Enhancing Patent Expertise through Automatic Matching with Scientific Papers

Kafil Hajlaoui, Pascal Cuxac, Jean-Charles Lamirel, Claire François

Enhancing Patent Expertise through Automatic Matching with Scientific Papers

Kafil Hajlaoui*, **Pascal Cuxac***, **Jean Charles Lamirel****, **Claire François***

Kafil.Hajlaoui@inist.fr, Pascal.Cuxac@inist.fr,
lamirel@loria.fr, Claire.Francois@inist.fr

(*) INIST CNRS, Vandœuvre-lès-Nancy, France
(**) INRIA team SYNALP-LORIA, Vandœuvre-lès-Nancy, France

Keywords: Supervised classification, Technological and scientific survey, Patents, KNN, Association rules

Abstract. This paper focuses on a subtask of the QUAERO[1] research program, a major innovating research project related to the automatic processing of multimedia and multilingual content. The objective discussed in this article is to propose a new method for the classification of scientific papers, developed in the context of an international patents classification plan related to the same field. The practical purpose of this work is to provide an assistance tool to experts in their task of evaluation of the originality and novelty of a patent, by offering to the latter the most relevant scientific citations. This issue raises new challenges in categorization research as the patent classification plan is not directly adapted to the structure of scientific documents, classes have high citation or cited topic and that there is not always a balanced distribution of the available examples within the different learning classes. We propose, as a solution to this problem, to apply an improved K-nearest-neighbors (KNN) algorithm based on the exploitation of association rules occurring between the index terms of the documents and the ones of the patent classes. By using a reference dataset of patents belonging to the field of pharmacology, on the one hand, and a bibliographic dataset of the same field issued from the Medline collection, on the other hand, we show that this new approach, which combines the advantages of numerical and symbolical approaches, improves considerably categorization performance, as compared to the usual categorization methods.

1 Introduction

Text categorization is a machine learning task which aims at automatically assigning predefined category labels to new upcoming free text documents with related characteristics [1]. Because of its numerous applications, text categorization has been one of

[1] http://www.quaero.org

the most studied branches within the field of machine learning [2]. Consequently, a variety of classification algorithms were developed and evaluated in applications such as mail filtering [3], opinion and feelings analysis [4], news [5] [6] or blogs [7] classification. Among the most often used learning methods exploited in that context, we may mention artificial neural networks [8] [9], K-nearest-neighbors (KNN) [10], decision trees [11] [12] [13], Bayesian networks [14] [15], support vector machines (SVM) [16], and more recently, boosting based methods [17] [18]. Although many methods developed for automatic text categorization have achieved significant accuracy when applied to simple text structure (for example emails, summaries, etc.), there are still many remaining challenges concerning classification of complex documents, especially when classification relies on imbalanced learning data.

A broad range of studies address the problem of Medline[2] database categorization. Most of these works focus on the importance of data preprocessing and data representation steps in the context of the text categorization task. In [19], the authors show that, in the case of a text representation based on the "bag of words" model, the weighting of the extracted terms significantly increases the performance of the classifiers. In order to classify Medline papers into predefined topics, Suomela and Andrade [20] restrict the extracted descriptors to predefined lexical classes (nouns, adjectives, verbs) and apply a word frequency scheme. Using specific Medline topics, the authors obtain a classification F-score of 65%. The same approach is further used by the Medline Ranker web-service [21] whose role is to extract a relevant list of Medline references starting from a set of keywords defined by the user. The study of Yin et al. [22] focuses both on the identification and on the extraction of protein interactions from Medline articles. For that purpose, documents are preprocessed using bigrams, in a first step, and SVM method is applied, in a second step. The authors obtain a performance of 50% true positives, and a recall rate of 51%. Recently, the Biocreactive III challenge proposed to classify Medline articles belonging to the biomedical field [23]. The best performance related to Medline data was obtained on this collection, with an accuracy of 89.2% and an F-score of 61.3%.

