A HMM Text Classification Model with Learning Capacity

A. SEARA VIEIRA[a], E.L. IGLESIAS[a], L. BORRAJO[a], AND R. ROMERO[a]

[a]*Computer Science Dept., Univ. of Vigo, Escola Superior de Enxeñería Informática, Ourense, Spain*

KEYWORD

Hidden Markov Model, Text classification, BioInformatics, Adaptive models

ABSTRACT

In this paper a method of classifying biomedical text documents based on Hidden Markov Model is proposed and evaluated. The method is integrated into a framework named BioClass. Bioclass is composed of intelligent text classification tools and facilitates the comparison between them because it has several views of the results.

The main goal is to propose a more effective based-on content classifier than current methods in this environment To test the effectiveness of the classifier presented, a set of experiments performed on the OSHUMED corpus are preseted. Our model is tested adding it learning capacity and without it, and it is compared with other classification techniques. The results suggest that the adaptive HMM model is indeed more suitable for document classification.

1. Introduction

Text classification is the process of using automated techniques to assign text samples into one or more of a set of predefined classes. In biomedicine area, text classification has experimented an increasing interest given the volume of information currently available, too big to be treated manually.

Thera are a lot of techniques used in the classification task: Support Vector Machines (SVM) [Harris, 2014, Gu et al., 2014, Maldonado et al., 2014], Naive Bayes classification [Farid et al., 2014, Wang et al., 2014b] and a number of other machine learning techniques.

In previous studies, we have proposed a model based on the Hidden Markov Models, called T-HMM [Seara Vieira et al., 2014]. This model aims to classify documents according to their content. The classifier is focused on distinguishing relevant and non-relevant documents from a dataset, and deals with the problem, common among search systems, of whether a document is relevant or not given the type of user query. T-HMM offers a simpler and parameterizable structure based on word relevance, which allows the model to be adjusted to future documents. Experimental results show that the application of this model appears to be promising.

In this article, an improved version of the T-HMM model is proposed, adding a feedback process to generate an adaptive model. The objective is to provide a solution to another kind of text classification problems where a previously created model needs to be adapted to each new document in an iterative learning frame.

Both systems, basic T-HMM and adaptive T-HMM, are integrated in a framework of intelligent text classification tools named BioClass [Romero et al., 2014]. With BioClass the user can work with preprocessed document corpus and visualize them, as well as apply different conventional reasoning

techniques to text classification. The incorporation of T-HMM models in the framework facilitates their comparison with other text classification models.

The rest of the paper is organized as follows: Section 2 presents the general process of classification. Section 3 explains the process of adaptation of the HMM model to the text classification that results in the T-HMM. Section 4 details the improved version of T-HMM with learning. Section 5 discusses the integration of the new techniques in BioClass and Section 6 presents the experiments and results obtained in the case study. Finally, in Section 7 we draw conclusions.

2. General process of classification

In a general process of biomedical text classification, it is necessary to identify the relevant documents for the research. To do so, the similarity between each new document to be classified and the documents already classified is usually measured. It is, therefore, a problem of supervised learning. In this case, the classification process is composed of two serial phases: the training and the test.

2.1 Training phase

The training phase is composed of three tasks: annotation, filter and training.

Annotation process is the basis of text and information retrieval. The process consists of reducing each document to a finite number of representative terms: those words that allow each document to be represented. There are several well-known techniques that can be used to infer the more significant feature words in each document: stemming, stopwords, bigramas, trigramas, use of dictionaries, etc. [Baeza-Yates and Ribeiro-Neto, 1999].

As a result of this process a document matrix is generated (sparse matrix) formed by N documents and M relevant attributes to each document. This way, each document is represented by its associate vector of attributes. In addition, a field is added to define the class to which the document belongs. In this article two types of classes, relevant and non-relevant have been considered. Those documents that are of interest to the research will belong to the relevant class and the rest belong to the non-relevant class.

The filter task tries to reduce the resulting group of data, eliminating those data that can contribute minimum information to the corpus, with minimal loss of expressive power. This type of task usually incorporates large quantity of algorithms that carry out statistical calculations trying to establish some type of term weighting.

The filter algorithms are usually divided into two big groups: filtered of instances and filtered of attributes. The instance filter algorithms reduce the number of documents belonging to each class. The attributes filter algorithms try to reduce the number of attributes to the most representatives. The use of both methods considerably decreases the dimensionality of the initial group and reduces the heavy computational load required. It is also necessary to mention that some of these algorithms, especially those of instance filtering, allow solving problems associated to the overtraining of the reasoning models.

