Text Representations for Patent Classification

Eva D'hondt[*]
Radboud University Nijmegen

Suzan Verberne[**]
Radboud University Nijmegen

Cornelis Koster[†]
Radboud University Nijmegen

Lou Boves[‡]
Radboud University Nijmegen

With the increasing rate of patent application filings, automated patent classification is of rising economic importance. This article investigates how patent classification can be improved by using different representations of the patent documents. Using the Linguistic Classification System (LCS), we compare the impact of adding statistical phrases (in the form of bigrams) and linguistic phrases (in two different dependency formats) to the standard bag-of-words text representation on a subset of 532,264 English abstracts from the CLEF-IP 2010 corpus. In contrast to previous findings on classification with phrases in the Reuters-21578 data set, for patent classification the addition of phrases results in significant improvements over the unigram baseline. The best results were achieved by combining all four representations, and the second best by combining unigrams and lemmatized bigrams. This article includes extensive analyses of the class models (a.k.a. class profiles) created by the classifiers in the LCS framework, to examine which types of phrases are most informative for patent classification. It appears that bigrams contribute most to improvements in classification accuracy. Similar experiments were performed on subsets of French and German abstracts to investigate the generalizability of these findings.

1. Introduction

Around the world, the patent filing rates in the national patent offices have been increasing year after year, creating an enormous volume of texts, which patent examiners

[*] Center for Language Studies, PO Box 9103, 6500 HD Nijmegen, the Netherlands. E-mail: e.dhondt@let.ru.nl.

[**] Center for Language Studies / Institute for Computing and Information Sciences, PO Box 9103, 6500 HD Nijmegen, the Netherlands. E-mail: s.verberne@let.ru.nl.

[†] Institute for Computing and Information Sciences, PO Box 9010, 6500 HD Nijmegen, the Netherlands. E-mail: kees@cs.ru.nl.

[‡] Center for Language Studies, PO Box 9103, 6500 HD Nijmegen, the Netherlands. E-mail: l.boves@let.ru.nl.

Submission received: 19 March 2012; revised submission received: 8 August 2012; accepted for publication: 19 September 2012.

doi:10.1162/COLI_a_00149

are struggling to manage (Benzineb and Guyot 2011). To speed up the examination process, a patent application needs to be directed to patent examiners specialized in the subfield(s) of that particular patent as quickly as possible (Smith 2002). This **preclassification** is done automatically in most patent offices, but substantial additional manual labor is still necessary. Furthermore, since 2010, the International Patent Classification[1] (IPC) is revised every year to keep track of recent developments in the various subdomains. Such a revision is followed by a reclassification of portions of the existing patent corpus, which is currently done mainly by hand by the national patent offices (Held, Schellner, and Ota 2011). Both preclassification and reclassification could be improved, and a higher consistency of the classifications of the documents in the patent corpus could be obtained, if more reliable and precise automatic text classification algorithms were available (Benzineb and Guyot 2011).

Most approaches to text classification use the bag-of-words (BOW) text representation, which represents each document by the words that occur in it, irrespective of their ordering in the original document. In the last decades much research has gone into expanding this representation with additional information, such as statistical phrases[2] (*n*-grams) or some forms of syntactic or semantic knowledge. Even though (statistical) phrases are more representative units for classes than single words (Caropreso, Matwin, and Sebastiani 2001), they are so sparsely distributed that they have limited impact during the classification process. Therefore, it is not surprising that the best scoring multi-class, multi-label[3] classification results for the well-known Reuters-21578 data set have been obtained using a BOW representation (Bekkerman and Allan 2003). But the limited contribution of phrases in addition to the BOW-representation does not seem to hold for all classification tasks: Özgür and Güngör (2010) found significant differences in the impact of linguistic phrases between short newswire texts (Reuters-21578), scientific abstracts (NSF), and informal posts in usenet groups (MiniNg): Especially the classification of scientific abstracts could be improved by using phrases as index terms. In a follow-up study, Özgür and Güngör (2012) found that for the three different data sets, different types of linguistic phrases have most impact. The authors conclude that more formal text types benefit from more complex syntactic dependencies.

In this article, we investigate if similar improvements can be found for patent classification and, more specifically, which types of phrases are most effective for this particular task. In this article we investigate the value of phrases for classification by comparing the improvements that can be gained from extending the BOW representation with (1) statistical phrases (in the form of bigrams); (2) linguistic phrases originating from the Stanford parser (see Section 3.2.2); (3) aboutness-based[4] linguistic phrases from the AEGIR parser (Section 3.2.3); and (4) a combination of all of these. Furthermore, we will investigate the importance of different syntactic relations for the classification task,

[1] The IPC is a complex hierarchical classification system comprising sections, classes, subclasses, and groups. For example, the "A42B 1/12" class label, which groups designs for bathing caps, falls under section A *"Human necessities,"* class 42 *"Headwear,"* subclass B *"Head coverings,"* group 1 *"Hats; caps; hoods."* The latest edition of the IPC contains eight sections, 129 classes, 639 subclasses, 7,352 groups, and 61,847 subgroups. The IPC covers inventions in all technological fields in which inventions can be patented.

[2] By a *phrase* we mean an index unit consisting of two or more words, generated through either syntactic or statistical methods.

[3] Multi-class classification is the problem of classifying instances into more than two classes. Multi-label signifies that documents in this test set are associated with more than one class, and must be assigned a set of labels during classification.

[4] The notion of *aboutness* refers to the conceptual content expressed by a dependency triple. For a more detailed description, see Section 3.2.3.

and the extent to which the words in the phrases overlap with the unigrams. We also investigate which syntactic relations capture most information in the opinion of human annotators. Finally, we perform experiments to investigate if our findings are language-dependent. We will then draw some conclusions on what information is most valuable for improving automatic patent classification.

2. Background

2.1 Text Representations in Classification

Lewis (1992) was the first to investigate the use of phrases as index terms for text classification. He found that phrases generally suffer from data sparseness and may actually cause classification performance to deteriorate. These findings were confirmed by Apté, Damerau, and Weiss (1994). With the advent of increasing computational power and bigger data sets, however, the topic has been revisited in the last two decades (Bekkerman and Allan 2003).

In this section we will give an overview of the major findings in previous research on the use of statistical and syntactic phrases for text classification. Except when mentioned explicitly, all the classification experiments reported here were conducted using the Reuters-21578 data set, a well-known benchmark of 21,578 short newswire texts for multi-class classification into 118 categories (a document has an average of 1.24 class labels).

2.1.1 Combining Unigrams with Statistical Phrases. For an excellent overview of the work on using phrases done up to 2002, see Bekkerman and Allan (2003), and Tan, Wang, and Lee (2002).

Because they contain more specific information, one might think that phrases are more powerful features for text classification. There are two ways of using phrases as index terms: either index terms only or in combination with unigrams. All experimental results, however, show that using only phrases as index terms leads to a decrease in classification accuracy compared with the BOW baseline (Bekkerman and Allan 2003). Both Mladenic and Grobelnik (1998) and Fürnkranz (1998) showed that classifiers trained on combinations of unigrams and *n*-grams composed of at most three words performed better than classifiers that only use unigrams; no improvement was obtained when using larger *n*-grams. Because trigrams are sparser than bigrams, most of the subsequent research has focused on optimizing the combination of unigrams and bigrams using different feature selection techniques.

