

Multi-classification of Patent Applications with Winnow

Cornelis H.A. Koster¹, Marc Seutter¹ and Jean Beney²

¹ University of Nijmegen, The Netherlands.
{kees,marcs}@cs.kun.nl

² Dept Informatique, INSA de Lyon, France.
jbeney@lisi.insa-lyon.fr

Abstract. The Winnow family of learning algorithms can cope well with large numbers of features and is tolerant to variations in document length, which makes it suitable for classifying large collections of large documents, like patent applications.

Both the large size of the documents and the large number of available training documents for each class make this classification task qualitatively different from the classification of short documents (newspaper articles or medical abstracts) with few training examples, as exemplified by the TREC evaluations.

This note describes recent experiments with Winnow on two large corpora of patent applications, supplied by the European Patent Office (EPO). It is found that the multi-classification of patent applications is much less accurate than the mono-classification of similar documents. We describe a potential pitfall in multi-classification and show ways to improve the accuracy. We argue that the inherently larger noisiness of multi-class labeling is the reason that multi-classification is harder than mono-classification.

1 Introduction

The automatic classification of patent applications is a challenging application area for automatic Text Categorization techniques (for an overview see [13]), due to the large number of features (many thousands of words per document) and the large number of labeled train- and test-documents available. Furthermore, the quality of the labeling is high: each document has been classified carefully by experts highly familiar with the domain at hand.

Traditional classification tasks, as exemplified by the Reuters dataset and most of the collections used in TREC, have been concerned with smaller documents (newspaper stories, medical abstracts) and classes for which few labeled examples are available (between one and a few hundreds). Because of these qualitative and quantitative differences, different problems may arise when classifying patent applications.

1.1 The EPO tests

A few years ago, the European Patent Office (EPO) in Rijswijk, the Netherlands, organised an open competition in order to assess the state-of-the-art in supervised learning for document classification (see [8]).

The first EPO test concerned mono-classification of a corpus of 16000 patent applications (EPO1) into 16 disjoint classes (*clusters* of directorates) with a thousand training examples each.

The second test involved a more realistic problem: a multi-classification with many classes, training on a larger number of documents and taking as test documents a large subset of a year's incoming documents.

The University of Nijmegen (KUN) participated in both EPO tests. We demonstrated the suitability of the Balanced Winnow algorithm for the task of classifying patent applications, comparing its performance to that of Rocchio in a number of experiments: a mono-classification of 16000 documents in 16 classes and a multi-classifications of 68418 documents in 44 classes. The largest corpus contained more than 500 000 different terms.

As we have reported in [6], the results of the mono-classification were surprisingly good, but the accuracy in multi-classification on quite similar data was found to be disappointing. In this note, we investigate the reasons for this disparity. Although we find ways to improve the accuracy of multi-classification (see section 3) we must conclude (in section 4) that there are inherent reasons that make it less accurate than mono-classification (section 2).

1.2 The classification system

The Linguistic Classification System (LCS) developed by the University of Nijmegen (KUN) in the course of the DORO and PEKING Esprit Projects³ is a generaliser system for the classification and routing of full-text documents.

The LCS system automatically learns a classification from a corpus of documents labeled with classes. After this training phase it can either be used for testing the classifier obtained on a test collection of documents with a known classification (usually held-out training data), or for producing a classification of new documents without known classification.

The LCS can perform both *mono-classification* (each document belongs to precisely one class) and *multi-classification* (each document belongs to zero or more classes).

The Winnow algorithm, implemented in the LCS, is a heuristical learning algorithm with nice mathematical convergence properties [5]. It iterates repeatedly over the training documents, computing for each class a vector of weights approximating an optimal linear separator between relevant and non-relevant documents [10]. According to [4], the balanced version of the Winnow algorithm can cope with large numbers of features and can tolerate large variations in document length because it uses both positive and negative weights.