Finally, the training task consists of training the reasoning system using a certain corpus. As previously mentioned, the annotation and filter tasks should be carried out over the corpus, and the sparse matrix resized to be sufficiently representative.

2.2 Test phase

The test phase is divided in three tasks similar to the previous ones: reannotation, filter and test.

The reannotation task creates the test matrix. Although follows the same action carried out for the annotation and building of the training matrix, it dif-

fers in that only the terms extracted from the training document corpus will be annotated.

The filter task on a test corpus is simpler than described in the previous phase because it uses the same group of data (training) to annotate the new corpus. Only if the two matrices are processed individually, when the matrices have their own group of annotations, will the attribute filtering process be applied in the two phases, obtaining as a result two matrices with the same attributes. This process is known as batchmode.

Finally, the test task uses the reasoning model previously trained with the new group of data, inferring whether the test documents are relevant. As with the present study, two classes are considered: Relevant and Non-Relevant. the Class field will contain one of these two values.

The final goal of this phase is to use the model trained in the previous stage to infer what documents of the test corpus can be relevant for the researcher.

3. T-HMM: An HMM applied to text classification

A study on the application of HMM in text classification and a new system designated T-HMM are presented in [Seara Vieira et al., 2014]. T-HMM is a customizable model with a simple structure, based on the relevance of the words that appear in the text, which can be adjusted to future documents.

An HMM can be defined as a diagram composed of a group of hidden states S and transitions between them. Each state can give an observation V with a given probability, and the transitions between states also have an associated probability, as shown in Fig. 1. The HMM model is considered a double random process. The first process describes the sequence of states that cannot be observed (forming the hidden layer), represented by the probabilities of transition. The second process associates each state with an observation, depending on the probability of its occurrence, giving rise to a sequence of observations (visible part of the HMM), as shown in Fig. 2.

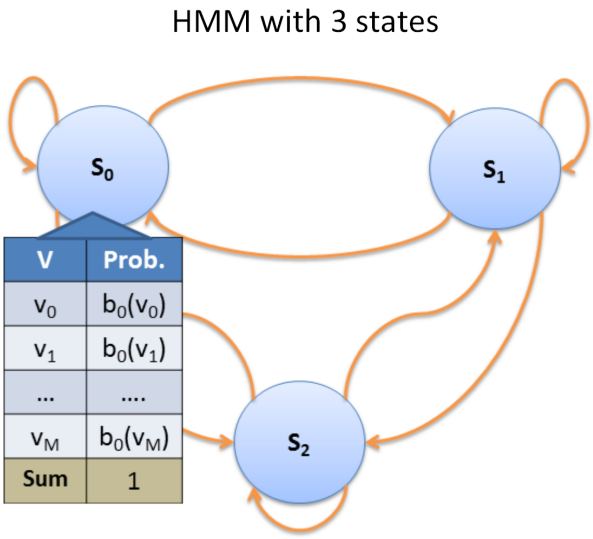

Figure 1: Example of an HMM with three states

Following the idea proposed by Kwan Yi *et al.* [Yi and Beheshti, 2009], T-HMM is a document generator. An HMM is implemented for each class: Relevant and Non-Relevant. Each model is then trained with documents belonging to the class that it represents. When a new document needs to be classified, the system evaluates the probability of this document being generated by each of the Hidden Markov Models. As a result, the class with the maximum probability value is selected and considered as the output class for the document.

In T-HMM, the observations are the feature words that represents the documents in the corpus (see Fig. 3 (Corpus with documents)). Based on the assumption that words do not have the same importance, hidden states in T-HMM reflect the difference in relevance (ranking) among words within a document. Each state represents a relevance level for words appearing in the corpus. That is, the most probable observations for the first state are the most relevant words in the corpus. The most probable ob-

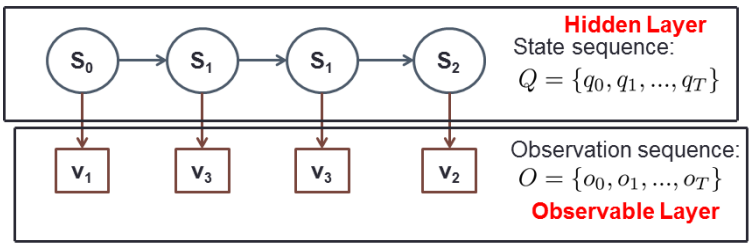

Figure 2: Sequence of states and observations in an HMM.

servations for the second state are the words holding the second level of relevance in the corpus, and so on. The number of states is a modifiable parameter that depends on the training corpus and how much flexibility the user wants to add to the model (see in detail in [Seara Vieira et al., 2014]).