2.1.2 Feature Selection. Obviously, unigrams and bigrams overlap: Bigrams are pairs of unigrams. Caropreso, Matwin, and Sebastiani (2001) evaluated the relative importance of unigrams and bigrams in a classifier-independent study: Instead of determining the impact of features on the classification scores, they scored all unigrams and bigrams using conventional feature evaluation functions to find the features that are most representative for the document classes. For the Reuters-21578 data set, they found that many bigram features scored higher than unigram features. These (theoretical) findings were not confirmed in subsequent classification experiments, however. When the bigram/unigram ratio for a fixed number of features is changed to favor bigrams, classification performance tends to go down. It appears that the information in the bigrams does not turn the unigrams redundant.

Braga, Monard, and Matsubara (2009) used a Multinomial Naive Bayes classifier to investigate classification performance with unigrams and bigrams by comparing multiview classification (the results of two independent classifiers trained with unigram and bigram features are merged) with monoview classification (unigrams and bigrams are combined in a single feature set).[5] They found that there is little difference between the output of the mono- and multiview classifiers. In the multiview classifiers, the unigram and bigram classifiers make similar decisions in assigning labels, although the latter generally yielded lower confidence values. Consequently, in the merge the unigram and bigram classifiers affirm each other's decisions, which does not result in an overall improvement in classification accuracy. The authors suggest combining unigrams only with those bigrams for which it holds that the whole provides more information than the sum of the parts.

Tan, Wang, and Lee (2002) proposed selecting highly representative and meaningful bigrams based on the Mutual Information scores of the words in a bigram compared with the unigram class model. They selected only the top 2% of the bigrams as index terms, and found a significant improvement over their unigram baseline, which was low compared to state-of-the-art results. Bekkerman and Allan (2003) failed to improve over their unigram baseline when using similar selection criteria based on the distributional clustering of unigram models. Crawford, Koprinska, and Patrick (2004) were not able to improve e-mail classification when using the selection criteria proposed by Tan, Wang, and Lee.

2.1.3 Combining Unigrams with Syntactic Phrases. Lewis (1992) and Apté, Damerau, and Weiss (1994) were the first to investigate the impact of syntactic phrases[6] as features for text classification. Dumais et al. (1998) and Scott and Matwin (1999) did not observe a significant improvement in classification on the Reuters-21578 collection when noun phrases obtained with a shallow parser were used instead of unigrams. Moschitti and Basili (2004) found that neither words augmented with word sense information, nor syntactic phrases (acquired through shallow parsing) in combination with unigrams improved over the BOW baseline. Syntactic phrases appear to be even sparser than bigrams. Therefore, it is not surprising that most papers concluded that classifiers using only syntactic phrases perform worse than the baseline, except when the BOW baseline is low for that particular classification task (Mitra et al. 1997; Fürnkranz 1999).

Deep syntactic parsing is a computationally expensive process, but thanks to the increase in computational power it is now possible to use phrases acquired through deep syntactic parsing in classification tasks. Nastase, Sayyad, and Caropreso (2007) used Minipar to generate dependency triples that are combined with lemmatized and unlemmatized unigrams to classify the 10 most frequent classes in the Reuters-21578 data set. Their criterion for selecting triples as index terms is document frequency ≥ 2. The small improvement over the lemmatized unigram baseline was not statistically significant. Özgür and Güngör (2010, 2012) achieve small but significant improvements when combining unigrams with a subset of the dependency types from the Stanford parser on three different data sets, including the Reuters-21578 set. They find that separate pruning levels (based on the term frequency–inverse document frequency [TF-IDF] score of the index units) for the unigrams and syntactic phrases influence

[5] The difference between multiview and monoview classification corresponds to what is called *late* and *early fusion* in the pattern recognition literature.

[6] The concept "syntactic phrase" can be given several different interpretations, such as noun phrases, verb phrases, predicate structures, dependency triples, and so forth.

classification accuracy. Which dependency relations prove most relevant for a classification task depends greatly on the language use in the different data sets: The informal MiniNG data set (usenet posts) benefits a little from "simple" dependencies such as *part*, denoting a phrasal verb, for example *write down*, while classification in the more formal Reuters-21578 (newswire) and NSF (scientific abstracts) data sets is more improved by using dependencies on phrase and clause level (adjectival modifier, compound noun, prepositional attachment; and subject and object, respectively). The highest-ranking features for the NSF data set are compound noun (nn), adjectival modifier (amod), subject (subj), and object (obj), respectively. Furthermore, they observe that splitting up more generic relator types (such as *prep*) into different, more specific, subtypes increases the classification accuracy.

2.2 Patent Classification

It is not possible to draw far-reaching conclusions from previous research on patent classification, because there is no tradition of using a "standard" data set, and a standard split of patent corpora in a training and test set. Furthermore, there are differences between the various experiments in task definitions (mono-label versus multi-label classification); the granularity of the classification (depth in the IPC hierarchy); and the choices of (sub)sets of data. Fall and Benzineb (2002) give an overview of the work done in patent classification research up to 2002 and of the commercial patent classification systems available; see Benzineb and Guyot (2011) for a general introduction to patent classification.

Larkey (1999) was the first to present a fully automated patent classification system, but she did not report her overall accuracy results. Larkey (1998) used a combination of weighted words and noun phrases as index terms to classify a subset of the USPTO database, but found no improvement over a BOW baseline. The weights were calculated as follows: Frequency of a word or phrase in a particular section times the manually assigned weight (importance) given to that section. The weights for each word or phrase were then summed across sections. Term selection was based on a threshold for these weights.

Krier and Zaccà (2002) organized a comparative study of various academic and commercial systems for patent classification for a common data set. In this informal benchmark Koster, Seutter, and Beney (2001) achieved the best results, using the Balanced Winnow algorithm with a word-only text representation. Classification is performed for 44 or 549 categories (which correspond to different levels of depth in the then used version of the IPC), with around 78% and 68% precision at 100% recall, respectively.

Fall et al. (2003) introduced the EPO-alpha data set, attempting to create a common benchmark for patent classification. Using only words as index terms, they tested different classification algorithms and found that SVM outperform Naive Bayes, k-NN, SNoW, and decision-based classifiers. They achieved P@3-scores[7] of 73% and 59% on 114 classes and 451 subclasses, respectively. They also found that when using only the first 300 words from the abstract, claims, and description sections, classification accuracy is increased compared with using the complete sections. The same data set was later used by Koster and Seutter (2003), who experimented with a combined

[7] Precision at rank 3 (P@3) signifies the percentage correct labels in the first three labels by the classifier to a given document.

representation of words and phrases consisting of head-modifier pairs.[8] They found that head-modifier pairs could not improve on the BOW-baseline: The phrases were too sparse to have much impact on the classification process.

Starting in 2009, the IRF[9] has organized CLEF-IP patent classification tracks in an attempt to bridge the gap between academic research and the patent industry. For this purpose the IRF has put a lot of effort into providing very large patent data sets,[10] which have enabled academic researchers to train their algorithms on real-life data. In the CLEF-IP 2010 classification track the best results were achieved by Guyot, Benzineb, and Falquet (2010). Using the Balanced Winnow algorithm, they achieved a P@1-score of 83%, while classifying on subclass level. They used a combination of words and statistical phrases (collocations of variable length extracted from the corpus) as index terms and used all available documents (in English, French, and German) in the corpus as training data. In the same competition, Derieux et al. (2010) came second (in terms of P@1). They also used a mixed document representation of both single words and longer phrases, which had been extracted from the corpus by counting word co-occurrences. Verberne, Vogel, and D’hondt (2010) and Beney (2010) experimented with a combined representation of words and syntactic phrases derived from an English and French syntactic parser, respectively. They both found that adding syntactic phrases to words improves classification accuracy slightly. Beney (2010) remarks that this improvement may be language-dependent. As a follow-up, Koster et al. (2011) investigated the added value of syntactic phrases. They found that **attributive phrases**, that is, combinations of adjective or nouns with nouns, were by far the most important syntactic phrases for patent classification. On a subset of the CLEF-IP 2010 corpus[11] they also found a small, but significant, improvement when adding dependency triples to words.

