When is the Time Ripe for Natural Language Processing for Patent Passage Retrieval?

Linda Andersson, Mihai Lupu, João Palotti, Allan Hanbury, and Andreas Rauber
Vienna University Of Technology
Vienna, Austria
{surname}@ifs.tuwien.ac.at

ABSTRACT

Patent text is a mixture of legal terms and domain specific terms. In technical English text, a multi-word unit method is often deployed as a word formation strategy in order to expand the working vocabulary, i.e. introducing a new concept without the invention of an entirely new word. In this paper we explore query generation using natural language processing technologies in order to capture domain specific concepts represented as multi-word units. In this paper we examine a range of query generation methods using both linguistic and statistical information. We also propose a new method to identify domain specific terms from other more general phrases. We apply a machine learning approach using domain knowledge and corpus linguistic information in order to learn domain specific terms in relation to phrases' Termhood values. The experiments are conducted on the English part of the CLEF-IP 2013 test collection. The outcome of the experiments shows that the favoured method in terms of PRES and recall is when a language model is used and search terms are extracted with a part-of-speech tagger and a noun phrase chunker. With our proposed methods we improve each evaluation metric significantly compared to the existing state-of-the-art for the CLEP-IP 2013 test collection: for PRES@100 by 26% (0.544 from 0.433), for recall@100 by 17% (0.631 from 0.540) and on document MAP by 57% (0.300 from 0.191).

Categories and Subject Descriptors

H.3.3 [**Information systems**]: [Information retrieval query processing]; I.2.7 [**Computing methodologies**]: [Natural language processing]

Keywords

Information Extraction, Natural Language Processing, Patent Retrieval, Text Mining

1. INTRODUCTION

CIKM'16 , October 24-28, 2016, Indianapolis, IN, USA

DOI: http://dx.doi.org/10.1145/2983323.2983858

Prior Art search (often referred to as simply Patent Retrieval) is interesting for both academic and commercial reasons. Academically, it is interesting because it brings together all aspects of the information retrieval (IR) science, from text retrieval to user and task analysis, including distributed and multimodal search. Commercially, it is interesting because intellectual property is a multi-trillion dollar business working on a relatively small collection of approximately 100 million patents [10]. In Prior Art search the patent experts carefully examine the first 100 to 200 retrieved documents based upon a session consisting of an iteration of Boolean search queries, including meta-data such as classification, application dates, etc. combined with key terms [11].

There are three main aspects associated with manually constructed search queries: familiarity with the search environment, domain expertise (i.e. knowing what query terms to use) and expertise of the type of search (invalidity search, freedom-to-operate and Prior Art search) [11]. These aspects will also to some degree be valid for automatic query generation.

In order to adapt general text retrieval systems to the patent domain, we need to incorporate domain knowledge, linguistic knowledge of the text genre, as well as knowledge of how a specific language represents domain specific concepts. The need of enhancing retrieval models with natural language processing (NLP) techniques in order to identify terminology, etc. has been addressed for domain specific IR in [15, 6]. We examine to what extent domain knowledge and linguistic information will help to generate better search queries, particularly when the task is to identify relevant paragraphs (as opposed to full patent documents). Due to the complexity of the patent text genre and the passage retrieval search task, we propose several different query generation methods accessing both linguistic and statistical knowledge, as well as making use of manually and automatically pre-defined lists of domain specific terms. In this paper, we compare and combine linguistic phrases with n-gram[1] methods, as well as incorporating domain specific meta-data in order to select more domain specific terms. A domain specific term or concept refers to a technical term or concept, which has a specific meaning in a scientific field (e.g. *composite cell* versus *blood mononuclear cell*). For instance, *composite cell* occurs 176 times in patents belonging to the Telecommunication sector but only 4 times in patents belonging to the Biotechnology sector. On the other hand,

[1]Here we refer to n-grams as representing an entire orthographical string unit of letter or digit sequences.

blood mononuclear cell occurs 5,600 times in Biotechnology patents but only twice in patents belonging to the Telecommunication sector. The contributions of this paper are:

1. We developed a novel method for domain specific terminology extraction (technical terms detection).
2. We demonstrate by combining words and noun phrases (NPs), especially phrases of a technical character, will enhance recall and PRES [17] compared to the current state-of-the-art methods for patent passage retrieval.
3. We demonstrate that words and phrases extracted only from the claims section of a patent document will enhance MAP and Precision at passage level.
4. We demonstrate that by merging results from a document index and a paragraph index, the overall retrieval performance will be enhanced.

