BioTextRetriever: yet another Information Retrieval system

Célia Talma Gonçalves^{1,3,5,6}, Rui Camacho^{1,2,4}, and Eugénio Oliveira^{1,2,3}
talma@fe.up.pt, rcamacho@fe.up.pt, eco@fe.up.pt

(1) Faculdade de Engenharia da Universidade do Porto, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
(2) Departamento de Engenharia Informática (DEI)
(3) LIACC
(4) LIAAD-INESCTEC
(5) Instituto Superior de Contabilidade e Administração do Porto, Rua Jaime Lopes Amorim, s/n, 4465-004 S. Mamede de Infesta
(6) Instituto Politécnico do Porto

Abstract. *It is of capital importance for every researcher to be aware of the work that has been done in his research area. However, finding "interesting/relevant" publications in the overwhelming amount of documents available in the Internet is quite difficult. We propose the use of Text Mining to address this information overload problem by automating the process of extracting relevant papers from very large repositories of scientific literature. We present in this paper, the automatic construction of a classifier capable of selecting the relevant papers among the whole MEDLINE, that is part of a software developed tool: BioTextRetriever. The empirical evaluation of the work shows a classifier's accuracies around 95%.*

Key words: Information Retrieval, Text Mining, Machine Learning, MEDLINE

1 Introduction

It is of capital importance for every researcher to be aware of the work that has been done in his research area. With the advent of the Internet, the amount of information available to everyone is usually overwhelming. Researchers have difficulty in accessing the right information amidst the overwhelming amount of documents available.

A frequent task for a Molecular Biologist researcher, when studying a new genomic or proteomic sequence, is to search for similar sequences and then look for their associated papers. The aim is to collect information related to the sequences under study. The search for such relevant papers, starting with the search for similar sequences, can be done on Web sites like the one of NCBI. However, the number of papers associated with stored sequences is rather small and not the most recent ones. A search on larger repositories is required. Due to the amount of information retrieved, a search in Google or MEDLINE using a set of keywords is most of the time disappointing. The number of papers is huge and a large percentage of them irrelevant.

We propose the use of Text Mining to address the information overload problem by automating the process of extracting the relevant parts of the scientific literature. This automation may increase the efficiency of searching for information and allow automated inference of new information.

2 State-of-the-Art

"The goal of biomedical text mining is to allow researchers to identify needed information more efficiently, uncover relationships obscured by the sheer volume of available information, and in general shift the burden of information overload from the researcher to the computer by applying algorithmic, statistical and data management methods to the vast amount of biomedical knowledge that exists in the literature as well as the free text fields of biomedical databases "[1]. We now present a set of tools specially designed for Text Mining applied to bioinformatics problems.

BioRAT [2] is a Biological Research Assistant for Text Mining, that accepts a query and autonomously finds a set of papers and highlights the most prominent facts in each paper. BioRAT combines tools to download papers, to extract information from papers and to design templates to allow this extraction (uses GATE (General Architecture for Text Engineering) which is a general purpose text engineering system based on NLP developed at Sheffield University.). When the user enters a query, a list of papers is presented to him and then the user constructs a template that helps in the extraction of the proper information. The output is presented in the XML format. BioRAT is a web-based friendly tool but it is restricted to documents in PDF format which is a limitation of this system and it also uses the traditional query search.

GoPubMed [1] [5] is a knowledge-based search engine for biomedical literature from PubMed. GoPubMed submits keywords to the PubMed database obtaining the correspondent biomedical literature to be indexed using the Gene Ontology and the Medical Subject Headings. GoPubMed uses several pre-processing techniques (stemming, tokenization, synonym detection). GoPubMed identifies relevant biomedical concepts associated with the query. GoPubMed does not rank results or provide importance scores for papers.