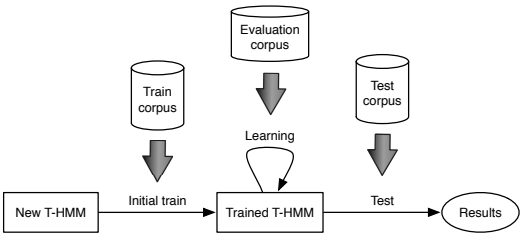

Figure 4: Evaluation process for adaptive T-HMM.

4. Adaptive T-HMM

In order to improve the ability of achieving good performance in text classification, we have added the learning capacity to the T-HMM in an innovate model named adaptive T-HMM. Adaptive T-HMM is able to learn from new documents, once trained. Thus, the model is updated with new data and the future classification of new documents will improve.

There are several studies on adaptive methods for text classification [Peng et al., 2008, Villmann et al., 2006, Wang et al., 2014a]. They show that the reasoning models are modified according to the acquired knowledge, changes in the environment or input data.

The Fig. 4 shows the process that the application of new classification model intends to solve.

Firstly, the adaptive T-HMM is trained with a training corpus (Train corpus). As result, a T-HMM classifier model with update capacity is created internally. The update capacity provides the model with the abilitiy to relearn and adapt the model to a new document corpus (Evaluation corpus). At the end of the process, the documents of the test corpus (Test corpus) are classified and the efficiency and precision of the model in the classification process are checked.

In the learning process the trained HMM model is adapted to a new document corpus (Learning corpus). The emission probability of the words in each HMM is adjusted according to the frequency of appearance of these words in the new documents. This adjustment is weighed by means of a parameter L. The bigger the parameter L is, the more important the new documents will be compared to the old ones. If L is too big it can cause overfitting on the new documents.

With a weight of 0.5, the value of the emission probabilities of the words would be equivalent to training the HMM initially with the union of the old documents and the new documents.

In addition, during the first step of the learning process, the model can perform an initial test on the evaluation corpus and select the incorrectly classified documents as the new documents to learn with. This is an option that is added in the learning step to learn only from the errors of classification and not from the entire evaluation corpus.