The paper is organized as follows: we first provide related work and linguistic theoretical background in Section 2. In Section 3, we present our experiment set, learning phrases' termhoodness, the query model, and indexes. The results are presented and discussed in Section 4, while in Section 5 we present our conclusion and suggest further work.

2. RELATED WORK

The patent retrieval research focus has mainly been on improving and developing methods and systems for supporting patent experts in the process of Prior Art search (i.e. retrieving patent documents that could invalidate a patent). The CLEF-IP track started in 2009 with the Prior Art Candidate Search track. In 2012, passage retrieval was introduced as the text mining task [20], the aim is not only to retrieve relevant documents but also extract those paragraphs (passages) in the relevant documents that are found most relevant. The passage retrieval task is more in line with the work of patent examiners during a validity search process, since the examiners need to identify both the prior art documents, as well as each specific paragraph within these documents which is be considered to be Prior Art [11].

2.1 NLP used in patent retrieval

Research involving IR and NLP shows that the shallow linguistic methods such as stop word removal, stemming, etc. yield significant improvements, while deeper linguistic analyses such as part of speech tagging (PoS), chunking, parsing, etc. could even decrease accuracy [6, 15]. Deeper linguistic methods have only been reported to achieve a significant improvement in retrieval performance for domain specific IR. However, the expected improvements have not been convincing compared to the extra effort that is required in order to use NLP in domain specific text genres.

Behind the usage of an NLP application is the assumption that the source data and target data have the same feature distribution [24]. Too many unseen events, i.e. words or syntactic constructions only occurring in the target data, will decrease the performance of an NLP application drastically. Therefore, extensive work is generally associated with domain adaptation of NLP applications, since it involves manual annotation of training data and creation of a ground truth for evaluation. However, there exist very few domain specific corpora, which NLP applications can be re-trained upon, and none of them covers the entire scientific fields present in the patent domain.

We can divide the NLP usage in patent text mining into two main categories – in order to improve the NLP itself [1, 3], or in order to improve an end-application [2, 6, 7, 26]. In the end-application studies NLP tools have either been given a moderate domain adaption or just been used as off-the-shelf modules. For instance, in [26] NLP applications were used as off-the-shelf modules in order to extract NPs for query generation, the authors did not mention how the NPs were identified (i.e. which PoS-tagger and NP chunker were used in the identification process). Still, their experiment showed by combining words and NPs compared to only using words the performance in recall and MAP increased significantly. Very few studies have conducted a direct evaluation of the NLP tools used in a pipeline setting for an end-tool (e.g. IR system, classification system) [3, 25].

In [25], a comparison between two different parsers was performed, AEGIR and the Connexor (Constraint) Functional dependency grammar (FDG) parser. A set of 100 randomly selected short patent sentences (5-9 words) were manually assessed based upon generated dependency triples (i.e. [word relation word] e.g. [damage ATTRIBUTE mucosal]). The F1-score for AEGIR was 0.47 and for Connexor FDG was 0.71. AEGIR is used in the PHASAR system and has been domain adapted towards the patent domain by increasing the lexicon coverage [6]. However, data is stored within a manually maintained database making the system both sensitive towards coverage and phrase weighting, which partly explains why the performance decreased when only the dependency triples in the CLEF-IP 2010 Prior Art search task were used. The mean average precision (MAP) value was 0.0386 for dependency triples and for the unigram method MAP was 0.0739 [6].

Within the PATExpert project several direct evaluations of the NLP tools used in the text mining pipeline have been conducted. The PATExpert data set consists of 1011 claims from Optical Recording Device and 486 claims from Machine Tool [7]. This data set has been re-used in several studies, see for instance [7, 3]. The aim of the project was to enrich patent data with semantic annotation, which could be used in different information extraction (IE) and IR applications. Initially, several NLP workbench modules were used as off-the-shelf modules, but by the end of the project, domain adaptation had been deployed to several tools [3]. Burga [3] examined PoS-tagger performance when applied on patent data. A PoS-tagger trained on part of the CoNLL 2007 test collection (a sub set of the Penn Treebank) dropped in accuracy from 97.69% (CoNLL) to 94.59% (patent text). Verb participles were especially discovered to be erroneously identified when functioning as adjectives or nouns (e.g. *coating method* and *slot-die coating*) [1, 3]. By correcting these systematically assigned PoS-tag errors made by the PoS-taggers, the performance of parsers and end-tools was improved.