3. Experimental Set-up

In this article, we investigate the relative contributions of different types of terms to the performance of patent classification. We use four different types of terms, namely, lemmatized unigrams, lemmatized bigrams (see Section 3.2.1), lemmatized dependency triples obtained with the Stanford parser (see Section 3.2.2), and lemmatized dependency triples obtained with the AEGIR parser (see Section 3.2.3). We will leave term (feature) selection to the preprocessing module of the Linguistic Classification System (LCS) which we used for all experiments (see Section 3.3). We will analyze the relation between unigrams and phrases in the class profiles in some detail, however (see Sections 4.2 and 4.3).

3.1 Data Selection

We conducted classification experiments on a collection of patent documents obtained from the CLEF-IP 2010 corpus,[12] which is a subset of the larger MAREC patent collection. The corpus contains 2.6 million patent documents, which roughly correspond

[8] Head-modifier pairs were derived from the syntactic analysis output of the EP4IR syntactic parser.

[9] Information Retrieval Facility, see `www.irf.com`.

[10] The CLEF-IP 2009, CLEF-IP 2010, and CLEF-IP 2011 data sets can be obtained through the IRF. The more recent data sets subsume the older sets.

[11] The same data set as will be used in this article. For a more detailed description, see Section 3.1.

[12] This test collection is available through the IRF (`http://www.ir-facility.org/collection`).

to 1.3 million individual patents, published between 1985 and 2001.[13] The documents in the collection are encoded in a customized XML format and may include text in English, French, and German. In addition to the standard sections of a patent document (title, abstract, claims, and description section), the documents also include meta-information on inventor, date of application, assignee, and so forth. Because our focus lies on text representation, we did not include any of the meta-data in our document representations.

The most informative sections of a patent document are generally considered to be the title, the abstract, and the beginning of the description (Benzineb and Guyot 2011). Verberne and D'hondt (2011) showed that using both the description and the abstract gives a small, but significant, improvement in classification results on the CLEF-IP 2011 corpus, compared with classification on abstracts only. The effort involved in parsing the descriptions is considerable, however: Because of the long sentences and the dense word use, a parser will have much more difficulty in processing text from the description section than from the abstracts. The titles of the patent documents also pose a parsing problem: These are generally short noun phrases that contain ambiguous PP-attachments that are impossible to disambiguate without any domain knowledge. This leads to incorrect syntactic analyses and, consequently, noisy dependency triple features. Because we are interested in comparing classification results for different text representations, and not in comparing results for different sections, we opted to use only the abstract sections of the patent document in the current article.

From the corpus, we extracted all files that contain both an abstract in English and at least one IPC class[14] in the *<classification-ipcr>* field. We extracted the IPC classes on the document level; this means that we did not include the documents where the IPC class is in a separate file than the English abstract. In total, there were 121 different classes in our data set. Most documents have been assigned one to three different IPC classes (on class level). On average, a patent abstract in our data set has 2.12 class labels. Previous cross-validation experiments on the same document set showed very little variation (standard deviation $< 0.3\%$) between the classification accuracies in different training-test splits (Verberne, Vogel, and D'hondt 2010). We therefore decided to use only one training and test set split.[15]

The final data set contained 532,264 abstracts, divided into two sets: (1) a training set (425,811 documents) and (2) a test set (106,453 documents). The distribution of the data over the classes is in accordance with the Pareto Principle: 20% of the classes cover 80% of the data, and 80% of the classes comprise only 20% of the data.

3.2 Data Preprocessing

Preprocessing included cleaning up character conversion errors like Expression (1) and removing claims and images references (Expression (2)) and list references

[13] Note the difference between a patent and a patent document: A patent is not a physical document itself, but a name for a group of patent documents that have the same patent ID number.

[14] For our classification experiments we use the codes on the class level in the IPC8 classification.

[15] The data split was performed using a perl script that randomly shuffles the documents and puts them into a train set and test set, while ensuring that the class distribution of the examples in the train set approximates that of the whole corpus. It can be downloaded as part of the LCS distribution.

(Expression (3)) from the original texts. This was done automatically, using the following regular expressions (based on Parapatics and Dittenbach 2009):

```
s/;gt&/>/g                                                    (1)

s/(\([ ]*[0-9][0-9a-z,.; ]*\))//g                             (2)

s/(\([ ]*[A-Za-z]\))//g                                       (3)
```

We then used a perl script to divide the running text into sentences, by splitting on end-of-sentence punctuation such as question marks and full stops. In order to minimize incorrect splitting, the perl script was supplied with a list of common English abbreviations and a list containing abbreviations and acronyms that occur frequently in technical texts, derived from the Specialist lexicon.[16]

3.2.1 Unigrams and Bigrams. The sentences in the abstract documents were converted to single words by splitting on whitespaces and removing punctuation. The words were then lemmatized using the AEGIR lexicon. Bigrams were created through a similar procedure. We did not create bigrams that spanned sentence boundaries. This resulted in approximately 60 million unigram and bigram tokens for the present corpus.

3.2.2 Stanford. The Stanford parser is a broad-coverage natural language parser that is trained on newswire text, for which it achieves state-of-the-art performance. The parser has not been optimized/retrained for the patent domain.[17] In spite of the technical difficulties (Parapatics and Dittenbach 2009) and loss of linguistic accuracy for patent texts reported in Mille and Wanner (2008), most patent processing systems that use linguistic phrases use the Stanford parser because its dependency scheme has a number of properties that are valuable for Text Mining purposes (de Marneffe and Manning 2008). The Stanford parser collapsed typed dependency model has a set of 55 different syntactic relators to capture **semantically contentful** relations between words. For example, the sentence *The system will consist of four separate modules* is analyzed into the following set of dependency triples in the Stanford representation:

```
det(system-2, The-1)
nsubj(consist-4, system-2)
aux(consist-4, will-3)
root(ROOT-0, consist-4)
num(modules-8, four-6)
amod(modules-8, separate-7)
prep_of(consist-4, modules-8)
```

The Stanford parser was compiled with a maximum memory heap of 1.2 GB. Sentences longer than 100 words were automatically skipped. Combined with failed parses this led to a 1.2% loss of parser output on the complete data set. Parsing the

[16] The lexicon can be downloaded at http://lexsrv3.nlm.nih.gov/Specialist/Summary/lexicon.html.

[17] For retraining a parser, a substantial amount of annotated data (in the form of syntactically annotated dependency trees) is needed. Creating such annotations is a very expensive task and beyond the scope of this article.

Table 1
Impact of lemmatization on the different text types in the training set (80% of the corpus).