EBIMed[2] [3] is a Web application that combines Information Retrieval and Extraction from MEDLINE. EBIMed's aim is to recognize protein/genes names, GO annotations, drugs and species from PubMed returned abstracts and semantically annotate these abstracts; EBIMed also extracts significant co-ocurrences between annotated entities. The EBIMed user interface is difficult to navigate, besides it only provides quicker results if the number of documents to analyse are limited.

[1] is available at `http://www.gopubmed.org`

[2] is available at `www.ebi.ac.uk/Rebholz-srv/ebimed`

FACTA (Finding Associated Concepts with Text Analysis) [15] is a Text Mining tool that searches for biomedical abstracts that are potentially relevant to a user query. FACTA can discover associations between biomedical concepts contained in MEDLINE articles; the analyses of the documents retrieved is based on a statistical method. Although it is an interesting and recent tool it only searches keywords, unlike the system we are proposing. The advantages of FACTA is that it is easy to use and delivers real-time responses while being able to accept flexible queries.

PubFocus[3] [11] performs a statiscal analysis of MEDLINE/PubMed search queries by enriching them with bibliometric indicators: the journal impact factor, the number of citations and the authors impact on the field of search. PubFocus provides a list of articles ranked by relevancy.

ReleMed[4] [12] is a search engine, from the University of Virginia's School of Medicine, that searches PubMed for medical literature and presents the results by relevancy based on keywords. Relemed finds articles that present a close relation among the search terms. ReleMed categorizes each retrieved citation into 8 different levels of relevance, depending on the frequency of occurrence of the search terms within the title, sentences of the abstract and MeSH. ReleMed presents the most relevant results first.

MedlineRanker [5] according to their developers [8] allows a flexible ranking in Medline for a topic of interest without expert knowledge. The MedlineRanker webserver requires as input a list of abstracts relevant to a particular topic and then the tool learns the most discriminative words (common words) in those abstracts. These words are used to score newly published articles. MedlineRanker uses a naïve bayiesan classifier for the learning and classification process. The authors claim that if the input contains closely related abstracts MedlineRanker returns relevant abstracts from the recent bibliography with high accuracy and that the tool processes thousands of abstracts from the Medline database in a few seconds, or millions in few minutes. According to the authors the MedlineRanker will produce more accurate results if the user provides a training set with enough abstracts (100-1000) to define the topic of interest.

A platform for Biomedical Text Mining (BioTM) called @Note [10] promotes a set of user-friendly tools for biomedical document retrieval, annotation and curation. Its main functional contributions are: the ability to process abstracts and full-texts; an information retrieval module enabling PubMed search and journal crawling; a pre-processing module with PDF-to-text conversion, tokenization and stopword removal; a semantic annotation schema; a lexicon-based annotator; a user friendly annotation view that allows to correct annotations and a Text Mining Module supporting dataset preparation and algorithm evaluation. This platform uses the GATE (General Architecture for Text Engineering) features fot the pre-processing and a low-level plugin (YALE) that includes the WEKA tool (a collection of Machine Learning algorithms) for the classification process.

[3] available at `www.pubfocus.com`

[4] `http://www.relemed.com`

[5] `http://cbdm.mdc-berlin.de/~medlineranker/about.html`

While the above works focus only on pre-processing, or classification or ranking, our work is intended to implement all of these procedures into just one tool. As far as we are concerned, there is not such a tool that classifies MEDLINE papers and ranks the results according to bibliometrics indicators.

3 BioTextRetriever Architecture

We have developed BioTextRetriever: a Web-based search tool for retrieving relevant literature in Molecular Biology and related domains from repositories such as MEDLINE. Figure 1 shows the overall processing of the BioTextRetriever tool. This paper focuses on step 4, i.e., the automatic construction of a classifier to filter out relevant papers from a very large repository.

