

# Patent classification experiments with the Linguistic Classification System LCS

Suzan Verberne, Merijn Vogel and Eva D’hondt

Information Foraging Lab  
Department of Computer Science  
Radboud University Nijmegen  
s.verberne@let.ru.nl

**Abstract.** In the context of the CLEF-IP 2010 classification task, we conducted a series of experiments with the Linguistic Classification System (LCS). We compared two document representations for patent abstracts: a bag-of-words representation and a syntactic/semantic representation containing both words and dependency triples. We evaluated two types of output: using a fixed cut-off on the ranking of the classes and using a flexible cut-off based on a threshold on the classification scores. Using the Winnow classifier, we obtained an improvement in classification scores when triples are added to the bag of words. However, our results are remarkably better on a held-out subset of the target data than on the 2000-topic test set. The main findings of this paper are: (1) adding dependency triples to words has a positive effect on classification accuracy and (2) selecting classes by using a threshold on the classification scores instead of returning a fixed number of classes per document improves classification scores while at the same time it lowers the number of classes needs to be judged manually by the professionals at the patent office.

## 1 Introduction

In this paper, we describe the classification experiments that we conducted in the context of the Intellectual Property (IP) track at CLEF (CLEF-IP<sup>1</sup>). In 2009, the track was organized for the first time with a prior art retrieval task. In 2010, a classification task was added to the track.

The goal of the classification task at CLEF-IP is to “classify a given patent document according to the International Patent Classification system (IPC)”. For the purpose of the track, the organization released a collection of 2.6 million patent documents pertaining to 1.3 million patents from the European Patent Office (EPO) with content in English, German and French. From the collection, 2000 documents (the ‘topics’) were held out as test set. The remainder of the corpus constitutes the target data, on which participants could develop their methods.

The target data comprise EPO documents with application dates older than 2002. Multiple documents pertaining to the same patent were not merged — the task was to classify them as individual documents. In the IPC system, documents are ordered hierarchically into sections, classes, subclasses, main groups and subgroups. In CLEF-IP, the classification task was to classify documents on the subclass level.

In our experiments, we focused on the use of different document representations in text classification. We compared the widely used bag-of-words model to a document representation based on syntactic/semantic terms, namely dependency triples (DTs). Dependency triples are normalized syntactic units consisting of two terms (a head and a modifier) and the syntactic relation between them (e.g. subject, object or attribute). For all experiments, we used the Linguistic Classification System (LCS).

In this notebook paper, we first explain which parts of the corpus we used and the how we prepared the data (Section 2). In Section 3, we describe our experiments and the classification settings used. We conclude with a discussion of the results (Section 4) and a plan for follow-up experiments (Section 5).

<sup>1</sup> <http://www.ir-facility.org/research/evaluation/clef-ip-10>

## 2 Data preparation

The data selection in our experiments was motivated by practical concerns: Since we wanted a comparison between classification experiments using bag-of-words and syntactic/semantic terms, the choice of data was limited to abstracts as these are the easiest and consequently the fastest to parse. We parsed all (over 500,000) English abstracts of the corpus in a couple of days. To parse all the claims and/or description sections would have taken considerably longer because of the extremely long and complex sentences used in these sections. [4].

We extracted from the corpus all files that contain both an abstract in English and at least one IPC class in the field <classification-ipcr>.<sup>2</sup> We extracted the IPC classes on the document level, not the invention level. This means that we did not include the inventions where the IPC class is in another file than the English abstract. We saved the abstract texts in plain text, and administrated the IPC classes in a separate file.

For the bag-of-words representation, we ran a simple normalization script that removed punctuation, capitalization and numbers from all abstract files. For the syntactic/semantic representation, we parsed the abstract texts with the AEGIR dependency parser [3]. AEGIR allows us to set a maximum parse time per sentence, which is useful since for longer (and hence more ambiguous) sentences the parsing speed goes down. The output of the parser is a list of dependency triples for each abstract that have undergone a number of normalizing transformations on the morphologic and syntactic level, such as the transformation from passive to active voice (hence the term ‘syntactic/semantic’).