	# tokens	# types (terms)			token/type (lem.)
		raw	lemmatized	gain	
unigram	48,898,738	160,424	142,396	1.12	343.39
bigram	48,473,756	3,836,212	3,119,422	1.23	15.54
Stanford	35,772,003	8,750,839	7,430,397	1.18	4.81
AEGIR	31,004,525	–	5,096,918	–	6.08

entire set of abstracts took 1.5 weeks on a computer cluster consisting of 60 2.4GHz cores with 4 GB RAM per core. The resulting dependency triples were stripped of the word indexes and then lemmatized using the AEGIR lexicon.

3.2.3 AEGIR. AEGIR[18] is a dependency parser that was designed specifically for robust parsing of technical texts. It combines a hand-crafted grammar with an extensive word-form lexicon. The parser lexicon was compiled from different technical terminologies, such as the SPECIALIST lexicon and the UMLS.[19] The AEGIR parser aims to capture the *aboutness* of sentences. Rather than outputting extensive linguistic detail on the syntactic structure of the input sentence as in the Stanford parser, AEGIR returns only the bare syntactic–semantic structure of the sentence. During the parsing process, it effectively performs normalization at various levels, such as typography (for example, upper and lower case, spacing), spelling (for example, British and American English, hyphenation), morphology (lemmatization of word forms), and syntax (standardization of the word order and transforming passive structures into active ones).

The AEGIR parser uses only eight syntactic relators and returns fewer unique triples than the Stanford parser. The parser is currently still under development; for this article we used the version AEGIR v.1.7.5. The parser was constrained to a time limit of maximum three seconds per sentence. This caused a loss of 0.7% of parser output on the complete data set. Parsing the entire set of abstracts took slightly less than a week on the computer cluster described above. The AEGIR parser has several output formats, among which its own dependency format. The example sentence used to illustrate the output of the Stanford parser is analyzed as follows:

```
[system,SUBJ,consist]
[consist,PREPof,module]
[module,ATTR,separate]
[module,QUANT,four]
```

3.2.4 Lemmatization. Table 1 shows the impact of lemmatization (using the AEGIR lexicon) on the distribution of terms for the different text representations. Lemmatization

[18] AEGIR stands for *Accurate English Grammar for Information Retrieval.* Using the AGFL compiler (found at `http://www.agfl.cs.ru.nl/`) this grammar can be compiled into an operational parser. The grammar is not freely distributed.

[19] The Unified Medical Language System contains a widely-used terminology of the biomedical domain and can be downloaded at `http://www.nlm.nih.gov/research/umls/`.

and stemming are standard approaches to decreasing the sparsity of features; stemming is more aggressive than lemmatization. Ozgür and Güngör (2009) showed that—when using only words as index terms—stemming (with the Porter Stemmer) appears to improve performance; stemming dependency triples did not improve performance, however.

We opted to use a less aggressive form of generalization: Lemmatizing the word forms. We found that the bigrams gain[20] most by lemmatizing the word forms, resulting in a higher token/type ratio. From Table 1 it can be seen that there are fewer triple tokens than bigram tokens: Whereas all the (high-frequency) function words are kept in the bigram representations, both dependency formats discard some function words in their parser output. For example, the AEGIR parser does not create triples for auxiliary verbs, and in both dependency formats, the prepositions become part of the relator. Consequently, the parsers will output fewer but more variable tokens, which results in lower token/type ratios and a lower impact of lemmatization.

3.3 Classification Experiments

The classification experiments were carried out within the framework of the LCS (Koster, Seutter, and Beney 2003). The LCS has been developed for the purpose of comparing different text representations. Currently, three classifier algorithms are available: Naive Bayes, Balanced Winnow (Dagan, Karov, and Roth 1997), and SVM-light (Joachims 1999). Verberne, Vogel, and D'hondt (2010) found that Balanced Winnow and SVM-light yield comparable classification accuracy scores for patent texts on a similar data set, but that Balanced Winnow is much faster than SVM-light for classification problems with a large number of classes. The Naive Bayes classifier yielded a lower accuracy. We therefore only used the Balanced Winnow algorithm for our classification experiments, which were run with the following LCS configuration, based on tuning experiments on the same data by Koster et al. (2011):

- Global term selection (GTS): Document frequency minimum is 2, term frequency minimum is 3. Although initial term selection is necessary when dealing with such a large corpus, we deliberately aimed at keeping as many of the sparse phrasal terms as possible.
- Local term selection (LTS): Simple Chi Square (Galavotti, Sebastiani, and Simi 2000). We used the LCS option to automatically select the most representative terms for every class, with a hard maximum of 10,000 terms per class.[21]
- After LTS the selected terms of all classes are *aggregated* into one combined term vocabulary, which is used as the starting point for training the individual classes (see Table 3).

[20] By "gain" we mean the decrease in number of types for the lemmatized forms compared to the non-lemmatized forms, which will result in higher corresponding token/type ratios.

[21] Increasing the cut-off to 100,000 terms resulted in a small increase in accuracy (F1 values) for the combined representations, mostly for the larger classes. Because the patent domain has a large lexical variety, a large amount of low-frequency terms in the tail of the term distribution can have a large impact on the accuracy scores. Because we are more interested in the relative gains between different text representations and the corresponding top terms in the class profiles than in achieving maximum classification scores, we opted to use only 10,000 terms for efficiency reasons.

Table 2
Impact of global term selection (GTS) criteria on the different text types in the training set (80% of the corpus).

	total # of terms	# of terms selected in GTS	% of terms removed in GTS
unigram	142,396	58,423[22]	58.97
bigram	3,119,422	1,115,170	64.25
stanford	7,430,397	1,618,478	78.22
AEGIR	5,096,918	1,312,715	74.24

- Term strength calculation: LTC algorithm (Salton and Buckley 1988) which is an extension of the TF–IDF measure.
- Training method: Ensemble learning based on one-versus-rest binary classifiers.
- Winnow configuration: We performed tuning experiments for the Winnow parameters on a development set of around 100,000 documents. We arrived at using the same setting as Koster et al. (2011), namely, $\alpha = 1.02$, $\beta = 0.98$, $\theta+ = 2.0$, $\theta- = 0.5$, with a maximum of 10 training iterations.
- For each document the LCS returns a ranked list of all possible labels and the attendant confidence scores. If the score assigned is higher than a predetermined threshold, the document is assigned that category. The Winnow algorithm has a default (natural) threshold equal to one. We configured the LCS to return a minimum of one label (with the highest score, even if it is lower than the threshold) and a maximum of four labels for each document.
- The classification quality was determined by calculating the Precision, Recall, and F1 measures per document/class combination (see, e.g., Koster, Seutter, and Beney 2003), on the document level (micro-averaged scores).

Table 2 shows the impact of our global term selection criteria for the different text representations. This first feature reduction step is category-independent: The features are discarded on the basis of the term and document frequencies over the corpus, disregarding their distributions for the specific categories. We can see that the token/type ratio of Table 1 is mirrored in this table: The sparsest syntactic phrases lose most terms. Although the Stanford parser output is the sparsest text representation, it has the largest pool of terms to select from at the end of the GTS process.

The impact of the second feature reduction phase is shown in Table 3. During local term selection, the LCS finds the most representative terms for each class by selecting the terms whose distributions in the sets of positive and negative training examples for that class are maximally different from the general term distribution. We can see that in the combined runs only around 50% of the selectable unigrams (after GTS) are

[22] For the BOW baseline, the GTS criteria resulted in a too small term set that could then be used as a starting point for the local term selection process for the individual classes. In such cases, the LCS has a back-off mechanism that automatically (re)selects terms that were initially discarded during GTS. In other words, the baseline classifier used terms that do not comply with the criteria in the GTS as described in the text. In the combination runs, enough terms remained after GTS and no unigrams or phrases that did not match the GTS criteria were selected.

