Polylingual Text Classification in the Legal Domain

TERESA GONÇALVES, PAULO QUARESMA*

SOMMARIO: *1. Introduction – 2. Concepts and Tools – 2.1. Automatic Text Classification – 2.2. Support Vector Machines – 3. Polylingual Approach to Text Classification – 3.1. Combining Monolingual Classifier – 3.2. Using Polylingual Classifiers – 4. Experiments – 4.1. Dataset Description – 4.2. Experimental Setup – 4.3. Monolingual Experiments – 4.4. Monolingual Combiner Experiments – 4.5. Polylingual Experiments – 5. Conclusions and Future Work*

1. INTRODUCTION

Current Information Technologies and Web-based services need to manage, select and filter increasing amounts of textual information. Text classification allows users, through navigation on class hierarchies, to browse more easily the texts of their interests. This paradigm is very effective both in filtering information as in the development of online end-user services.

Since the number of documents involved in these applications is large, efficient and automatic approaches are necessary for classification. A Machine Learning approach can be used to automatically build the classifiers. The construction process can be seen as a problem of supervised learning: the algorithm receives a relatively small set of labelled documents and generates the classifier. Several algorithms have been applied, such as decision trees, linear discriminant analysis and logistic regression, the naïve Bayes algorithm and Support Vector Machines (SVM). Besides having a justified learning theory describing its mechanics, with respect to text classification SVM are known to be computationally efficient, robust and accurate.

Because of the globalization trend, an organization or individual often generates, acquires and archives the same document written in different languages (i.e., polylingual documents); moreover, many countries adopt multiple languages as their official languages. If these polylingual documents are organized into existing categories one would like to use this set of preclassified documents as training documents to build models to classify newly arrived polylingual documents.

* T. Gonçalves is Auxiliar Professor at the Department of Computer Science of the University of Évora; P. Quaresma is Associated Professor at the same Department.

For multilingual text classification, some prior studies address the challenge of cross-lingual text classification. However, prior research has not paid much attention to using polylingual documents yet. This study is motivated by the importance of providing polylingual text classification support to organizations and individuals in the increasingly globalized and multilingual environment.

We propose a method that combines different monolingual classifiers in order to get a new classifier as good as the best monolingual one which has the ability to deliver all the best performance measures (precision, recall and F1) possible.

This methodology was applied and evaluated on a set of legal documents from the EUR-Lex site. We collected documents for two anglo-saxon languages (English and German) and two roman ones (Italian and Portuguese), obtaining four different sets. The obtained results were quite good, indicating that combining different monolingual classifiers may be a promising approach to the problem of classifying documents written in several languages.

The paper is organized as follows: Section 2. describes the main concepts and tools used in our approach and Section 3. introduces several the methodology to get classifiers that use polylingual text documents. Section 4. presents the document collection used for evaluation, describes the experimental setup and evaluates the obtained results. Finally, Section 5. presents some conclusions and points out possible future work.

2. Concepts and Tools

This section introduces the Automatic Text Classification approach and the classification algorithm and software tool used in this work.

2.1. Automatic Text Classification

Originally, research in Automatic Text Classification addressed the binary problem, where a document is either relevant or not w.r.t. a given category. However, in real-world situations the great variety of different sources and hence categories usually poses a multi-class classification problem, where a document belongs to exactly one category from a predefined set. Even more general is the multi-label problem, where a document can be classified into more than one category.

In most multi-label text classification problems while some categories can

by "easily" learned, there are others that present low performance measures. This can be due to the fact that those categories are more difficult to learn in that specific language. Another problem concerns the precision/recall tradeoff: some classifiers present better precision in expense of worse recall, others exhibit the opposite behavior. If documents are available in different languages, we can use several sources of knowledge to try to overcome those problems.

In order to be fed to the learning algorithm, documents must by pre-processed to obtain a more structured representation. The most common approach is to use a bag-of-words representation[1] where each document is represented by the words it contains, with their order and punctuation being ignored. Normally, words are weighted by some measure of word's frequency in the document and, possibly, the corpus. In most cases, a subset of words (stop-words) is not considered, because their role is related to the structural organization of the sentences and does not have discriminating power over different classes and some works reduce semantically related terms to the same root applying a lemmatizer.

Fig. 1 shows the bag-of-words representation for the sentence "The provisions of the Agreement shall be applied to goods exported from South Africa to one of the new Member States".

