Improving imbalanced scientific text classification using sampling strategies and dictionaries

L. Borrajo[1], R. Romero[1], E. L. Iglesias[1*] and C. M. Redondo Marey[2]

[1]Univ. of Vigo, Computer Science Dept., Campus As Lagoas s/n, 32004 Ourense, Spain

[2]Complejo Hospitalario Universitario de Vigo, Vigo, Spain

Summary

Many real applications have the imbalanced class distribution problem, where one of the classes is represented by a very small number of cases compared to the other classes. One of the systems affected are those related to the recovery and classification of scientific documentation.

Sampling strategies such as *Oversampling* and *Subsampling* are popular in tackling the problem of class imbalance. In this work, we study their effects on three types of classifiers (Knn, SVM and Naive-Bayes) when they are applied to search on the PubMed scientific database.

Another purpose of this paper is to study the use of dictionaries in the classification of biomedical texts. Experiments are conducted with three different dictionaries (BioCreative, NLPBA, and an ad-hoc subset of the UniProt database named Protein) using the mentioned classifiers and sampling strategies.

Best results were obtained with NLPBA and Protein dictionaries and the SVM classifier using the *Subsampling* balancing technique. These results were compared with those obtained by other authors using the TREC Genomics 2005 public corpus.

1 Introduction

Due to the ever-increasing amount of scientific articles in the biomedical domain, Text Mining has been recognized as one of the key technologies for future research. With an overwhelming amount of textual information in biomedicine, there is a need for effective and efficient literature mining and knowledge discovery that can help to gather and make use of the knowledge encoded in text documents.

Scientific papers are a primary source of information for investigators to know the current status in a topic or compare their results with other colleagues. However, mining of biomedical texts remains to be a great challenge. Firstly by the huge volume of scientific databases. Secondly, their imbalanced nature which only a small number of relevant papers to each user query.

The imbalance problem exists in a broad range of experimental data, but only recently has attracted close attention for researchers [8, 35]. Data imbalance occurs when the majority class is represented by a large portion of all the examples, while the other, the minority one, has only a small percentage [31]. When a text classifier encounters an imbalanced document corpus, the performance of machine learning algorithms often decreases [22].

*To whom correspondence should be addressed. Email: eva@uvigo.es

doi:10.2390/biecoll-jib-2011-176

Sampling strategies such as over- and *Subsampling* are popular in tackling the problem of class imbalance [3, 12, 32, 40]. The *Subsampling* algorithm decreases artificially the number of samples that belongs to majority class, while the *Oversampling* algorithm redistributes the number of samples that belongs to minority taking into account the majority class. In this work, we study effects of *Subsampling* and *Oversampling* on three classifiers widely used in the field of text mining (K nearest neighbours (Knn), Support Vector Machine (SVM) and Naive-Bayes).

Another purpose of this paper is to analyse the behaviour of dictionaries when classifying biomedical texts. Synonymy is one of the most important relations found between different terminologies and is critical for building text mining systems of high quality. Dictionaries which list synonymous terms, likes WordNet [25], Longman's dictionary [9] or NLPBA dictionary [13] have been found to be useful for improving the results of information retrieval systems [24]. Several authors have used dictionaries in the process of classification of biomedical texts [1, 7, 23, 30] obtaining good results.

In our case, a comparison is made using three different dictionaries (BioCreative [19], NLPBA [13] and an ad-hoc subset of UniProt named Protein) [5] with the classifiers mentioned above. To perform tests and to compare results with those obtained by other authors we have used the *Text Retrieval Conferences (TREC)* [34] public corpus. In 2005, TREC provided a set of evaluation tasks to know the state of the art of applying information extraction techniques to problems in biology. Specifically, the goal of the TREC Genomics Track was to create test collections for evaluation of information retrieval and related tasks in the genomics domain [18].

The rest of this paper is constructed as follows: Section 2 describes the architectural model proposed to scientific text classification. Experimental results are given in Section 3, and in Section 4 a comparative with other authors is made. Finally Section 5 concludes this paper.

2 Model

The Fig. 1 shows an schema of the proposed biomedical text classification process. As observed, the architecture is divided into four tasks that are explained below.

2.1 Annotation task

This first task processes documents extracting the most relevant keywords. The annotation process can be quite complex depending on the techniques to apply. In this research, an annotation plugin called Abner-Tagger [28], provided by GATE (General Architecture for Text Engineering) [15], is used.

GATE includes components for language processing tasks, e.g. parsers, morphology, tagging, information retrieval tools, information extraction components for different languages, and many others. In our experiments, tokeniser and stemmer have been used. *Tokeniser* splits text into simple tokens, such as numbers, punctuation, symbols, and words of different types (e.g. with an initial capital, all upper case, etc.). *Stemmer* process annotates each token with its stem. GATE stemmers are based on the Porter stemmer for English [26].

doi:10.2390/biecoll-jib-2011-176

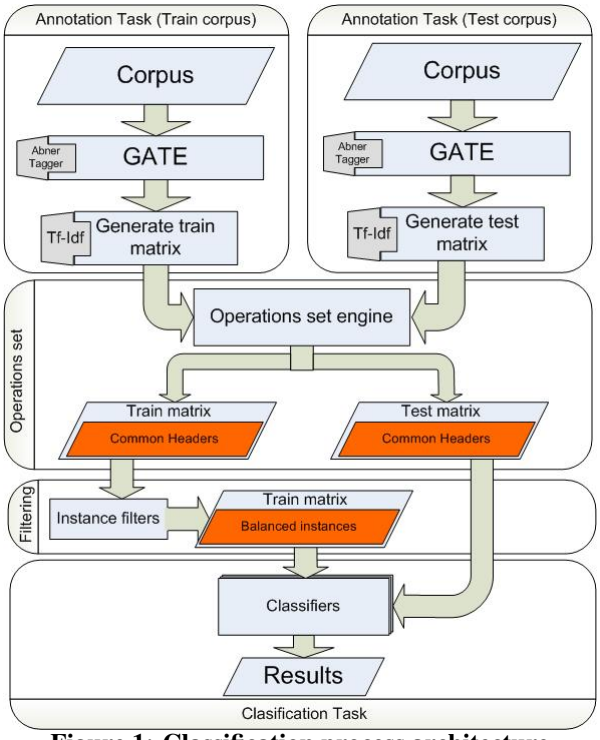


Figure 1: Classification process architecture

The *entity recognizer* (Abner-Tagger) allows to use different dictionaries in order to preprocess the documents and to tag entities. The available dictionaries for the current distribution are based on the NLPBA [13] and the BioCreative [19] corpora.

