detail the process of feature selection and weighting.

Parts of Speech and Roots. There are several tools to identify the part of speech of each word and to get its root (word stemming). To achieve the best performance, this paper proposes using both a PoS tagger [2] and a dictionary[3] working together.

Category Descriptors. Training document collections used for classification purposes are usually built in a manual way. That is, human beings assign documents to one or more categories depending on the classification they are dealing with. To help this process, and to be sure that different people use a similar criteria, each class is represented by a set of keywords (*category descriptors*) which identifies the subject of the documents that belong to that category. A document containing some of these keywords should reinforce its relation with particular categories.

Nouns and Adjectives. There are types of words whose contribution is not important for classification tasks (e.g. articles or prepositions) because concepts are typically symbolized in texts within noun phrases (Scott and Matwin, 1999). Also, if we have a look at the category descriptors, we can observe that almost all the words are nouns and adjectives. So, it makes sense to think that the word types used to describe the subject of each category should be also the word types to be extracted from the training documents to identify the category they belong to.

We must assume that it is almost impossible to detect every noun phrase. Moreover, technical corpus are continuously being updated with new words

[2] SVMTool (Màrquez and Giménez, 2004)

[3] *www-formal.standford.edu/jsierra/cs193l-project/ morphological-db.lisp*

and abbreviations. We propose considering these unknown terms as nouns because they are implicitly uncommon and discriminative.

Words and Expressions. When a word along with its adjoining words (a phrase) is considered towards building a category profile, it could be a good discriminator. This tight packaging of words could bring in some semantic value, and it could also filter out words occurring frequently in isolation that do not bear much weight towards characterizing that category (Kongovi et al., 2002). Moreover, it may be useful to group the words so that the number of terms in an expression (TL or text length) can be taken as a new relevance measure.

Non-descriptive expressions. The presence of neutral or void expressions can be avoided by using a fixed list of stopwords (Granitzer, 2003). However, if we only have a general list, some terms may be left out of it. We show that building this list automatically is not only possible but convenient.

Document's Title. The documents to be categorized have a title which briefly summarizes the contents of the full document in only one sentence. Some algorithms would discard an expression that only appears once or twice in a couple of titles because its relevance cannot be confirmed [4]. This paper proposes not only not discarding it, but giving it more importance.

4 FEATURE SELECTION AND WEIGHTING SCHEMA

There are two main processes involved in the task of building the category prototype vector for each category. First, the training data is analyzed in order to detect and extract the most relevant expressions (*expression selection*). These expressions will be used as dimensions of the category prototype vectors. Second, the category prototypes are weighted according to the training set (*expression weighting*).

4.1 Average-based Discrimination

An average or central tendency of a set (list) of values refers to a measure of the "middle value" of the data set. In our case, having a set E of n expressions where each expression is weighted according to a measure W $(\{w_1 \ldots w_n\})$, the average of the set E for the measure W (that we denote as $\overline{W}$) is defined as the arithmetic mean for the values w_i as follows:

$$\overline{W} = \frac{\sum_{i=1}^{n} w_i}{n}$$

Average discrimination uses the average of a measure W over a set E as threshold for discarding those elements of E whose weight w_i is higher (*H-Average discrimination*) or lower (*L-Average discrimination*) than $\overline{W}$ depending on the selected criteria. In the context of this work, this technique will be applied on different measures for selecting the most representative or discriminative expressions from the training data.

4.2 Expression Selection

The most relevant expressions are selected from the training data by using the *average discrimination measure* of different characteristics as cutting threshold.

4.2.1 Selecting Valid Terms

This process detects and extracts relevant expressions from each document as follows:

1. The words are reduced to their roots.
2. Only nouns and adjectives are taken into consideration. Any other part of speech (verbs, prepositions, conjunctions, etc.) is discarded. For this purpose, a PoS tagger and a dictionary are used. The words which are not found in the dictionary are considered to be nouns.
3. Sentences are divided into expressions: sequences of nouns or adjectives terminating in a noun. In regular expression form this is represented as "{Adjective, Noun}* Noun". For instance, the expressions extracted from "Ultrasound examinations detect cardiac abnormalities" are:

ultrasound
ultrasound examination
examination
cardiac abnormality
abnormality

This process will give us the set of valid terms (VT) in the whole collection. From now on the words 'term' and 'expression' are used interchangeably.