Table 3
Impact of local term selection (LTS) criteria in the training set (80% of the corpus).

		# of terms after GTS	# of terms after LTS
baseline	uni	58,423	69,476
unigrams + bigrams	uni	58,423	23,753
	bi	1,115,170	300,826
unigrams + stanford triples	uni	58,423	26,630
	stanford	1,618,478	424,204
unigrams + AEGIR triples	uni	58,423	29,348
	AEGIR	1,312,715	409,851

Table 4
Classification results on CLEF-IP 2010 English abstracts, with ranges for a 95% confidence interval. Bold figures indicate the best results obtained with the five classifiers. (P: Precision; R: Recall, F1: F1-score).

	P	R	F1
weighted random guessing	*6.09% ± 0.14*	*6.04% ± 0.14*	*6.06% ± 0.14*
unigrams	76.27% ± 0.26	66.13% ± 0.28	70.84% ± 0.27
unigrams + bigrams	79.00% ± 0.24	70.19% ± 0.27	74.34% ± 0.26
unigrams + Stanford triples	78.35% ± 0.25	69.57% ± 0.28	73.70% ± 0.26
unigrams + AEGIR triples	78.51% ± 0.25	69.18% ± 0.28	73.55% ± 0.26
all representations	**79.51% ± 0.24**	**71.11% ± 0.27**	**75.08% ± 0.26**

selected as features during LTS. This means that the phrases replace at least a part of the information contained in the possible unigrams.

4. Results and Discussion

4.1 Classification Accuracy

Table 4 shows the micro-averages of Precision, Recall, and F1 for five classification experiments with different document representations. To give an idea of the complexity of the task we have included a random guessing baseline in the first row.[23] We found that extending a unigram representation with statistical and/or linguistic phrases gives a significant improvement in classification accuracy over the unigram baseline. The best-performing classifier is the one that combines all four text representations. When adding only type of phrase to unigrams, the unigrams + bigrams combination is significantly better than the combinations with syntactic phrases. Combining all four representations boosts recall, but has less impact on precision.

[23] The script used to calculate the baseline can be downloaded at `http://lands.let.ru.nl/~dhondt/`. We used a weighted randomization that takes the category label distributions and label frequency distributions into account.

Table 5
Penetration of the bigrams and triples in the B60 class profiles (in % of terms at given rank).

		rnk10	rnk20	rnk50	rnk100	rnk1000
bigrams		3.0	4.0	48.0	45.0	70.5
stanford		0.0	1.0	24.0	26.0	48.0
AEGIR		0.0	0.5	20.0	25.0	44.9
all representations	bigrams	2.0	2.0	34.0	36.0	43.2
	stanford	0.0	0.0	4.0	6.0	13.0
	AEGIR	0.0	0.0	2.0	4.0	18.0

The results are similar to Özgür and Güngör’s (2012) findings for scientific abstracts: Adding phrases to unigrams can significantly improve classification. The text in the patent corpus is vastly different from the newswire text in the Reuters corpus. Like scientific abstracts, patents are full of jargon and terminology, often expressed in multi-word units, which might favor phrasal representations. Moreover, the innovative concepts in a patent are sometimes described in generalized terms combined with some specifier (to ensure larger legal scope). For example, a *hose* might be referred to as a *watering device*. The term *hose* can be captured with a unigram representation, but the multi-word expression cannot. The difference with the results on the Reuters-21578 data set (discussed in Section 2.1.1), however, may not completely be due to genre differences: Bekkerman and Allan (2003) remark that the unigram baseline for the Reuters-21578 task is difficult to improve upon, because in that data set a few keywords are enough to distinguish between the categories.

4.2 Unigram versus Phrases

In this section we investigate whether adding phrases suppresses, complements, or changes unigram selection. To examine the impact of phrases in the classification process, we analyzed the class profiles[24] of two large classes (H04 – Electric Communication Technique; and H01 – Basic electric elements) that show significant improvements in both Precision and Recall[25] for the bigram classifier compared with the unigram baseline. We look at (1) the overlap of the single words in the class profiles of the unigram and combined representations; and (2) the overlap of the single words and the words that make up the phrases (hereafter referred to as **parts**) within the class profile of one text representation.

4.2.1 Overlap of Unigrams. The class profiles in the baseline unigram classifier contained far fewer terms (< 20%) than the profiles in the classifiers that combine unigrams and phrases. This could be expected from the data in tables 2 and 3.

Unigrams are the highest ranked[26] features in the combined representation class profiles (see Table 5). Furthermore, words that are important terms for unigram classification also rank high in the combined class profiles: On average, there is an 80%

[24] A class profile is the model built by the LCS classifier for a class during training. It consists of a ranked list of terms that contribute most to distinguishing members from a class from all other classes.

[25] H04: P: + 3.09%; R: + 1.83%; H01: P: + 3.61%; R: + 5.14%.

[26] The rank of a term is based on the decreasing order of mass assigned to that term in the class profile. (See Section 4.2.2.)

overlap of the top 1,000 most important words in unigram and combined representation class profiles. This decreases to 75% when looking at the 5,000 most important words. This shows that the classifier tends to select mostly the same words as important terms for the different text representation combinations. The relative ranking of the words is very similar in the class profiles of all the text representations. Thus, adding phrases to unigrams does not result in replacing the most important unigrams for a particular class and the improvements in classification accuracy must derive from the additional information in the selected phrases.

4.2.2 Overlap of Single Words and Parts of Bigrams. Like Caropreso, Matwin, and Sebastiani (2001), we investigated to what extent the parts of the high-ranked phrases overlap with words in the unigrams + bigrams class profile. We first looked at the lexical overlap of the words and the parts of the bigrams in the H01 unigrams + bigrams class profile. Interestingly, we found a relatively low overlap between the words and the parts of the phrases: For the 20 most important bigrams, only 11 of the 32 unique parts of the bigrams overlap with the 100 most important single word terms; in the complete class profile only 56% of the 10,387 parts of the bigrams overlap with the 9,064 words in the class profile. This means that a large part of the bigrams contains complementary information not present in the unigrams in the class profile.

To gain a deeper insight into the relationship between the bigrams and their parts, we also looked at the **mass** of the different terms in the class profiles. The mass of a term for a certain class is the product of its TF–IDF score and its Winnow weight for that class; ″mass″ provides an estimate of how much a term contributes to the score of documents for a particular class. We can divide the terms into three main categories:

(a) mass(partA) $\geq$ mass(partB) $\geq$ mass(bigram);

(b) mass(partA) $\geq$ mass(bigram) $>$ mass(partB);

(c) mass(bigram) $>$ mass(partA) $\geq$ mass(partB).

We note that 50% of the top 1,000 highest ranked bigrams fall within category (b) and typically consist of one part with high mass accompanied by a part with a low mass, which can be a function word (for example *a_transmitter*), or a general term (for example, *device* in *optical_device*). The highest ranked bigrams can be found in category (a) where two highly informative words are combined to form very specific concepts, for example, *fuel_cell*. These are specifications of a more general concept that is typical for that class in the corpus. The bigrams in this category are similar to those investigated by Caropreso, Matwin, and Sebastiani (2001) and Tan, Wang, and Lee (2002). Though highly ranked, they only make up a small subset (22%) of the important bigram features.