Up to now, patents evaluation is a manual operation that involves groups of experts with in-deep expertise of the related field. This evaluation is mainly based on references to relevant scientific papers (articles, theses, books ...) associated to the patents. The automatic classification of publications in patent classes can thus represent a valuable help for the experts. However, this task is not a traditional classification task because the classification structure (i.e. the patent classification scheme) does not directly fit with the data to be classified (i.e. the scientific papers). To cope with that problem, two alternative approaches can be used. A first approach consists in creating a gateway between the publications classification plan and the patents classification plan. Nonetheless, this approach is difficult to implement because it involves the intensive use of complex tree comparison techniques (here, the classification plans), and consequently, an intensive use of complementary human expertise. A second approach consists in developing a classification system that directly uses the patents

[2] http://www.ncbi.nlm.nih.gov/pubmed/

classification plan. Such an approach is founded on the assumption that scientific papers that are cited in a patent are strongly related to the patent field, and consequently to the classification code of the latter. In this context, the training dataset would consist of the whole set of scientific citations extracted from the patents of the considered field. However, one potential barrier of this approach is that the learning classes might not necessarily have a homogeneous quantity of patents and thus might not provide an homogeneous amount of learning data (i.e. cited papers) leading thus to face with an imbalance classification problem. Moreover, in a focused domain, patent classes might have high citation or cited topics overlaps leading to additional class similarity problem.

In the following sections, we describe a complete experiment of automatic classification of scientific papers based on an initial patents dataset. The experimental dataset is related to the field of pharmacology and the bibliographic references cited in the patents are extracted from the Medline collection. In the first section, we describe the dataset construction process and we illustrate the resulting phenomena of imbalance learning examples and class similarity. By applying usual categorization methods, we then illustrate, in the second section, the influence of the term extraction strategy on the classification results. Two approaches are particularly discussed. The first one is based on the direct use of Medline indexing keywords associated to the bibliographic records. The second is based the construction of an index from the titles and abstracts of the records by the use of NLP tools. This section highlights that the best performance are obtained with the K-Nearest Neighbor algorithm (KNN) in our context. To cope with the class imbalance and similarity problems, we present in the third section a modified KNN algorithm named KNNBA-2T which is based on the exploitation of association rules between data descriptors and patent class labels. We show that this algorithm provides better classification accuracy than the original KNN algorithm in our context. In section 4, we perform a complementary test of the KNNBA-2T algorithm in combination with resampling techniques. In this test, we exploit our former dataset as well as 6 other UCI datasets and compare the results of KNNBA-2T with resampling with a broader range of other usual algorithms. The final section draws our conclusion and perspectives.

2 Building and indexation of the corpora

Our main experimental resource is issued from the QUAREO project. It is a collection of patents related to the pharmacology domain and including bibliographic references. This resource consists of 6387 patents in XML format, grouped into 15 subclasses of the A61K class (medical preparation). As shown in Figure 1, we begin by extracting the citations from the patents. From 6387 patents, we extracted 25887 references such as databases, books, encyclopedia and scientific articles. In a second step, we query the Medline database with the extracted citations related to the scientific articles. In such a way, we obtained 7501 articles. This represents a recall of 90% for this type of references. Each article is then labeled by the class code of the citing patent and the set of labeled articles represents our final training dataset.

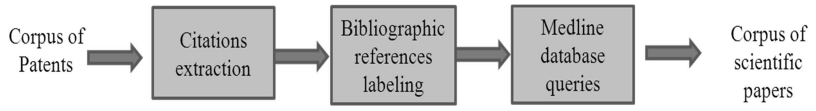

Fig. 1. Building steps for the training dataset

Figure 2 summarizes the distribution of documents of the training dataset relatively to the different class codes. From that information, one might conclude that one of the important criteria of selection of the classification method will be its ability to process imbalance data. By the fact, the distribution of references between classes is very heterogeneous. The smallest class contains only 22 articles (A61K41 class) whilst the bigger one has more than 2500 (A61K31 class).

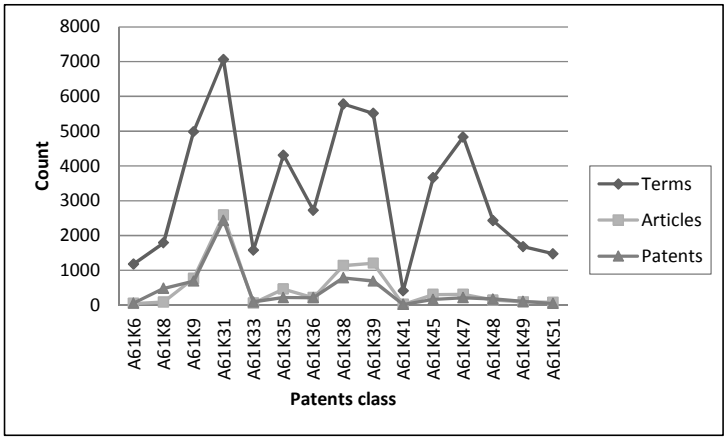

Fig. 2. Distribution of the training data in the patents classes.