Figure 1 is an example of a small original text, the normalized text in the bag-of-words representation and the triples in the syntactic/semantic representation. For the experiments with the syntactic/semantic representation, the triples are concatenated to the words. Table 1 gives general statistics on the target data and the test data.

Original text	words	triples
Heat is stored	heat is stored	[IT,SUBJ,store] [store,OBJ,heat]
at a steady	at a steady	[store,PREPat,temperature]
temperature using	temperature using	[temperature,ATTR,steady]
calcium chloride	calcium chloride	[temperature,DET,a] [chloride,ATTR,calcium]
hexahydrate and	hexahydrate and	[hexahydrate,ATTR,chloride]
up to 20 percent	up to percent	[hexahydrate,ATTR,using] [up,PREPto,20 percent]
strontium chloride	strontium chloride	[assist,OBJ,crystallization]
hexahydrate	hexahydrate	[chloride,ATTR,strontium]
to assist	to assist	[hexahydrate,ATTR,chloride]
crystallisation.	crystallisation	[hexahydrate,SUBJ,assist]

**Fig. 1.** Part of the original text from the abstract of document EP-0011358-A1.txt (left) and the two document representations that we created: normalized text (words) and syntactic/semantic terms (triples)

**Table 1.** Statistics on the target data and test data (topic set)

	target data	test data
# of files	2 680 604	2 000
# of files with an English abstract and IPC-R class	532 274	2 000
% of abstract files with empty parser output (max. parse time 10 secs)	3.5	4.6
# of different IPC-R subclasses	629	476
Average number of classes per file	2.7	2.3

<sup>2</sup> IPC-R is the IPC Reform classification, sometimes also called IPC8. See [http://www.intellogist.com/wiki/IPC\\_Classification\\_System](http://www.intellogist.com/wiki/IPC_Classification_System).

### 3 Classification experiments with LCS

For our classification experiments, we use the Linguistic Classification System (LCS)<sup>3</sup> [2, 1]. The LCS can perform both mono-classification (each document belongs to precisely one class) and multi-classification. In the training phase, the LCS takes as input a file which list the paths to the classification files followed by their classes. After this training phase the LCS can be used for testing the classifier obtained on a test collection of documents with known classes (usually held-out training data), or for producing a classification of new documents without known classes.

#### 3.1 Experimental set-up

Three classifiers have been implemented in the LCS: Naive Bayes, Winnow or  $SVM^{light}$ . We experimented with both Winnow and  $SVM^{light}$  and we found that their classification accuracy scores are comparable but that  $SVM^{light}$  is much slower. For example, in order to train a model based on 425 819 abstracts that belong to 629 different subclasses, Winnow needed around two hours (independent of the document representation used) while  $SVM^{light}$  spent six and a half hours on the same task. Therefore, we decided to use Winnow for the CLEF-IP experiments.

Winnow has a number of parameters that can be tuned:  $\alpha$ ,  $\beta$  and *maxiters* (the number of training iterations). After some tuning around the default values, we decided to use  $\alpha = 1.02$  and  $\beta = 0.98$ . For *maxiters*, we experimented with three and ten iterations, and found that the classification accuracy still improved somewhat after the third iteration. Therefore we decided to use ten iterations

In the case of multi-classification, LCS is flexible with respect to the number of classes that is returned per document. Internally, it produces a full ranking of classes for each document in the test set. The user can regulate the selection of classes with three parameters: (1) a threshold that puts a lower bound on the classification score for a class to be selected, (2) the maximum number of classes selected per document ('maxranks') and (3) the minimum number of classes selected per document ('minranks'). We kept the selection threshold to 1.0 (which is the default). Based on the average number of classes per document in the target data, we decided to set *maxranks* = 4. Setting *minranks* = 1 assures that each document is assigned at least one class, even if all classes have a score below the threshold.