The bigrams in category (c) (27%) are typically made up from low-ranked single words, such as *mobile_station*. Interestingly, most bigram parts in this subset do not occur as word terms in the unigram and bigram class profiles, but occur in the negative class profiles (a selection of terms that are considered to describe anything *but* that particular class). The complementary information of bigram phrases (compared to unigrams) is contained in this set of bigrams.

4.3 Statistical versus Linguistic Phrases

Results in Section 4.1 indicate that bigrams are most important additional features, but the experiment combining all four representations showed that dependency triples do complement bigrams. In this section we examine what information is captured by the different phrases and how this accounts for the differences in classification accuracy.

4.3.1 Class Profile Analysis. We first examined the differences between the statistical phrases and the two types of linguistic phrases to discover what information contained in the bigrams leads to better classification results. We performed our analysis on the different class profiles of B60 ("Vehicles in general"), a medium-sized class, which most clearly shows the advantage of the bigram classifier compared to the classifiers with linguistic phrases.[27]

All four class profiles with phrases contain roughly the same set of unigrams (between 78% to 91% overlap) that occur quite high in the corresponding unigram class profile. The AEGIR class profile contains 10% more unigrams than the other combined representation class profiles; these are mainly words that appear in the negative class profile of the corresponding unigram classifier. As in class H01, the relative position of the words remains the same. The absolute position of the words in the list, however, does change: Caropreso, Matwin, and Sebastiani (2001) introduced a measure for the effectiveness of phrases as terms, called the **penetration**, that is, the percentage of phrases in the top k terms when classifying with both words and phrases.

Comparing the penetration levels at the various ranks for the different classifiers, we can see that the classification results correspond with the tendency of a classifier to select phrases in the top k terms. Interestingly, we see a large disparity in the phrasal features that are selected by the combination classifier. The preference for bigrams is mirrored by the penetration levels of the unigrams + bigrams classifier which has selected more bigrams at higher ranks in the class profile than the classifiers with the linguistic phrases. This is in line with the findings of Caropreso, Matwin, and Sebastiani (2001) that penetration levels are a reasonable way to compute the contribution of n-grams to the quality of a feature set. On average, the linguistic phrases have much smaller weight in the class profiles than the bigrams and, consequently, are likely to have a smaller impact during the classification process. For the combination run, however, it seems that a long tail of small-impact features does improve classification accuracy.

Linguistic analysis of the top 100 phrases in the profiles of class B60 shows that all classifiers select similar types of phrases. We manually annotated the bigrams with the correct syntactic dependencies (in the Stanford collapsed typed dependency format) and compared these with the syntactic relations expressed in the linguistic phrases. The results are summarized in Table 6.

It appears that noun phrases and compounds such as *circuit board* and *electric device* are by far the most important terms in the class profiles. Interestingly, phrases that contain a determiner relation (e.g., *the device*) are deemed equally important in all four different class profiles. It is unlikely that this is a semantic effect, that is, that the determiner relation provides additional semantic information to the nouns in the phrases, but rather it seems an artefact of the abundance of noun phrases which occur in patent texts. We also looked into the lexical overlap between the parts of the different types of phrases. We found that the selected phrases encode almost exactly the same information in all three representations: There is an 80% overlap between the parts of

[27] Precision is 77.34% for unigrams+bigrams, 75.67% for unigrams+Stanford, 73.47% for unigrams+AEGIR, and 77.38% for unigrams+bigrams+Stanford+AEGIR. The Recall scores are essentially equal for all three, that is, 68.81%, 68.38%, 69.7%, and 70.18%, respectively.

Table 6
Distribution of the top 100 statistical and syntactic phrases in the B60 class profiles.

grammatical relation	bigrams	stanford	AEGIR	combination
noun–noun compounds	41	48	62[28]	44[28]
adjectival modifier	11	8		
determiner	34	28	27	41
subject	6	4	6	9
prepositions	2	4	1	2
<other>	7	8	4	4

the top 100 most important phrases. This decreases only to 75% when looking at the 10,000 most important phrases.

Given that the class profiles select the same set of words and contain phrases with a high lexical overlap, therefore, how do we explain the marked differences in classification accuracy between the three different representations? These must stem from the different combinations of the words in the phrasal features. To examine in detail how the features created through the different text represenations differ, we conducted a feature quality assessment experiment against a manually created reference set.

4.3.2 Human Quality Assessment Experiment. To gain more insight in the syntactic and semantic relations that are considered most informative by humans, we conducted an experiment in which we asked human annotators to select the five to ten most informative phrases[29] for 15 sentences taken at random from documents in the three largest classes in the corpus. We then compiled a reference set consisting of 70 phrases (4.6 phrases per sentence) which were considered as "informative" by at least three out of four annotators. Of these, 57 phrases were noun–noun compounds and 11 were combinations of an adjectival modifier with a noun. None of the annotators selected phrases containing determiners.

We created bigrams from the input and extracted head–modifier pairs[30] from the parser output for the sentences in the test set. We then compared the overlap of the generated phrases with the reference phrases. We found that bigrams overlap with 53 of the 70 reference phrases; Stanford triples overlap with 62 phrases and AEGIR triples overlap with 57 phrases. Although three data points are not enough to compute a formal measure, it is interesting to note the correspondence with the number of terms kept for the three text representations after Local Term Selection (see Table 3). The fact that the text representation with the smallest number of terms after LTC and with the smallest overlap with "contentful" phrases in a text as indicated by human annotators still yields the best classification performance suggests that not all "contentful" phrases are important or useful for the task of classifying that text. This finding is reminiscent of the fact that the "optimal" summary of a text is dependent on the goal with which the summary was produced (Nenkova and McKeown 2011).

Only 15% of the phrases extracted by the human annotators contain word combinations that have long-distance dependencies in the original sentences. This suggests

[28] As mentioned in Section 3.2.3 the AEGIR parser uses a more condensed dependency output format. The Stanford's *nn* and *amod* are collapsed into the *attributive* (ATTR) relation.

[29] "Phrase" was defined as a combination of two words that both occur in the sentence, irrespective of the order in which they occur in the sentence.

[30] Head–modifier pairs are syntactic triples that are stripped of their grammatical relations.

that the most meaningful phrases are expressed in local dependencies, that is, adjacent words. Consequently, syntactic analysis aimed at discovering meaning expressed by long-distance dependencies can only make a small contribution. A further analysis of the phrases showed that the smaller coverage of the bigrams is due to the fact that some of the relevant noun–noun combinations are missed because function words, typically determiners or prepositions, occur between the nouns. For example, the annotators constructed the reference phrase *rotation axis* for the noun phrase *the rotation of the second axis*. This reference phrase cannot be captured by the bigram representation. When intervening function words are removed from the sentences, the coverage of the resulting bigrams on the reference set rises[31] to 59 phrases (more than AEGIR, and almost as many as Stanford). Despite the fact that generating more phrases does not necessarily lead to better classification performance, we intend to use bigrams stripped of function words as additional terms for patent classification in future experiments.

The analysis also revealed an indication why syntactic phrases may lead to inferior classification results: Both syntactic parsers consistently fail to find the correct structural analysis of the long and complex noun phrases such as *an implantable, inflatable dual chamber shape retention tissue expander*, which are frequent in patent texts. Phrases like this contain many compounds in an otherwise complex syntactic structure, namely

[an [implantable, inflatable [[dual chamber] [shape retention] [tissue expander]]]].