2.2 The patent text genre

All patents contain a rich set of meta-data such as citation (citing prior art), assignee (person or company), inventor (persons), date, address, and classification code (e.g. International Patent Classification) etc. The International Patent Classification schema (IPC) reflects a semantic interpretation regarding technical domains and organized in a taxonomy structure [12, 14]. IPC codes are suitable for cluster-based retrieval since it can be a basis for semantic clustering.

The patent text genre is associated with several interesting

linguistic characteristics such as huge differences in length, strictly formalized document structure (both semantic and syntactic), and extensive use of domain specific terms. The number of new concepts introduced in the patent domain by using phrases is also very high compared to other genres [21]. Furthermore, the vocabulary diversity within different technical fields [10, 14] makes it problematic to use standard domain specific terminology dictionaries or general language resources such as WordNet. In [25] it was observed that in terms of individual token coverage there is no significant difference between general English and the English used in patent claims. The (new) domain specific terminology is more likely introduced in the form of complex NPs.

Rhetorically, a patent document consists of four main textual components (title, abstract, description, and claim), each with a different communication goal. The abstract gives a short and general summary, where umbrella (broad) terms are used. The description gives elaborative background information on the invention. Finally, the claims section describes the essential component of the invention and has its own very special conceptual, syntactic and stylistic/rhetorical structure [14].

2.3 Domain Specific Terminology and Multi word terms

The majority of entities in technical English dictionaries consist of terms with more than one word [23]. The technical multi-word phrases consist of NPs containing common adjectives, nouns and occasionally prepositions (e.g. 'of'). In technical English text, the word formation using noun compounds is often deployed in order to expand the working vocabulary, without creating new words [21, 23]. The noun compounds could either be an orthographical unit (e.g. *bookcase*), or combined with hyphenation (e.g. *mother-in-law*) or a multi-word unit (MWU) (e.g. *crash landing*).

The noun compounding strategy causes not only unseen events on the morphological level (words) with new orthographical units, it also generates a diversity of syntactic structures among NPs (e.g. verb participle being used as nouns and adjectives), which is problematic for NLP applications [6, 1, 3].

The complexity of the NPs increases in patent text due to the high density of technical terminology in terms of MWUs. Among the MWUs we speak of multi-word terms (MWTs), which are phrases characterised by a very strong bond between the words that form them [1, 23, 6]. A MWT generally represents a domain specific concept e.g. *complex programmable logic device*, while a MWU include more general phrases such as *the green house* and *the method of the invention*. Ultimately, it is the MWTs that are most important and most efforts go in the direction of identifying such terms as opposed to the more generic MWUs. For IR, MWTs should in fact be considered terms in the traditional sense (i.e. denoting a specific meaning). Technical concept and domain specific concept can also be represented by a single word (e.g. *bradycardia*), in this paper we are only interested in termhoodness among domain specific terms of type MWUs (i.e. MWTs).

The most successful techniques for term extraction involve supervised learning methods [2]. These methods require access to human annotators in order to establish a training set. The labelling task is both time-consuming and costly. Moreover, for domain specific fields, the annotation task requires both linguistic knowledge, as well as domain specific knowledge. Technical terms are more ambiguous to label than other better defined named entity categories (e.g. company names, brands, geographic locations, medical treatments) [2]. In [2], NPs composed of common English words such as *rear cross frame member* or *memory data processor* would only be labelled technical terms if they were thought to refer to a reasonably independent artefact. However, this definition is cumbersome, due to the extensive use of paraphrasing in patent text, as observed in [19]. The scholarly term *word processor* would be referred to as *document editing device* and therefore the latter MWT would not be recognized as a technical term. For instance, the MWT *word processor* has a Wikipedia entry but *document editing device* has not.

There are also several unsupervised techniques for extracting technical terms: pure statistical methods such as conditional random field (CRF), mutual information (MI) and linguistic methods using lexico-syntactic filters [23]. The method still considered to be the state-of-the-art is combining statistical measures and linguistic filters, i.e. the C-value [9]. The C-value claims to reflect a MWU's termhoodness, i.e. the degree to which a MWU is actually a MWT. In our case, this translates to an indicator of the degree to which a phrase should be considered a domain specific concept. The C-value computation consists of two parts: 1) a linguistic filter and 2) a nested statistical weight technique. The linguistic filter could either be a closed filter i.e. only permitting phrase sequences of specific PoS-tagging sequences, or an open filter using an NP-chunker. The C-value statistical measure assigns the termhood value to a candidate string. It computes the statistical characteristics of the candidate string, according to following formula:

$$\text{C-value(a)} = \begin{cases} log_2|a| \cdot f(a), & \text{if } a \text{ is not nested,} \\ log_2|a| \cdot \left(f(a) - \frac{1}{P(T_a)} \sum_{b \in T_a} f(b) \right) & \text{otherwise} \end{cases} \quad (1)$$

where:

- a is the candidate string,
- $|a|$ is the length of the candidate string a (in number of words),
- $f(a)$ is its total frequency of occurrence in the corpus,
- T_a is the set of extracted candidate terms that contain a,
- $P(T_a)$ is the number of these candidate terms,
- $f(b)$ is the frequency of the candidate term b that contains a.