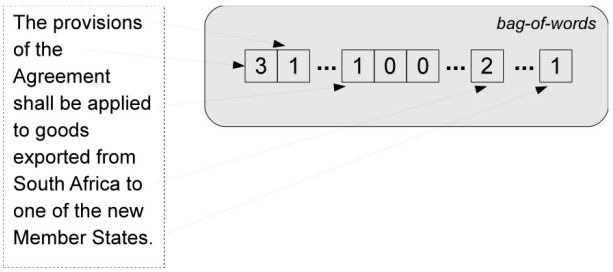

Fig. 1 – Bag-of-words representation

Research interest in this field has been growing in the last years. Several machine learning algorithms were applied, such as decision trees[2], linear dis-

[1] G. SALTON, A. WONG, C. YANG, *A Vector Space Model for Information Retrieval*, in "Communications of the ACM", 1975, Vol. 18, pp. 613-620.

[2] R. TONG, L.A. APPELBAUM, *Machine Learning for Knowledge-based Document Routing*, in "Proceedings of TRC'94, 2nd Text Retrieval Conference", 1994.

criminant analysis and logistic regression[3], the naïve Bayes algorithm[4] and Support Vector Machines (SVM)[5]. Joachims[6] says that using SVMs to learn text classifiers is the first approach that is computationally efficient and performs well and robustly in practice. There is also a justified learning theory that describes its mechanics with respect to text classification.

2.1.1. Multilingual Text Classification

While most text classification studies focus on monolingual documents, some point to multilingual text classification. From these, the great majority address the challenge of cross-lingual text classification where the classification model relies on monolingual training documents and a translation mechanism to classify documents written in another language[7]. A technique that takes into account all training documents of all languages when constructing a monolingual classifier for a specific language is proposed in Wei et al.[8].

[3] H. SCHÜTZE, D. HULL, J. PEDERSEN, *A Comparison of Classifiers and Document Representations for the Routing Problem*, in "Proceedings of SIGIR'95, 18th International Conference on Research and Development in Information Retrieval" (ACM), 2005, pp. 229-237.

[4] D. MLADENIĆ, M. GROBELNIK, *Feature Selection for Unbalanced Class Distribution and Naïve Bayes*, in "Proceedings of ICML'99, 16th International Conference on Machine Learning", 1999, pp. 258-267.

[5] T. JOACHIMS, *Transductive Inference for Text Classification Using SVM*, in "Proceedings of ICML'99", cit.

[6] T. JOACHIMS, *Learning to Classify Text Using Support Vector Machines*, Berlin, Heidelberg, Springer, 2002, 228 p.

[7] N. BEL, C. KOSTER, M. VILLEGAS, *Cross-lingual Text Categorization*, in "Proceedings of ECDL'03, 7th European Conference on Research and Advanced Technology for Digital Libraries", 2003, pp. 126-139; L. RIGUTINI, M. MAGGINI, B. LIU, *An EM Based Training Algorithm for Cross-Language Text Categorization*, in "Proceedings of WI'05, IEEE/WIC/ACM International Conference on Web Intelligence (IEEE Computer Society)", 2005, pp. 529-535; C.H. LEE, H.C. YANG, *Construction of Supervised and Unsupervised Learning Systems for Multilingual Text Categorization*, "Expert Systems Applications", Vol. 36, 2009, n. 2, pp. 2400-2410.

[8] C. WEI, H. SHI, C. YANG, *Feature Reinforcement Approach to Polylingual Text Categorization*, in "Proceedings of the International Conference on Asia Digital Libraries" (LNCS Springer), 2007, pp. 99-108.

2.2. Support Vector Machines

Support Vector Machines, a learning algorithm introduced by Vapnik and coworkers[9], was motivated by theoretical results from statistical learning theory: it joins a kernel technique with the structural risk minimization framework.

Kernel techniques comprise two parts: a module that performs a mapping from the original data space into a suitable feature space and a learning algorithm designed to discover linear patterns in the (new) feature space. The *kernel function*, that implicitly performs the mapping, depends on the specific data type and domain knowledge of the particular data source.

The *learning algorithm* is general purpose and robust. It's also efficient since the amount of computational resources required is polynomial with the size and number of data items, even when the dimension of the embedding space (the feature space) grows exponentially[10]. A mapping example is illustrated in Fig. 2.