For a parser it is impossible to parse this correctly without knowing which word sequences are actually compounds. That knowledge might be gleaned from the frequency with which sequences of nouns and adjectives occur in a given domain. For the time being, the Stanford parser (and the AEGIR parser, to a lesser extent) will parse any noun phrase by attaching the individual words to the right-most head noun, resulting in the following analysis:

[an [implantable, [inflatable [dual [chamber [shape [retention [tissue expander]]]]]]].

This effectively destroys many of the noun–noun compounds, which are the most important features for patent classification (see Table 6). Bigrams are less prone to this type of "error."

These findings are confirmed when looking at the overlap of the word combinations: Although there is high lexical overlap between the phrases of the different representations (80% overlap of the parts of phrases in Section 4.3.1), the overlap of the word combinations that make up the phrases is much lower: Only 33% of the top 1,000 phrases are common between all three representations.

4.4 Stanford versus AEGIR Triples

The performance with the unigrams + Stanford triples is not significantly different from the combination with AEGIR triples. Because the AEGIR triples are slightly less sparse (see Table 1), we expected that these would have an advantage over Stanford triples. Most of the normalization processes that make the AEGIR triples less sparse concern syntactic variation on the clause level, however. But as was shown in Section 4.3,

[31] This result is language-dependent: English has a fairly rigid phrase-internal word order but for a more synthetic language with a more variable word order, like Russian, bigram coverage might suffer from the variation in the surface form.

Table 7
Classification results on CLEF-IP 2010 French and German abstracts, with ranges for 95% confidence intervals.

		P	R	F1
French	unigrams	70.65% ± 0.68	61.40% ± 0.73	65.70% ± 0.70
	unigrams + bigrams	**72.31% ± 0.67**	**62.58% ± 0.72**	**67.09% ± 0.69**
German	unigrams	**76.44% ± 0.34**	**65.82% ± 0.38**	**70.73% ± 0.37**
	unigrams + bigrams	76.39% ± 0.34	65.41% ± 0.38	70.47% ± 0.37

the most important terms for classification in the patent domain are found in the noun phrase, where Stanford and AEGIR perform similar syntactic analyses. Although Stanford's dependency scheme is more detailed (see Table 6), the noun-phrase internal dependencies in the Stanford parser map practically one-to-one onto AEGIR's set of relators, resulting in very similar dependency triple features for classification. Consequently, there is no normalization gain in using the AEGIR dependency format to describe the internal structure of the noun phrases.

4.5 Comparison with French and German Patent Classification

We found that phrases contribute to improving classification on English patent abstracts. The improvement might be language-dependent, however, because compounds are treated differently in different languages. A compounding language like German might benefit less from using phrases than English. To estimate the generalizability of our findings, we conducted additional experiments in which we compared the impact of adding bigrams to unigrams for both French and German.

Using the same methods described in sections 3.1 and 3.2, we extracted and processed all French and German abstracts from the CLEF-IP 2010 corpus, resulting in two new data sets that contained 86,464 and 294,482 documents, respectively (Table 7). Both data sets contained the same set of 121 labels and had label distributions similar to the English data set. The sentencing script was updated with the most common French and German abbreviations to minimize incorrect sentence splitting. The resulting sentences were then tagged using the French and German versions of the TreeTagger.[32] From the tagged output, we extracted the lemmas and used these to construct unigrams and bigrams for both languages. We ran the experiments with the LCS using the settings reported in Section 3.3.

The results show a much smaller but still significant improvement for using bigrams when classifying French patent abstracts and even a deterioration for German. Due to the difference in size between the English and French data set it is difficult to draw hard conclusions on which language benefits most from adding bigrams. It is clear, however, that our findings are not generalizable to German (and probably other compounding languages).

5. Conclusion

In this article we have examined the usefulness of statistical and linguistic phrases for patent classification. Similar to Özgür and Güngör's (2010) results for scientific

[32] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/.

abstracts, we found that adding phrases to unigrams significantly improves classification results for English. Of the three types of phrases examined in this article, bigrams have the most impact, both in the experiment that combined all four text representations, and in combination with unigrams only.

The abundance of compounds in the terminology-rich language of the patent domain results in a relatively high importance for the phrases. The top phrases across the different representations were mostly noun–noun compounds (for example *watering device*), followed by phrases containing a determiner relation (for example *the module*) and adjective modifier phrases (for example *separate module*).

The information in the phrases and unigrams overlaps to a large extent: Most of the phrases consist of words that are important unigram features in the combined profile and that also appear in the corresponding unigram class profile. When examining the H01 class profiles, however, we found that 27% of the selected phrases contain words that were not selected in the unigram profile (see Section 4.2.2).

When comparing the impact of features created from the output of the aboutness-based AEGIR parser with those from the Stanford parser, we found the latter resulted in slightly (but not significantly) better classification results. AEGIR's normalization features are not advantageous (compared with Stanford) in creating noun-phrase internal triples, which are the most informative features for patent classification.

The parsers were not specifically trained for the patent domain and both experienced problems with long, complex noun phrases consisting of sequences of words that can function as adjective/adverb or noun and that are not interrupted by function words that clarify the syntactic structure. The right-headed bias of both syntactic parsers caused problems in analyzing those constructions, yielding erroneous and variable data. As a consequence, parsers may miss potentially relevant noun–noun compounds and noun phrases with adjectival modifiers. Because of the highly idiosyncratic nature of the terminology used in the patent domain, it is not evident whether this problem can be solved by giving a parser access to information about the frequency with which specific noun–noun, adjective–noun, and adjective/adverb–adjective pairs occur in technical texts. Bigrams, on the other hand, are less variable (as seen in Table 1) and therefore yield better classification results. This is the more important point because the dependency relations marked as important for understanding a sentence by the human annotators consist mainly of pairs of adjacent words.

We also performed additional experiments to examine the generalizability of our findings for French and German: As could be expected, compounding languages like German which express complex concepts in "one word" do not gain from using bigrams.

In line with Bekkerman and Allan (2003) we can conclude that with the large quantities of text available today, the role of phrases as features in text classification must be reconsidered. For the automated classification of English patents at least, adding phrases and more specifically bigrams significantly improves classification accuracy.

References

Apté, Chidanand, Fred Damerau, and Sholom Weiss. 1994. Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems*, 12(3):233–251.

Bekkerman, Ron and John Allan. 2003. Using bigrams in text categorization. Technical Report IR-408, Center of Intelligent Information Retrieval, University of Massachusetts, Amherst.

Beney, Jean. 2010. LCI-INSA linguistic experiment for CLEF-IP classification track. In *Proceedings of the Conference on Multilingual and Multimodal Information Access Evaluation (CLEF 2010)*, Padua.

Benzineb, Karim and Jacques Guyot. 2011. Automated patent classification. In Mihai Lupu, Katja Mayer, John Tait, and Anthony J. Trippe, editors, *Current Challenges in Patent Information Retrieval*, volume 29. Springer, New York, pages 239–261.

Braga, Igor, Maria Monard, and Edson Matsubara. 2009. Combining unigrams and bigrams in semi-supervised text classification. In *Proceedings of Progress in Artificial Intelligence, 14th Portuguese Conference on Artificial Intelligence (EPIA 2009)*, pages 489–500, Aveiro.

Caropreso, Maria Fernanda, Stan Matwin, and Fabrizio Sebastiani. 2001. A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. In A. G. Chin, editor, *Text Databases & Document Management*. IGI Publishing, Hershey, PA, pages 78–102.