4.2.2 Computing Term, Document and Category Frequencies

Our starting point is m training collections, each one containing the training documents belonging to each category. Every subset is processed separately to compute the frequencies for each expression in all the

[4] A threshold of 3 is usually chosen (Granitzer, 2003), which means that terms not occurring at least within 3 documents are discarded before learning

categories. This way, the values required by the algorithm to get the final weights could be put in a matrix where columns correspond to expressions and rows correspond to categories:

	e_1	e_2	.	e_n	
c_1	TF_{11}, DF_{11}	TF_{12}, DF_{12}	.	TF_{1n}, DF_{1n}	N_1
c_2	TF_{21}, DF_{21}	TF_{22}, DF_{22}	.	TF_{2n}, DF_{2n}	N_2
...	...	...	.	...	...
c_i	TF_{i1}, DF_{i1}	TF_{i2}, DF_{i2}	.	TF_{in}, DF_{in}	N_i
...	...	...	.	...	...
c_m	TF_{m1}, DF_{m1}	TF_{m2}, DF_{m2}	.	TF_{mn}, DF_{mn}	N_m
	CF_1	CF_2	.	CF_n	

where:

- c_i = category i.
- n = number of expressions extracted from all the training documents.
- m = number of categories.
- TF_{ij} = Term Frequency of the expression e_j, that is, number of times that the expression e_j appears in all the training documents for the category c_i.
- DF_{ij} = Document Frequency of the expression e_j, that is, number of training documents for the category c_i in which the expression e_j appears.
- CF_j = Category Frequency of the expression e_j, that is, number of categories in which the expression e_j appears.
- N_i = number of expressions extracted from the training documents of the category c_i.

Every CF_j and N_i can be easily calculated from TF_{ij} by:

$$CF_j = \sum_{i=1}^{m} x_{ij} \; ; \; N_i = \sum_{j=1}^{n} x_{ij} \; ; \; x_{ij} = \begin{cases} 1 & \text{if } TF_{ij} \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

Across this paper, some examples are shown with the expressions obtained from the training process. The TF and DF corresponding to each expression are put together between brackets, i.e. (TF, DF).

Frequencies for Expressions in the Titles. Some experiments have been performed to get an appropriate factor which increases the weight of the expressions that appear in document's titles (Ruiz-Rico et al., 2006). Doubling TF and DF is proved to be a consideration which optimizes the performance.

4.2.3 Getting the Most Representative Terms (MRT)

The expressions obtained from documents are associated to the categories each document belongs to in the training collection. As a result, we will get m sets of expressions, each one representing a specific category.

For example, after analysing every document associated to the category "Carcinoid Heart Disease", some of the representative expressions are:

carcinoid disease (1,1)	tricuspid stenosis (1,1)
carcinoid heart (26,8)	ventric. enlargement (1,1)
carcin. heart disease (26,8)	ventricular failure (4,1)
carcinoid syndrome (8,3)	ventricular volume (1,1)
carcinoid tumour (3,2)	ventric. vol. overload (1,1)

For each category, we have to select the terms which best identify each category. Three criteria are used to carry out this selection:

- *Predominance inside the whole corpus*. The more times a term occurs inside the full training collection, the more important it is. L-Average discrimination using TF and DF over all the expressions in the courpus ($\overline{TF}$, $\overline{DF}$) is used to identify and select the best terms (BT) across the whole corpus.
- *Discrimination among categories*. The more categories a term represents, the less discriminative it is. Expressions appearing in more than half of the categories are not considered discriminative enough. Some authors use fixed stopword lists (Granitzer, 2003) for discarding expressions during the learning and classification processes. Our approach produces this list automatically so that it is adjusted to the number of categories, documents and vocabulary of the training collection.