As seen in Eq. 1, the C-value is based on the frequency of a occurring as an NP or as a subset of a larger NP. In our experiments we use the C-value as seed for a machine learning (ML) algorithm due to the fact that computing C-value for an entire collection is extremely time consuming, since it requires to assign almost the entire collection with at least PoS-tags and NP brackets.

2.4 Query Generation

A patent document is too long to be used directly as a search query; it needs to be transformed into a reasonable length [26]. There are several factors that need to be analyzed during this transformation [5] such as: number of terms that should be extracted; how terms should be selected from the different text sections (title, abstract, description and claim), which term weight techniques should

be used e.g. term frequency (tf), inverse document frequency (idf) or a combination of $log(tf) \cdot idf$. In [26] it was shown that the best section to extract search terms from is a smaller part of the description section referred to as the brief summary field <bsum>. However, this text field only exist in US patents, European patents do not have this text field.

Cetintas and Si [5] conducted extensive experiments on the TExt REtrieval Conference (TREC) 2009 Chemical track, which also contained a Prior Art task. The optimal query generation method was found to be 20 to 30 search terms, weighted by $log(tf) \cdot idf$, and using all text sections as a base for search term extraction, but giving extra weight to the text sections abstract, description and claim.

Phrases are frequently used to narrow the scope of a topic, since phrases distinguish between different meanings of polysemous terms (e.g. *cell* in *blood cell* versus *composite cell*) [8]. Contrary to the general assumption that phrases or n-grams improve precision oriented measurements, it has been reported in patent retrieval that phrases tend to improve recall oriented measurement [6, 18, 26]. Only in [26] were phrases (NPs) when combined with words shown to improve MAP.

However, since a majority of MWTs are composed of general English words, they are more exposed to paraphrasing. Each member could be substituted with a synonym and this could lead to data sparseness (too low frequency) issues [19]. Therefore, Mahdabi et al. [18] used a statistical proximity n-gram (skip-gram) method to reduce the data spareness issue. Furthermore, the skip-grams were given weights based upon IPC information in order to reduce polysemous terms. The final search terms were selected based on their occurrence in a specific IPC class, as well as their internal closeness to each other within the documents in the collection. Mahdabi et al. [18] reported Recall@1000 of 0.659, MAP@1000 of 0.105 and PRES@1000 of 0.554 on the CLEF-IP 2010 Prior Art Search task. Also a skip-gram method was used in a patent classification task by D'hondt [6], who observed that the most effective features were skip-grams filtered with a PoS-tagger, preferably adapted to the patent domain. D'hondt reported that to only use unigrams would infer more noise for patent classification compared to the Reuters-2158 data set for text classification, where unigrams were shown to be a more effective feature than phrasal features.

Finally, Luo and Yang [16] concluded that to use only adjectives and nouns as query terms was a too aggressive query generation method. Instead, they selected hyphenation words based on the hypothesis that the hyphened words would have a technical signature since they generally occur as nouns and to some degree disambiguate otherwise polysemous words. Their best method achieved on the CLEP-IP 2013 test collection a PRES@100 of 0.433 and recall@100 of 0.540 on document level and on passage level the Precision(P) of 0.213 and MAP(P) of 0.132. MAP(P) and Precision(P) are a micro version of the standard measurements MAP and Precision, first computing average precision and precision for each relevant passage in a single relevant retrieved document [20]. The results reported by Luo and Yang are the current state-of-the-art for the CLEF-IP 2013 patent passage retrieval task.