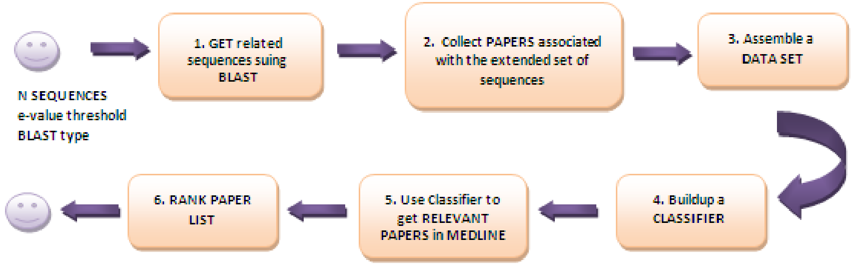

1. GET related sequences suing BLAST
2. Collect PAPERS associated with the extended set of sequences
3. Assemble a DATA SET
N SEQUENCES
e-value threshold
BLAST type
6. RANK PAPER LIST
5. Use Classifier to get RELEVANT PAPERS in MEDLINE
4. Buildup a CLASSIFIER

Fig. 1. Sequence of steps implemented by BioTextRetriever

In order to understand the procedure involved in step 4 of the tool we provide a context and summarize the previous steps. BioTextRetriever accepts a set of genomic or proteomic sequences and returns an ordered set of references to scientific papers reporting work considered relevant for the study of the given sequences. With the input sequences, and using the NCBI BLAST tool, a set of similar sequences is collected from PubMed (step 1).

Each of the similar sequences has a set of papers associated. Using the extended set of sequences the tool collects (step 2), the following information from MEDLINE: pmid, journal title, journal ISSN, article title, abstract, list of authors, list of keywords, list of Medical Subject Headings[6] (MeSH) terms and publication date. Considering the scope of this research work we are only taking into account paper references that have an abstract available in MEDLINE.

To the set of relevant papers (associated with the similar sequences) that represent the “positive examples” we add an equal number of “negative examples” (less relevant/ irrelevant papers).

We have done a set of experiments to determine a proper way of collecting the negative examples (details are in [9]). In this previous study [9], we have considered three possible solutions to collect the negative examples, which were the

[6] The MeSH list is a set of controlled terms for the indexation of MEDLINE articles

Near-Misses Value method (NMV), the MeSH Random Value method (RMV) and the Random Value method (RV).

In the NMV method, from the number of positive examples returned, we divide them (based on the e-value introduced by the user) into positive examples and negative examples. The positive examples are the ones that have e-value lower then the e-value introduced by the user. The negative examples (if they exist) are the ones that are greater than this e-value and lower then a relaxation value defined by us. To proper distinguish between positive and near-misses examples, we discard a a "no man's land" zone that is 10% of the number of "not similar" sequences associated with the introduced sequence, and we gather the ones that are farthest away from the relevant ones.

The MRV method combines the previous method with some N randomly selected irrelevant papers generated by the LDB. However, these irrelevant papers must have maximum number of MeSH terms in common with the relevant papers. The idea is to generate papers that although being far apart (as far as "e-value" criteria is concerned), still have something in common with the relevant papers to improve classifier robusteness.

The RV approach randomly collects papers from MEDLINE in a number equal to the number of relevant papers. We guarantee that in this set of irrelevant papers there are no relevant papers (from the ones associated to the original sequences). And these randomly selected papers have also the maximum number of MeSH terms in common with the positive relevant papers. The main idea is to obtain negative examples but that are somehow related to the sequences introduced.

After step 2 we have a "proto data set". The proto data set is pre-processed and converted into a data set in step 3. The pre-processing techniques applied include: Handling Synonyms; Stop-words removal; Word validation using a dictionary and Stemming.

Step 5, the use of the classifier to retrieve papers from the whole MEDLINE, and 6, the ranking of of retrieved papers is outside the scope of this paper. The classifier is constructed in step 4 as described in the next sections.

4 Classifier Construction Process

This section addresses the question of how to construct a classifier to filter relevant papers in MEDLINE. To address this problem we have considered to combine different partitions of the original data set with different ways of using the Machine Learning algorithms (either isolated or in an ensemble).