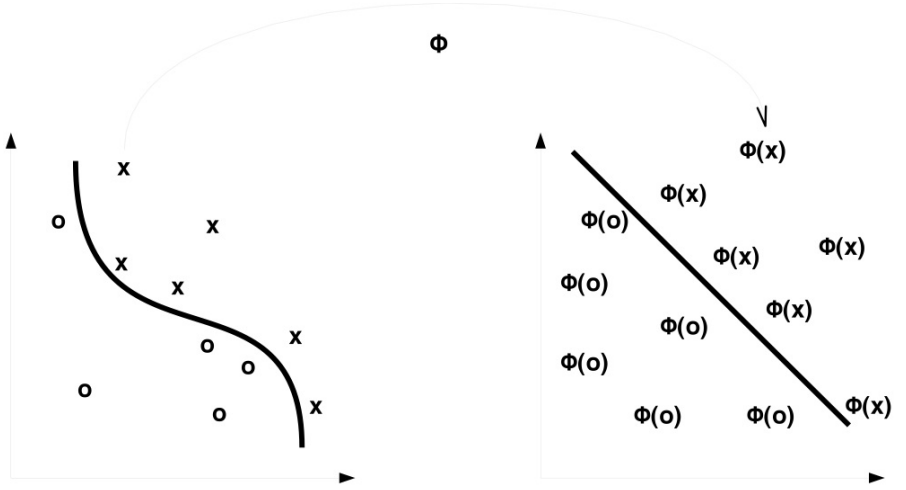

Fig. 2 – The SVM approach: kernel transformation

Four key aspects of the approach can be highlighted as follows:

[9] C. CORTES, V. VAPNIK, *Support-vector Networks*, in "Machine Learning", Vol. 20, 1995, n. 3, pp. 273-297.

[10] J. SHAWE-TAYLOR, N. CRISTIANINI, *Kernel Methods for Pattern Analysis*, Cambridge, Cambridge University Press, 2004, 476 p.

- data items are embedded into a vector space called the feature space;
- linear relations are discovered among the images of the data items in the feature space;
- the algorithm is implemented in a way that the coordinates of the embedded points are not needed; only their pairwise inner products;
- the pairwise inner products can be computed efficiently directly from the original data using the kernel function.

The *structural risk minimization* (SRM) framework creates a model with a minimized VC (Vapnik-Chervonenkis) dimension. This developed theory[11] shows that when the VC dimension of a model is low, the expected probability of error is low as well, which means good performance on unseen data (good generalization). In geometric terms, it can be seen as a search to find, between all decision surfaces (the $\mathcal{T}$-dimension surfaces that separate positive from negative examples) the one with maximum margin, that is, the one having a separating property that is invariant to the most wide translation of the surface. This property can be enlighten by Fig. 3 that shows a 2-dimensional problem. Boxed examples are called *support vectors* since they are the onlty ones that define the decision surface (all other examples are irrelevant to the decision surface definition).

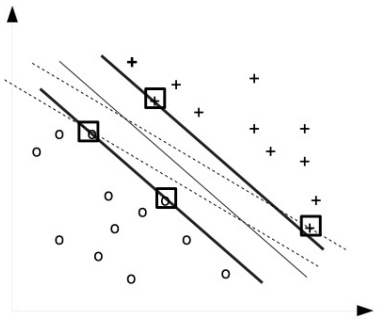

Fig. 3 – Maximum margin: the induction of support vector classifiers

SVM can also be derived in the framework of the regularization theory instead of the SRM one. The idea of regularization, introduced by Tikhonov and Arsenin[12] for solving inverse problems, is a technique to restrict the

[11] V. VAPNIK, *Statistical Learning Theory*, New York, John Wiley and Sons, 1998, 736 p.

[12] A.N. TIKHONOV, V.Y. ARSENIN, *Solution of Ill-Posed Problems*, Washington DC, John Wiley and Sons, 1977, 272 p.

(commonly) large original space of solutions into compact subsets.

2.2.1. Classification Software

As classification software we used SVM^{light}[13]. It is a C implementation of SVM that that allows solving classification, regression and ranking problems, handles many thousands of support vectors and several hundred-thousands of training examples and supports standard kernel functions besides letting the user define its own.