NLPBA (Natural Language Processing in Biomedical Applications) corpus contains articles extracted from MEDLINE database using the MeSH terms *human*, *blood cells* and *transcription factors*. Abstracts were annotated for the entity classes *protein*, *DNA*, *RNA*, *cell line* and *cell type* [28].

BioCreative is an annotation passage retrieval corpus of human proteins and contains one entity subsuming *genes* and *gene products* (proteins, RNA, etc.). The annotations are based on Gene Ontology (GO) terms. The passages were extracted from full text articles of the Journal of Biological Chemistry, and evaluated regarding the context of each passage within the whole document, meaning that those passages where highlighted within the full text articles.

On the other hand, we also have generated an ad-hoc protein-based dictionary using a subset of the UniProt database [5], named Protein. In short, only proteins known to be associated with lung, colon and breast cancer are being included. They should also appear in the UniProt Knowledge-base (UniProtKB), the central hub for the collection of functional information on proteins, with accurate, consistent and rich annotation.

As a result of the annotation task, a dataset compounded by vectors (named *sparse matrix*) is generated. In this matrix, each row is a mathematical representation of a document. Inside of the matrix generation process we have used the well-known normalization technique called *tf-idf*. This normalization process tries to weight a single attribute taking in account its relevance over a document and the corpus.

As mentioned, regarding the corpus the provided by TREC 2005 competition [4] was selected.

Test and train sparse matrices were generated for scientific articles about alleles of mutant phenotypes in combination with BioCreative, NLPBA and Protein dictionaries.

2.2 Operation sets

Once the test and train sparse matrices are generated, it is necessary to analyse their similarity and computational cost. During the classification process, the train and test matrices must include the same number of attributes in a particular order. When dictionaries are too large, as in our case, it may occur that during the re-annotation process (train matrix headers), most of the relevant attributes belonging to the train matrix do not belong to the second one. This situation generates a second matrix with useless data. We have solved this problem applying a mathematical intersection over the matrices in order to reduce their dimensionalities and make them computable.

2.3 Instance Filtering

Before classification we apply balancing techniques in order to make an improvement about the results and try to reduce the over-fitting. As mentioned, instance filtering represents a powerful tool in case of over-training over a single class, i.e. if the dataset is represented by a very small number of samples compared to the other classes, which is usually of great interest.

Subsampling and *Oversampling* techniques are tested here, because they allow to balance the number of instances that belongs to each class, and apply algorithms in order to decrease the number of attributes.

2.4 Classification

Different machine learning methods have been used for document classification: K-nearest neighbour (Knn) [2, 6, 37], Logistic regression [29], Support Vector Machines (SVM) [6, 10, 21, 29, 39], Naive Bayes classification [6, 10, 38], neural networks [36], instance-based learning [20], and a number of other machine learning technique [27].

Different implementations of the various machine learning methods are available. For example, the Weka toolkit [17] comprises a suite of machine learning tools that can be used for classification [33]. As mentioned, our tests were carried with three types of classifiers: Knn, Naive-Bayes and SVM.

The K-nearest neighbour algorithm is a supervised machine learning technique where each new instance is classified based on majority of K-nearest neighbour category. It bases that estimation calculating the distance between instances that belongs to the learned data model against the second one. In our implementation we have taken in account options like neighbour distances, the number of them, or the nearest neighbour search algorithm to use.

The Naive Bayes classifier technique is based on the Bayesian theorem. It results particularly useful when the data dimensionality is really high. Despite its simplicity, Naive Bayes can often outperform more sophisticated classification methods. There are multiple versions of the Naive

Bayes classifier that provide support for different options regarding the data distribution: normal (Gaussian), kernel, multinomial, and multivariate multinomial. In our approach, a simple classifier with capabilities for data discretization and kernel distributions is used.

SVM method is a supervised learning algorithm proposed by Vladimir Vapnik and his coworkers [14]. For a binary classification task with classes {+1, -1}, given a train set with n classlabeled instances, $(x_1, y_1), (x_2, y_2), ..., (x_i, y_i), ..., (x_n, y_n)$, where x_i is a feature vector for the ith instance, and y_i indicates the class. A SVM classifier learns a linear decision rule, which is represented using a hyper-plane. The tag of an unlabelled instance x is determined by the side of the hyperplane on which x lies [6].

We have used an implementation of a SVM with four kernels (lineal, polynomial, radial and sigmoid).

2.5 Model Evaluation

To represent the results in the whole process we employed the well-known recall, precision and F-measure (harmonic mean between Recall and Precision) evaluation measures applied in text categorization [16]. We also calculate the Utility measure (1) that contains coefficients for the utility of retrieving a relevant and a non relevant document. It is composed by the best possible score U_{max} and the raw score U_{raw}(2), where U_r (3) is the relative utility of relevant document, and U_{nr} is the relative utility of non relevant document. For our purposes, we assume that U_{nr} is -1.

$$U_{norm} = \frac{U_{raw}}{U_{max}} \quad (1)$$

$$U_{raw} = (U_r \cdot relevant\ docs\ retrieved) + (U_{nr} \cdot nonrelevant\ docs\ retrieved) \quad (2)$$

$$U_r = \frac{all\ possible\ negatives}{all\ possible\ positives} \quad (3)$$

3 Experimental Results

In this section, results using different dictionaries in the annotation process and their impact over classification algorithms are discussed.

The annotation process is supported by GATE and Abner-tagger pluggings. As commented, we generate models training two dictionaries, BioCreative and NLPBA, and producing a group of annotations per document. On the other hand, we generate manually another model based on proteins. Regarding balancing techniques (filtering task) we use a random selection algorithm to remove instances in order to accomplish the distribution factor between each class.

