

# Phrase-based Document Categorization

Cornelis H.A. Koster<sup>1</sup>, Jean G. Beney<sup>2</sup>, Suzan Verberne<sup>1</sup>, and Merijn Vogel<sup>1</sup>

<sup>1</sup> Computing Science Institute ICIS, Univ. of Nijmegen, The Netherlands  
{kees,s.verberne,merijnv}@cs.ru.nl

<sup>2</sup> Dept. Informatique, LCI, INSA de Lyon, France  
jean.beney@insa-lyon.fr

**Abstract.** (Chapter in Springer book "Current Challenges in Patent Information Retrieval", to appear in May 2011) This paper takes a fresh look at an old idea in Information Retrieval: the use of linguistically extracted phrases as terms in the automatic categorization of documents, and in particular the pre-classification of patent applications. In Information Retrieval, until now there was found little or no evidence that document categorization benefits from the application of linguistics techniques. Classification algorithms using the most cleverly designed linguistic representations typically did not perform better than those using simply the bag-of-words representation.

We have investigated the use of dependency triples as terms in document categorization, according to a dependency model based on the notion of aboutness and using normalizing transformations to enhance recall.

We describe a number of large-scale experiments with different document representations, test collections and even languages, presenting evidence that adding such triples to the words in a bag-of-terms document representation may lead to a statistically significant increase in the accuracy of document categorization.

## 1 Introduction

The document representation most widely used today in Information Retrieval (IR) is still the *bag-of-words* model, both in traditional query-based retrieval and in automatic document categorization<sup>3</sup>. In this document representation, the order of the words in a document plays no role at all, whereas our intuition tells us that important information about the category of a document must be available in certain groups of words found in the document. It is as if we are trying to determine the function of a ruined building (church? castle? monastery? dwelling? shop? railway station?) by dismantling it into bricks and then looking exclusively at the frequency distribution of the individual types of bricks, disregarding any coherent substructures (the bag-of-bricks model). Intuitively it seems obvious that larger units (combinations of words into meaningful phrases) could serve as better terms for the categorization, yet there is surprisingly little evidence corroborating this intuition.

---

<sup>3</sup> The terms Categorization and Classification are used interchangeably.

In this paper we describe a successful attempt to improve the accuracy of classification algorithms for patent documents by using, besides words, dependency triples (see section 2) as features in the document representation.

### 1.1 Previous Work

The use of linguistically derived phrases as indexing terms has a long history in IR (for an overview see [25]). Many different kinds of linguistic phrases have been tried, with at most a modest success [4]. The predominant feeling about the value of NLP to IR, as voiced in [25], is that only ‘shallow’ linguistic techniques like the use of stop lists and lemmatization are of any use to IR; the rest is a question of using the right statistical techniques.

In the literature on document classification two forms of linguistic phrases are distinguished:

- *statistical phrases*, chunks or collocations: sequences of  $k$  non-stop words occurring consecutively [7], stemmed and ordered bigrams of words [6], even collocations taken from a specially prepared domain terminology [3];
- *syntactic phrases*, identified by shallow parsing [1, 17], template matching, finite state techniques [10] or by “deep” parsing [9, 26, 22, 12].

In spite of all these efforts, over the years no classification experiment using syntactic phrases has shown a marked improvement over the use of single keywords, at least for English. Only [22] recently reported a positive effect in classifying the Reuters-21578 collection. In the case of statistical phrases, the experience is only a little bit more positive [6, 17].

In this paper, we show that the use of syntactic phrases can significantly improve the accuracy of patent classification when each document is represented by a bag of words and dependency triples, generated according to a new aboutness-based dependency model, using deep parsing and transduction.

In the remainder of this section we describe the IR notion of aboutness. We propose an indirect way to measure aboutness and formulate the aboutness hypothesis. In section two of this paper we will describe the use of dependency triples as terms for IR, introducing and motivating a dependency model based on the notion of aboutness and discussing the syntactic normalization of phrases, which is crucial for achieving recall.

In section three we describe the software and data resources used in our experiments: the test corpora, parsers and classification engines used. In section four and five we describe our experiments to test the effect of using dependency triples according to this model in the classification of patent documents, investigating the plausibility of the aboutness hypothesis and presenting the experimental results. Finally, we formulate our conclusions.

### 1.2 The Notion of Aboutness

The notion of *aboutness* is highly central to Information Retrieval: the user of a retrieval system expects the system, in response to a query, to supply a list of documents which are *about* that query.

A model-theoretic basis for the notion of aboutness was described in [5]:

An information carrier  $i$  will be said to be *about* information carrier  $j$  if the information borne by  $j$  holds in  $i$

The rather abstract notion of “information carrier” can denote a single term, but also a composition of terms into a structured query or a document. Practical retrieval systems using words as terms are in fact based on a surprisingly simple-minded notion of aboutness:

If the word  $x$  occurs in the document then the document is *about*  $x$ .

This notion may appear highly naive, but it leads directly to the vector-space model when a measure for the *similarity* between the query and the document is introduced, based on the aboutness of words, the term frequency and document frequency.

For phrase-based retrieval, we need a notion of aboutness appropriate for phrases, which in its simplest form can be defined in the same way:

If the phrase  $x$  occurs in the document then the document is *about*  $x$ .

But the problem with all these definitions is that they are not concrete enough to compare the aboutness of different document representations.

Although intuitively it seems obvious that linguistic phrases provide a more informative document representation than keywords, the above formulation is not helpful in deciding what phrases from a document to choose and how to represent them.

### 1.3 Measuring Aboutness

In fact we cannot measure aboutness directly for lack of a well-defined measure, but we can compare the accuracy (precision and recall) achieved in the categorization of a given set of documents using different document representations: *the representation that leads to the highest accuracy must best capture the aboutness of the documents.*

Text Categorization provides a good vehicle for the study of document representations, because many suitable test sets are available and accuracy can be measured objectively. Categorization does not suffer from the problem of measuring recall, unlike query-based search. Classification algorithms with a firm statistical basis allow objective and repeatable experiments, that can easily be replicated by others.