SVM^{light} can also train SVMs with cost models[14] and provides methods for assessing the generalization performance efficiently – precision, recall and XiAlpha-estimates for error rate[15]. This tool has been used on a large range of problems, including text classification[16], image recognition tasks, bioinformatics and medical applications. Many of these tasks have the property of sparse instance vectors and by using a sparse vector representation, it leads to a very compact and efficient representation.

3. Polylingual Approach to Text Classification

Having documents in several languages, one can adopt a naïve approach by considering the problem as multiple independent monolingual text classification problems. This naïve approach only employs the training documents of one language to construct a monolingual classifier for that language and ignores all training documents of other languages. When a new document in a specific language arrives, one select the corresponding classifier to predict appropriate category(s) for the target document. However, the independent construction of each monolingual classifier fails to use the opportunity offered by polylingual training documents to improve the effectiveness

[13] T. Joachims, *Making Large-scale SVM Learning Practical*, in Schölkopf B., Burges C.J.C., Smola A.J. (eds.), "Advances in Kernel Methods - Support Vector Learning", Cambridge, MA, MIT Press, 1999; also available at http://svmlight.joachims.org.

[14] P.B.K. Morik, T. Joachims, *Combining Statistical Learning with a Knowledge-based Approach – A Case Study in Intensive Care Monitoring*, in "Proceedings of ICML-99", cit.

[15] T. Joachims, *Estimating the Generalization Performance of a SVM Efficiently*, in "Proceedings of ICML-00", 17th International Conference on Machine Learning, MIT Press, 2000; T. Joachims, *Learning to Classify Text Using Support Vector Machines*, cit.

[16] T. Joachims, *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*, in "Proceedings of ECML'98, 10th European Conference on Machine Learning", 1998, pp. 137–142; T. Joachims, *Transductive Inference for Text Classification Using SVM*, cit.

of the classifier.

3.1. Combining Monolingual Classifier

One way of using polylingual training documents to obtain a classifier is to team up monolingual classifiers. We propose the following strategies for the combination system:

- the sum of SVMs output values;
- the F_1 weighted sum of SVMs output values;
- the F_1 weighted sum of SVMs decisions.

The above measures could also be used to draw decisions when considering a voting strategy of the monolingual classifiers.

3.2. Using Polylingual Classifiers

Another approach to use polylingual training documents is to get a polylingual classifier explicitly. We tried two different ways to obtain that classifier:

- a corpus that incorporates documents from all languages;
- a document representation that incorporates the information from all languages (just like a big document that incorporates the monolingual ones).

4. EXPERIMENTS

This section introduces the dataset, describes the experimental setup and presents the obtained results for the legal concepts classification task.

4.1. Dataset Description

For testing the proposed methodology, experiments were run over a set of European Union law documents. These documents were obtained from the EUR-Lex site[17] within the "International Agreements" section, belonging to the "External Relations" subject matter. From all available agreements we chose the ones with full text (not just bibliographic notice) obtaining a set of 2714 documents (dated from 1953 to 2008).

[17] See http://eur-lex.europa.eu/en/index.htm.

Since agreements are available in several languages we collected them for two anglo-saxon languages (English and German) and two roman ones (Italian and Portuguese), obtaining four different *corpora*: "eurlex-EN", "eurlex-DE", "eurlex-IT" and "eurlex-PT". Tab. 1 presents the total number and average per document of tokens (running words) and types (unique words).

	tokens		types	
corpus	total	per doc	total	per doc
`eurlex-EN`	10699234	3942	73091	570
`eurlex-DE`	10145702	3728	133191	688
`eurlex-IT`	10665455	3929	96029	636
`eurlex-PT`	9731861	3585	86086	567

Tab. 1 – Total number and average per document of tokens and types for each corpus

Each document is classified onto several ontologies: the "EUROVOC descriptor", the "Directory code" and the "Subject matter". In all available classifications each document can be assigned to several categories. For our classification problem we used the first level of the "Directory code" classification, considering only categories with at least 50 documents. Tab. 2 shows each category along with the number of documents assigned.

id	*name*	*# of docs*
2	Customs Union and free movement of goods	209
3	Agriculture	390
4	Fisheries	361
7	Transport policy	81
11	External relations	2628
12	Energy	58
13	Industrial policy and internal market	55
15	Environment, consumers and health protection	138
16	Science, information, education and culture	99

Tab. 2 – Number of documents assigned to each category

4.2. Experimental Setup

The experiments were done using a bag-of-words representation of documents, the SVM algorithm was run using SVM^{light} with a linear kernel and other default parameters and the model was evaluated using a 10-fold stratified cross-validation procedure with significance tests done with a 90% confidence level.