³ <http://www.cs.kun.nl/peking>

2 EPO1: Mono-classification

The EPO1 corpus of patent material in English consists of 1000 abstracts and 1000 full-text documents for each of 16 classes. Some statistics are given in the following table:

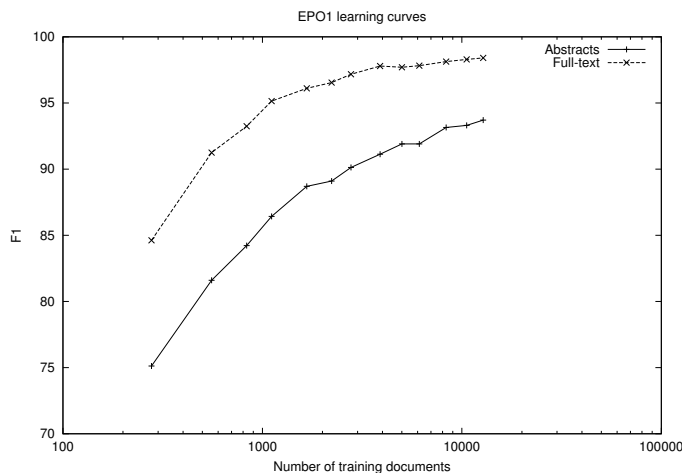
	documents	lines	words	characters
full-text	16000	7.6M	73.3M	461.1M
abstracts	16000	0.2M	2.0M	12.3M

The material is very homogeneous in document size, and evenly distributed over the classes. The abstracts, with an average length of 129 words, are in descriptive prose without tables or formulae. The full text documents are longer (avg 4580 words) and contain some non-linguistic material like tables and formulae.

The only pre-processing applied to the documents was the replacement of capital letters by the corresponding lowercase letters (decapitalization) and the replacement of special characters by spaces. In particular, no term stemming was performed and no stoplist was applied. We relied on the term selection to eliminate redundant features.

From the labeled full-text documents (the original trainset) a new train and test set was obtained by randomizing the order and splitting the set (80% training, 20% held out for testing). This training set was trained in epochs of increasing size and the resulting classifier tested on the remaining 20%.

The *learning curves* (micro-averaged precision⁴ against number of documents trained) for abstracts and full-text documents are as follows:



⁴ The *macro-averaged* precision is the average over all classes of the precision per class, whereas the *micro-averaged* precision is computed by lumping the classes together. In mono-classification micro-averaged precision equals recall (any document which is incorrectly assigned to a class is missing in another class).

The result is encouraging, with precision on full-text documents exceeding 98 percent at 800 train examples per class, in spite of the large size of the documents. The classes appear to be sufficiently disjoint, and the noise on the labeling appears to be small. At very low numbers of training examples (280 examples, an average of 17.5 examples per class) the precision is already 85 percent. Although learning slows down after about 3000 documents trained, there is still a gradual improvement.

The precision on abstracts is definitely lower. This contradicts (at least for patent applications) the expectation that abstracts should be easier to classify than the corresponding documents because they contain fewer digressions and the words used are more carefully chosen, thus more pertinent. On the contrary, in a patent abstract the lawyers go out of their way to use inclusive, general language. In that light, 93% precision is not bad.

3 EPO2: Multi-classification

The EPO2 corpus contains 68418 labeled documents in English, both in the full-text form and in the form of abstracts. Some statistics:

	documents	lines	words	characters
full-text	68418	44.7M	401.3M	2607.3M
abstracts	68418	1.0M	9.0M	59.5M

For these documents, three different labelings were provided by EPO, of which we consider only one in this paper: a multi-class labeling with 44 directorates. A *directorate* is an organizational unit within EPO, responsible for all patent applications in a certain domain, but many patents belong to more than one domain.

The pre-processing applied to the documents was the same as in the EPO1-test, but this time all terms with a term frequency < 4 were eliminated (TF selection).

The EPO2 test was harder than the previous one, due to the large size of the test material but in particular due to the fact that this test concerned *multi-classification* rather than mono-classification.

3.1 Multi-classification issues

Conceptually, multi-classification (where each document belongs to zero or more classes) appears to be simpler than mono-classification (where each document belongs to precisely one class), since it can be seen as training an independent binary classifier for each class. In mono-classification, each document is assigned to the class for which it achieves the highest score, which requires the scores of different classes to be comparable.

In multi-classification, each class must have its own *threshold* to determine whether a document with a certain score belongs to that class. In fact the training procedure for multi-classification is the same as that for mono-classification

(apart from the fact that each document now may belong to more than one class), but the *selection* of classes for documents is more complicated: rather than assigning each document to the class for which it has the highest score, in multi-classification it is necessary to determine for each class an optimal threshold [1, 14] to distinguish between relevant and non-relevant documents for that class.

Another difference concerns the measurement of accuracy: In multi-classification, precision and recall are no longer equal, and we shall use the *micro-averaged F1 value* as a measure of accuracy. The F1-value is a kind of harmonic average of precision and recall.