Successful classification algorithms are purely based on the frequency distribution of the *content words* (non stop-words) occurring in the different categories (forming the “language models”), and therefore on aboutness, instead of being based on any deep semantic analysis and reasoning. Statistics is a good and cheap alternative for the Semantics which is as yet unattainable.

## 1.4 The Aboutness Hypothesis

According to Information Retrieval folklore, the best classification terms are the *content words*: the words from the open categories, in particular nouns and adjectives. The words from the closed categories are just stop words. In a classification experiment reported in [2], it was indeed found that nouns, verbs and to a lesser degree adjectives and adverbs are the only lexical words that contribute measurably to the aboutness of a text.

These observations lead us to the following *aboutness hypothesis*:

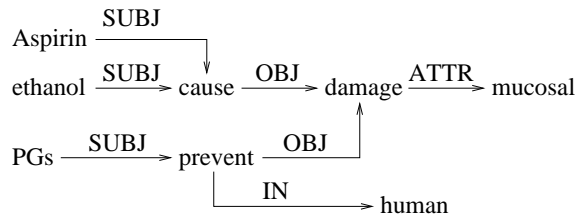
- of all the words in a text, the words from the open categories (nouns, verbs, adjectives, adverbs) carry most of the aboutness;
- of all possible dependency triples (words from the text joined by a syntactic relator), the triples containing only words from the open categories will carry most of the aboutness;
- we expect a better classification result when using triples as terms in addition to words than when using words alone.

We are going to investigate the elements of this hypothesis in one of the hardest document classification tasks: the classification of patent documents.

## 2 Dependency Graphs and Dependency Triples

By a *dependency graph* for a phrase we mean an acyclic graph (a tree with possibly additional confluent arcs) whose nodes are marked with words from that phrase and whose arcs are marked with directed relations. We are particularly interested in *syntactic relations*, those that can reliably be found by parsing.

As an example, we assign the following dependency graph to the sentence 'In humans, PGs prevent the mucosal damage caused by aspirin and ethanol':



A dependency graph represents the main structure of a sentence in an abstract way, much more compactly than a constituent tree (parse tree), in terms of certain syntactic relations between words of the sentence. It represents the compositional structure of the sentence, over which semantic relations can be constructed relatively easily. In that sense it is close to a semantic representation.

By a *dependency triple* we mean a triple [word,relation,word], which forms part of a dependency graph, from which it can be obtained by *unnesting* the graph to the triples contained in it. Triples are a syntactic way of representing phrases, closely related to the Head/Modifier pairs often used in IR (e.g. [9]).

## 2.1 The aboutness-based dependency model

The following table shows an example triple for each of the most important dependency relations:

subject relation	[device,SUBJ,comprise]
object relation	[cause,OBJ,lesion]
predicate relation	[aspirin,PRED,NSAID]
attribute relation (adj.)	[damage,ATTR,mucosal]
attribute relation (noun)	[breaker,ATTR,crust]
prepos relation (noun)	[coating,PREPof,albumin]
prepos relation (verb)	[charge,PREPinto,container]
prepos relation (adj.)	[related,PREPto,serotonin]
modifier relation (verb)	[extending,MOD,angularly]
modifier relation (adj.)	[efficient,MOD,highly]

Notice that in each of those relations both the head and the modifier are content words. In this respect, the aboutness-based dependency model is quite different from traditional linguistically-motivated dependency models that describe dependency relations between *all* words in the sentence, including punctuation, such as Minipar [19] and Link Grammar [24]; it is closer to the collapsed version of the Stanford Dependency Parser [21] output. Furthermore, the *subject* is taken as the head of a clause rather than the verb. This has some advantages: it allows one head to be the subject of several verbal clauses, and makes it easy to express certain raising constructions.

## 2.2 Factoids

The above dependency relations express only the factual content of the sentence, without further frills like time and modality of the verbs, topicalization, the argumentation structure (as expressed by conjunctions) and the linear word order, which are lost in this graph representation. Aboutness-based dependency graphs are well-suited to represent *factoids*, simple sentences expressing a (purported) fact. A typical factoid pattern is ‘who did what to whom with what’.

From the above example sentence, the following factoids can be derived:

- Aspirin causes mucosal damage
- ethanol causes mucosal damage
- PG’s prevent [that] damage in humans.

## 2.3 Normalization

In writing a text, people avoid literal repetition: they will go out of their way to choose another word, another turn of phrase, using anaphora, synonymy, hypernymy, periphrasis, metaphor and hyperbole in order to avoid the literal repetition of something they have said or written before. From a literary standpoint this

is wonderful, but it gives complications in IR: we have to compensate for this human penchant for variety.

Rather than using literal phrases taken from the text as terms, we therefore reduce each phrase to a normal form which expresses only the bare bones of its aboutness. We eliminate all dispensable ornaments and undo such morphological, syntactic and semantic variation as we can, while conserving the aboutness.

The *syntactic normalization* is in our case expressed in the grammar underlying our parser, using a mechanism for describing compositional transduction: every construct in the grammar is described along with its transduction to the output format (dependency graphs with content words as heads and modifiers).

- all elements which do not contribute to the aboutness of the text are elided during transduction: articles and determiners, quantifiers, auxiliary verbs, conjunctions - which is much like applying a stop list;
- embedded constructions such as relative clauses and participial constructions, like the one in the example sentence, are expanded into additional basic sentences (and therefore SUBJ and OBJ relations);
- a preposition linking two content words is treated as a parameter to the PREP relation, bringing the content words cw1 and cw2 together into one triple [cw1,PREPpreposition,cw2], rather than producing triples like

```
[content-word-1, PREP,preposition]
[preposition,PREP',content-word-2]
```

in which the preposition (a non-content word) occurs as a head or modifier;

- for the same reason, in triples involving a copula the copula is elided: ‘*the car is red*’ leads to the triple [car,ATTR,red], just like ‘*a red car*’;
- one of the most effective normalizing transformations is *de-passivation*: transforming a passive sentence into an active sentence with the same aboutness. By this transformation, the sentence *renal damage caused by Aspirin* is considered equivalent to *Aspirin causes renal damage*.