  In this case, the set of category discriminative terms (CDT) for a category is obtained by removing expressions that are representative in more than half of the categories. That is, for every category, an expression e_j will be removed if:

  $$CF_j > (m/2+1)$$

  where m stands for the number of categories.
- *Predominance inside a specific category*. The more times a term occurs inside a category, the more representative it is for that particular category. L-Average discrimination using TF and DF values over all the expressions in each category i ($\overline{TF_i}$, $\overline{DF_i}$) are used to identify the best terms in a category (BTC_i).

So, we propose using these TF, DF and CF measures for dimensionality reduction as follows. For each category i:

1. Select the set of terms that are predominant inside the corpus (BT).
2. Select the set of terms that are discriminant among categories (CDT).
3. Select the set of terms that are predominant into this category (BTC_i).

The most representative terms of the category ($MRTC_i$) are obtained from the intersection of the three enumerated sets of terms:

$$\{MRTC\}_i = \{BT\} \cap \{CDT\} \cap \{BTC\}_i$$

As a result, we will get a subset of expressions for each category. For example, the category "Carcinoid Heart Disease" is identified by the following expressions:

carcinoid (44,8)	heart disease (40,9)
carcinoid heart (26,8)	tricuspid valve (11,5)
carcinoid heart disease (26,8)	

4.3 Expression Weighting

At this point, we have the most relevant terms for each of the m categories in the training set. These expressions are now weighted in order to measure their respective importance in a category. This process is accomplished as follows.

4.3.1 Normalization

The corpus of each category has its own characteristics (e.g. different number of training documents, longer or shorter expressions). So, we should not use the TF, DF and TL values directly obtained from the corpus. They can be normalized so that the final weights do not depend on the size of each category's training set neither on the differences on the averaged length over the representative expressions.

As also stated in (Singhal et al., 1996), we consider that expressions whose frequencies and lengths are very close to the average, are the most appropriate, and their weights should remain unchanged, i.e. they should get unit or no normalization. By selecting an average normalization factor as the pivot, normalized values for TF, DF and TL (TFn, DFn and TLn) are calculated in terms of proportion between the total values and the average over all the expressions in the category:

$$TFn_{ij} = \frac{TF_{ij}}{\overline{TF_i}} \;\; ; \;\; DFn_{ij} = \frac{DF_{ij}}{\overline{DF_i}} \;\; ; \;\; TLn_{ij} = \frac{TL_j}{\overline{TL_i}}$$

where i stands for the category c_i and j for the expression e_j respectively.

Normalized values higher than 1 indicate relevance higher than the average, therefore they point to quite significant expressions.

4.3.2 Expressions Matching Category Descriptors

Normalized values measure how much a term stands out over the average. Since category descriptors are special expressions which can be considered more important than the average, if an expression e_j contains some of the category descriptors of c_i, its normalized frequencies and length should be 1 or higher. To assure this, TFn, DFn and TLn are set to 1 for category descriptors with normalized weights lower than 1.

4.3.3 Weighting

All the proposed values are put together to get a single relevance measure (weight). The proposed weighting schema contains much more information than the common TFIDF approach. Usually, a single set of features is extracted from the training data, and each feature is assigned a single weight. We extract an individual set of features per category, obtaining also different weights for the same expression in different categories.

Every expression e_j is weighted for each of the categories c_i according the following formula:

$$w_{ij} = \frac{(TFn_{ij} + DFn_{ij}) \cdot TLn_{ij} \cdot TFnew_j}{CF_j}$$

where $TFnew_j$ stands for the single number of times that the expression e_j appears in the current document which is being represented as a vector. The greater w_{ij} becomes, the more representative e_j is for c_i. By following the intuitions explained in section 3, this equation makes the weight grow proportionally to the term length and frequencies and makes it lower when the term is more distributed among the different categories.

Since the goal is building a binary classifier, we must have a class c_p representing the positive samples in the training set. Intuitively, to get an even more separable case, the weights of the expressions representing c_p should be calculated differently from the ones representing other categories. For the latter case, we propose accumulating the weight of the negative classes. More formally, we obtain the weight w_j of the expression e_j as following:

$$w_j = \begin{cases} w_{pj} & \text{if } e_j \in c_p \\ \sum_{i=1}^{m} w_{ij} \; \forall i \neq p & \text{otherwise} \end{cases}$$

where w_{pj} stands for weight calculated from the positive samples, and $\sum_{i=1}^{m} w_{ij} \;\; \forall i \neq p$ represents the weight calculated from negative samples.