As mentioned, in the classification process we have used an implementation of a SVM with four kernels (lineal, polynomial, radial and sigmoid). In order to support the software part we have tested different implementations, but finally a library called LibSVM [11] that implements a kernel method based on costs (C-SVM) is used.

Furthermore, we use some parameters like *probability estimates* which means that it will be generated probabilities instead of $[-1/+1]$ values per each case for SVM output classification. Another interesting parameter is *normalize*, which permits to scale each attribute values between $[-1, +1]$.

For the Naive Bayes (NBayes) implementation we use one provided by Weka. It has two important options, *useKernelEstimator* and *useSupervisedDiscretization*. The first one uses an estimator based on a kernel to recalculate the data vectors in a new feature space. The second one discretizes all data avoiding normalized values, converting them from numerical to nominal. For our tests we setted these values to off, because we did not get good results changing their values.

K-nearest neighbour implementation was supported using the same software package like NBayes. We consider to use a KDTree search algorithm combined with an Euclidean distance in order to provide an algorithm core support. On the other hand, distances between neighbours were weighted using the inverse of the distance. At last, we got the best results using seven neighbours.

Our results, based on a contrast between the classification process and the impact using different dictionaries in the annotation task, are now presented. In order to represent results, some plots will be shown. They have been grouped by filtering tecniques: umbalanced data, *Subsampling*, and *Oversampling*.

To understand signatures used in horizontal axis, the following acronyms for SVM are used: N means Normalize, P means Probabilistic, NP is equal to both Normalize and Probabilistic, and $G[X]$ is Gamma parameter with X corresponding to values of its parameter.Tests for others classifiers do not have any parameter attached.

In addition, we have added statistical significance tables in order to justify the results obtained in tests. These tables use the same signatures related to dictionaries and classifiers commented, and include statistical tests based on χ^2 and $Fisher$ correlation.

3.1 Unbalanced process

Using an unbalanced classification process means that the number of instances which belongs to each class does not suffer any modification before to apply the reasoning model.

These tests are not remarked with plots because results were poor. Utility results were equal to zero using almost all classifiers, except Naive Bayes which got some positive responses with the three dictionaries.results were less than 0.1 due to obtain few relevant documents identified as itself. It explains why using the other measure(F-measure) we got these results.

3.2 Subsampling filtering

For *Subsampling* distribution values we use different scale factors (10:1, 5:1, 4:1, 3:1, 2:1, 1:1), getting best results per method setting parameters to an uniform distribution (factor equal to 1:1).

In Fig. 2, results based on Utility measure are represented. Best results were achieved using Protein and NLPBA dictionaries, with 0.7762 and 0.7238 values respectively, and the peaks

were obtained using a Knn classifier. However, in some cases the Utility measure may not be conclusive, like this situation, because these results are based on a precision close to $0,051$. Tanking a look to the *SVM polynomial normalized*, a bit poor results were achieved but with a more high precision.

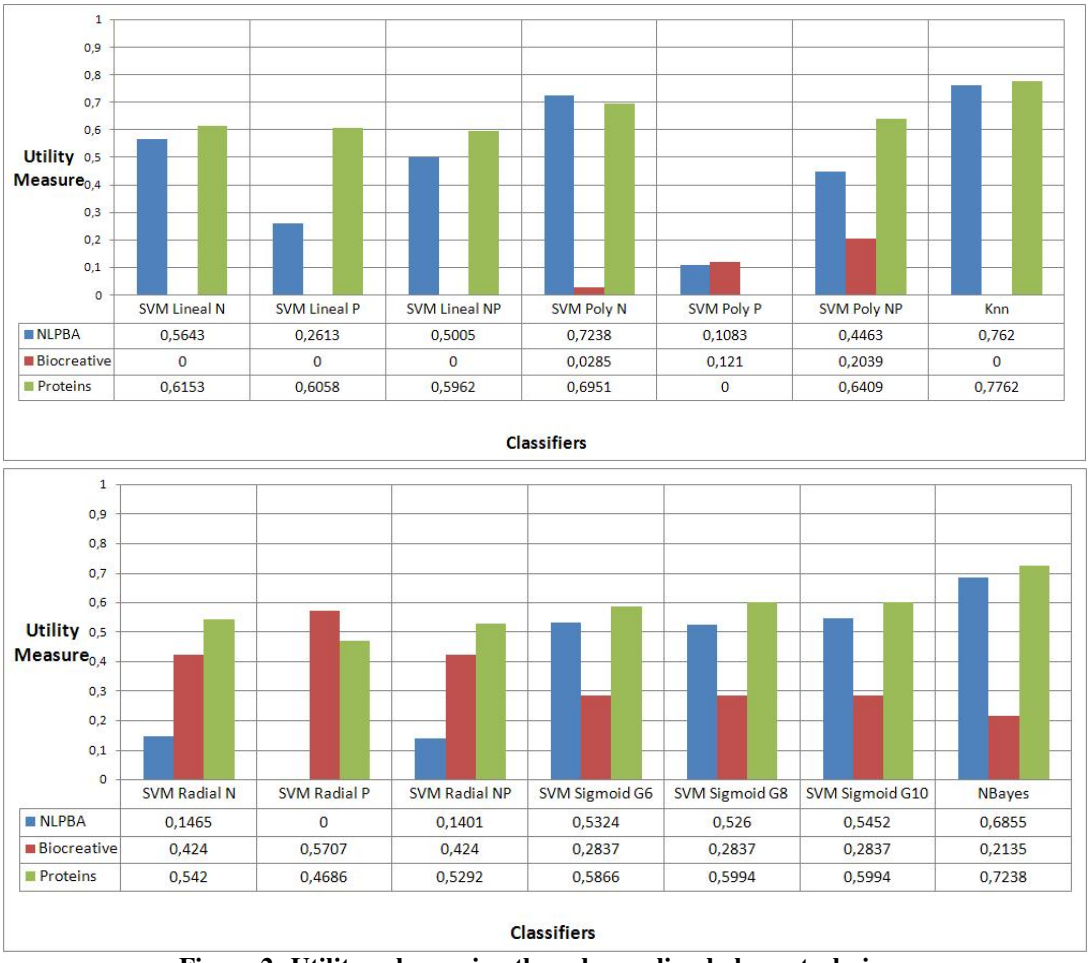
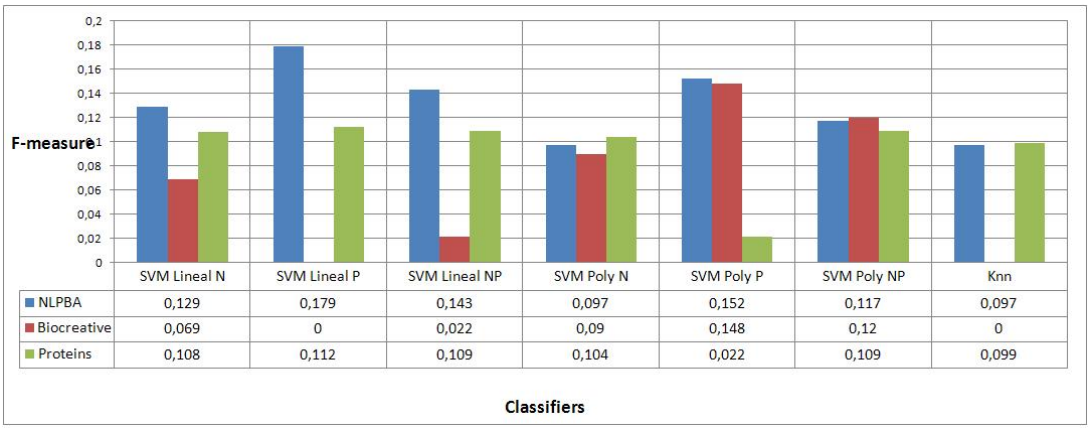