2.2 Data Representation

As we have mentioned it before, for text classification, the choice of a document model is a crucial step. A common approach is to use a document model called "bag of words", in which the only exploited information is the presence and/or the frequency of terms. In our case, we translate the bag of words model into a vectorial representation, as it has been proposed by Salton [24]. Following this approach, each article of the dataset is represented as a vector in an N-dimensional space, where N is the total number of terms (features) issued from the articles collection. The whole text collection is then represented as a $(N+1)\ J$ matrix, where J is the number of articles in the collection. Each line j of this matrix is an N-dimensional bag of word vector for the article j, plus its class label.

If a feature i does not occur in the article j, then the relevant matrix element a_{ij} is zero, otherwise it is assigned a positive value or weight. The way to calculate this weight depends on the scheme used for feature representation. The weight is 0 or 1

for the binary scheme. On its own side, the standard frequency weighting scheme is based on terms document occurrences. However, with such scheme, too much importance could be given to descriptors that appear frequently in many documents and which are, consequently, unrepresentative for each single document. Another weighting scheme, called the TF.IDF (Term Frequency Inverse Document Frequency) is thus often used in literature [26] [27] [28]. This scheme evaluates the importance of a term according to its frequency in the document (TF = Term Frequency) weighted by its frequency in the corpus (IDF = Inverse Document Frequency).

$$Tf.Idf(t_k, D_j) = TF(t_k, D_j) \times Idf(t_k)$$

where $TF(t_k, D_j)$ is the number of occurrences of t_k in D_j, and,

$$Idf(t_k) = \log \frac{|S|}{DF(t_k)}$$

where $|S|$ is the documents number in the corpus and $DF(t_k)$ is the number of documents containing t_k.

At the following stage, we built features according to two different approaches, the first one relying on keywords found in documents, and the second one relying on the lemmas extracted from the document abstracts by the use of an NLP tool. The objective of this last approach is to improve the representation of documents' content. To do this, we use the TreeTagger tool [25] developed by the Institute for Computational Linguistics of the University of Stuttgart. This tool is both a lemmatizer and a tagger. A lemmatizer associates a lemma, or a syntactic root, to each word in the text and a tagger automatically annotates text with morpho-syntactic information. In our case, the document are firstly lemmatized and the tagging process is performed on lemmatized items (in the case when a word is unknown to the lemmatizer, its original form is conserved). The punctuation signs and the numbers identified by the tagger are deleted. A sample output of the TreeTagger program is given in figure 3.

The	DT	the
most	RBS	most
widely	RB	widely
used	VVN	use
therapeutic	JJ	therapeutic
modality	NN	modality
is	VBZ	be
chemical	JJ	chemical
pleurodesis	**NN**	**<unknown>**

Fig. 3. Example of a sentence labeled and lemmatized by TreeTagger[3]

[3] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/Penn-Treebank-Tagset.pdf

The feature selection according to grammatical categories allows identifying salient features for the documents classification according to document types or opinions. Moreover, this approach permits to consistently reduce the description space. We thus choose to apply it in our experiment.

3 Classification

To evaluate the relevance of different indexation and weighting methods, we chose to use three different classifiers: a K-nearest-neighbors (KNN) classifier relying on Euclidean distance, a (SVM) classifier and a probabilistic classifier (Naive Bayes). These three supervised machine learning algorithms are known to provide the best results for text classification [29] [30]. In our case, we have exploited them in the Weka environment[4].

In all tests, we have applied different weighting techniques, according to type of extracted descriptors. For the lemmas we mainly use the standard frequency and the TF.IDF techniques. Applying the TF technique on keywords would be meaningless, because the indexing on documents is not redundant. Therefore we use solely the Boolean or IDF technique in this case.

For the features based on lemmas, we have performed several experiments by switching the selected grammatical categories (A: Adjective, N: Noun, NA: Noun + Adjective, NV: Noun + Verb, VA: Verb + Adjective, NVA: Noun + Verb + Adjective).

The classification results are expressed in terms of precision and recall. A precision of 100% means that all articles are classified in the correct category. The recall is the percentage of correct answers that are given. These measures are calculated after applying a 10-fold cross-validation process (90% of the corpus is used for learning and 10% for testing).