*Morphological normalization* by lemmatization is applied to the nouns and main verbs occurring in the resulting triples. As an example, the triples

```
[model,SUBJ,stand] [stand,PREPat>window]
```

can be obtained from ‘the model stood at the window’, ‘a model standing at the window’ and from ‘two models were standing at the window’. However the phrase ‘a standing order’ leads to a triple [order,ATTR,standing] because in this case ‘standing’ is not a main verb.

By these normalizing transformations, we try to improve the recall while surrendering little or no precision.

## 2.4 Bag of triples and bag of words

In any language, there are many more triples (pairs of words with a relator between them) than words. When these are used as terms in a bag-of-triples

model, the classification algorithm must be able to cope with very high numbers of features, demanding sophisticated Term Selection. But for the same reason, the feature space is very sparse. Triples may be high-precision terms, but they threaten to have a very low recall: a triple is never more frequent than the lowest frequency of the words from which it is composed. Low-frequency triples abound.

In classifying documents on e.g. gastric ulcers, it is therefore not immediately clear whether a triple like [ulcer, ATTR, gastric] is a more accurate search term than the co-occurrence of the words 'ulcer' and 'gastric'. It will probably have a higher precision but also certainly a lower recall. Compared to the literal string 'gastric ulcer' the triple will have the same precision but a slightly higher recall, since it catches also phrases like 'most of the ulcers appeared gastric'. Does the gain in precision compensate for the loss in recall? We need experimentation in order to find out whether triples work.

### 3 Resources used

In this section we describe the resources used in the experiments: the corpora, the classification system and the parser. The next section describes the experiments that were carried out and their results.

#### 3.1 The CLEF-IP corpora

The Intellectual Property Evaluation Campaign (CLEF-IP)<sup>4</sup> is an ongoing benchmarking activity on evaluating retrieval techniques in the patent domain. CLEF-IP has two main goals:

- to create a large test collection of multi-lingual European patents
- to evaluate retrieval techniques in the patent domain

In 2010, CLEF-IP uses a collection of  $\pm 2$  million patent documents with text in English, German, and French, part of which was used in our experiments. In the remainder, we shall designate this corpus by the name CLIP10. The total number of files in the CLIP10 corpus is 2,680,604, in 3 different languages. We classified only the abstracts, focusing on the two languages for which we have a parser, and taking only those documents for which an IPC-R classification is available. We used two sub-corpora:

- CLIP10-EN: abstracts taken from all files in English having one or more IPC-R classifications
- CLIP10-FR: the same for French.

We considered only those classes and subclasses for which at least one document was available. Some statistics about these sub-corpora:

---

<sup>4</sup> See [www.ir-facility.org/research/evaluation/clef-ip-10](http://www.ir-facility.org/research/evaluation/clef-ip-10).

	CLIP10-EN	CLIP10-FR
nmb of documents	532,274	55,876
avg nmb of words per document	119.5	121.2
nmb of classes	121	118
nmb of subclasses	629	617
avg nmb of classes/doc	2.13	1.32
min/max nmb of classes/doc	1/14	1/7
avg nmb of subclasses/doc	2.72	1.42
min/max nmb of subclasses/doc	2/19	1/7

### 3.2 The English parser AEGIR

The experiment on CLIP10-EN made use of a new hybrid dependency parser for technical English, which combines a rule-based core grammar (Accurate English Grammar for IR) and a large lexicon of technical terms with various disambiguation mechanisms. The experiments described here were its first large scale test. AEGIR is a further development of the EP4IR parser (English Phrases for Information Retrieval) [14]. In those experiments involving words, the words in the documents were case-normalized but not lemmatized; punctuation was removed.

The words occurring in the triples were case-normalized, and *cautiously* lemmatized: nouns were brought into their singular form, and main verbs brought into their infinitive form. When not used as main verb (e.g. “related”), participles were not lemmatized. Verb particles were attached to the verb lemma (e.g. “turn\_back”).

The parser was running on a relatively small parallel computer (4 CPU’s with 4 cores each) with a very large main memory (128 Gigabytes).

### 3.3 The French parser FR4IR

The French documents were parsed with the French parser FR4IR developed by Jean Beney at the INSA de Lyon. A description of the FR4IP parser can be found at [www.agfl.cs.ru.nl/FR4IP](http://www.agfl.cs.ru.nl/FR4IP) where there is also a demonstration version. The parser is not at present in the public domain, but for scientific purposes a copy of the parser can be obtained from [jean.beney@insa-lyon.fr](mailto:jean.beney@insa-lyon.fr). It is still very much under development.

The preprocessing of the French documents was similar to that for English. All experiments with French documents were performed on the Large Data Collider (LDC) of the Information Retrieval Facility ([www.ir-facility.org](http://www.ir-facility.org)).

### 3.4 The LCS Classification Engine

The Linguistic Classification System (LCS) used in these experiments [13] was developed at the University of Nijmegen during the IST projects DORO (Esprit 22716, 1997 to 1999) and PEKING (IST-2000-25338, 2000 to 2002) for the experimentation with different document representations. In the current TM4IP

project <sup>5</sup> it was re-implemented in Java. For most of the experiments we used the Balanced Winnow algorithm [20, 8], a child of the Perceptron, which is very efficient and highly robust against large but sparse feature sets. Some of the experiments were also performed using the Support Vector Machine algorithm (SVM light [11]).

As is the case for other classification engines, the performance of the classifiers depends on certain tuning parameters. In the experiments with Winnow in this section the parameters were chosen based on some small tuning experiments on training data:

Promotion parameters:  $\alpha = 1.02$ ,  $\beta = 0.98$   
Thick threshold parameters:  $\theta^- = 0.5$ ,  $\theta^+ = 2.0$   
Number of Winnow iterations = 10  
Global term selection:  $DF > 1$ ,  $TF > 2$   
Local term selection: Simplified ChiSquare, max 10 000 terms/category.

For SVM, we used  $C = 2$ ; we only experimented with the linear kernel.