5 EVALUATION

Comparison to previous works is proposed using SVM against the simple BOW. We have represented the input data as feature vectors under the proposed schema (noun phrases) to make comparisons using a well-known training corpus such as the OHSUMED collection. Results will show that our method increases substantially the classification performance.

5.1 Classification Algorithm

For more accurate comparisons against previous works (Granitzer, 2003), SVM^{light} software (Joachims, 1999) has been used for evaluation purposes as a baseline classifier. All default parameters are selected except the cost-factor (“-j”) (Joachims, 2003), which controls the relative weighting of positive to negative examples, and thus provides a way to compensate for unbalanced classes. Leave-one-out cross-validation (LOO) (turned on by “-x 1” parameter) is used to compute a training set contingency table corresponding to each setting of “-j”. SVM^{light} is run multiple times for each category, once for each of the resulting values from 0.25 to 4.0 with 0.25 increments (e.g. 0.25, 0.5, 0.75 ... 3.75, 4.0).

Since SVM yields better error bounds by using euclidean norm (Zu et al., 2003), all feature vectors (both in the training and test set) are normalized to euclidean length 1.

5.2 Data Sets

The OHSUMED collection consists of 348,566 citations from medical journals published from 1987 to 1991. Only 233,445 documents contain a title and an abstract. Each document was manually assigned to one or several topics, selected from a list of 14,321 MESH terms. Since automating this process leads to a quite difficult classification problem, most of the authors use smaller data sets. We have chosen the diseases (Joachims, 1998) and heart diseases sub-trees (Granitzer, 2003).

For the diseases hierarchy, MESH terms below the same root node are grouped, leading to 23 categories. The first 10,000 documents in 1991 which have abstracts are used for training, and the second 10,000 are used for testing.

For the heart diseases sub-tree, the categories which have no training documents are discarded, leaving only 16,592 documents and 102 possible categories. The documents from 1987 to 1990 are used as the training set, and the 1991 ones are used as the test set.

5.3 Evaluation Measures

The algorithm performance has been evaluated through the standard BEP and F1 measures (Joachims, 2000):

$$F_1 = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision}$$

where recall is defined to be the ratio of correct assignments by the system divided by the total number of correct assignments, and precision is the ratio of correct assignments by the system divided by the total number of the system’s assignments. The precision and recall are necessarily equal (BEP) when the number of test examples predicted to be in the positive class equals the true number of positive test examples.

The relevant list of topics for each category is evaluated first, and the average performance score is calculated for all documents (micro-averaged) and for all categories (macro-averaged).

5.4 Relevance of the Parameters for the Classification Task

To compute the relevance of the parameters, the micro-averaged F1 performance is obtained from the original algorithm. After removing each parameter individually, the evaluation is performed again and the percentage of deterioration from the original algorithm is calculated.

Figure 1 reflects the influence of the main characteristics for the final categorization results. The following points describe the conditions applied for the different evaluations along with their associated labels in this figure:

- **Phrases**: Expressions are made of single words.
- **Phr.PoS**: Expressions are made of one single word of any part of speech (no dictionary nor PoS tagger are used).
- **Titles**: Expressions in the titles have the same weight as the other ones.
- **Cat.Des.**: Category descriptors do not have any influence for weighting the expressions.
- **TF-DF, TFnew, CF, TL**: TFn_{ij}, DFn_{ij}, $TFnew_j$, CF_j and TLn_j respectively do not have any effect during the weighting process. This is achieved by modifying the equation given in section 4.3.3 to omit in each case the indicated value:
  $TF - DF \Rightarrow w_{ij} = \frac{TLn_{ij} \cdot TFnew_j}{CF_j}$
  $TFnew \Rightarrow w_{ij} = \frac{(TFn_{ij} + DFn_{ij}) \cdot TLn_{ij}}{CF_j}$

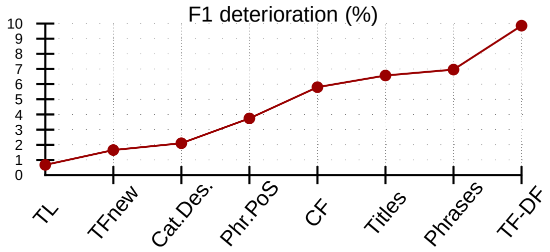

Figure 1: Deterioration in the performance after removing each parameter. Parameters are put from left to right in increasing order, from the least to the most relevant one. Tests performed over the diseases sub-tree data set.