Tables 1 and 2 show the obtained results in terms of precision (P) and recall (R). They illustrate that the best results in our corpus are obtained with the KNN method combined with an indexation based on the lemmas involving the three grammatical categories (Nouns, Verbs, Adjectives) and using TF-IDF weighting scheme. The obtained measures are 61% for precision and 55% for recall. However, these results can still be considered as far from satisfying ones. Such results can be explained by the imbalanced data distribution between the classes (see figure 2), but also by the fact that the classes are very similar one to another. To highlight that problem, we computed class/class similarity using cosine correlation and drew the resulting class/class similarity distribution (figure 4). This distribution clearly shows that it might be difficult for any classification model to precisely detect the right class: more than 70% of classes' couples have a similarity between 0.5 and 0.9.

[4] http://www.cs.waikato.ac.nz/ml/weka/index.html

Table 1. Classification results related to indexing by keywords

KNN				NB				SVM			
Boolean		IDF		Boolean		IDF		Boolean		IDF	
P	R	P	R	P	R	P	R	P	R	P	R
0.39	*0.39*	*0.39*	0.43	0.4	**0.47**	**0.43**	0.44	0.4	0.45	0.4	0.45

Table 2. Classification results related to indexing by lemmas

	KNN				NB				SVM			
	Frequency		TF.IDF		Frequency		TF.IDF		Frequency		TF.IDF	
Type	P	R	P	R	P	R	P	R	P	R	P	R
A	0.42	0.36	0.42	0.36	0.38	0.2	*0.37*	*0.18*	0.45	0.46	0.45	0.46
N	0.5	0.41	0.52	0.4	0.43	0.31	0.44	0.28	0.54	0.55	0.54	0.55
NA	0.55	0.4	0.57	0.39	0.45	0.36	0.46	0.36	0.55	0.55	0.55	0.55
NV	0.49	0.38	0.52	0.38	0.44	0.35	0.44	0.31	0.53	0.54	0.53	0.54
NVA	0.6	0.54	**0.61**	**0.55**	0.44	0.34	0.45	0.34	0.54	0.55	**0.55**	**0.55**

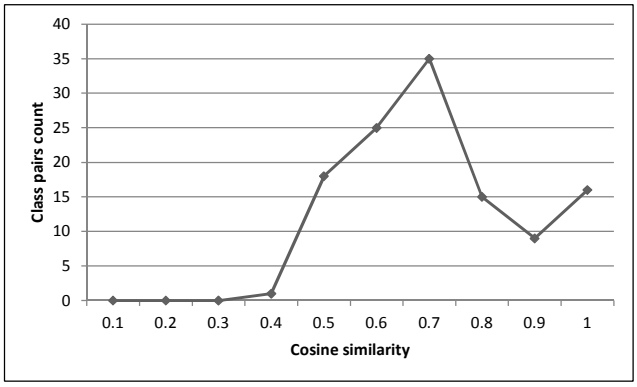

Fig. 4. Class to class similarity distribution

We therefore propose, in the next section, an improvement of the approach, based on the best method, namely the KNN method. The goal of this improvement is to take into account the specific characteristics of the corpus: the important imbalance between classes and the high similarity between them.

4 The KNNBA-2T method

Our improvement of the KNN algorithm is based on the exploitation of association rules. We firstly present a general definition of association rules. We then present a new approach for calculating the weight of class attributes or features, by using a special type of association rules. We finally present a new algorithm, called *KNNBA-2T*, inspired by the method previously developed by Mordian et al. [31].

4.1 Association rules

The association rules extraction approach is a method for discovering relevant relationships between two or more variables. This method is based on local laws and requires no user intervention (it lets the system self-organizing). It allows identifying, from a set of transactions, a set of rules that express a possibility of association between different items (words, attributes, concepts …). A transaction is a series of items expressed in a given order. In addition, transactions can be of different lengths. The relevance of a rule of such extracted association is measured by its index of support and its index of confidence.

For an association rule: $X \rightarrow Y$ the indices of support and confidence are defined by the following two equations:

$$Support = P(X \bigcup Y),\ Confidence = P(Y|X)$$

where $P(X \bigcup Y)$ is the probability that a transaction contains both X and Y, and is the conditional probability of Y knowing that it is X.