In combining triples and words, both were simply considered as literal terms (the two term lists were concatenated). We did not try to give them different weights.

Since patents may have multiple IPC-R labels, we perform a multiclassification. After training classifiers for each category, the test documents were given a score by each classifier and each test document that scored better than a certain threshold for a certain category was assigned to that category.

The Winnow algorithm has a natural threshold 1 for each category. Each test document was assigned to at least one category (the one with the highest score) and at most five categories for which its score was greater than 1. Micro-averaged Precision, Recall and F1 were then computed in the usual way by comparing these categories with the ones for which the document is labeled.

This procedure reflects a pre-classification situation: each incoming document must be sent to precisely the right examiners.

For other measures (Recall or Precision at  $k$  documents, etc) for all categories a full ranking of all test documents is produced.

## 4 Experimental Results

In the following experiments we have investigated the aboutness hypothesis, using document classification:

- what is the relative contribution of the open word classes?
- which dependency relations contribute most to the aboutness?
- is classification on words plus triples more accurate than classification on words alone?

---

<sup>5</sup> [www.phasar.cs.ru.nl](http://www.phasar.cs.ru.nl)

Our main experiment concerns the comparison of three document representations: words alone, triples alone and words plus triples. As the measure of accuracy we used the Micro-averaged F1, assuming a loss function attaching equal weight to precision and recall, which is representative for applications in patent pre-classification.

#### 4.1 Experimental setup

The experiments in this section were each performed 10 times (10-fold cross-evaluation). In each experiment on unseen documents the documents were split at random into a train set (80%) and a test set (20%). After training on all train and test sets, we determined the micro-averaged value and standard deviation of the precision, recall and F1.

For the experiments with seen data we trained on a random subset of 80% of the documents and used a quarter of these train documents as test set. As usual, the terms ‘seen’ and ‘unseen’ refer to testing the accuracy of the classifier on the train set used and on the held-out test sets, respectively.

In the tables, a number in brackets indicates the standard deviation. The rightmost column shows for each representation the improvement with respect to the baseline.

We ran these experiments with both the Winnow and SVM classifiers, in order to compare the effect of triples for those algorithms. Winnow and SVM are both known to cope well with large numbers of terms (in this case triples).

#### 4.2 Winnow – Results for English

##### Testing on seen data

Algorithm	P	R	F1	improvement
testing on seen documents				
Winnow	CLIP10-EN/classes			
words alone	83.36(0.08)	72.05(0.23)	77.30(0.10)	baseline
triples alone	88.61(0.09)	73.82(0.19)	80.54(0.15)	+3.24
words+triples	89.65(0.08)	80.41(0.06)	84.78(0.07)	+7.48
Winnow	CLIP10-EN/subclasses			
words alone	83.93(0.04)	65.15(0.21)	73.36(0.13)	baseline
triples alone	90.80(0.02)	69.51(0.05)	78.74(0.03)	+5.38
words+triples	91.64(0.04)	76.48(0.05)	83.38(0.04)	+10.02

The accuracy (F1) on *seen* documents has in all cases improved over that of words alone, both for triples alone and for words plus triples. This improvement may be attributed to the fact that there are so many more different triples than different words:

	different terms	after term selection
words	115,604	74,025
triples	7,639,799	1,177,134
words+triples	8,671,497	1,526,353

Using so many terms, it becomes easier to construct a class profile which models the differences between the classes. But these good results on seen data are not important for practical applications, since they do not carry over to unseen data.

**Testing on unseen data** On unseen data, the picture is indeed somewhat different.

Algorithm	P	R	F1	improvement
testing on unseen documents				
Winnow	CLIP10-EN/classes			
words alone	76.33(0.14)	66.35(0.13)	70.99(0.07)	baseline
triples alone	74.26(0.08)	58.71(0.09)	65.57(0.07)	-5.42
words+triples	78.25(0.07)	69.06(0.09)	73.37(0.06)	+2.37
Winnow	CLIP10-EN/subclasses			
words alone	72.78(0.13)	55.96(0.16)	63.27(0.07)	baseline
triples alone	71.80(0.09)	48.64(0.11)	57.99(0.09)	-5.28
words+triples	75.72(0.10)	59.31(0.13)	66.52(0.08)	+3.25

The results on *unseen* data are representative for the intended application (pre-classification of patent applications). The base line (words) may look disappointing, but it is quite hard to improve upon it. Attempts using lemmatization, stop lists etc (not reported here) have consistently given no significant improvement.

As is to be expected, the classification on the 121 main classes is (by about 7 points) better than that on 629 subclasses. But in both cases the difference (of 2.37 or 3.25 percent points) between the base line of words and the triples plus words is many times the standard deviations. Therefore the difference is statistically highly significant – that is our main result.

Note that an accuracy below the base line is achieved when (unwisely) using only triples as terms. Both precision and recall are much lower for triples-only than for words, and so is the F1. Experts [9, 17] agree that linguistic terms should be *added* to the word terms, not used instead. Still, there is some progress: in our previous experiments [12] with Head/Modifier pairs we found the accuracy using only pairs to be much lower; probably the quality of the linguistic document representation and the parser has in the mean time improved, but not enough to “beat” the baseline (but see section 4.5).

### 4.3 SVM – Results for English

For performance reasons, we ran the experiments with SVM on classes with four-fold instead of ten-fold cross-evaluation, and the experiment on subclasses only once, without cross-evaluation (one experiment classifying subclasses using words and triples took 22 hours with SVM, compared to 2 hours with Winnow), but since the standard deviations appear to be quite low, we felt that performing a ten-fold cross-evaluation was not necessary.