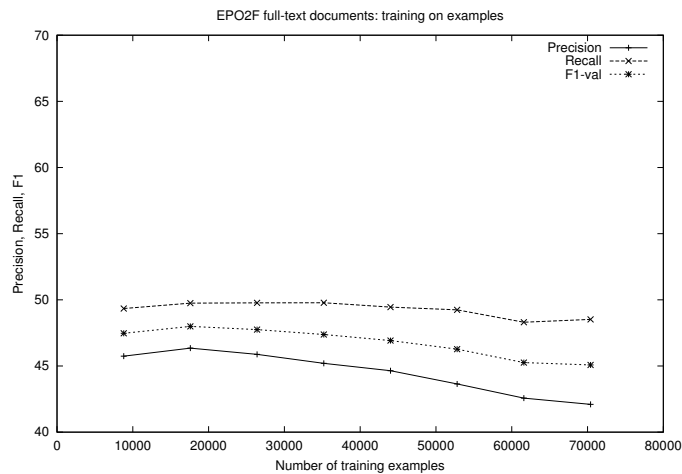
In the “directorate” classification, an average of 1.43 classes is assigned to each of the 68418 training documents. Therefore each document is a (positive) example for (on the average) 1.43 classes. Calling each <document, class> combination an *example*, the total number of examples is 88000, distributed in such a way that each of the 44 directorates has 2000 examples.

The distribution of the number of classes per document is as follows:

number of classes	1	2	3	4	5	6	7	> 7
number of docs	51981	13745	2323	306	46	13	4	0

3.2 Training on Examples

The 88000 examples were randomly split 80/20 in a train and test set, and classifiers were trained using the Winnow algorithm in eight epochs. The following learning curve (precision, recall and F1-value as a function of the number of documents trained) resulted:



The accuracy reached is low, below 50%. It can be seen that, as more documents are trained, precision, recall and F1-value at first increase, but after some 20000 examples all three deteriorate again. We must be doing something wrong.

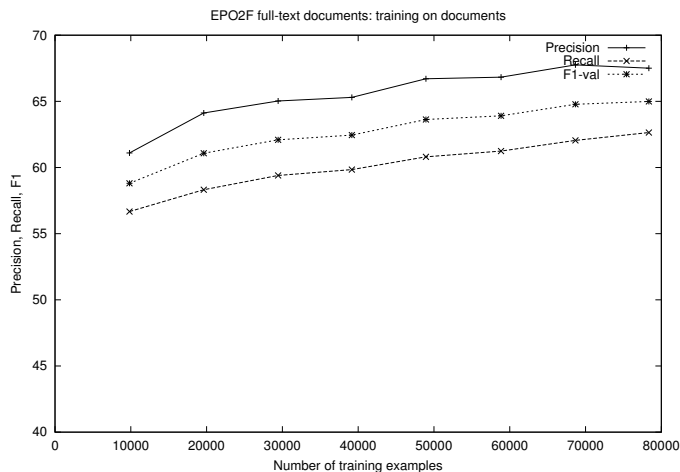
3.3 Training on Documents

When training on examples, we do not in any way take into account that some examples pertain to the same document. We are training independent binary classifiers. In training for a certain class, all examples involving that class are seen as *positive* examples, whereas all examples belonging to other classes are seen as *negative* examples for that class. As a consequence, any document belonging to more than one class will be trained both as a positive example and as a negative example for each class to which it belongs!

This will be detrimental for any classification algorithm based on the Vector Space Model (computing the score of a document for some class as an inproduct between class term weights and document term strengths).

In order to prevent a document from being a counterexample to itself, we should be *training on documents* rather than examples, seeing the document as a positive example for each class to which it belongs, and as a negative example for all classes to which it does not belong.

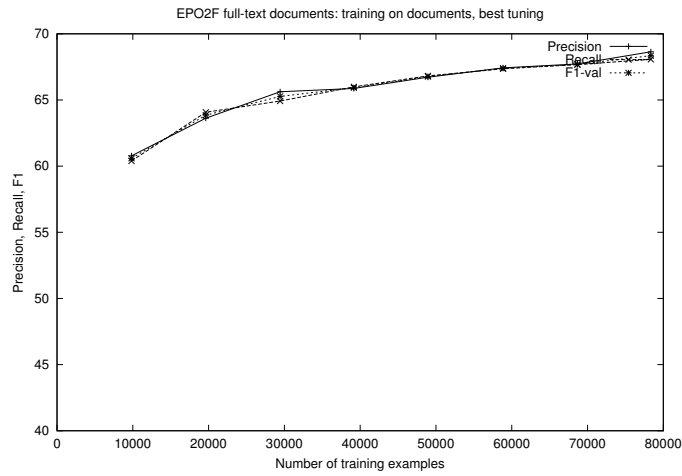
Modifying the train procedure accordingly, and shuffling the documents rather than the examples, we obtain a much better learning curve:



3.4 Training with optimal tuning

In the preceding two graphs, as in our original experiment, precision and recall are far apart, with the measure of accuracy (F1) precariously in the middle. In principle it might be caused by a bad choice of thresholds, but the LCS computes on the train set for each class that threshold value which maximizes F1 (not necessarily the point where precision equal recall). The Winnow algorithm has a number of parameters, and the weak point is that we took their values from literature.

After tuning the Winnow parameters on the train set, following the same procedure as in [7], we obtained a much better learning curve:



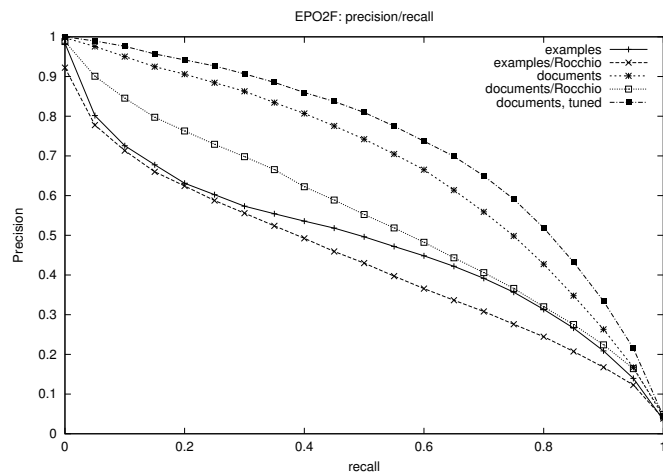
Precision and recall now practically coincide, and when training on 80% of the documents an accuracy of .68 is reached.

4 Further experiments

It is obvious that training on documents results in a better accuracy than training on examples, and tuning can improve the accuracy further, but we may ask ourselves whether this really corresponds to a better ranking of the documents.

4.1 Ranking

In order to study the quality of the ranking, we have made the following Precision/Recall graph (using the treceval utility of TREC) for Winnow trained on examples, on documents and tuned, adding also the curves for the Rocchio algorithm (on examples and documents) for comparison:



The graphs show clearly that Winnow outperforms Rocchio all the way (as was found in [6]), but also that training on documents rather than examples makes a marked improvement for both algorithms. It is evidently not a good idea to consider multi-class training as the training of independent binary classifiers.

And better tuning of Winnow makes another important improvement at all recall levels. But compared to section 2, the graphs still show that the accuracy achieved in the mono-classification of the EPO1 data is much larger than for the multi-classification of the EPO2 data. Is there an inherent difference between these forms of classification that causes the difference in accuracy?

4.2 Noisy labeling

We contend that the difference lies mainly in the *noise* in the labeling of the training data. Among the sources of labeling noise mentioned in [11], many are the same for both forms of classification, but multi-classification is particularly prone to *quantization error* (caused by the fact that labels have to be either present or absent, no gradual judgements are allowed).

In the case of the EPO1 corpus, the labeling noise is particularly small because it contains only documents which were unequivocally mono-classified by the experts. A labeling with a “main category” and (zero or more) secondary categories will be much more noisy.

If some percentage of the labels in the trainset have been doled out incorrectly or inconsistently, *no* classification algorithm can be expected to make fewer errors than that percentage in any experiment with held-out data.

In order to obtain an indication of the labeling noise in the EPO1 corpus we therefore performed an experiment on “seen” corpus (training and testing on the same random subset of 50% of the documents), with the following result:

	precision	recall	F1-value
Winnow abstracts	99.61	99.61	99.61
Winnow full-text	100.00	100.00	100.00

Winnow can “explain” the class given to most abstracts and all full-text documents (in the sense that it can find term weights such that they bring the document score to the right side of the threshold). Of course this can also be interpreted as evidence of overtraining on the seen documents. But at least, it is possible to train a classifier which has 100% accuracy on the EPO1 documents.