The first efficient method for extracting such rules was introduced by Agrawal for the analysis of the market basket through the Apriori algorithm [32]. The operation of this algorithm can be decomposed into two phases:

1) Searches for all the "patterns" or frequent itemsets that appear in the database with a frequency greater than or equal to a threshold defined by the user, called minsup.
2) Generation, from these common patterns, of all the association rules with a measure of confidence greater than or equal to a threshold defined by the user, named minconf.

4.2 The KNNBA-2T algorithm

KNNBA is an improvement of the KNN algorithm. The objective is to assign weights to each attribute by using association rules. For that purpose, we used the association rules that help to identify the most representative terms of a given class. Each transaction consists of all of the extracted terms (attributes) and the label of the class. After the generation of the rules, we are keeping the rules of the type:

$$Attribute \rightarrow Class \text{ and } Attribute_1, Attribute_2 \rightarrow Class$$

The rules composed of three attributes are rare and thus not determinative.

The principle behind this approach is that if two attributes (here *Attribute1* and *Attribute2*) are associated together to a class, the relevance (i.e. the information power) of each of the two attributes deducted from their association must be considered as more important than the one of each single attribute.

The first version of our algorithm (KNNBA-1T) is similar to the Mordian et al. algorithm [31]. It takes into account the rules composed of a single attribute (term). In

the second version of our algorithm (KNNBA-2T), we first compute two attributes rules and we further apply the former principle by deriving single attribute rules from two attributes rules.

After the rule extraction step, a weight can be associated to each attribute *i* (i.e. feature) [31]. It is computed as:

$$W[i] = \left(\frac{1}{1 - G\sup_{[i]}}\right)$$

where G_sup represents the greatest support of the attribute *i*.

As compared to KNN, the new formula for calculating the distance used in the KNNBA2T takes the weight of the attributes into account and thus becomes:

$$D(a,b) = \sqrt{\sum_{i=1}^{n} W[i] \times (x_{ai} - x_{bi})^2}$$

where *a* and *b* are two documents, and x_{ai} and x_{bi} represent the term *i* of each document vector.

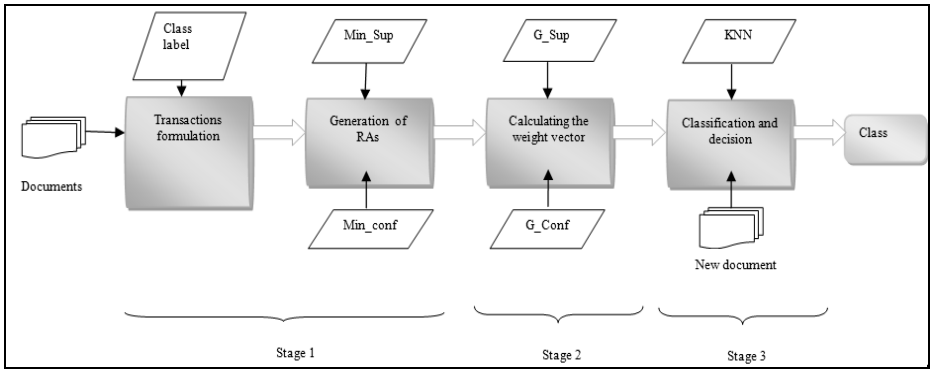

Fig. 5. General operating process of the KNNBA approach

The general process of KNNBA-2T approach is summarized in figure 5 and is composed of three stages:

Stage 1: this phase consists of two steps. The first step is the construction of transactions representing entries to generate the association rules. Each document is transformed into a transaction, consisting of all the representative document descriptors associated with the label of the class. The second step is the generation of the rules of association through using an Apriori algorithm [1].

Stage 2: in this phase, we seek to generate an attribute weight vector from the description of the documents. For that purpose, a group of 15 rules (15 corresponding to the number of classes) is built for each attribute and the most relevant rule (the highest support, the highest confidence) is used to compute the attribute weight.

Stage 3: this phase consists in applying the KNN algorithm with the added extension. To predict the class of a new document by the inter-document similarity calculation, we take into account the weight vector generated in the previous stage.

From an overall perspective, our approach extends the K-nearest-neighbors method in two ways:

1) First, a new weighting scheme of descriptors is introduced, according to their informational weight in relation with their distribution in all the classes.
2) Second, the vote of the closest neighbors is based on a vote function extended by the weight vector W. This extension uses the strength for each term to activate the classes.