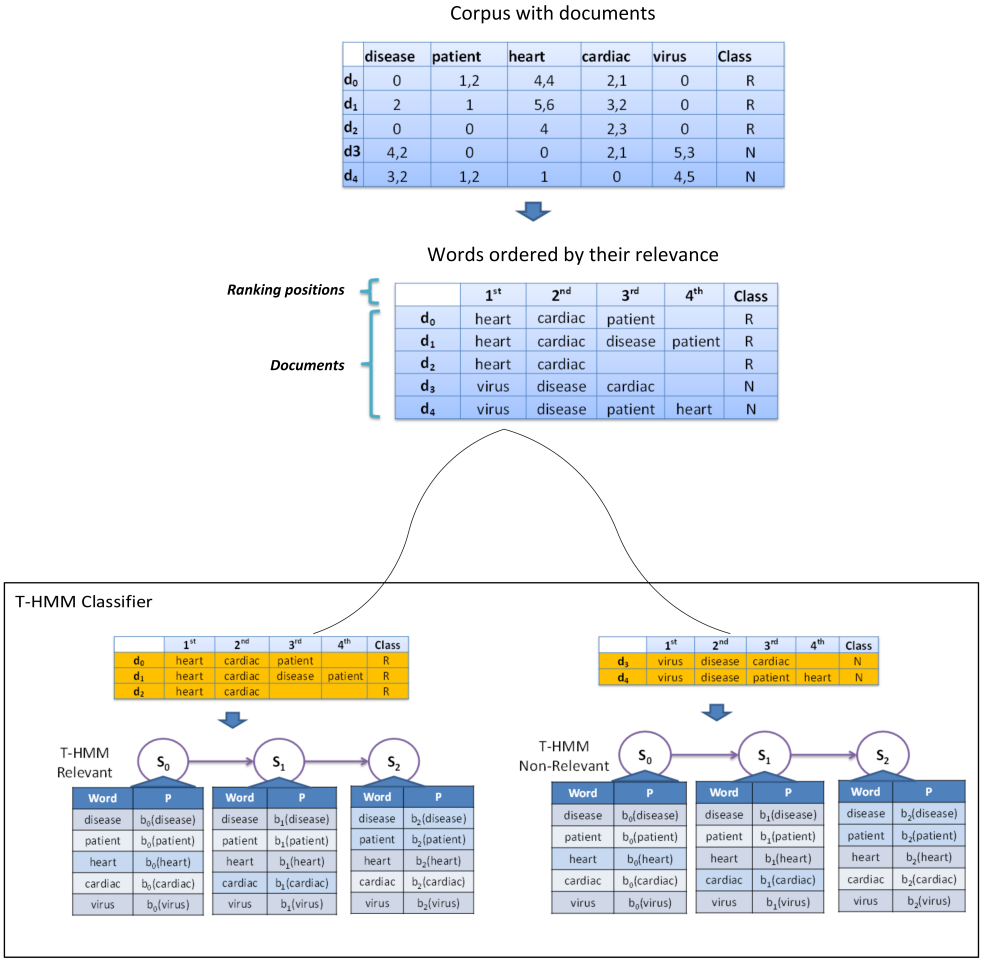

Figure 3: Building and training of the T-HMM.

5. Implementation of the text classification algorithms

5.1 Base framework: BioClass

T-HMM and adaptive T-HMM are integrated into the BioClass framework. BioClass [Romero et al., 2014] is a platform focused on the application of reasoning models in text classification issues. It is designed to work with the results obtained from an information retrieval process in a document database, as shown in Fig. 6. The preprocess step leads to a manageable representation of the documents. With these data as input, BioClass offers multiple filtering and machine learning algorithms to handle the automatic classification problem.

In Fig. 5 the architecture of BioClass is shown in detail. The main functionalities of this tool are:

1. Load sparse matrices: Firstly, the preprocessed datasets (sparse matrices) used for training and testing of classifiers are obtained. BioClass enables the addition of different data access connectors. It currently supports CSV and ARFF [Garner, 1995] file types.

2. Data filtering: With filtering algorithms it is

possible to carry out operations on datasets in order to reduce the noise and the dimensionality of the input data, significantly improving the classification processes. Filters are divided into two types: documents and attributes. The document filter makes it possible to handle the number of documents that belong to each class (relevant or non-relevant). In order to support these operations, *subsampling* and *oversampling* algorithms are implemented. The subsampling algorithm artificially decreases the number of samples that belong to the majority class. The oversampling algorithm redistributes the number of samples that belong to the minority class, taking the majority class into account [Anand et al., 2010, Zhang and Mani, 2003]. The attribute filter reduces the number of attributes of the corpus, trying to decrease the dimensionality without losing expressive power, using set operations (union, intersect, minus), chi-squared distribution, Information Gain or rule-based systems, among others.

3. Classification: BioClass supports different reasoning algorithm implementations and provides methods for parameterizing, training and testing. In addition to T-HMM and adaptive T-HMM, two Support Vector Machines (Cost-SVM and Nu-SVM) [Osuna et al., 1997], Naive Bayes [Domingos and Pazzani, 1997] and k-Nearest Neighbor classifiers [Dasarathy, 1991] are implemented.

4. Visualization and storage of results: The tool provides a graphical user interface that allows the user to select among various reasoning models or filters for data processing, and to edit or manage datasets and classifiers. Additionally, it provides a user-friendly interface to analyze the obtained results from a classification process.

BioClass proposes a workflow that guides the user through the entire process (see Fig. 6). The optional "Learn" step takes into account models like the adaptive T-HMM, which can learn from new sets of documents after they are trained.

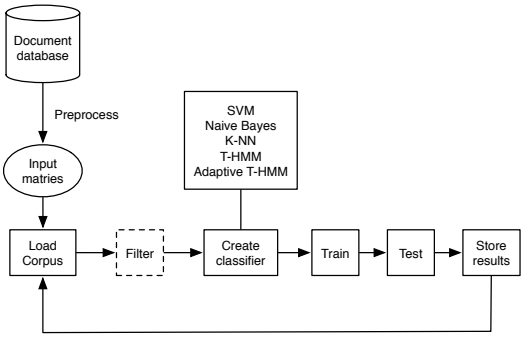

Figure 6: Workflow used by BioClass in a supervised classification process. The "Learn" step is applied only for the adaptive T-HMM.

5.2 Tools and technologies

To carry out the implementation of the T-HMM and the adaptive T-HMM algorithms the next technologies were used:

1. Java: The algorithms were developed with Java version 6 programming language. To facilitate the implementation process we use Eclipse, an Integrated Development Environment (IDE). The Swing tool was also used to develop the graphical interfaces.