Table 1: Number of features in relation with micro-averaged F1 performance. Results obtained over the 23 diseases categories.

	# features	F1
Noun words	2055	63.9
Any words	2221	66.0
Noun phrases	24823	68.6

$$CF \Rightarrow w_{ij} = (TFn_{ij} + DFn_{ij}) \cdot TLn_{ij} \cdot TFnew_j$$

$$TL \Rightarrow w_{ij} = \frac{(TFn_{ij} + DFn_{ij}) \cdot TFnew_j}{CF_j}$$

Figure 1 indicates how relevant each parameter is for the whole categorization process. It shows the percentage of deterioration in the performance when each parameter is removed from the algorithm.

The TF-DF measures lead the graph, meaning that term frequencies have a crucial significance as known from many other previous works. The use of phrases or noun phrases instead of single words of any type is the second most important parameter. The increment of the weights for those expressions in titles also improves significantly the performance. Category Frequency is the fourth most important parameter, which confirms that the more categories an expression represents, the less discriminative it is.

Documents from other corpora may be better represented by taking only single words as features and using simple weighting schemas (Dumais et al., 1998). However, it is not the same for OHSUMED. As far as we know, the results here presented are the best ever achieved, leading us to the conclusion that for some type of data such as MEDLINE documents, we should try to increment the number of features and the amount of information they contain, as confirmed in table 1.

6 RESULTS

Next tables show the results obtained in previous works followed by the results achieved by applying the new proposed algorithm for feature selection and weighting over the same training and test sets. The best values are in boldface.

Table 2: Break even point on 5 most frequent categories and micro-averaged performance over all 23 diseases categories (Joachims, 1998).

	SVM (words)	SVM (noun phrases)
Pathology	**58.1**	52.7
Cardiovascular	77.6	**80.9**
Immunologic	73.5	**77.1**
Neoplasms	70.7	**81.5**
Digestive system	73.8	**77.5**
Micro avg (23 cat.)	66.1	**68.6**

Table 3: Averaged F1 performance over 102 heart diseases categories (Granitzer, 2003).

	SVM (words)	SVM (noun phrases)
Micro avg	63.2	**69.9**
Macro avg	50.3	**55.5**

Table 2 shows the micro-averaged BEP performance calculated over the 23 diseases categories. Noun phrases gets a global 3.8% improvement, also outperforming almost all categories individually.

Table 3 contains both the micro and macro averaged F1 performance over the 102 categories of the heart diseases sub-tree. For this corpus we have achieved more than 10% improvements (10.6% for micro and 10.3% for macro measures respectively).

7 CONCLUSIONS

Using a proper feature selection and weighting schema is known to be decisive. This work proposes a particular way to choose, extract and weight special n-grams from documents in plain text format so that we get a high performing representation. Moreover, the new algorithm is fast, easy to implement and it contains some necessary adjustments to automatically fit both existing and incoming MEDLINE documents.

REFERENCES

Aas, K. and Eikvil, L. (1999). Text categorisation: A survey. Technical report, Norwegian Computer Center.

Antonie, M. and Zaane, O. (2002). Text document categorization by term association. *IEEE International Conference on Data Mining (ICDM)*, pages 19–26.

Basili, R., Moschitti, A., and Pazienza, M. T. (2000). Language-sensitive text classification. In *Proceeding of RIAO-00, 6th International Conference "Recherche d'Information Assistee par Ordinateur"*, pages 331–343, Paris, FR.

Buckley, C. (1993). The importance of proper weighting methods. *In M. Bates, editor, Human Language Technology Morgan Kaufman*.