Our proposed extension is thus founded on the general idea that the observations for training data, which are particularly close to the new observation (y, x) from test data, must have a higher weight in the decision that the neighbors that are farther from the pair (y, x). This strategy differs from for the standard KNN method in which only the K-nearest-neighbors influence the prediction, but the influence is identical for each of the neighbors, irrespective of their degree of similarity with (y, x). In order to achieve our new goal, the distances on which the search for neighbors is based on are thus transformed in a first step according to the strength (i.e. power) of the term to activate a class.

Table 3. Comparison of classification results with KNN and KNNBA algorithms

KNN		KNNBA-1T		KNNBA-2T	
P	R	P	R	P	R
0.61	*0.55*	0.65	0.65	**0.67**	**0.67**

Table 3 illustrates the precision and recall results obtained after application of the three KNN algorithms (KNN, KNNBA-1T, KNNBA-2T) on our reference dataset with the use of NVA lemmas and TF-IDF weighting. The best results are obtained with the KNNBA-2T algorithm, when compared to the KNN and the KNNBA-1T algorithms. We find that the percentage of correctly classified documents rises from 61% to 65% with KNNBA-1T and to 67% with KNNBA 2T. Our adapted methods thus significantly improve the classification performance on our dataset.

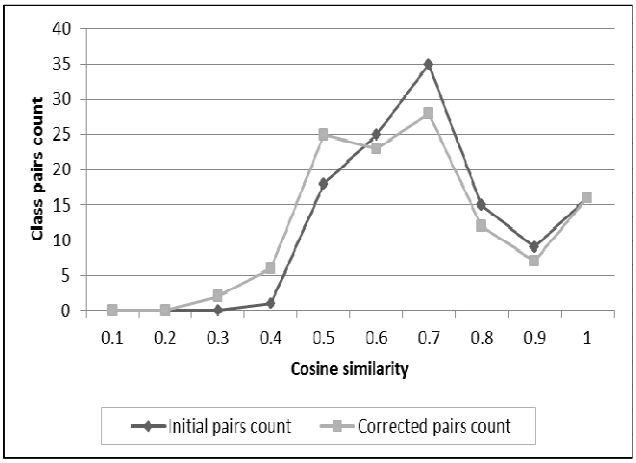

Fig. 6. Correction of the class imbalance and class similarity using the KNNBA-2T approach

Our new KNNBA-2T approach provides correction capabilities both for terms distribution within classes (i.e. class imbalance) and for class similarity. The correction of class similarity is illustrated at figure 6. However, we also remarked that the smoothing of terms distribution is not effective on the largest class (A61K31) which is always a majority class. As a result, our approach still tends to ignore small classes while concentrating on classifying the larger ones accurately.

Even if the class similarity problem remains difficult to solve in our context, because of the overlapping nature of the exploited patent classification, one complementary approach can be used to better solve the class imbalance problem. Hence, resampling methods are very commonly employed for dealing with such problem. Their advantage over other methods is that they are external and thus easily portable and very simple to implement. Resampling is usually conducted using the two following strategies: oversampling consists of copying existing training examples at random and adding them to the training set until a full balance is reached. Undersampling consisted of randomly removing existing examples until a full balance is reached [33][34].

In the next section we thus provide an extended test of the KNNBA-2T algorithm by combining it with a resampling technique. In this test, we exploit our former dataset as well as 6 other UCI datasets and compare the results of KNNBA-2T with resampling with a broader range of other usual algorithms.

5 Extended experimental results

In this new experiment we have made use of a combination of the KNNBA-2T method with a resampling technique. The exploited resampling technique is the Weka resample algorithm which is an undersampling algorithm suitable for decreasing the

influence of very large classes. We also extend the range of comparison by using a broader range of classification techniques (including neural network (ANN) and J48 algorithms). Apart of our former dataset, we also exploit 6 complementary reference datasets issued from the UCI machine learning database collection. The overall characteristics of the experimental datasets are presented in table 4.

Table 4. Description of the datasets used in the experiments

	Dataset	Size	Nb. of attributes	Nb. of classes
1	NVA+Resample	7501	2463	15
2	Breast-cancer-w	699	11	2
3	Car	1728	6	4
4	Ecoli	336	6	8
5	Glass	214	10	6
6	Nursery	266	8	5
7	Zoo	101	18	7

All algorithms are executed in the following similar conditions:

1. The Weka resample (undersampling) algorithm is applied.
2. The ten-fold cross validation method is used.
3. The Weka default parameters are used for all of the algorithms.
4. The number of neighbors is set to 10 (k=10) for the KNN-based algorithms.