Crawford, Elisabeth, Irena Koprinska, and Jon Patrick. 2004. Phrases and feature selection in e-mail classification. In *Proceedings of the 9th Australasian Document Computing Symposium (ADCS)*, pages 59–62, Melbourne.

Dagan, Ido, Yael Karov, and Dan Roth. 1997. Mistake-driven learning in text categorization. In *Proceedings of 2nd Conference on Empirical Methods in NLP*, pages 55–63, Providence, RI.

de Marneffe, Marie-Catherine and Christopher Manning. 2008. The Stanford Typed Dependencies representation. In *Coling 2008: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8, Manchester.

Derieux, Franck, Mihaela Bobeica, Delphine Pois, and Jean-Pierre Raysz. 2010. Combining semantics and statistics for patent classification. In *Proceedings of the Conference on Multilingual and Multimodal Information Access Evaluation (CLEF 2010)*, Padua.

Dumais, Susan, John Platt, David Heckerman, and Mehran Sahami. 1998. Inductive learning algorithms and representations for text categorization. In *Proceedings of the Seventh International Conference on Information and Knowledge Management (CIKM '98)*, pages 148–155, Bethesda.

Fall, Caspar J. and Karim Benzineb. 2002. Literature survey: Issues to be considered in the automatic classification of patents. Technical report, World Intellectual Property Organization, Geneva.

Fall, Caspar J., Atilla Törcsvári, Karim Benzineb, and Gabor Karetka. 2003. Automated categorization in the international patent classification. *ACM SIGIR Forum*, 37(1):10–25.

Fürnkranz, Johannes. 1998. A study using n-gram features for text categorization. Technical Report OEFAI-TR-98-30, Austrian Research Institute for Artificial Intelligence, Vienna.

Fürnkranz, Johannes. 1999. Exploiting structural information for text classification on the WWW. In *Proceedings of Advances in Intelligent Data Analysis (IDA-99)*, pages 487–497, Amsterdam.

Galavotti, Luigi, Fabrizio Sebastiani, and Maria Simi. 2000. Experiments on the use of feature selection and negative evidence in automated text categorization. In *Proceedings of Research and Advanced Technology for Digital Libraries, 4th European Conference*, pages 59–68, Lisbon.

Guyot, Jacques, Karim Benzineb, and Gilles Falquet. 2010. Myclass: A mature tool for patent classification. In *Proceedings of the Conference on Multilingual and Multimodal Information Access Evaluation (CLEF 2010)*, Padua.

Held, Pierre, Irene Schellner, and Ryuichi Ota. 2011. Understanding the world's major patent classification schemes. Paper presented at the PIUG 2011 Annual Conference Workshop, Vienna, 13 April.

Joachims, Thorsten. 1999. Making large-scale support vector machine learning practical. In Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, editors, *Advances in Kernel Methods*. MIT Press, Cambridge, MA, pages 169–184.

Koster, Cornelis, Jean Beney, Suzan Verberne, and Merijn Vogel. 2011. Phrase-based document categorization. In Mihai Lupu, Katja Mayer, John Tait, and Anthony J. Trippe, editors, *Current Challenges in Patent Information Retrieval*, volume 29. Springer, New York, pages 263–286.

Koster, Cornelis, Marc Seutter, and Jean Beney. 2001. Classifying patent applications with winnow. In *Proceedings Benelearn 2001*. pages 19–26, Antwerpen.

Koster, Cornelis, Marc Seutter, and Jean Beney. 2003. Multi-classification of patent applications with winnow. In Manfred Broy and Alexandre V. Zamulin, editors, *Perspectives of Systems Informatics*: 5th International Andrei Ershov Memorial Conference, volume 2890 of *Lecture Notes in*

Computer Science. Springer, New York, pages 546–555.

Koster, Cornelis and Mark Seutter. 2003. Taming wild phrases. In *Proceedings of the 25th European conference on IR research (ECIR'03)*, pages 161–176, Pisa.

Krier, Marc and Francesco Zaccà. 2002. Automatic categorization applications at the European patent office. *World Patent Information*, 24(3):187–196.

Larkey, Leah. 1998. Some issues in the automatic classification of U.S. patents. In *Working Notes of the Workshop on Learning for Text Categorization, 15th National Conference on AI*, pages 87–90, Madison, WI.

Larkey, Leah S. 1999. A patent search and classification system. In *Proceedings of the Fourth ACM Conference on Digital Libraries (DL'99)*, pages 179–187, Berkeley.

Lewis, David D. 1992. An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '92)*, pages 37–50, Copenhagen.

Mille, Simon and Leo Wanner. 2008. Making text resources accessible to the reader: The case of patent claims. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech.

Mitra, Mandar, Chris Buckley, Amit Singhal, and Claire Cardie. 1997. An analysis of statistical and syntactic phrases. In *Proceedings of RIAO'97 Computer-Assisted Information Searching on Internet*, pages 200–214, Montreal.

Mladenic, Dunja and Marko Grobelnik. 1998. Word Sequences as Features in Text-Learning. In *Proceedings of the 17th Electrotechnical and Computer Science Conference (ERK98)*, pages 145–148, Ljubljana.

Moschitti, Alessandro and Roberto Basili. 2004. Complex linguistic features for text classification: A comprehensive study. In Sharon McDonald and John Tait, editors, *Advances in Information Retrieval*, volume 2997 of *Lecture Notes in Computer Science*. Springer, New York, pages 181–196.

Nastase, Vivi, Jelber Sayyad, and Maria Fernanda Caropreso. 2007. Using dependency relations for text classification. Technical Report TR-2007-12, University of Ottawa.

Nenkova, Ani and Kathleen McKeown. 2011. Automatic summarization. *Foundations and Trends in Information Retrieval*, 5(2–3):103–233.

Ozgür, Levent and Tunga Güngör. 2009. Analysis of stemming alternatives and dependency pattern support in text classification. In *Proceedings of Tenth International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2009)*, pages 195–206, Mexico City.

Özgür, Levent and Tunga Güngör.
2010. Text classification with the support of pruned dependency patterns. *Pattern Recognition Letters*, 31(12):1598–1607.

Özgür, Levent and Tunga Güngör.
2012. Optimization of dependency and pruning usage in text classification. *Pattern Analysis and Applications*, 15(1):45–58.

Parapatics, Peter and Michael Dittenbach. 2009. Patent claim decomposition for improved information extraction. In *Proceedings of the 2nd International Workshop on Patent Information Retrieval (PAIR'09)*, pages 33–36, Hong Kong.

Salton, Gerard and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing Management*, 24(5):513–523.

Scott, Sam and Stan Matwin. 1999. Feature engineering for text classification. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML '99)*, pages 379–388, Bled.

Smith, Harold. 2002. Automation of patent classification. *World Patent Information*, 24(4):269–271.

Tan, Chade-Meng, Yuan-Fang Wang, and Chan-Do Lee. 2002. The use of bigrams to enhance text categorization. *Information Processing and Management*, 38(4):529–546.

Verberne, Suzan and Eva D'hondt. 2011. Patent classification experiments with the Linguistic Classification System LCS in CLEF-IP 2011. In *Proceedings of the Conference on Multilingual and Multimodal Information Access Evaluation (CLEF 2011)*, Amsterdam.

Verberne, Suzan, Merijn Vogel, and Eva D'hondt. 2010. Patent classification experiments with the Linguistic Classification System LCS. In *Proceedings of the Conference on Multilingual and Multimodal Information Access Evaluation (CLEF 2010)*, Padua.