2. AIBench [Glez-Peña et al., 2010]: AIBench is a lightweight, non-intrusive, Model View Controller-based Java application framework that eases the connection, execution and integration of operations with well-defined input/output. It is a powerful programming model to develop applications quickly because the logic can be decoupled from the user interface. AIBench allows the research effort to be centered on the algorithm development process, leaving the design and implementation

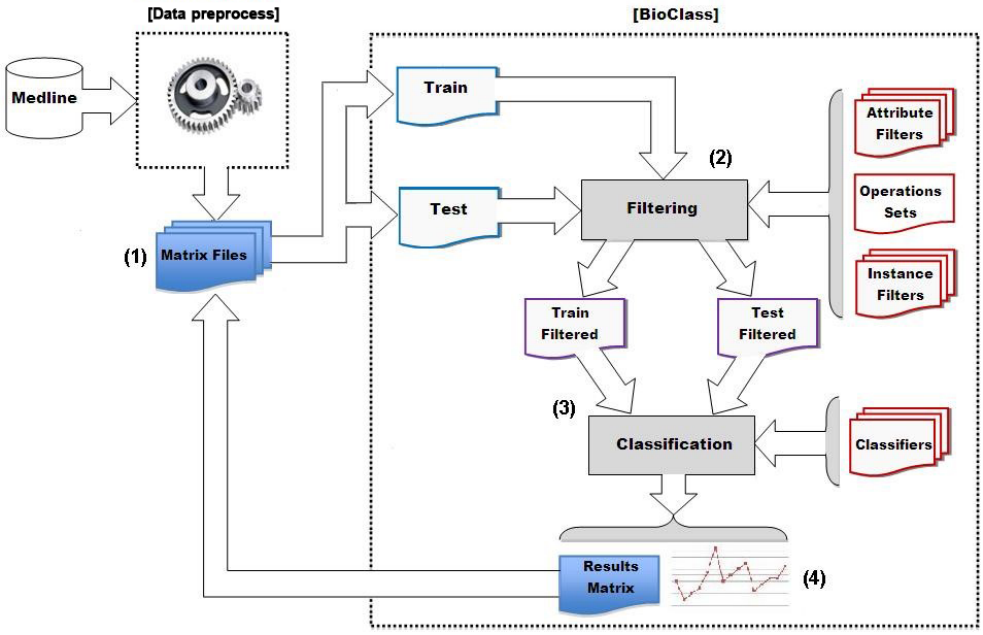

Figure 5: BioClass architecture.

of input-output data interfaces in the background. The different modules of BioClass are extensions of the AIBench platform.

3. JaHMM [Francois,]: JaHMM is a general purpose Java implementation of Hidden Markov Models related algorithms. The library is interesting in research because algorithms can be easily modified. The use of JaHMM was essential in our implementation , because it allowed to modify the model and implementation of the Viterbi, Forward-Backward and Baum-Welch algorithms [Rabiner, 1990], among others.

4. Weka [Garner, 1995]: To develop some functionalities related to document preprocessing, Weka (Waikate Environment for Knowledge) has been used. Weka is a collection of machine learning algorithms for data mining tasks that allows the discovery of patterns in large datasets and the extraction of information.

6. Results

In this section we demonstrate the usefulness of the proposed tool in a real-world scenario. Firstly, we define the document corpora used in the experiments and its preprocessing. Later, the use of the BioClass is shown in a complete classifying process with different reasoning models, including the adaptive T-HMM technique and its learning step.

6.1 OHSUMED Corpus

The Ohsumed test collection, initially compiled by Hersh *et al.* [Hersh et al., 1994], is a subset of the MEDLINE database, a bibliographic database of medical literature maintained by the National Library of Medicine. It contains $348,566$ references consisting of fields such as titles, abstracts, and MeSH descriptors from 279 medical journals published between 1987 and 1991. The set of medical abstracts for the year 1991 is taken as our initial corpus.

6.2 Data preprocessing

Each document of the initial corpus has one or more associated disease categories. In order to adapt them to our scheme, which consists of distinguishing relevant documents from non-relevant ones, we select one of these categories as relevant and consider the others as non-relevant. If a document has been assigned two or more categories and one of them is considered relevant, then the document itself is considered relevant and is excluded from the set of non-relevant documents.

The Neoplasms (C04) category and the Immunologic (C20) category are selected as the relevant categories in the OHSUMED corpus. One corpus is created per category and other documents that do not have the tagged relevant category are considered the non-relevant documents.

Once the documents are organized, it is necessary to format them into a vector of feature words in which elements describe the occurrence frequency of that word. All the different words that appear in the training corpus are candidates for input features.

In order to reduce the input feature size to train the classifier, the standard text preprocessing techniques are used. A predefined list of stopwords (common English words) is removed from the text, and a stemmer based on the Lovins stemmer [Lovins, 1968] is applied. Then, words occurring in less than 10 documents of the entire training corpus are also removed.