We have considered and evaluated five possible alternatives for these combinations. We now describe the different alternatives and present their evaluation in Section 4. The different alternatives considered different amounts of data to construct the classifiers and the use of single classifiers or ensembles of classifiers. In all cases a 10-fold cross-validation method was used to evaluate the alternatives. In all experiments the base algorithms are the ones in Table 1.

Table 1. Machine Learning algorithms used in the study.

Tool	Algorithm	Type
ZeroR	Majority class predictor	Rule learner
smo	Sequential Minimal Optimization	Support Vector Machines
rf	Random Forest	Ensemble
ibk	K-nearest neighbours	Instance-based learner
BayesNet	Bayesan Network	Bayes learner
j48	Decision tree (C4.5)	Decision Tree learner
dtnb	Decision table/Naïve bayes hybrid	Rule learner
AdaBoost	Boosting algorithm	Ensemble learner
Bagging	Bagging algorithm	Ensemble learner
Ensemble Selection	Combines several algorithms	Ensemble learner

We have developed and used a wrapper to tune each of the Machine Learning algorithms' parameters and also to apply different parameters for each of the stand alone algorithms. In the following discussion we use T to representing the number of basic algorithms (6 in our case), and M to represent the number of Ensemble learners (3 in our case). Table 2 presents, in a schematic way, the five alternatives considered in this study that we now describe in detail.

Alternative 1

In alternative 1 all algorithms are individually applied to the whole data (shown in Table 2).

Alternative 2

In Alternative 2 (shown in Table 2) the whole data to train T basic classifiers (C_i) that were then combined in an ensemble. For the ensemble we have used three well known algorithms: AdaBoost, Bagging and Ensemble Selection. Different "ensemble parameters" were tested as explained latter in Section 5.

Alternative 3

The third alternative (shown in Table 2) divides the original data set into T equal and balanced parts, where T is the number of basic classifiers. Each basic algorithm runs on a single partition of the data set. At the end we ensemble the results of the T classifiers adding a voting scheme algorithm to the wrapper. We have implemented the simple plurality vote scheme [6], where each base classifier assigns a vote for its prediction, and the example is classified in the class with more votes.

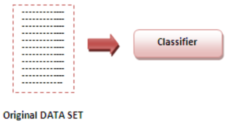
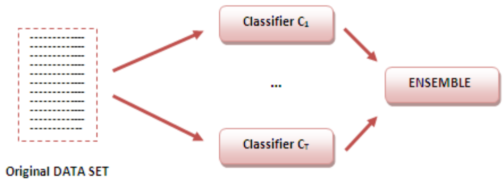
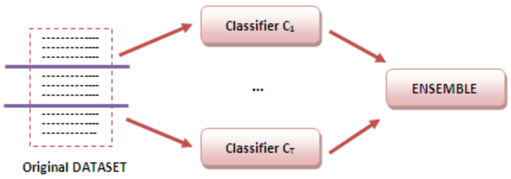
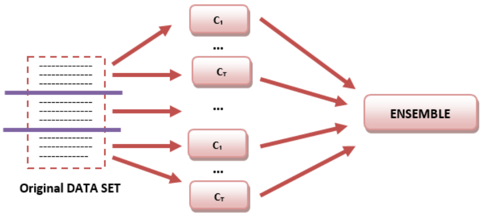
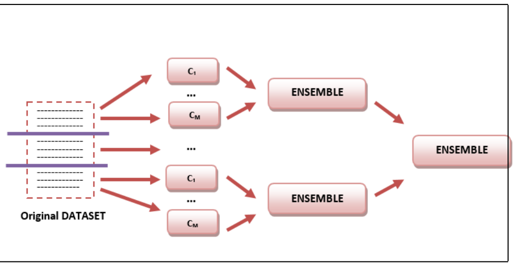
Alternative 4

Table 2 shows the fourth alternative that also divides the original data set into T equal parts as in alternative 3. But in this alternative each part of the original data will be used to train all the different learning algorithms, and the ensemble is made in the end. Thus we will train T classifiers in each part of the original data set and in the end we will ensemble T * T classifiers. Like alternative 3 we have used the same simple plurality voting scheme for the ensemble algorithm.