The boundary condition “each document belongs to precisely one class” provides important information, as can be seen by weakening it to “each document belongs to at most one class”. The results for this 0:1 multi-classification of seen EPO1 documents are shown in the next table.

	precision	recall	F1-value
Winnow abstracts	99.87	98.83	99.35
Winnow full-text	100.00	99.80	99.90

The precision is no longer equal to the recall and some .1% of the documents obstinately refuse to be classified in the right category. This is caused by the fact that documents are no longer forced into a class even when their score falls under the threshold of that class. In experiments with a totally different corpus [2], we found that 0:3 multi-classification of unseen mono-classified documents gave a few percent less accuracy than mono-classification.

Training on “seen” EPO2 corpus (0:3 multi-classification, training and testing on the same 80% of the corpus), we obtained the following results:

	precision	recall	F1-value
Winnow abstracts	73.03	72.21	72.62
Winnow full-text	80.12	79.16	79.63

These figures can be taken as an upper limit on the accuracy achievable in multi-classification of EPO2, and are far below those for mono-classification. In this sense, multi-classification is harder than mono-classification.

5 Conclusions and further work

The experiments described above confirm that the Winnow algorithm is suitable for the classification of large collections of large documents, such as patent applications, and for this purpose is preferable to Rocchio.

The accuracy in the mono-classification of full-text documents was much better than that of abstracts, contradicting the expectation that abstracts would be easier to classify, because of the more expressive terminology used and the lack of digressions.

We found a potential pitfall in multi-classification (example-based instead of document-based training. Although we managed to diminish the difference in precision and recall between mono-classification and multi-classification by tuning the Winnow parameters, the former remained much more accurate. We argued that this is caused by noise in the labeling of train documents, which is much larger for multi- than for mono-classification, due to the arbitrary labeling of border cases.

The documents were represented quite conventionally, as a bag of words, without any linguistic pre-processing. In particular, no semantical ontology was used and, in contrast to [9], the internal structure of the documents was completely ignored. In spite of this simple document representation, the accuracy of mono-classification is very good.

Our later experiments with other corpora and more informative document representations did not substantiate the expectation that the use of phrases [7], lemmatization and collocations [2] would make an important improvement to the accuracy of automatic classification.

References

1. A. Arampatzis and A. van Hameren (2001), The Score-Distributional Threshold Optimization for Adaptive Binary Classification Tasks. Proceedings ACM SIGIR 2001, pp. 267-275.
2. Nuria Bel, Cornelis H.A. Koster and Marta Villegas (2003), Cross-Lingual Text Categorization. Proceedings ECDL 2003, Springer LNCS 2769, pp 126-139.
3. W.W. Cohen and Y. Singer (1999), Context-sensitive learning methods for text categorization. *ACM Transactions on Information Systems* 13, 1, 100-111.
4. I. Dagan, Y. Karov, D. Roth (1997), Mistake-Driven Learning in Text Categorization. Proceedings 2nd Conference on Empirical Methods in NLP, pp. 55-63.
5. A. Grove, N. Littlestone, and D. Schuurmans (2001), General convergence results for linear discriminant updates. *Machine Learning* 43(3), pp. 173-210.
6. C.H.A. Koster, M. Seutter and J. Beney (2001), Classifying Patent Applications with Winnow, Proceedings Benelearn 2001, Antwerpen, 8pp. <http://cnts.uia.ac.be/benelearn2001/>
7. C.H.A. Koster and M. Seutter (2002), Taming Wild Phrases. Proceedings 25th European Conference on IR Research (ECIR 2003), Springer LNCS 2633.
8. M. Krier and F. Zaccà (2002), Automatic Categorisation Applications at the European Patent Office. *World Patent Information* 24, pp. 187-196.
9. L. S. Larkey (1999), A patent search and classification system. Proceedings of DL-99, 4th ACM Conference on Digital Libraries, pp 179-187.
10. N. Littlestone (1988), Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2, pp. 285-318.
11. C. Peters and C.H.A. Koster (2003), Uncertainty-based Noise Reduction and Term Selection in Text Categorisation, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems (IJUFKS) Vol. 11, No. 1, pp 115-137.
12. J.J. Rocchio (1971), Relevance feedback in Information Retrieval, In: Salton, G. (ed.), *The Smart Retrieval system - experiments in automatic document processing*, Prentice - Hall, Englewood Cliffs, NJ, pp 313-323.
13. F. Sebastiani (2001), Machine learning in automated text categorization. *ACM Computing Surveys*, Vol 34 no 1, 2002, pp. 1-47.
14. Y. Zhiang and J. Callan (2001), Maximum Likelihood Estimation for Filtering Thresholds, Proceedings of ACM SIGIR 2001, pp. 294-302.