Finally, we end up with a matrix similar to Fig. 3 (Corpus with documents). Rows correspond to documents and columns to feature words. The value of an element in a matrix is determined by the number of occurrences of that feature word (column) in the document (row). This value is adjusted using the TF-IDF statistic (Term Frequency-Inverse Document Frequency, [Baeza-Yates and Ribeiro-Neto, 1999]) in order to measure the word relevance. The application of TF-IDF decreases the weight of terms that occur very frequently in the collection and increases the weight of terms that occur rarely.

In addition, in order to evaluate the reasoning models, each corpus is randomly divided into three different subsets:

- Training Corpus (50% of the original corpus), used to train all the evaluated classifiers.
- Evaluation Corpus (10% of the original corpus). The adaptive T-HMM classifier is adjusted with this subset to show its learning capability.
- Test Corpus (40% of the original corpus), used as the test corpus in all classifiers.

6.3 Execution

Once the document corpus is in a format that the BioClass framework can handle, the complete supervised classification process with the proposed tool is described as follows:

1. Load Corpus: The three different matrices (Train, Evaluation and Test) are loaded in the platform. The tool has a "Clipboard" section where reasoning models, results and corpus are organized as a labeled tree (see Fig. 7). The user can examine the data and employ it in multiple classifying operations.
2. Create Classifier: The platform offers multiple reasoning models to be created and parameterized. For evaluation purposes, we define an adaptive T-HMM with a number of states $n = 30$, and a factor $f = 0.25$, which are those that experimentally achieved the best results with the corpus. In addition, we create Naive Bayes, *k*-NN, SVM (Support Vector Machines) and basic T-HMM classifiers. The configuration for Naive Bayes, *k*-NN and SVM models are those utilized by default in the WEKA environment [Sierra Araujo, 2006]. Naive Bayes performs best without using any extra features, and when the number

of neighbors is 3 for *k*-NN. A RBF kernel is selected for SVM as it is used in related studies with the OHSUMED corpus [Joachims, 1998].

3. Train: All the models (T-HMM, adaptive T-HMM, Naive Bayes, k-NN and a SVM) are trained with the training corpus created in the preprocessing step.

4. Learn: In the case of the adaptive T-HMM, the "Learn" operation is applied. This takes the Evaluation corpus to perform the adjustment process defined in the Section 4 for each document in the set. The learning weight in this experiment is set to $L = 0.5$. In addition, all documents from the evaluation corpus are used to perform the learn process instead of using only misclassified documents.

5. Test: The trained models are tested with the Test corpus created in the preprocessing step, classifying all the documents in this set. The results are stored in the platform and can be displayed in the same way as shown in Fig. 7. Evaluation measures can also be viewed in this result visualization.

Table 1 and Table 2 collects all the results obtained for each classifier and corpus. To evaluate the effectiveness of the models, F-measure is used, which is the weighted harmonic mean of recall and precision: evaluation measures commonly utilized in text classification and information retrieval [Baeza-Yates and Ribeiro-Neto, 1999].

As can be seen, adaptive T-HMM achieves a logical improvement when the learning process is applied, and the model outperforms both the Naive Bayes and *k*-NN approaches in the selected evaluation measure. On the other hand, F-measure values in the Non-relevant category of SVM are superior to the other classifiers. However, the ability to retrieve relevant instances is more important than achieving a better average precision, since the relevant documents are in a much lower proportion. Thus, measures related to that category must be considered.

Table 1: Results achieved by each executed reasoning model in the automatic classification of the processed OHSUMED corpus with category C04.

Model	Non-relevant F-measure	Relevant F-measure
k-NN	0.759	0.449
Naive Bayes	0.878	0.704
SVM	0.929	0.744
T-HMM	0.920	0.781
Adaptive T-HMM	0.925	0.787

Table 2: Results achieved by each executed reasoning model in the automatic classification of the processed OHSUMED corpus with category C20.

Model	Non-relevant F-measure	Relevant F-measure
k-NN	0.910	0.219
Naive Bayes	0.875	0.497
SVM	0.954	0.565
T-HMM	0.920	0.607
Adaptive T-HMM	0.929	0.624

Finally, in Fig. 8 a comparison of the experiments is shown, using the interface of BioClass.

7. Conclusions

This study presents a novel adaptive text classification model based on Hidden Markov Models. It applies a reasoning system based on word relevance and distinguishes relevant documents from a set of unlabeled data. The model allows to be adjusted for future classifiable data.