Figure 2: Utility values using the subsampling balance technique

Fig. 3 compares results in the same way that previous ones but using F-measure. In the Figure, only results for relevant documents are included.

The result peaks for Utility and related to Protein and NLPBA dictionaries were obtained using a SVM polynomial kernel normalized and the K-nearest neighbour. But, if we take a look to F-measure, conclusions are different. In case of Knn, 100% of relevant documents were correctly classified but having a Precision equal to 0.051. This means that we have a lot of false positives for both dictionaries, Protein and NLPBA.

Best results were achieved by classifiers and dictionaries which have gotten balanced results between F-measure and Utility. In general, it is difficult to explain results in which the test dataset has problems about unbalanced instances, because Precision and Recall oscillate too much. From a point of view of F-measure, normalized SVM based on a linear kernel got best results, followed by the same SVM but with a polynomial kernel.It means that the mean between Recall and Precision is good, but we know that it is not a guaranty of quality.In order to justify statistically these results, correlation values in table 1 are less than **0.0001** and therefore extremely statistic significant.



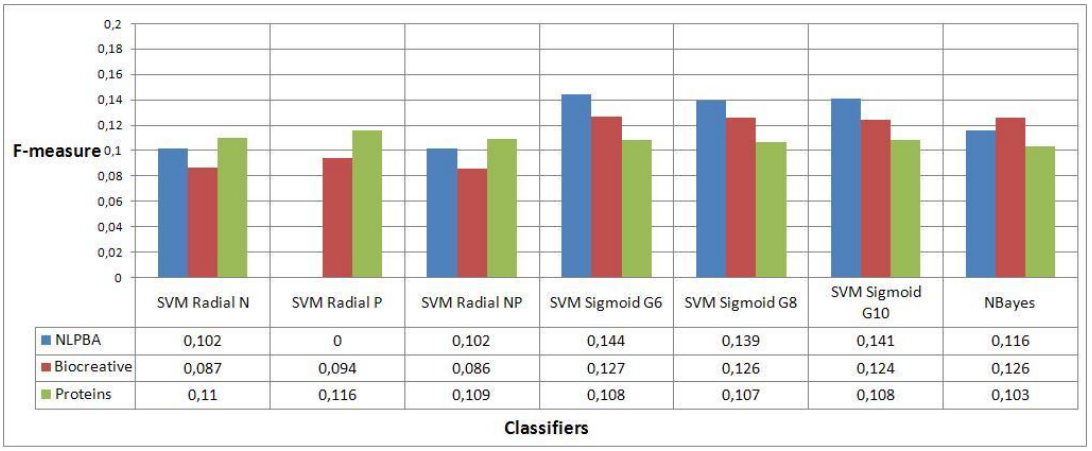


Figure 3: F-measure values using the subsampling balance technique

Following the same reasoning with the sigmoid kernel, it got a bit poor results compared with others, but the balance between each measures were good(χ^2 values between 70.629 to 60.031 for NLPBA), achieving results between 0.52 and 0.6 in case of Utility, and 0.139 to 0.144 regarding F-measure. That means accuracy for this kernel type was good even obtaining good results with all dictionaries (see table 3 for statistical significance). A final consideration for this kernel is that it got good results in almost all situations with small variations. In other words, changing parameters values its results were pretty much stable or similar.

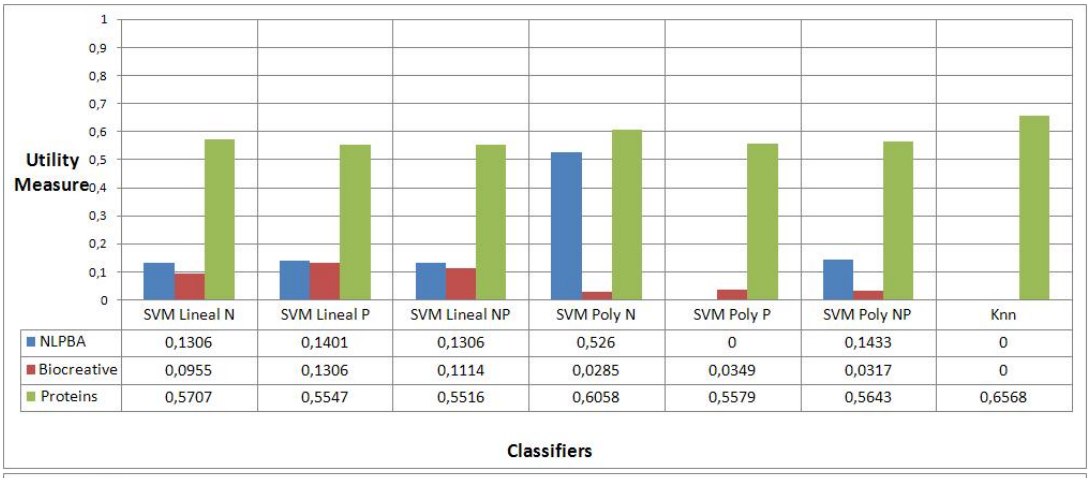
Table 1: Statistical significance using Subsampling