Algorithm	P	R	F1	improvement
testing on seen documents				
SVM	CLIP10-EN/classes			
words alone	90.32(0.07)	72.31(0.10)	80.32(0.09)	baseline
triples alone	94.80(0.07)	84.73(0.04)	89.48(0.05)	+9.16
words+triples	94.93(0.03)	77.71(0.05)	85.46(0.04)	+5.14
SVM	CLIP10-EN/subclasses			
words alone	92.30	68.84	78.86	baseline
triples alone	96.66	87.75	91.9	+13.04
words+triples	98.71	91.23	94.82	+15.96
testing on unseen documents				
SVM	CLIP10-EN/classes			
words alone	81.09(0.12)	62.23(0.14)	70.42(0.14)	base line
triples alone	78.34(0.09)	58.35(0.11)	66.88(0.10)	-3.53
words+triples	84.54(0.01)	63.05(0.04)	72.23(0.02)	+1.81
SVM	CLIP10-EN/subclasses			
words alone	78.22	52.88	63.10	base line
triples alone	77.86	51.17	61.76	-1.34
words+triples	82.29	59.18	68.84	+5.74

The improvement in accuracy on seen documents was even larger for SVM than for Winnow. On unseen documents, a similar improvement when adding triples was found as was the case for Winnow, somewhat smaller for classes but larger for subclasses.

#### 4.4 Results for French

Again we trained on 80% of the documents and tested on 20%, with 10-fold cross-evaluation. We did not measure the accuracy on seen documents and used only Winnow. The results on *unseen* documents were as follows.

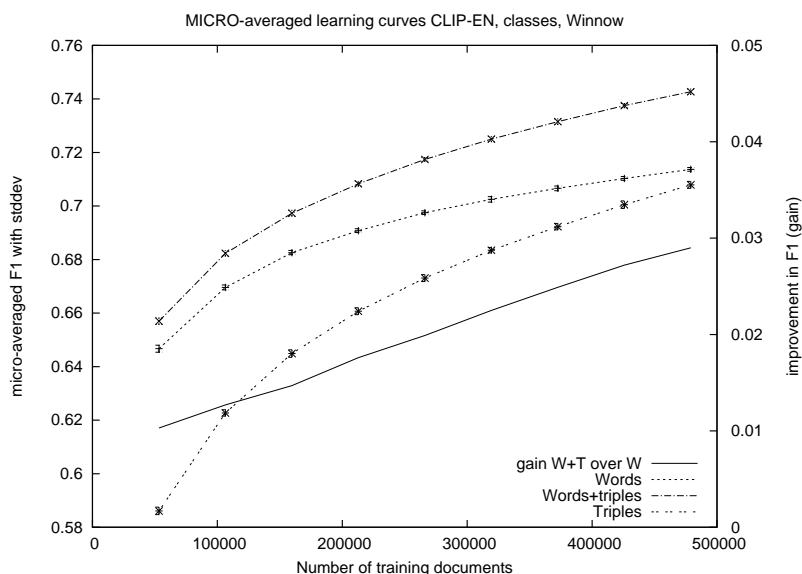
Algorithm	P	R	F1	improvement
testing on unseen documents				
Winnow	CLIP10-FR/classes			
words alone	69.94(.21)	66.95(.24)	68.41(0.19)	baseline
triples alone	54.41(.23)	46.28(.30)	50.02(0.26)	-18.40
words+triples	72.84(.23)	67.50(.22)	70.07(0.20)	+1.66
Winnow	CLIP10-FR/subclasses			
words alone	65.83(0.20)	55.47(0.22)	60.21(0.18)	baseline
triples alone	46.45(0.47)	35.74(0.32)	40.27(0.27)	-19.96
words+triples	68.67(0.30)	56.01(0.30)	61.70(0.28)	+1.49

The baseline (words) is much lower on these French documents than for the CLIP10-EN documents because there are much fewer train documents (see also following section). The results for triples-only are quite bad, but again the addition of triples to the words does improve the classification result.

#### 4.5 The number of train documents

In this section we investigate the effect of the number of documents trained on the accuracy obtained. The following graph is a *learning curve*, obtained by testing on a 10% test set and training on increasing numbers of train documents, for the three different document representations and for two different measures of accuracy. The experiment was performed using Winnow on CLIP10-EN (classes) with 10-fold cross-validation.

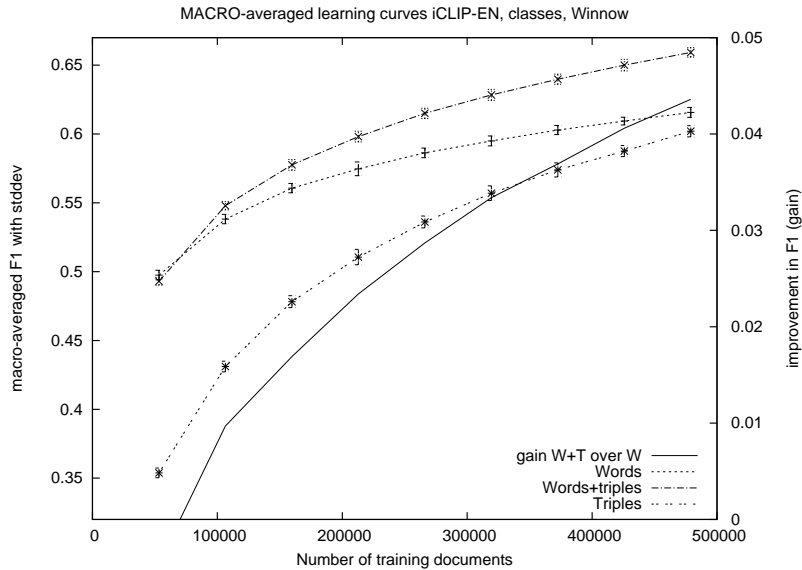
**The effect for the large classes** In the first experiment, as in the experiments reported earlier, the *micro*-averaged F1 is reported (averaging over all <document,class> pairs).



Adding triples to words always gives an improvement and the improvement grows with the number of documents trained. Initially, the classification power of triples alone is lower than that of words, but it grows faster. It appears that, with 100000 more train documents, the triples alone could achieve a better classification than the words alone, but we could not verify this.

The improvement given by words + triples over words alone (the gain) grows about linearly with the number of train documents. This is good news for patent classification, where millions of train documents are available.