Experimental results show that the application of the adaptive T-HMM appears to be promising. In automatic classification of OHSUMED corpus, adaptive T-HMM outperforms commonly used text classification techniques, like Naive Bayes, and achieves

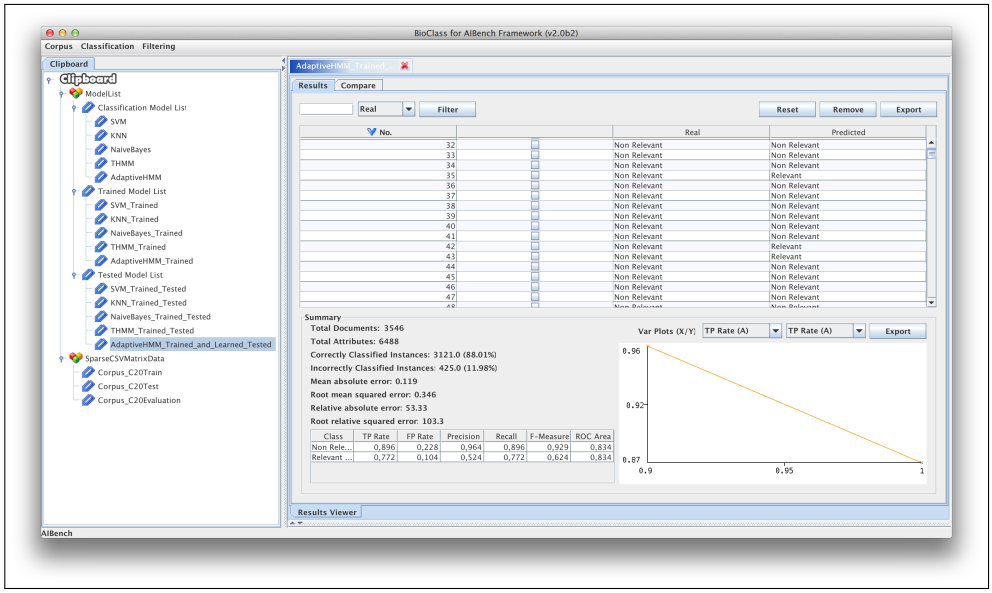

Figure 7: Visualization of results of the adaptive T-HMM in BioClass.

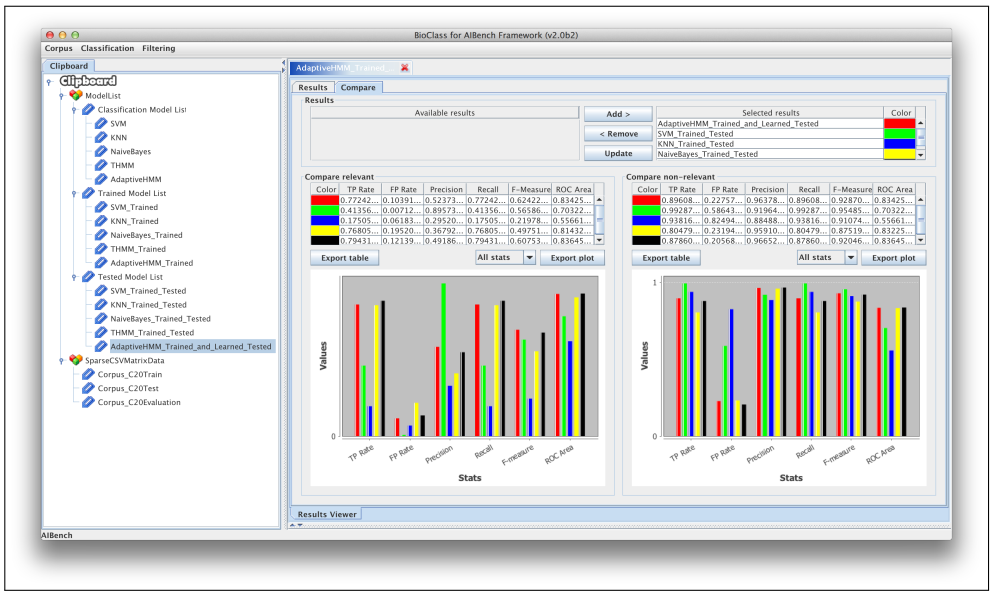

Figure 8: Comparison of results of the experiments in BioClass.

comparable results to those reached by SVM approach.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgements

This work has been funded from the following projects:

- European Union Seventh Framework Programme [FP7/REGPOT-2012-2013.1] under grant agreement 316265, BIOCAPS.

- Ministry of Economy and Competitiveness, under grant agreement TIN2013-47153-C3-3-R, Integration platform of intelligent techniques to analyze biomedical information.