Alternative 5

The fifth alternative (shown in Table 2), divides the data set into M equal parts (where M is the number of ensemble algorithms). For each sub set an ensemble is made using the best parameters of the basic algorithms (obtained in alternative 1). In the end, we make an ensemble of these M ensembles using a majority voting schema.

Table 2. The five Alternatives analised in the study.

<table>
<tr><th colspan="2">Alternative 1</th></tr>
<tr><td colspan="2"> </td></tr>
<tr><th>Alternative 2</th><th>Alternative 3</th></tr>
<tr><td> </td><td> </td></tr>
<tr><th>Alternative 4</th><th>Alternative 5</th></tr>
<tr><td> </td><td> </td></tr>
</table>

5 Evaluating the alternatives

The data sets used in the evaluation of the experiments are characterized in Table 3. In all experiments a 10-fold cross-validation was used. As standard procedure in Machine Learning we have generated several data sets based on sequences that belong to five different domains: S (RNASE), Hemoglubin (H), (ERYT), Hypertension (HYP), Blood Pressure (BP) and Lung Disease (LUNG).

5.1 Alternative 1

In the first alternative basic algorithms were used with the whole data and their results compared. We have used the WEKA classifiers: smo, rf, ibk, bayesnet, j48,

Table 3. Characterization of data sets used to assess the 5 alternatives.

Data Sets	Number of Attributes	Positive Examples	Negative Examples	Total Examples
HYP11	1706	130	130	260
HYP21	1944	194	194	388
BG11	1546	97	97	194
BG21	1631	115	115	230
BG31	1859	149	149	298
LUNG21	1535	120	120	240

and dtnb. This first alternative allows the evaluation of stand alone algorithm's performance with the data. Within this alternative only we have also considered the use of the ILP system Aleph[14].

Table 4 shows the accuracy results obtained using the WEKA and ILP classifiers.

As a global result we can see that all algorithms have, in general, very good performances, well above the majority class predictor (that is around 50%). We can also conclude that *BayesNet*, *dtnb*, *j48* and *rf* are the best algorithms. According the the t-student test there is however no statistical difference between them. We can also conclude that the *ibk* algorithm performed worse.

Within this first alternative we have considered the use of an ILP system. The Aleph ILP system was applied to the same data sets. All the information was encoded in Prolog. To estimate the predictive quality of the classification models we performed 4 fold cross-validation.

The set of files built to give to the Aleph system were generated from the ARFF files and from the information stored in our local MEDLINE database, which we have used in our previous experiments using propositional-based algorithms. We have encoded the features used in the propositional learners as background knowledge and have added a set of additional predicates. Table 4 presents the accuracy results between all the propositional algorithms and ILP results. To compare the propositional and ILP results obtained, we have used the statistical t-student test ($\alpha = 0.05$) [13].

Table 4. Results for Alternative 1 using accuracy (%).

Data Sets	ZeroR	smo	rf	ibk	bayesnet	j48	dtnb	ILP
HYP11	50.0	81.9(5.5)	86.5(5.2)	55.4(6.1)	91.2(4.1)	87.3(6.6)	84.6(6.3)	84.3(3.2)
HYP21	49.0	91.3(3.2)	93.8(4.4)	59.3(6.4)	93.2(3.8)	92.5(4.6)	90.2(4.3)	91.2(3.5)
BG11	48.5	86.6(9.5)	92.2(4.5)	54.6(12.9)	93.3(3.5)	93.2(3.6)	93.2(3.6)	90.6(5.3)
BG21	47.8	85.7(8.5)	92.2(5.3)	51.3(10.6)	93.5(4.7)	94.4(5.0)	95.7(3.6)	91.8(3.2)
BG31	49.7	84.6(7.6)	88.3(3.9)	53.7(11.0)	91.6(4.0)	91.0(5.5)	91.3(5.0)	86.9(3.4)
LUNG21	50.0	83.8(8.2)	90.8(6.8)	66.3(13.0)	90.8(5.5)	88.3(6.8)	91.3(5.0)	88.7(3.2)
Overall Average	49.2	85.6(7.1)	90.6(5.0)	56.8(10.0)	91.3(4.7)	91.1(5.4)	91.1(4.6)	88.9(3.6)