Dumais, S. T., Platt, J., Heckerman, D., and Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. In Gardarin, G., French, J. C., Pissinou, N., Makki, K., and Bouganim, L., editors, *Proceedings of CIKM-98, 7th ACM International Conference on Information and Knowledge Management*, pages 148–155, Bethesda, US. ACM Press, New York, US.

Granitzer, M. (2003). Hierarchical text classification using methods from machine learning. Master's thesis, Graz University of Technology.

Joachims, T. (1998). Text categorization with support vector machines: learning with many relevant features. In Nédellec, C. and Rouveirol, C., editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 137–142, Chemnitz, DE. Springer Verlag, Heidelberg, DE. "Lecture Notes in Computer Science" series, number 1398.

Joachims, T. (1999). *Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning.* Software available at *http://svmlight.joachims.org/*.

Joachims, T. (2000). Estimating the generalization performance of a svm efficiently. In Langley, P., editor, *Proceedings of ICML-00, 17th International Conference on Machine Learning*, pages 431–438, Stanford, US. Morgan Kaufmann Publishers, San Francisco, US.

Joachims, T. (2003). *Support Vector and Kernel Methods*. SIGIR 2003 Tutorial.

Kongovi, M., Guzman, J. C., and Dasigi, V. (2002). Text categorization: An experiment using phrases. In *Proceedings of ECIR-02, 24th European Colloquium on Information Retrieval Research*, pages 213–228, Glasgow, UK.

Màrquez, L. and Giménez, J. (2004). A general pos tagger generator based on support vector machines. *Journal of Machine Learning Research*. Software available at *www.lsi.upc.edu/ nlp/SVMTool*.

Moschitti, A. and Basili, R. (2004). Complex linguistic features for text classification: A comprehensive study. In *Proceedings of ECIR-04, 26th European Conference on Information Retrieval Research*, pages 181–196. Springer Verlag, Heidelberg, DE.

Ruiz-Rico, F., Vicedo, J. L., and Rubio-Sánchez, M.-C. (2006). Newpar: an automatic feature selection and weighting schema for category ranking. In *Proceedings of DocEng-06, 6th ACM symposium on Document engineering*, pages 128–137.

Scott, S. and Matwin, S. (1999). Feature engineering for text classification. In Bratko, I. and Dzeroski, S., editors, *Proceedings of ICML-99, 16th International Conference on Machine Learning*, pages 379–388, Bled, SL. Morgan Kaufmann Publishers, San Francisco, US.

Sebastiani, F. (1999). A tutorial on automated text categorisation. In Amandi, A. and Zunino, R., editors, *Proceedings of ASAI-99, 1st Argentinian Symposium on Artificial Intelligence*, pages 7–35, Buenos Aires, AR. An extended version appears as (**?**).

Singhal, A., Buckley, C., and Mitra, M. (1996). Pivoted document length normalization. *Department of Computer Science, Cornell University, Ithaca, NY 14853*.

Tan, C.-M., Wang, Y.-F., and Lee, C.-D. (2002). The use of bigrams to enhance text categorization. *Information Processing and Management*, 38(4):529–546.

Tesar, R., Strnad, V., Jezek, K., and Poesio, M. (2006). Extending the single words-based document model: a comparison of bigrams and 2-itemsets. In *DocEng '06: Proceedings of the 2006 ACM symposium on Document engineering*, pages 138–146, New York, NY, USA. ACM Press.

Yang, Y. and Liu, X. (1999). A re-examination of text categorization methods. In Hearst, M. A., Gey, F., and Tong, R., editors, *Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval*, pages 42–49, Berkeley, US. ACM Press, New York, US.

Z. Yang, Z. Lijun, Y. J. and Zhanhuai, L. (2003). Using association features to enhance the performance of naive bayes text classifier. *Fifth International Conference on Computational Intelligence and Multimedia Applications, ICCIMA '03*, pages 336–341.

Zu, G., Ohyama, W., Wakabayashi, T., and Kimura, F. (2003). Accuracy improvement of automatic text classification based on feature transformation. In *Proceedings of DOCENG-03, ACM Symposium on Document engineering*, pages 118–120, Grenoble, FR. ACM Press, New York, US.