Dictionaries / Classifiers	NLPBA			BioCreAtIvE			Proteins		
	P-value(F)	P-value(χ^2)	χ^2	P-value(F)	P-value(χ^2)	χ^2	P-value(F)	P-value(χ^2)	χ^2
SVM Lineal N	$\leqslant$ **0.0001**	$\leqslant$ **0.0001**	45.816	**0.0030**	**0.0017**	9.807	**0.0005**	**0.0008**	11.192
SVM Lineal P	$\leqslant$ **0.0001**	$\leqslant$ **0.0001**	86.373	0.1000	0.2021	1,627	$\leqslant$**0.0001**	$\leqslant$**0.0001**	17.239
SVM Lineal NP	$\leqslant$ **0.0001**	$\leqslant$ **0.0001**	64.364	0.0234	0.1811	1.789	**0.0002**	**0.0003**	13.140
SVM Poly N	0,8797	0.9897	0	**0.0079**	**0.0063**	7.465	0.0019	0.0035	8.540
SVM Poly P	$\leqslant$ **0.0001**	$\leqslant$ **0.0001**	46.784	$\leqslant$**0.0001**	$\leqslant$**0.0001**	40.646	0.0408	0.2945	1.099
SVM Poly NP	$\leqslant$ **0.0001**	$\leqslant$ **0.0001**	17.088	$\leqslant$**0.0001**	$\leqslant$**0.0001**	23.740	$\leqslant$**0.0001**	$\leqslant$**0.0001**	15.537
SVM Radial N	0.0388	0.0445	4.038	0.1245	0.1296	2.297	**0.0005**	**0.0007**	11.388
SVM Radial P	1.0000	0.8166	0.054	0.6429	0.6874	0.162	$\leqslant$**0.0001**	$\leqslant$**0.0001**	16.751
SVM Radial NP	0.0353	0.0393	4.246	0.0601	0.0649	3.407	**0.0014**	**0.0020**	9.573
SVM Sigmoid G6	$\leqslant$ **0.0001**	$\leqslant$ **0.0001**	70.629	$\leqslant$**0.0001**	$\leqslant$**0.0001**	22.282	**0.0006**	**0.0009**	11.049
SVM Sigmoid G8	$\leqslant$ **0.0001**	$\leqslant$ **0.0001**	60.031	$\leqslant$**0.0001**	$\leqslant$**0.0001**	20.879	**0.0011**	**0.0017**	9.890
SVM Sigmoid G10	$\leqslant$ **0.0001**	$\leqslant$ **0.0001**	68.330	$\leqslant$**0.0001**	$\leqslant$**0.0001**	19.536	**0.0007**	**0.0011**	10,607
Knn	1.0000	1.0000	0	0.1000	0.2021	1.627	0.0994	0.1292	2.302
NBayes	$\leqslant$ **0.0001**	$\leqslant$ **0.0001**	35.800	$\leqslant$**0.0001**	$\leqslant$**0.0001**	19.243	**0.0008**	**0.0020**	9.578

3.3 Oversampling filtering

In our experiments, the *Oversampling* process provided by Weka is used with a distribution bias values between $1.0, 0.75, 0.5, 0, 25, 0.0$. Best results were obtained when the number of instances per class are near to an uniform distribution.

Like *Subsampling*, first of all we are going to talk about the Utility Measure. Fig. 4 shows that best results were achieved using the Protein dictionary with 0.7174 value, but not much better than using *Subsampling*. In addition, K-nearest neighbour got some positive results only for the Protein dictionary, but in any case it was never better than other classifiers.



	SVM Lineal N	SVM Lineal P	SVM Lineal NP	SVM Poly N	SVM Poly P	SVM Poly NP	Knn
NLPBA	0,1306	0,1401	0,1306	0,526	0	0,1433	0
Biocreative	0,0955	0,1306	0,1114	0,0285	0,0349	0,0317	0
Proteins	0,5707	0,5547	0,5516	0,6058	0,5579	0,5643	0,6568

	SVM Radial N	SVM Radial P	SVM Radial NP	SVM Sigmoid G6	SVM Sigmoid G8	SVM Sigmoid G10	NBayes
NLPBA	0,0445	0,0126	0,0413	0,2709	0,2932	0,3124	0,711
Biocreative	0	0	0	0,2263	0,2422	0,255	0,7015
Proteins	0,5165	0,4655	0,5069	0,5771	0,5803	0,5803	0,7174

Figure 4: Utility values using the Oversampling balance technique

Regarding F-measure, relevant results are shown in Fig. 5. Result peaks using Utility were justified by Protein and NLPBA dictionaries using a SVM polynomial kernel normalized and Naive Bayes.Taking a look to the table 1, most p-values for Proteins and NLPBA are extremely statistically significant,less than 0.0001.

But, focusing on F-measure different results are obtained. As *Subsampling*, best results were archived by classifiers and dictionaries which have gotten balanced results between F-measure and Utility.