The accuracy results obtained in Table 4 are very similar. The t-student test gave no statistical significant difference between *dtnb*, *bayesnet*, *j48* and *rf*. Thus according to these results ILP is the fourth best algorithm. Regarding the results we have decided not to use ILP in the following alternatives because we did not achieve very good results in this first alternative and is very time consuming.

5.2 Alternative 2

The second alternative is quite similar to the first one. The difference is that an ensemble of the basic classifiers is used. This alternative uses the best parameter combinations found in alternative 1 for each basic classifier. In the end the ensemble is made using the WEKA's ensemble classifiers Bagging, AdaBoost and Ensemble Selection. Results are shown in Table 5.

Table 5. Ensemble's Accuracy results in Alternative 2. The bold values are statistically different from the second best values.

Data Sets	**AdaBoost**	**Bagging**	**Ensemble Selection**
HYP11	91.2 (4.1)	90.0 (6.3)	91.2 (4.1)
HYP21	**95.1 (1.9)**	**95.1 (3.3)**	89.7 (6.5)
BG11	93.8 (3.4)	94.3 (4.6)	94.3 (5.1)
BG21	95.7 (3.6)	94.4 (4.6)	93.9 (5.1)
BG31	93.9 (3.2)	91.6 (4.8)	92.3 (5.0)
LUNG21	93.8 (4.1)	92.5 (4.7)	92.5 (4.3)
Overall Average	93.9 (3.4)	93.0 (4.7)	92.3 (5.0)

The t-student test gave statistical significance between the average results and the ZeroR algorithm.

Table 6. Ensemble's Precision/Recall/F-Measure results in Alternative 2. The bold values are statistically different from the second best values.

Data Sets	**AdaBoost**	**Bagging**	**Ensemble Selection**
HYP11	94(8) / 90(8) / 91(4)	94(8) / 91(8) / 90(7)	95(8) / 90(8) / 91(4)
HYP21	97(7) / 97(5) / 95(2)	93(7) / 98(3) / 95(3)	92(8) / 90(1) / 89(7)
BG11	95(8) / 97(4) / 94(3)	99(4) / 94(8) / 94(6)	99(3) / 92(7) / 94(6)
BG21	95(8) / 97(4) / 94(3)	99(4) / 94(8) / 94(6)	99(3) / 92(7) / 94(6)
BG31	97(6) /93(5) / 94(3)	98(4) / 94(6) / 91(4)	96(7) / 90(6) / 92(6)
LUNG21	95(8) / 96(7) / 93(5)	100(0.0) / 93(8) / 92(5)	93(8) / 92(7) / 92(5)
Overall Average	96(6) / 95(6) / 94(3)	97(4) / 94(7) / 93(5)	96(6) / 91(6) / 92(6)

As a global result we can see that all algorithms have in general very good performance, well above the majority class predictor (see the ZeroR result in Table 4). As expected through the literature [4] ensemble learners have a higher and uniform performance than base learners.

The three ensemble learners used (Bagging, AdaBoost and Ensemble Selection) according to the t-student test ($\alpha = 0.05$) have no statistically significant difference. Thus we conclude that one can use either one of the three proposed ensemble learners.