4.2.1. Document Representation

To represent each document we used the bag-of-words approach. Document's representation was obtained by mapping all numbers to the same token and using the *tf-idf* weighting function normalized to unit length. This well known measure weights word w_i in document d as:

$$tf\text{-}idf(w_i,d) = tf(w_i,d)\ln\frac{N}{df(w_i)}$$

where $tf(w_i,d)$ is the w_i word frequency in document d, $df(w_i)$ is the number of documents where word w_i appears and N is the number of documents in the collection.

4.2.2. Stratified Cross-validation

The cross-validation (CV) is a model evaluation method where the original dataset is divided into k subsets (in this work, $k = 10$), each one with (approximately) the same distribution of examples between categories as the original dataset (stratified CV). Then, one of the k subsets is used as the test set and the other $k-1$ subsets are put together to form a training set; a model is built from the training set and then applied to the test set. This procedure is repeated k times (one for each subset). Every data point gets to be in a test set exactly once, and gets to be in a training set $k-1$ times. The variance of the resulting estimate is reduced as k is increased.

4.2.3. Performance Measures

To measure learner's performance we analyzed precision, recall and the F_1 measures[18] of the positive class. These measures are obtained from con-

[18] G. SALTON, A. WONG, C. YANG, *A Vector Space Model for Information Retrieval*, cit., pp. 613-620.

tingency table of the classification (prediction *vs.* manual classification). *Precision* is the number of correctly classified documents (true positives) divided by the number of documents classified into the class (true positives plus false positives).

Recall is given by the number of correctly classified documents (true positive) divided by the number of documents belonging to the class (true positives plus false negatives).

F_1 is the weighted harmonic mean of precision and recall and belongs to a class of functions used in information retrieval, the $F_\beta\text{-}measure$. F_β can be written as follows:

$$F_\beta(h) = \frac{(1+\beta^2)prec(h)rec(h)}{\beta^2 prec(h) + rec(h)}$$

For each performance measure we calculated the micro- and macro-averaging values of the top ten categories. *Macro-averaging* corresponds to the standard way of computing an average: the performance is computed separately for each category and the average is the arithmetic mean over the ten categories.

Micro-averaging does not average the resulting performance measure, but instead averages the contingency tables of the various categories. For each cell of the table, the arithmetic mean is computed and the performance is computed from this averaged contingency table.

4.3. Monolingual Experiments

To support our claim, as baseline we built classifiers for each language. Tab. 3 shows the average precision, recall and F_1 measures for each corpus and each category (boldface values are significantly worse than the best value obtained). Last line presents the average values over all nine classes.

For the precision values we can notice that the Portuguese dataset has values with no significant difference with the "best" for all classes; all other languages perform worse for some classes (English: $c2$, $c4$ and $c16$; German: $c12$ and $c16$; Italian: $c2$, $c7$ and $c12$). With this in mind one can say that the Portuguese language generates the best precision classifiers.

Concerning recall, it's the English and German languages that consistently present the best values; Italian and Portuguese while equally good for some classes, are worse for others (Italian: $c2$ and $c3$; Portuguese: $c2$, $c3$ and $c4$).

	precision				*recall*				F_1			
id	EN	DE	IT	PT	EN	DE	IT	PT	EN	DE	IT	PT
2	**.919**	.957	**.922**	.929	.651	.665	**.580**	**.565**	.758	.783	**.702**	**.701**
3	.916	.928	.938	.942	.818	.805	**.705**	**.503**	.863	.861	**.803**	**.655**
4	**.956**	.966	.980	.971	.934	.906	.914	**.823**	.944	.934	.945	.890
7	.846	.870	**.793**	.813	.568	.543	.518	.482	.672	.662	.608	.590
11	.973	.973	.973	.973	.998	.997	.998	.997	.985	.985	.985	.985
12	.958	**.874**	**.877**	.921	.638	.707	.670	.600	.763	.781	.745	.716
13	.942	.933	.933	.944	.382	.309	.300	.320	.538	.459	.436	.461
15	.909	.922	.917	.902	.725	.732	.725	.732	.803	.815	.805	.806
16	**.862**	**.883**	.916	.941	.778	.798	.718	.647	.806	.832	.785	**.753**
avg	.828	.832	.825	.926	.721	.718	.613	.567	.792	.790	.681	.656

Tab. 3 – Average precision, recall and F_1 values for each monolingual classifier

The F_1 measure presents the same behavior as recall, being the only difference the classes where the Portuguese language performs worse ($c2$, $c3$ and $c16$).