We present the results on four experiments with LCS:

1. Classifying abstracts from the target data in the bag-of-words (words-only) representation into IPC-R subclasses
2. Classifying abstracts from the target data in the syntactic/semantic (words+triples) representation into IPC-R subclasses
3. Classifying abstracts from the test data in the bag-of-words (words-only) representation into IPC-R subclasses
4. Classifying abstracts from the test data in the syntactic/semantic (words+triples) representation into IPC-R subclasses

For experiments 1 and 2, we randomly split the target data: we used 80% of the data for training the classifier and 20% for testing. We repeated this four times with different random splits and calculated the mean and standard deviation over the four outcomes in order to get a measure for the reliability of the results. For experiments 3 and 4, we applied classification models which were previously trained on a random 80% of the target data to the 2000 abstracts from the test data, after the relevance assessments for the topics had been released by the organization.

#### 3.2 Results

We present the results in terms of precision ( $P$ ), recall ( $R$ ) and their harmonic mean ( $F_1$ ) for two types of output: (a) the classes that were selected using the threshold on classification scores in

---

<sup>3</sup> A demo of the application can be found at <http://ir-facility.net/news/linguistic-classification-system-prototype/> for registered IRF members.

LCS and (b) the classes that were returned using a fixed cut-off point in the class ranking. For the threshold-based cut-off, precision and recall are calculated using:

$$P = \frac{|relevant\ classes \cap selected\ classes|}{|selected\ classes|} \quad (1)$$

$$R = \frac{|relevant\ classes \cap selected\ classes|}{|relevant\ classes|} \quad (2)$$

For the fixed cut-off, precision and recall are calculated using:

$$P@n = \frac{|relevant\ classes \cap classes\ returned@n|}{|classes\ returned@n|} \quad (3)$$

$$R@n = \frac{|relevant\ classes \cap classes\ returned@n|}{|relevant\ classes|} \quad (4)$$

We chose  $n = 4$  as a cut-off point for evaluating the ranking because it best compares to our parameters for the threshold-based cut-off in LCS ( $maxranks = 4$ ). In addition to that, we give the results in terms of  $P@1$  and  $R@50$  because precision is especially relevant in the high ranks and recall in the longer tail. We also give Mean Average Precision (MAP) for each of the experiments. The results for the target data and the test data are in Table 2 and 3 respectively.

**Table 2.** Classification results using Winnow on abstracts from a held-out subset of the target data. P, R and F are averages over four random 80–20 splits of the data. Between brackets is the standard deviation. All numbers are percentages. Boldface marks the results that are discussed in the next section.

	Threshold-based cut-off			Fixed cut-off					MAP
	P	R	F <sub>1</sub>	P@1	P@4	R@4	R@50	F <sub>1</sub> @4	
1. words-only	67.63 (0.17)	61.28 (0.15)	<b>64.30</b> (0.08)	80.91%	47.90%	70.41%	90.06%	<b>57.01%</b>	0.717
2. words+triples	73.64 (0.08)	61.74 (0.13)	<b>67.16</b> (0.07)	83.11%	50.21%	73.70%	93.73%	<b>59.73%</b>	0.755

**Table 3.** Classification results using Winnow on abstracts from the test data (2000 topics). All numbers are percentages. Boldface marks the results that are discussed in the next section.

	Threshold-based cut-off			Fixed cut-off					MAP
	P	R	F <sub>1</sub>	P@1	P@4	R@4	R@50	F <sub>1</sub> @4	
3. words-only	60.06	52.06	<b>55.77</b>	69.95%	37.46%	64.60%	87.61%	<b>47.42%</b>	0.665
4. words+triples	61.52	52.08	<b>56.41</b>	71.85%	38.36%	66.16%	89.59%	<b>48.56%</b>	0.685

## 4 Discussion

We compare the classification results from three different points of view: (1) the two document representations (words-only vs. words and triples), (2) the target data vs. the test data and (3) the threshold-based cut-off vs. the fixed cut-off for the class ranking.