5.3 Alternative 3

The Alternative 3 applies the following steps to obtain the presented results. In the first step we apply cross-validation to the original data set. The second step divides each of the data sets into train/test sets. The third divides the data set (the train data sets) into T equal parts, where T is the number of classifiers. To each of the T parts one of the base learning algorithms of alternative 1 was used creating a model. In the end the models were used in the test parts created in the second step. Then we make an ensemble of the T classification results through the wrapper-ensemble developed. This wrapper-ensemble implements the voting algorithm on the ensemble. We have implemented the simple plurality vote scheme [7], where each base classifier assigns a vote for its prediction, and the example is classified in the class with more votes. Table 7 presents the results obtained, and from which we can conclude that the accuracy of this alternative is around 87%. Results in Table 7 (Alt 3).

5.4 Alternative 4

The fourth alternative, like alternative 3, divides the data set into T equal parts where T is the number of classifiers. The difference from alternative 3 is that in alternative 4, T classifiers are constructed for each of the T parts. T * T classifiers are constructed in total. In the end we make an ensemble using the voting algorithm of the wrapper-ensemble. Table 7 presents the results obtained and from which we can conclude that the accuracy of this alternative is around 89%. Results in Table 7 (Alt 4).

5.5 Alternative 5

The fifth, and last alternative, like alternative three and four, also divides the data set into M equal parts where M is the number of ensemble algorithms. Just like alternative fourth applies the WEKA's propositional algorithms to each divided subset, then uses the WEKA's ensembles algorithms for the ensemble of the T parts. In the end uses the wrapper-ensemble to make an ensemble of an ensemble and then applies the voting scheme algorithm. Table 7 (Alt 5) presents the results obtained and from which we can conclude that the accuracy of this alternative is around 90%.

6 Comparing the five alternatives

The proposed alternatives include the combination of different partitions of the data together with different types of classifiers.

Table 7 shows the accuracies of the best classifier of each of the five alternatives. For Alternative 1 and Alternative 2 the accuracies result from the best algorithm so it may result from different algorithms.

From the results presented in the accuracy Table 7, we can say that ensemble learners performed better, as expected, than base learners. Although alternative 2 has the highest average over the set of data sets, alternative 2 is not statistically significantly different from alternative 1. Figure 2 is a graphical representation that compares the five alternatives in terms of accuracy results.

Table 7. Best accuracies achieved in each alternative.

Data Set	Alt1	Alt2	Alt3	Alt4	Alt5
HYP11	91.2 (4.1)	91.2 (4.1)	78.6 (4.8)	80.5 (4.8)	84.6 (4.3)
HYP21	93.8 (4.4)	95.1 (1.9)	92.0 (1.4)	94.5 (0.8)	94.1 (3.2)
BG11	93.3 (3.5)	94.3 (4.6)	87.4 (4.3)	89.9 (0.2)	91.0 (1.3)
BG21	95.7 (3.6)	95.7 (3.6)	84.4 (2.5)	85.1 (2.1)	88.2 (3.2)
BG31	91.6 (4.0)	93.9 (3.2)	86.3 (1.1)	91.2 (3.4)	93.5 (3.2)
LUNG21	91.3 (5.0)	93.8 (4.1)	84.4 (6.2)	87.1 (6.6)	87.3 (6.4)
Overall Average	92.8 (4.1)	**94.0 (3.7)**	85.6 (3.4)	88.1 (3.0)	89.8 (3.6)

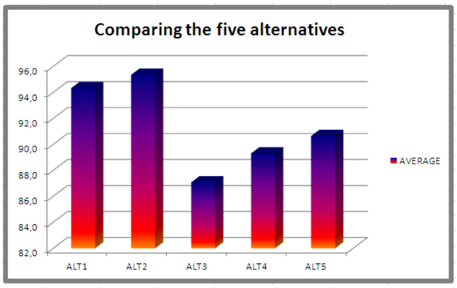


Fig. 2. Comparing the five alternatives accuracies average.