Table 5. Comparison of accuracy of KNNBA-2T with other algorithms

Dataset	KNNBA-2T	KNN	NB	NN	J48	SVM
NVA+Resample	**77.78**	70.2	70.1	71.56	*68.27*	71.56
Breast-cancer-w	96.89	96.42	95.99	95.27	*94.56*	**96.99**
Car	**95.21**	93.51	*85.53*	99.53	92.36	93.75
Ecoli	**88.31**	86.01	85.41	86.01	*84.22*	84.22
Glass	**68.89**	66.35	*48.59*	67.75	66.82	56.07
Nursery	**98.58**	97.58	*95.08*	99.83	98.06	96.93
Zoo	**98.41**	88.11	95.04	96.03	*92.07*	96.03

In all experiments, the accuracy of each algorithm is based on the percentage of correctly classified documents. The complete results are presented in the table 5.

Our new experiments highlight that the resampling method significantly improves the performance of the KNNBA-2T method on our reference dataset of scientific papers (+10 points of precision). Even if it is not presented in the table, similar improvement can been observed for the other methods and for the other datasets. Table 5 also highlights that the overall results of our KNNBA-2T algorithm are above average

(most of the time the best) on the other UCI datasets. However, the most important advantage as regards to the other methods is observed for large test collections in which classes include a large number of attributes. In this case our method clearly allows reducing the class representation space by selecting the relevant attributes.

6. Discussion and Conclusion

The classification of scientific papers in a patents' classification plan is a real challenge as such classification plan is very detailed and not very suitable with respect to the scientific documents. In this paper we presented a new method for supervised classification derived from the KNN method. This method, which we named KNNBA-2T, operates a classes' descriptor term weighting based on association rules induced by these terms. We applied it on a dataset of bibliographic articles from the Medline database in order to classify them within a classification plan of patents belonging to the field of pharmacology. This new method offers very interesting performance for our study as compared to existing methods, especially when it is combined with resampling techniques. Nevertheless, the resulting class imbalance and the similarity of the class description remains a major problem still hampering the performance improvement of automatic classification of articles within the international patents' plan. Therefore, we undertook new experiments in order to combine our method with vocabulary extension techniques based on domain ontologies. In our context, such ontology as Mesh which is associated to Medline resource represents a good candidate.

Acknowledgment: This work was done under the program QUAERO[5] supported by OSEO[6] national agency of research development.

References

1. Cohen, A.M. et Hersh, W.R.: A survey of current work in biomedical text mining. Briefings in Bioinformatics 6. pp. 57-71 (2005)
2. Hillard, D. Purpura, S. et Wilkerson, J.: An active learning framework for classifying political text. In Annual Meeting of the Midwest Political Science Association. Chicago (2007)
3. Cormack, G.V. et Lynam, T.R.: Online supervised spam filter evaluation. ACM Transactions on Information Systems. 25(3):11 (2007)
4. Pang, B. et Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval. 2(1-2):1–135 (2008)
5. Purpura, S. et Hillard, D.: Automated classification of congressional legislation. Proceedings of the international conference on Digital government research. pp. 219–225 (2006)

[5] http://www.quaero.org
[6] http://www.oseo.fr/

6. Evans, M., McIntosh, W., Lin, J. et Cates, C.: Recounting the courts? Applying automated content analysis to enhance empirical legal research. Journal of Empirical Legal Studies. 4(4):1007–1039 (2007)

7. Durant, K. et Smith, M.: Predicting the Political Sentiment of Web Log Posts Using Supervised Machine Learning Techniques Coupled with Feature Selection. In Advances in Web Mining and Web Usage Analysis: 8th International Workshop on Knowledge Discovery on the Web. Webkdd 2006. pp. 187–206. Philadelphia. Springer-Verlag, NY (2007)

8. Wiener, E., Pedersen, J.O. et Weigend, A.S.: A Neural Network Approach to Topic Spotting. Symposium on document analysis and information retrieval. pp. 317-332 (1995)

9. Schütze, H., Hull, D.A. et Pedersen, J.O.: A Comparison of Classifiers and Document Representations for the Routing Problem. Proceedings of the 18th Annual ACM SIGIR Conference. pp. 229-337 (1995)