4.4. Monolingual Combiner Experiments

From all possible combiners (see Section 3.1.), there is one that, for all classes, persistently generated the best F_1 values: the F_1 weighted sum of SVMs decisions.

Tab. 4 shows, for each performance measure its results compared with the "best" monolingual classifiers (boldface values are significantly worse than the corresponding combiner ones): the Portuguese one for precision, and the English and German one for recall and F_1. Last line equally presents the average values over all classes.

From the average values, one can easily see that precision is higher than recall and that the best monolingual classifier depends on what performance measure one is considering. Nevertheless, the combined classifier has all performance measures very similar to the best monolingual classifier.

Significant tests show that, for all classes and all performance measures, there is no significant difference between the "best" monolingual classifier and the corresponding combined classifier. Moreover, in most cases the confidence interval is broader than the corresponding combined classifier.

	precision		*recall*			F_1		
id	PT	comb	EN	DE	comb	EN	DE	comb
2	.937	.940	.651	.665	.675	.758	.783	.786
3	.943	.924	.818	.805	.813	.863	.861	.865
4	.971	.963	.934	.906	.928	.944	.934	.945
7	.806	.863	.568	.543	.568	.672	.662	.676
11	.973	.973	.998	.997	.998	.985	.985	.985
12	.938	.929	.638	.707	.672	.763	.781	.780
13	.967	.900	.382	.309	.327	.538	.459	.480
15	.908	.904	.725	.732	.754	.803	.815	.822
16	.947	.865	.778	.798	.778	.806	.832	.819
avg	.926	.915	.721	.718	.724	.792	.790	.795

Tab. 4 – Precision, recall and F_1 values of best monolingual classifiers compared with combiner ones

4.5. Polylingual Experiments

The results obtained for both approaches of incorporating polylingual information during classifier generation (see Section 3.2.) showed an average increase of around 2% for F_1 when using a document representation that incorporates information from all languages. This approach obtained better F_1 values for classes $c2$, $c3$, $c7$, $c15$ and $c16$, with no significant difference for classes $c4$, $c11$ and $c12$; only class $c13$ got worse results.

Comparing this best setting with the best one from previous experiments (that combines monolingual classifiers using a F_1 weighted sum of SVMs decisions), one can see that there is no significant difference for all measures and classes. Tab. 5 shows these values.

5. Conclusions and Future Work

A proposal to make polylingual text classification was presented and evaluated using two different approaches: combining monolingual classifiers and generating polylingual ones. The methodology uses SVM classifiers to associate concepts to legal documents and for the first approach uses a decision function that combines monolingual classifiers in order to obtain, for each class, a classifier as good as the best monolingual classifier of each performance measure; for the second approach generates a document description that incorporates knowledge from several languages.

The baseline experiments allows one to conclude that some languages

	precision		*recall*		F_1	
id	comb	poly	comb	poly	comb	poly
1	.940	.945	.675	.660	.786	.777
2	.924	.926	.813	.803	.865	.860
3	.963	.962	.928	.922	.945	.942
4	.836	.828	.568	.593	.676	.691
5	.973	.973	.998	.998	.985	.985
6	.929	.902	.672	.638	.780	.747
7	.900	.895	.327	.309	.480	.459
8	.904	.915	.754	.775	.822	.839
9	.865	.876	.778	.788	.819	.830
avg	.915	.914	.724	.721	.795	.792

Tab. 5 – Precision, recall and F_1 values of polylingual and combiner best experiments

generate classifiers with better precision values (Portuguese language) while others generate classifiers with better recall ones (English and German languages).

By combining all monolingual classifiers one obtains a classifier as good as the best monolingual one. This combined classifier can even be considered better than the others since it has the ability to deliver all the best performance measures (precision, recall and F_1) unlike using one monolingual classifier.

Considering the use of a polylingual classifier with a richer document description that incorporates knowledge from several languages, it is possible to say that it does not bring any further improvements when compared to the combination of monolingual classifiers.

Regarding future work, we intend to extend this work to other collections and domains. It will be important to evaluate if these results are bound to this collection or, as expected, are true for other collections and domains.