7 Conclusions

This article focused on the construction of the classifier, the main part of step 4 of BioTextRetriever's architecture. The most important part of the study presented in this paper was the exaustive study of the possibilities to address the construction of the classifier figuring out the best alternative of the experimented alternatives. The first alternative shows that the best stand alone classifiers were *dtnb*, *j48* and *rf*. We can also conclude that the application of ILP performed worst than most of the others. Regarding the best alternative we concluded that using ensemble learners (Alternative 2) achieves the best results.

References

1. Aaron M. Cohen and William R. Hersh, *A survey of current work in biomedical text mining*, Brief Bioinform **6** (2005), no. 1, 57–71.
2. David P. A. Corney, Bernard F. Buxton, William B. Langdon, and David T. Jones, *Biorat: extracting biological information from full-length papers*, Bioinformatics (2004), no. 17, 3206–3213.
3. Rebholz-Schuhmann D., Kirsch H., Arregui M., Gaudan S., Riethoven M., and Stoehr P., *Ebimed - text crunching to gather facts for proteins form medline*, Bioinformatics (2007), e237–e244.
4. Thomas G. Dietterich, *Ensemble methods in machine learning*, Proceedings of the First International Workshop on Multiple Classifier Systems (London, UK, UK), MCS '00, Springer-Verlag, 2000, pp. 1–15.
5. Andreas Doms and Michael Schroeder, *Gopubmed: Exploring pubmed with the geneontology*, Nucleic Acid Research (2005), W783–W786.

6. Saso Džeroski and Bernard Ženko, *Is combining classifiers with stacking better than selecting the best one?*, Mach. Learn. **54** (2004), 255–273.
7. Saso Dzeroski and Bernard Zenko, *Is combining classifiers with stacking better than selecting the best one?*, Machine Learning (2004), no. 3, 255–273.
8. Jean-Fred Fontaine, Adriano Barbosa-Silva, Martin Schaefer, Matthew R. Huska, Enrique M. Muro, and Miguel A. Andrade-Navarro, *MedlineRanker: flexible ranking of biomedical literature*, Nucl. Acids Res. (2009), no. suppl_2, W141–146.
9. Célia Talma Gonçalves, Rui Camacho, and Eugenio Oliveira, *From sequences to papers: An information retrieval exercise*, ICDM Workshops (Myra Spiliopoulou, Haixun Wang, Diane J. Cook, Jian Pei, Wei Wang, Osmar R. Zaïane, and Xindong Wu, eds.), IEEE, 2011, pp. 1010–1017.
10. Anália Lourenço, Rafael Carreira, Sónia Carneiro, Paulo Maia, Daniel Glez-Peña, Florentino Fdez-Riverola, Eugénio C. Ferreira, Isabel Rocha, and Miguel Rocha, *@note: A workbench for biomedical text mining*, Journal of Biomedical Informatics **42** (2009), no. 4, 710–720.
11. Maksim Plikus, Zina Zhang, and Cheng M. Chuong, *PubFocus: semantic MEDLINE/PubMed citations analytics through integration of controlled biomedical dictionaries and ranking algorithm*, BMC Bioinformatics (2006), no. 1, 424.
12. Mir Siadaty, Jianfen Shu, and William Knaus, *Relemed: sentence-level search engine with relevance score for the MEDLINE database of biomedical articles*, BMC Medical Informatics and Decision Making (2007), no. 1.
13. Mark D. Smucker, James Allan, and Ben Carterette, *A comparison of statistical significance tests for information retrieval evaluation*, Proceedings of the sixteenth ACM conference on Conference on information and knowledge management (New York, NY, USA), CIKM '07, ACM, 2007, pp. 623–632.
14. A. Srinivasan, *The aleph manual, 2003. available from http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn/aleph.*
15. J Tsuruoka, Tsujii J., and Ananiadou S., *Facta: A text search engine for finding biomedical concepts.*, (2008).