Finally, we would like to remark again the results gotten by Sigmoid kernel. It obtains a bit poor results compared with the peaks denoted by lineal kernel, but stable results in all tests (Utility

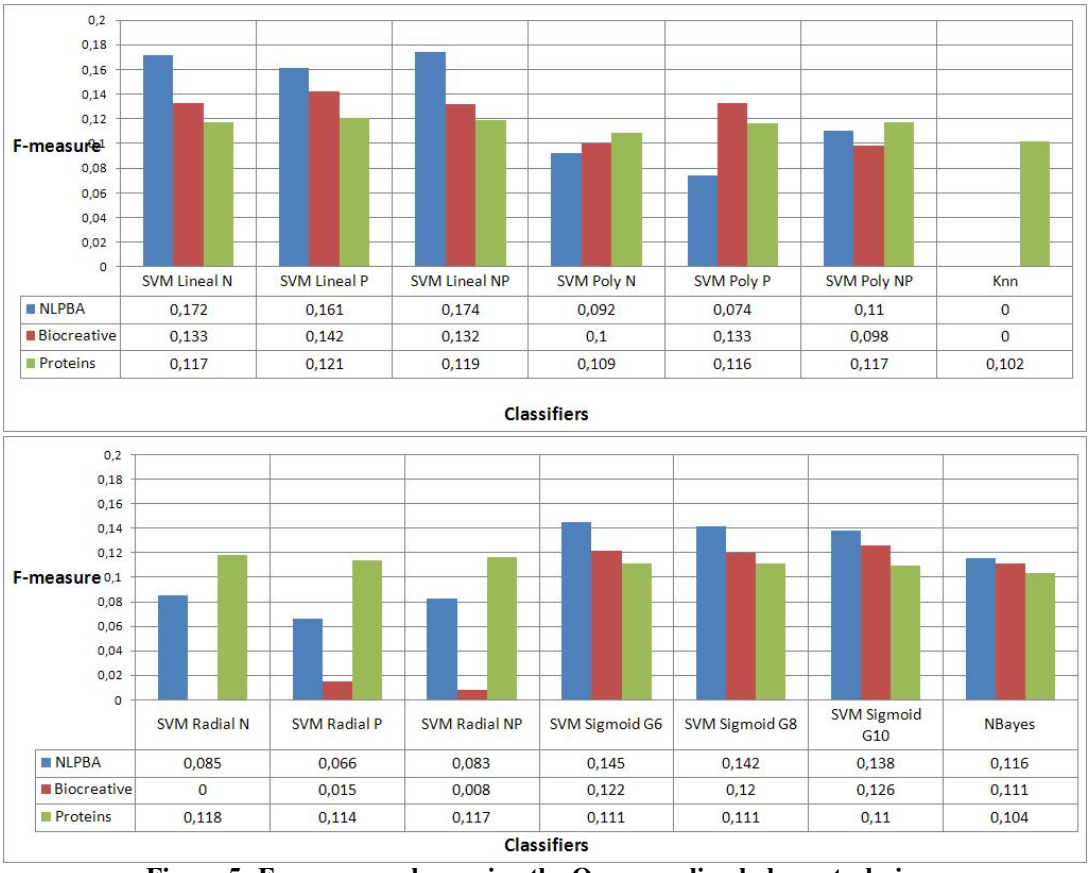


Figure 5: F-measure values using the Oversampling balance technique

between 0.2709 to $0,3114$, and F-measure between $0,138$ to $0,145$). Making a comparative between statistical results shown in tables 1 and 2, we can see that correlation coefficients values based on Proteins and NLPBA dictionaries are pretty much the same.

Table 2: Statistical significance using Oversampling

Dictionaries / Classifiers	NLPBA			BioCreAtIvE			Proteins		
	P-value(F)	P-value(χ^2)	χ^2	P-value(F)	P-value(χ^2)	χ^2	P-value(F)	P-value(χ^2)	χ^2
SVM Lineal N	**⩽0.0001**	**⩽0.0001**	69.900	**⩽0.0001**	**⩽0.0001**	27.991	**⩽0.0001**	**⩽0.0001**	24.402
SVM Lineal P	**⩽0.0001**	**⩽0.0001**	55.342	**⩽0.0001**	**⩽0.0001**	34.511	**⩽0.0001**	**⩽0.0001**	29.196
SVM Lineal NP	**⩽0.0001**	**⩽0.0001**	73.323	**⩽0.0001**	**⩽0.0001**	25.977	**⩽0.0001**	**⩽0.0001**	26.093
SVM Poly N	0.3410	0.3780	0.777	**0.0005**	**0.0001**	14.835	**0,0003**	**0.0005**	12.302
SVM Poly P	**0.0015**	**0.0007**	11.532	**⩽0.0001**	**⩽0.0001**	25.370	**⩽0.0001**	**⩽0.0001**	21.215
SVM Poly NP	**0.0041**	**0.0038**	8.372	0.4670	0.6106	0.259	**⩽0.0001**	**⩽0.0001**	23.781
SVM Radial N	0.0994	0.1167	2.461	0.3425	0.5116	0.431	**⩽0.0001**	**⩽0.0001**	21.628
SVM Radial P	0.4074	0.4575	0.552	0.1223	0.2437	1.359	**⩽0.0001**	**⩽0.0001**	14.790
SVM Radial NP	0.1222	0.1452	2.122	0.4670	0.6106	0.259	**⩽0.0001**	**⩽0.0001**	19.330
SVM Sigmoid G6	**⩽0.0001**	**⩽0.0001**	41.999	**⩽0.0001**	**⩽0.0001**	16.674	**⩽0.0001**	**⩽0.0001**	14.409
SVM Sigmoid G8	**⩽0.0001**	**⩽0.0001**	39.361	**⩽0.0001**	**⩽0.0001**	15.071	**⩽0.0001**	**0.0002**	14.132
SVM Sigmoid G10	**⩽0.0001**	**⩽0.0001**	36.556	**⩽0.0001**	**⩽0.0001**	19.971	**⩽0.0001**	**0.0002**	13.606
Knn	-	-	-	-	-	-	0,0295	0,0341	4.489
NBayes	**⩽0.0001**	**⩽0.0001**	39.955	**⩽0.0001**	**⩽0.0001**	25.907	**0.0005**	**0.0013**	10.321

4 Comparative with another studies

In order to know how good our results are, we have analysed the results published by other authors. Initially we looked for approaches as similar as possible to our work. That is to say,

works with a preprocess data (with the same dictionaries) and data balanced (sampling) similar to ours. As we did not find any work gathering these characteristics, we also researched on similar works published in the TREC Genomics 2005 Communications. As discussed before, the use of sampling strategies considerably improved our results. Therefore, will be chosen these results to compare our work with another contributions.

The Table 3 shows results in the TREC competition and those more important obtained in our experiments (marked in gray). The first two entries in the table correspond to the best values obtained in the competition, and the last three ones correspond to minimum, average and maximum values considering all participants.