**The effect for smaller classes** The micro-average is dominated by the large classes, those with many documents. Therefore the previous graph does not show whether the effect for smaller classes is also positive. We therefore also show a graph of the *macro*-averaged accuracy (averaged over the classes) which assigns equal weight to all classes. In this experiment, 5-fold cross-validation was used.



The macro-averaged F1 is lower than the micro-averaged F1 (it is dominated by the smaller classes, which have (much) fewer train documents) but the gain in using triples is now larger (except for very small numbers of train documents). It appears from this experiment that triples may work even better for the smaller classes than for the large classes.

## 5 The aboutness of words and triples

In this section we report on a number of experiments to shed light on the behaviour of triples as classification terms. Because here we are more interested in insight than in representiveness, we use a smaller subset of CLIP10-EN, consisting of only the documents in class A61 of the IPC-R, a total of 73869 documents in 13 subclasses, varying in size from 208 to 38396 documents. We trained Winnow classifiers for its 13 subclasses on those documents (80% train, 20% test), and used the classification for the subclass A61B in those experiments in this section which concern a single class profile.

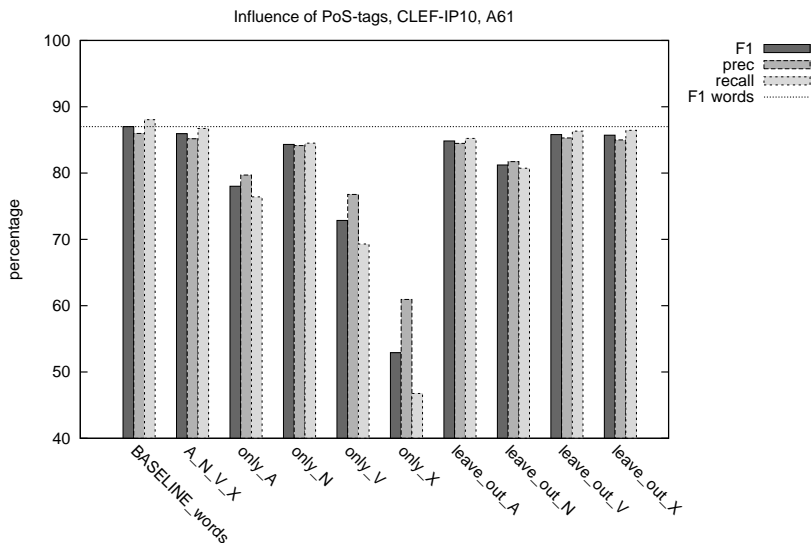
For these experiments with much fewer documents we used the default values for the Winnow parameters ( $\alpha = 1.1$ ,  $\beta = 0.9$ ,  $\theta^- = 0.5$ ,  $\theta^+ = 2.0$ , number of iterations = 3,  $DF > 1$ , no local term selection)

### 5.1 The Aboutness of Word Categories

In [2], the contribution to the accuracy by one category of words was measured by classifying a given corpus using only words from that category as terms. The category of words was determined using the Brill tagger. Of course this tagging was not very precise, because the tagger takes only local context into account.

We have repeated this experiment on CLIP10-EN class A61, using the AEGIR parser as a tagger: during the parsing, the parser will disambiguate the Part-Of-Speech (PoS) of every ambiguous word in the sentence. Both the Brill tagger and the parser can (and will) make errors, but in our approach we can be sure that each occurrence of a word obtains a Part-of-Speech which is appropriate in its syntactic context.

The following graphs show the results of the classification of A61 (80/20 split, 10-fold cross-evaluation). In each case we measured the accuracy (F1) for the classification of the subclass A61B, either using *only* the words with a certain PoS or *excluding* them. We use the letters A, N, V and X to stand for the open word categories: adjective, noun, verb and adverb.



As was to be expected, the one-POS-only experiments show that the nouns have by far the highest aboutness, followed by the adjectives and the verbs. Adverbs have very little aboutness.

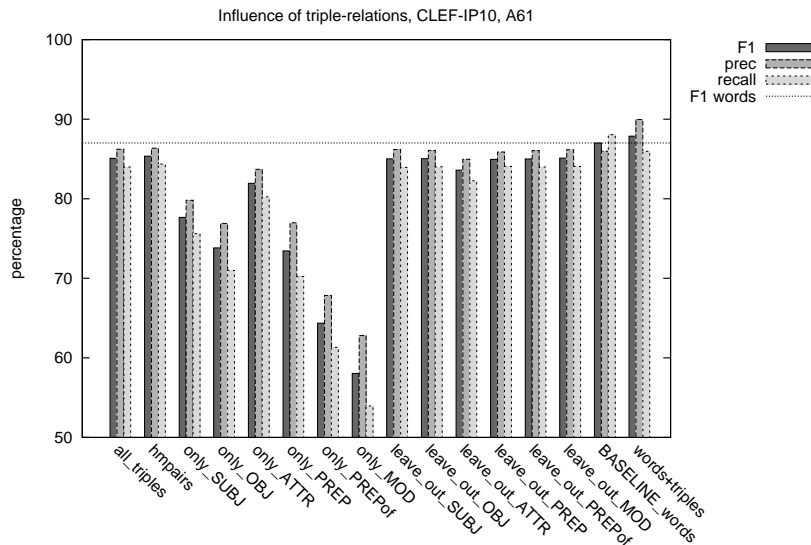
The leave-one-out results show less variation. The only word category that should not be left out are the nouns, the others make little or no difference. The accuracy when using all words is slightly better than when taking only those words that belong to one of the open categories (A+N+V+X), probably indicating some non-content words (e.g. numbers) that carry some aboutness.

## 5.2 The Aboutness of Relations

By a *relation* we here mean the set of all triples having a specific relator (e.g. ATTR or SUBJ). We want to measure for each relation what is its contribution to the accuracy of a classification.

We experimented again on the CLIP10-ENA61 corpus, this time representing each document by a bag of dependency triples, but either using *only* the triples

of some relation or *leaving out* all triples belonging to that relation. The results for the subclass A61B are shown in the following graphs:



The ATTR relation by itself has the largest aboutness, because it represents the internal structure of the noun phrase and most of the terminologically important terms are noun phrases. Then follow the PREP, OBJ and SUBJ relations (which may be because they have a noun as head or modifier). The MOD relation, which has a verb as head, makes the smallest contribution to accuracy, but it is also the least frequent.