- University of Vigo, under grant agreement 14VI05, TDAB Systems development support for decision making under biomedical scope

References

Anand, A., Pugalenthi, G., Fogel, G. B., and Suganthan, P. N., 2010. An approach for classification of highly imbalanced data using weighting and undersampling. *Amino acids*, 39:1385–1391.

Baeza-Yates, R. A. and Ribeiro-Neto, B., 1999. *Modern Information Retrieval*. Addison-Wesley Longman.

Dasarathy, B. V., 1991. *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. IEEE Computer Society Press, Los Alamitos, CA.

Domingos, P. and Pazzani, M., 1997. On the Optimality of the Simple Bayesian Classifier under Zero-One Loss. *Machine Learning*, 29:103–130.

Farid, D. M., Zhang, L., Rahman, C. M., Hossain, M., and Strachan, R., 2014. Hybrid decision tree and naÃŕve Bayes classifiers for multi-class classification tasks. *Expert Systems with Applications*, 41(4):1937–1946. doi:10.1016/j.eswa.2013.08.089.

Francois, J. *Jahmm - An implementation of HMM in Java*.

Garner, S. R., 1995. WEKA: The Waikato Environment for Knowledge Analysis. In *Proc. of the New Zealand Computer Science Research Students Conference*, pages 57–64.

Glez-Peña, D., Reboiro-Jato, M., Maia, P., Rocha, M., Díaz, F., and Fdez-Riverola, F., 2010. AIBench: A rapid application development framework for translational research in biomedicine. *Computer Methods and Programs in Biomedicine*, 98:191–203.

Gu, N., Wang, D., Fan, M., and Meng, D., 2014. A kernel-based sparsity preserving method for semi-supervised classification. *Neurocomputing*, 139:345–356. doi:10.1016/j.neucom.2014.02.022.

Harris, T., 2014. Credit scoring using the clustered support vector machine. *Expert Systems with Applications*, 42(2):741–750. Cited By 2.

Hersh, W. R., Buckley, C., Leone, T. J., and Hickam, D. H., 1994. OHSUMED: An Interactive Retrieval Evaluation and New Large Test Collection for Research. In *SIGIR*, pages 192–201.

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

Lovins, J. B., 1968. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11:22–31.

Maldonado, S., Weber, R., and Famili, F., 2014. Feature selection for high-dimensional class-imbalanced data sets using Support Vector Machines. *Information Sciences*, 286:228–246. doi:10.1016/j.ins.2014.07.015.

Osuna, E., Freund, R., and Girosi, F., 1997. Support Vector Machines: Training and Applications. Technical report.

Peng, T., Zuo, W., and He, F., 2008. SVM based adaptive learning method for text classification from positive and unlabeled documents. *Knowledge and Information Systems*, 16(3):281–301.

Rabiner, L. R., 1990. Readings in speech recognition. chapter A tutorial on hidden Markov models and selected applications in speech recognition, pages 267–296. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Romero, R., Vieira, A., Iglesias, E., and Borrajo, L., 2014. BioClass: A Tool for Biomedical Text Classification. In *8th International Conference on Practical Applications of Computational Biology & Bioinformatics*, volume 294 of *Advances in Intelligent Systems and Computing*, pages 243–251. Springer.

Seara Vieira, A., Iglesias, E. L., and Borrajo, L., 2014. T-HMM: A Novel Biomedical Text Classifier Based on Hidden Markov Models. In *8th International Conference on Practical Applications of Computational Biology & Bioinformatics*, volume 294 of *Advances in Intelligent Systems and Computing*, pages 225–234. Springer.

Sierra Araujo, B., 2006. *Aprendizaje automático: conceptos básicos y avanzados: aspectos prácticos utilizando el software Weka*. Pearson Prentice Hall.

Villmann, T., Schleif, F., and Hammer, B., 2006. Comparison of relevance learning vector quantization with other metric adaptive classification methods. *Neural Networks*, 19(5):610–622.

Wang, J., Belatreche, A., Maguire, L., and McGinnity, T., 2014a. An online supervised learning method for spiking neural networks with adaptive structure. *Neurocomputing*, 144(0):526–536.

Wang, S., Jiang, L., and Li, C., 2014b. Adapting naive Bayes tree for text classification. *Knowl Inf Syst*. doi:10.1007/s10115-014-0746-y.

Yi, K. and Beheshti, J., 2009. A hidden Markov model-based text classification of medical documents. *Journal of Information Science*, 35(1):67–81.

Zhang, J. and Mani, I., 2003. kNN Approach to Unbalanced Data Distributions: A case study involving Information Extraction. In *Proc. of the ICML'2003 workshop on learning from imbalanced datasets*.