Table 3: Comparative based on TREC Genomics 2005 results, sorted by utility measure

Tag	Precision	Recall	F-Measure	Utility
aibamadz05 [4]	0.4669	0.9337	0.6225	0.8710
ABBR003SThr [29]	0.4062	0.9458	0.5683	0.8645
ASVMN03 [29]	0.4019	0.9127	0.5580	0.8327
aNLMB [6]	0.3391	0.9398	0.4984	0.8320
Knn + Protein *(Subsampling)*	0.0520	0.9800	0.0990	0.7762
aQUT14 [41]	0.3582	0.8675	0.5070	0.7760
Knn + NLPBA *(Subsampling)*	0.0510	1.0000	0.0970	0.7620
aMUSCUIUC3 [42]	0.4281	0.8072	0.5595	0.7438
Naive Bayes + Protein *(Subsampling)*	0.0550	0.9530	0.1030	0.7238
SVM Poly N *(Subsampling)*	0.0510	0.9530	0.0970	0.7238
aUCHSCnb1En3 [10]	0.5080	0.7651	0.6106	0.7215
NBayes + NLPBA *(Oversampling)*	0.0620	0.9370	0.1160	0.7110
NBayes + BioCreative *(Oversampling)*	0.0590	0.9250	0.1110	0.7015
SVM Lineal N + NLPBA *(Subsampling)*	0.0700	0.7550	0.1290	0.5643
SVM Sigmoid G10 + NLPBA *(Subsampling)*	0.0780	0.7310	0.1410	0.5452
SVM Lineal NP + NLPBA *(Subsampling)*	0.0800	0.6760	0.1430	0.5005
SVM Poly NP + NLPBA *(Subsampling)*	0.0650	0.6090	0.1170	0.4463
aUCHSCsvm [10]	0.7957	0.4458	0.5714	0.4391
aNLMF [6]	0.2219	0.5301	0.3129	0.4208
Minimum	*0.2191*	*0.2500*	*0.2387*	*0.2009*
Median	*0.3572*	*0.8931*	*0.5065*	*0.7773*
Maximum	*0.7957*	*0.9578*	*0.6667*	*0.8710*

It is interesting to note that these studies have used preprocessing techniques like chi-square [41] to reduce features dimensionality or another kind of thresholds like *Rcut, Pcut or Scut* [6, 29], but in any case balance techniques have been used.

As observed in the table, the classification and balancing techniques chosen for the approach presented here are effective, since better results that similar research works are obtained.

5 Conclusions

In this study we have used three different dictionaries, two balancing algorithms and several reasoning models. These tools were used in order to classify biomedical texts.

First of all, we can say that the use of dictionaries during the preprocessing task is positive. We made tests using only some data mining techniques, such as stemming, tokenizing and a mix between them, getting disappointment results or even uncomputable.

Although in some occasions the dictionaries did not improve the results, as in the case of *SVM* classifier with *Polynomial kernel* and *NLPBA* dictionary, generally their use is essential to get good results. Therefore, best results were obtained with dictionaries *NLPBA* and nearly *Protein*. However *BioCreative* dictionary was disappointed by almost all classifiers.

Regarding balancing techniques, *Subsampling* and *Oversampling* algorithms to solve the unbalanced nature of this information, and improve the classification results were applied. We got best results using the *Subsampling* technique for almost all scenarios, except for *Naive Bayes* which got better results using *Oversampling*. Anyway, the use of class-balancing techniques permits to obtain better results than without it.

At last, reasoning models got different results depending on the preprocessing techniques used. We used a *Cost-SVM* classifier with several kernels, *Naive Bayes* and *Knn*, showing their results through *F-measure* and *Utility* measures. As commented, it is necessary to observe them at the same time, because they are not conclusive individually.

The experimental results show that, regarding to the utility measure, the best classifier is based on the *Polynomial kernel*. On the other hand, taking in account other measures (precision, recall and f-measure) the best balanced result were achieved by the probabilistic SVM with linear and sigmoid kernels.

Acknowledgements

This work has been partially funded by the Spanish Ministry of Science and Innovation, the Plan E from the Spanish Government and the European Union from the ERDF (TIN2009-14057-C03-02).

References

[1] A. Abi-Haidar and L. M. Rocha. Biomedical article classification using an agent-based model of t-cell cross-regulation. In *ICARIS*, pages 237–249, 2010.

[2] D. W. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithms. In *Machine Learning*, pages 37–66, 1991.

[3] A. Anand, G. Pugalenthi, G. B. Fogel, and P. N. Suganthan. An approach for classification of highly imbalanced data using weighting and undersampling. *Amino Acids*, 39(5):1385–1391, 2010.

[4] R. K. Ando, M. Dredze, and T. Zhang. Trec 2005 genomics track experiments at ibm watson. In *In Proceedings of TREC 2005. NIST Special Publication*, 2005.

[5] R. Apweiler, A. Bairoch, C. H. Wu, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, M. J. Martin, D. A. Natale, C. O'Donovan,

N. Redaschi, and L.-S. L. Yeh. Uniprot: the universal protein knowledgebase. *Nucleic Acids Res*, 32(Suppl 1):D115–D119, 2004.

[6] A. R. Aronson. Fusion of knowledge-intensive and statistical approaches for retrieving and annotating textual genomics documents. In *Proc TREC 2005*, pages 36–45, 2005.

[7] R. Bai, X. Wang, and J. Liao. Extract semantic information from wordnet to improve text classification performance. In *AST/UCMA/ISA/ACN*, pages 409–420, 2010.

[8] R. Barandela, J. S. Sánchez, V. García, and E. Rangel. Strategies for learning in class imbalance problems. *Pattern Recognition*, 36(3):849–851, 2003.

[9] B. Boguraev, T. Briscoe, J. Carroll, D. Carter, and C. Grover. The derivation of a grammatically indexed lexicon from the longman dictionary of contemporary english. In *Proceedings of the 25th annual meeting on Association for Computational Linguistics*, pages 193–200, Morristown, NJ, USA, 1987. Association for Computational Linguistics.

[10] J. G. Caporaso. Concept recognition and the trec genomics tasks. the fourteenth text retrieval. In *Conference Proceedings (TREC 2005). Gaithersburg, MD: National Institute for Standards and Technology*, 2005.