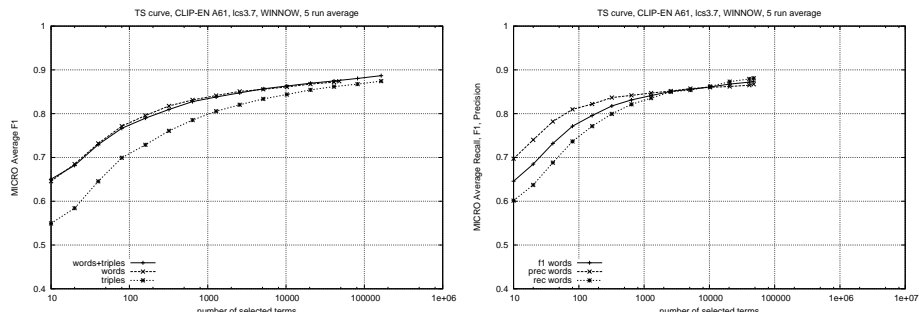
The leave-one-out experiments all give a very similar result. Only leaving out the ATTR relations leads to a marked drop in accuracy with respect to using all triples. Intuitively it is clear that the relations are not very independent terms, since the head of one triple is very often the modifier of another, or vice versa.

We also investigated the question whether the triple representation (with explicit relators) is better than the older Head/Modifier representation (obtained here by simply dropping the relator); the second graph from the left (marked hmpairs) shows a slightly higher recall and accuracy for Head/Modifier pairs.

### 5.3 Words and triples as terms

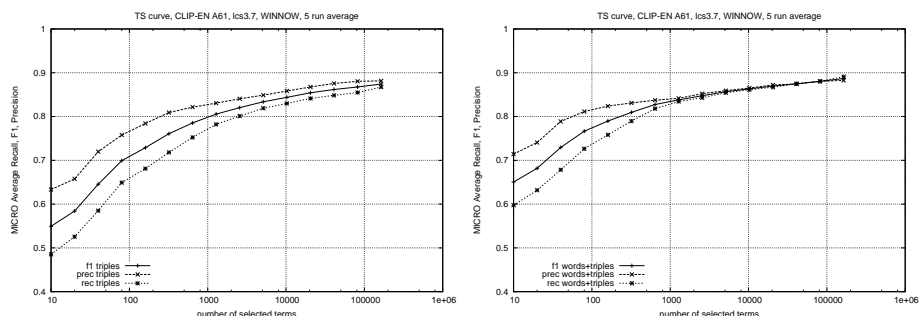
It appears that the use of triples as terms may lead to a better classification result, but many more of them are needed. Therefore we have compared the effectiveness of words and triples in dependence on the *number* of terms used, in a Term Selection setting: we performed the same experiment (classifying A61 subclasses with Winnow with words alone, triples alone and words plus triples) while selecting for each subclass only the top  $n = 2^k$  terms using the simple Chi-square Term Selection (TS) criterium [23].

We show two graphs; the first is the resulting *TS curve* (F1 against number of terms) for the three representations, and the second shows in more detail the TS curve for words. The variance is so small that we did not show it in the graphs.



The TS curve shows that the triples do not start contributing to the accuracy (F1) until more than 10000 terms are used; for fewer terms, using only words gives slightly better results. At the highest number of terms (beyond the end of the curve for words), triples plus words gives about 5 points improvement, while the F1 for triples alone equals that for words alone.

In the second graph, concerning words alone, we show also precision and recall. For low numbers of terms, precision exceeds recall, but between 2000 and 10000 terms the two curves cross, and above 10000 terms recall is larger than precision (but remember that this represents an average over 13 subclasses). Next, we look at the TS curves for the two other representations.



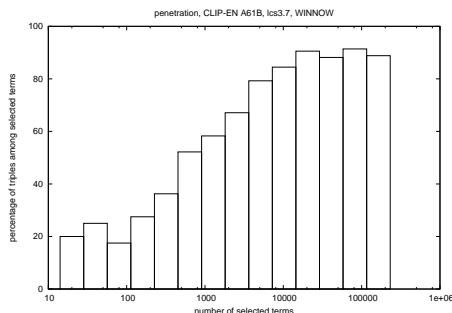
Using words plus triples, the cross-over point between precision and recall is now much higher than for words (between 40000 and 100000 terms). The graph shows that, when adding triples to words as terms, no term selection should be applied, because that may diminish the benefit.

For triples alone, precision is larger than recall in all cases, but it is possible (and looks likely) that a cross-over will occur for many more terms.

#### 5.4 The penetration of triples

Another measure for the effectiveness of linguistic classification terms which has been used in literature [6] is the *penetration* of such terms in the class profile:

in our case, the percentage of triples among the top  $n$  terms when classifying on both words and triples. We have measured it by classifying the subclasses of A61 (once) and then measuring the percentage of triples among the terms in the profile of A61B, in power-of-two buckets.



Among the 10 top terms are no triples, but after that more and more triples appear. After about 1000 terms the triples have a high penetration, but their presence does not improve the classification until very many triples are used. Inspection of the profile shows that, for fewer terms, the triples mostly replace words with more or less the same aboutness (as was found in [6]).

## 5.5 Discussion

The experiments described here were performed for the comparison of different document representations: words and words plus triples. For practical patent classification, of course many other considerations are important.

In classifying two different corpora of patent abstracts in two languages and with two different classification algorithms, a statistically significant increase in accuracy was found when adding triples to the word features (“bag of words and triples”). Further experiments are needed to see whether the same improvements are found when classifying other sections of the patents (e.g. the Claims) or non-patent documents.

In a previous article [15], we have described a similar series of experiments on patent abstracts and also on full-text documents, in which the results were very similar. For those experiments we used the EP4IR parser and LCS on the EPO2 corpus [16]. The presently used CLEF-IP corpus has the advantage of being publicly available, which makes it easier for others to duplicate our experiments.