10. Yang, Y. et Chute, C.G.: An example-based mapping method for text categorization and retrieval. ACM Trans. Inform. Syst. 12: 252-277 (1994)

11. Lewis, D.D. et Ringuette, M.: Comparison of two learning algorithms for text categorization. In Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval (SDAIR'94). pp. 81-93 (1994)

12. Quinlan, J.R.: Induction of decision trees. Machine Learning. 1(1). pp. 81-106 (1986)

13. Apte, C., Damerau, F. et Weiss, S. M.: Text mining with decision rules and decision trees. In Proceedings of the Conference on Automated Learning and Discovery. Workshop 6: Learning from Text and the Web (1998)

14. Lewis, D.D.: An Evaluation of Phrasal and Clustered Representations on a Text Categorization Task. In Proceedings of the 15th Annual ACM SIGIR Conference. pp. 37-50 (1992)

15. Joachims, T.A.: Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. In Proceedings of ICML-97. 14th International Conference on Machine Learning (1997)

16. Joachims, T.A.: Text categorization with support vector machines: Learning with many relevant features. In proceedings of the European conference on Machine learning. pp. 137-142 (1998)

17. Schapire, R., Singer, Y. et Singhal, A.: Boosting and Rocchio applied to text filtering. In Proceedings of the 21th Annual ACM SIGIR Conference (1998)

18. Iyer, R., Lewis, D., Schapire, R., Singer, Y. et Singhal, A.: Boosting for document routing. In Proceedings of the 9th International Conference on Information and Knowledge Management (2000)

19. Lan, M., Tan, C.L., Su, J. et Low, H.B.: Text representations for text categorization: a case study in biomedical domain. In IJCNN: International Joint Conference on Neural Networks (2007)

20. Suomela, B.P. et Andrade, M.A.: Ranking the whole MEDLINE database according to a large training set using text indexing. BMC Bioinformatics 6:75 (2005)

21. Fontaine, J.F., Barbosa-Silva, A., Schefer, M., Huska, M.R., Muro, E.M. et Andrade-Navarro, M.A.: MedlineRanker: flexible ranking of biomedical literature. Nucleic Acids Res 37(Web Server issue):141-146 (2009)

22. Yin, L., Xu, G., Torii, M., Niu, Z., Maisog, J.M., Wu, C., Hu, Z. et Liu, H. : Document classification for mining host pathogen protein-protein interactions. Artif Intell Med 49(3):155-160 (2010)

23. Krallinger, M., Vazquez, M., Leitner, F., Salgado, D. et Valencia, A.: Results of the BioCreative III (Interaction) Article Classification Task. In Proceedings of the Third BioCreative Workshop. Bethesda. USA. 13-15 September 2010 (2010)

24. Salton, G.: Automatic processing of foreign language documents. Prentice-Hall. Englewood Cliffs. NJ (1971)

25. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In Proceedings of the International Conference on New Methods in Language Processing. pp. 44-49 (1994)

26. Vincarelli, A.: Indexation de documents manuscrits. In Proceedings du Colloque International Francophone sur l'Ecrit et le Document (CIFED06). pp. 49-53 (2006)

27. Salton, G. et Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing Management. pp. 513-523 (1988)

28. Spark-Jones, K.: A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation. pp. 11-21 (1972)

29. Sebastiani, F.: A tutorial on automated text categorisation. In A. Amandi, A. and Zunino, R. editors. Proceedings of the 1st Argentinian Symposium on Artificial Intelligence (ASAI'99). pp. 7-35 (1999)

30. Yang, Y. et Liu, X.: A reexamination of text categorization methods. In Proceedings of the 22th Annual ACM SIGIR Conference. pp. 42-49 (1999)

31. Mordian, M. et Baarani, A. KNNBA: k-Nearest Neighbor Based Association Algorithm. University of Isfahan. Iran (2009)

32. Agrawal, R. et Srikant, R.: Fast algorithms for mining association rules in large data bases. In. Bocca, J.B., Jake, M. and Zaniolo, C. Editors. Proceeding of 20th VLDB Conference, Chile (1994)

33. Tomek, I.: Two modifications of CNN. IEEE Trans. Syst. Man. Cybern. SMG-6(11):769-772 (1976)

34. Kubat, M. et Matwin, S.: Addressing the curse of imbalanced training sets: one-sided selection. In ICML. pp. 179-186 (1997)