[11] J. T. Chang, S. Raychaudhuri, and R. B. Altman. Including biological literature improves homology search. In *In Pacific Symposium on Biocomputing*, pages 374–383, 2001.

[12] N. V. Chawla, N. Japkowicz, and A. Kotcz. Editorial: special issue on learning from imbalanced data sets. *SIGKDD Explorations*, 6(1):1–6, 2004.

[13] N. Collier, P. Ruch, and A. Nazarenko, editors. *JNLPBA '04: Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, Morristown, NJ, USA, 2004. Association for Computational Linguistics.

[14] C. Cortes and V. Vapnik. Support-vector networks. In *Machine Learning*, pages 273–297, 1995.

[15] H. Cunningham. GATE, a General Architecture for Text Engineering. *Computers and the Humanities*, 36:223–254, 2002.

[16] W. B. Frakes and R. A. Baeza-Yates, editors. *Information Retrieval: Data Structures and Algorithms*. Prentice-Hall, Englewood Cliffs, NJ 07632, USA, 1992.

[17] S. R. Garner. Weka: The waikato environment for knowledge analysis. In *In Proc. of the New Zealand Computer Science Research Students Conference*, pages 57–64, 1995.

[18] W. Hersh, A. Cohen, J. Yang, R. T. Bhupatiraju, P. Roberts, and M. Hearst. Trec 2005 genomics track overview. In *In TREC 2005 notebook*, pages 14–25, 2005.

[19] L. Hirschman, A. Yeh, C. Blaschke, and A. Valencia. Overview of BioCreAtIvE: critical assessment of information extraction for biology. *BMC Bioinformatics*, 6(Suppl 1):S1, 2005.

[20] M. Iwayama and T. Tokunaga. Cluster-based text categorization: a comparison of category search strategies. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '95, pages 273–280, New York, NY, USA, 1995. ACM.

[21] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In C. Nédellec and C. Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 137–142, Heidelberg et al., 1998. Springer.

[22] P. Kang and S. Cho. EUS SVMs: Ensemble of Under-Sampled SVMs for Data Imbalance Problems. In I. King, J. Wang, L. Chan, and D. Wang, editors, *Neural Information Processing*, volume 4232 of *Lecture Notes in Computer Science*, chapter 93, pages 837–846. Springer Berlin Heidelberg, 2006.

[23] Y. Liu, P. Scheuermann, X. Li, and X. Zhu. Using wordnet to disambiguate word senses for text classification. In *Proceedings of the 7th international conference on Computational Science, Part III: ICCS 2007*, pages 781–789, Berlin, Heidelberg, 2007. Springer-Verlag.

[24] J. McCrae and N. Collier. Synonym set extraction from the biomedical literature by lexical pattern discovery. *BMC Bioinformatics*, 9:159, 2008.

[25] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. Introduction to wordnet: An on-line lexical database. *Journal of Lexicography*, 3(4):235–244, 1990.

[26] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, July 1980.

[27] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[28] B. Settles. Abner: An open source tool for automatically tagging genes, proteins, and other entity names in text. *Bioinformatics*, 21(14):3191–3192, 2005.

[29] L. Si and T. Kanungo. Thresholding strategies for text classifiers: Trec 2005 biomedical triage task experiments. the fourteenth text retrieval. In *Conference Proceedings (TREC 2005). 2005. Gaithersburg, MD: National Institute for Standards and Technology*, 2005.

[30] A. Sureka, P. Mirajkar, P. Teli, G. Agarwal, and S. Bose. Semantic based text classification of patent documents to a user-defined taxonomy. In R. Huang, Q. Yang, J. Pei, J. Gama, X. Meng, and X. Li, editors, *Advanced Data Mining and Applications*, volume 5678 of *Lecture Notes in Computer Science*, pages 644–651. Springer Berlin / Heidelberg, 2009.

[31] S. Tan. Neighbor-weighted k-nearest neighbor for unbalanced text corpus. *Expert Systems with Applications*, 28(4):667–671, 2005.

[32] Y. Tang, Y. Zhang, and N. V. Chawla. Svms modeling for highly imbalanced classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(1):281–288, 2009.

[33] L. S. Venkata, S. Mukherjea, and D. Punjani. Biomedical document triage: Automatic classification exploiting category specific knowledge. In *TREC*, 2005.

[34] E. M. Voorhees and L. P. Buckland, editors. *Proceedings of the Fourteenth Text REtrieval Conference, TREC 2005, Gaithersburg, Maryland, November 15-18, 2005*, volume Special Publication 500-266. National Institute of Standards and Technology (NIST), 2005.

[35] G. M. Weiss. Mining with rarity: a unifying framework. *SIGKDD Explor. Newsl.*, 6:7–19, June 2004.

[36] E. D. Wiener, J. O. Pedersen, and A. S. Weigend. A neural network approach to topic spotting. In *Proceedings of SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval*, pages 317–332, Las Vegas, US, 1995.

[37] Y. Yang. A study on thresholding strategies for text categorization. In *Proceedings of SIGIR-01, 24th ACM International Conference on Research and Development in Information Retrieval*, pages 137–145. ACM Press, 2001.

[38] Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 42–49, New York, NY, USA, 1999. ACM.

[39] C. Zhaif. Uiuc/musc at trec 2005 genomics track. the fourteenth text retrieval. In *Conference Proceedings (TREC 2005). Gaithersburg, MD: National Institute for Standards and Technology*, 2005.

[40] J. Zhang and I. Mani. knn approach to unbalanced data distributions: A case study involving information extraction. In *In Proceedings of the ICML'2003 workshop on learning from imbalanced datasets*, 2003.

[41] Z. H. Zheng. Applying probabilistic thematic clustering for classification in the trec 2005 genomics track. the fourteenth text retrieval. In *Conference Proceedings (TREC 2005). 2005. Gaithersburg, MD: National Institute for Standards and Technology. http://trec.nist.gov/pubs/trec14/papers/queensu.geo.pdf*, 2005.

[42] G. Zhou, D. Shen, J. Zhang, J. Su, and S. Tan. Recognition of protein/gene names from text using an ensemble of classifiers. *BMC Bioinformatics*, 6(Suppl 1):S7, 2005.