With respect to the first point, we observe a significant improvement in classification performance on the target data when we add triples to the bag of words:  $F_1$  increases from 64.30 (with a standard deviation of 0.08) to 67.16 (with a standard deviation of 0.07). However, on the test data, this difference is much smaller and probably not significant.<sup>4</sup>

<sup>4</sup> We cannot measure standard deviations for the test data because the topic set is too small to split up and compare the results on random subsets of it.

That brings us to the second point: the results for the target data and test data are very different from each other. Overall classification scores are lower for the topic test set than they are on a held-out set from the target data ( $F_1$  for words-only is 55.77 compared to 64.30). Inspection of the files in both sets shows that all files included in the test data are newer than the ones in the target data. This was done by the CLEF-IP organization to reflect the realistic task of classifying incoming patent applications using a model trained on existing patents. The fact that models trained on older abstracts are a better fit on contemporary abstracts than on more recent abstracts suggests that the content of the patents belonging to a specific subclass has changed over time.

It is more difficult to explain why the improvement gained from adding triples to words is smaller for the test data than it is for the target data. Table 1 shows that in the test data more abstracts had empty parser output than in the target data but this difference is small (4.6% and 3.5% respectively). We checked the output of the parser for the topic abstracts but we have no reason to believe that the topic abstracts were that much more difficult to parse as to result in less reliable triplets. This leaves us with the option that the smaller improvement is (at least partly) due to coincidence. There are only 2000 topic abstracts that are classified in 476 different IPC-R classes. A different selection of 2000 abstracts could easily lead to a few percent change in the classification accuracy.

Finally, we compared the results on the ranking with fixed cut-off to the results for the threshold-based cut-off. We see that class selection using a threshold on the classification score has a positive effect on both the precision and the recall, and hence on the  $F_1$  score (64.30% compared to 57.01% at rank 4 for words-only on the target data). Selecting classes by using a threshold on the classification scores for the classes instead of returning a fixed number of classes per document leads to better classification while a lower number of classes needs to be judged manually.

## 5 Follow-up experiments

For the proceedings of CLEF-IP 2010, we plan to conduct follow-up experiments in two directions.

First, we will investigate why the improvement gained from adding triples to words is smaller for the test data than it is for the target data. We plan to look into (1) the distribution of IPC classes in the test data compared to the target data, (2) the subset of IPC classes that are covered by the target data but not by the test data and (3) the impact of triples compared to words in the class profiles of these classes.

In order to find out whether the differences between the results for the test data and the target data are due to coincidence, we plan to create at least five test sets of 2000 abstracts extracted from the same time slice of the MAREC corpus as the supplied topic test set. Then we will classify these sets using the same models trained on the target data in order to obtain the variation of the classification accuracy on test sets of 2000 abstracts.

Finally, we plan to set up a series of tuning experiments for the threshold parameter in LCS on a held-out development set, to see if we can gain additional improvement from optimizing the class selection.

## References

1. C.H.A. Koster and J.G. Beney. Phrase-based document categorization revisited. In *Proceedings of the 2nd international workshop on Patent information retrieval*, pages 49–56. ACM, 2009.
2. C.H.A. Koster, M. Seutter, and J. Beney. Multi-classification of patent applications with Winnow. *Lecture Notes in Computer Science*, pages 545–554, 2003.
3. Nelleke Oostdijk, Suzan Verberne, and Cornelis H.A. Koster. Constructing a broad coverage lexicon for text mining in the patent domain. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC 2010)*. European Language Resources Association (ELRA), 2010.
4. Suzan Verberne, Eva D’hondt, Nelleke Oostdijk, and Cornelis H.A. Koster. Quantifying the Challenges in Parsing Patent Claims. In *Proceedings of the 1st International Workshop on Advances in Patent Information Retrieval (AsPIRe 2010)*, pages 14–21, 2010.