It is interesting to note that all parsers used in the experiment are still under development. It appears likely that when the parsers reach more accuracy on the very complicated documents from the patent world, the effect of using triples may increase further.

The abundance of different triples each having only a low frequency still appears to be the main reason why the classification on triples alone gives disappointing results. We are now investigating whether this problem can be overcome by using some form of term clustering along the lines of [18].

The results on the aboutness of word categories and relations require further analysis, but it is definitely plausible that the aboutness hypothesis is well founded. Certainly not busted, we claim.

## 6 Conclusions

We have formulated the aboutness hypothesis, stating essentially that the open word categories are the carriers of aboutness, and introduced an aboutness-based dependency model, which differs from the more descriptive dependency models preferred in linguistics in that the heads and modifiers of the dependency graph involve only words from the open categories.

Dependency triples following this aboutness model are derived by syntactic parsing of the documents, followed by a transduction to dependency graphs and unnesting of the graphs to triples.

We have performed an experimental investigation on patent abstracts in English and French from CLEF-IP 2010. Our experiments showed that using aboutness-based dependency triples as additional features in document classification did lead to a statistically significant increase in classification accuracy, for larger as well for smaller classes. This improvement was found for different document collections in different languages and for different classification algorithms. The larger the number of train examples, the larger the improvement was found to be.

In a classification setting, we have compared the aboutness of different categories of words and dependency triples. We also compared the behaviour of words and triples using term selection and measured their penetration. All these experiments yielded rich evidence for the aboutness hypothesis, but this evidence still has to be analyzed theoretically and explained more deeply.

The two parsers and the LCS classification system used in the experiments are available from the authors for research purposes.

## References

1. M. Alonso, J. Vilares, and V. Darriba. On the usefulness of extracting syntactic dependencies for text indexing. *Springer LNCS 2464*, pg. 3–11, 2002.
2. A. Arampatzis, T. Van der Weide, C.H.A. Koster, and P. Van Bommel. An Evaluation of Linguistically-motivated Indexing Schemes. In *Proceedings of BCS-IRSG 2000 Colloquium on IR Research, 5th-7th April 2000*, Sidney Sussex College, Cambridge, England, 2000.
3. N. Bel, C.H.A. Koster, and M. Villegas. Cross-Lingual Text Categorization. In *Proceedings ECDL 2003, Springer LNCS 2769*, pg. 126–139, 2003.
4. T. Brants and Google Inc. Natural Language Processing in Information Retrieval. In *Proceedings CLIN 2003*, pg. 1–13.
5. P. Bruza and T. Huibers. Investigating Aboutness Axioms using Information Fields. In *Proceedings SIGIR 94*, pg. 112–121, 1994.
6. M. Caropreso, S. Matwin, and F. Sebastiani. A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. *Text Databases and Document Management: Theory and Practice*, pg. 78–102, 2000.

7. W. Cohen and Y. Singer. Context sensitive learning methods for text categorization. In *Proceedings of the 19th Annual International ACM Conference on Research and Development in Information Retrieval*, pg. 307–315, 1996.
8. I. Dagan, Y. Karov, and D. Roth. Mistake-Driven Learning in Text Categorization. In *Proceedings of the Second Conference on Empirical Methods in NLP*, pg. 55–63, 1997.
9. J. Fagan. *Experiments in automatic phrase indexing for document retrieval: a comparison of syntactic and non-syntactic methods*. PhD thesis, Cornell University, 1988.
10. G. Grefenstette. Light parsing as finite state filtering. In *Workshop on Extended finite state models of language, Budapest, Hungary, August 1996. ECAI'96.*, 1996.
11. T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European Conference on Machine Learning (ECML-98)*, pg. 137–142. Springer Verlag, 1998.
12. C.H.A. Koster and M. Seutter. Taming Wild Phrases. In *Proceedings ECIR2003*, pg. 161–176. Springer LNCS 2633, 2003.
13. C.H.A. Koster, M. Seutter, and J.G. Beney. Multi-classification of Patent Applications with Winnow. In *Proceedings PSI 2003, Springer LNCS 2890*, pg. 545–554, 2003.
14. C.H.A. Koster, M. Seutter, and O. Seibert. Parsing the Medline Corpus. In *Proceedings RANLP 2007*, pg. 325–329, 2007.
15. C.H.A. Koster, J. G. Beney. Phrase-Based Document Categorization Revisited. In *Proceedings of The 2nd International Workshop on Patent Information Retrieval (PAIR 2009) at CIKM 2009*, pg. 49-55, 2009.
16. M. Krier and F. Zaccà. Automatic Categorization Applications at the European Patent Office. *World Patent Information*, (24):187–196, 2002.
17. D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings ACM SIGIR'92*, 1992.
18. D. Lewis and B. Croft. Term Clustering of Syntactic Phrases. *Proceedings SIGIR '90*, pp. 385-404, 1990.
19. D. Lin. Dependency-based evaluation of MINIPAR. In *Workshop on the Evaluation of Parsing Systems*. Granada, Spain, 1998.
20. N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, (2):285–318, 1988.
21. M. De Marneffe, B. Maccartney and C. D. Manning. Generating typed dependency parses from phrase structure parses, In *Proceedings LREC 2006*, 2006.
22. V. Nastase, J. Sayyad Shirabad and M.F. Caropreso. Using Dependency Relations for Text Classification. University of Ottawa SITE Technical Report TR-2007-12, 13 pages, 2007.
23. F. Sebastiani (2002), Machine learning in automated text categorization, ACM Computing Surveys, Vol 34 no 1, 2002, pp. 1-47.
24. D. D. Sleator and D. Temperley. Parsing English with a link grammar. In *Third International Workshop on Parsing Technologies*
25. K. Spairck Jones. The role of NLP in Text Retrieval. In T. Strzalkowski, editor, *Natural Language Information Retrieval*, pg. 1–24. Kluwer, 1999.
26. T. Strzalkowski. Natural Language Information Retrieval. *Information Processing and Management*, 31(3):397–417, 1995.