3. EXPERIMENT SET UP

We have in previous sections provided linguistic characteristics of the patent text genre and how English forms (new) MWTs. We need to keep this in mind when drafting an automatic query generating method. We also reported that if NLP tools are to be used they need to be adapted to the patent domain, otherwise it may hamper the performance of the end-tools, as reported in [6, 3]. In previous literature, several query generation methods have been explored, but none of them have combined domain knowledge, linguistic characteristics of the text genre, as well as general linguistic knowledge of English word formation strategies. In [16], hyphenation was used to identify technical terms. In [18], IPC information in combination with skip-grams was used. In [6] skip-grams were combined with linguistic information. However, in order to improve over state-of-the-art methods for patent passage retrieval, the query generation methods have to integrate MWTs into the search term selection process, as well as transforming a patent topic (a patent application document) into a sizable search query. We would like our experiment to answer three questions:

- How shall we transform a patent document into search query length with the maximum of 100 tokens?
- Which terms should be used, phrases and words or n-grams?
- Are there some MWUs that are better than others (i.e. MWTs) as search terms and how can we identify them without using too time consuming methods?

We now present all of the tested methods. Throughout this section, bold font anchors will be used to define the runs described in the next section (Results).

```
<topic ucid: EP-1287743-A2 query: PSG-47>
  (freezing OR start OR liquid OR dough OR glucose OR
  bake-off OR coating OR foodstuff OR pre-glaze OR syrum)
  AND
  ("complex sugar"~5 OR "glucose syrup"~5 OR "dough
  product"~5 OR "dough mixture"~5  OR "form liquid"~5
  OR "pre-glaze composition"~5 OR "coating step"~5 OR
  "coating part outer surface dough mixture"~9)
```

Figure 1: Example of search query syntax using the select handler in Solr.

3.1 The test Collection

The CLEF-IP collection contains approximately 2.6 million XML documents (representing 1.5 million patents). In this study we experiment on the 50 English topics of CLEF-IP 2013 passage retrieval search task [20]. Patent topics consist of one or more claims, which were manually selected by the track organisers based on patent examiner's search reports. Evaluation was performed both at document level and at passage level. At document level, the Qrels were the set of patents cited by the examiner in the search report. For the passage retrieval task these citations are combined with each relevant paragraph to form a Qrel i.e. a relevant document can be part of several Qrels depending upon the number of relevant paragraphs manually identified in each document. Paragraphs are represented by their XPath (henceforth passages).

3.2 Index Setup

We created two different indices, one based upon passages and one based on documents. All English passages and documents were indexed with Solr 4.7.2 using a white

space tokenizer, the default English stop word list and the Emin stemmer, provided by Solr. In this experiment we selected three IR models, each model represents a type of category model, Vector Space Model (Vector), Probabilistic (BM25), and Language Model with Jelinek-Mercer smoothing method (LMJM). For all selected IR models, we used the default settings, since the main purpose of this study is to experiment with automatic query generation and not system optimization. For the query we used the Solr RequestHandler *select* with the Boolean operator OR among the set of words and phrases. In the methods where both words and phrases were used we connected the syntax with AND (see Figure 1). Skip-grams were denoted using the Lucene query syntax (e.g. *"simple example"∼5* denotes that the terms in the preceding phrase need to be found within 5 locations of each other).

3.3 Learning Termhood

We established a manual domain specific terminology sample set consisting of 4,400 instances: 2,700 phrases labelled as MWTs and 1,700 phrases labelled as not MWTs. Due to the time consuming processing of assigning PoS-tag and identifying NP boundaries in all sentences where a specific phrase exists, we only created C-values for a smaller set. On average, a patent sentence took between 2 and 4 seconds to process through the NLP pipeline. Therefore, we decided to set up a smaller experiment in order to see if it was possible to substitute the C-value with less time consuming statistical computations. We sampled a set of 637 terms (415 MWTs, 222 not MWTs i.e. only being MWUs). The aim with the smaller experiment was to minimise the loss in performance when excluding C-value. We used the ML software Weka 3.6 and elaborated on 13 different features. The numerical features we computed were: phrase length, document frequency (DF), MI, C-value and a set of statistical measurements associated with the IPC distribution (henceforth IPC-distribution-values) of a phrase. We only used one non-numerical feature, 'syntax', which consists of the PoS patterns of each sample phrase. Syntax_freq refers to the given frequency of that particular NP pattern (e.g. 'JJ NN NNS', 'JJ NN' etc.) observed in the sample set. For the final experiment we used the larger sample set (4,400). We also switched to a random forest classifier and for features we selected phrase length, DF, MI, IPC-distribution-values.

The features related to IPC-distribution-values need further explanation. We chose to compute different statistical measurements based upon the IPC codes assigned to the documents in which a particular phrase occurs. For each phrase, we computed an IPC-frequency, as follows: for each IPC, we counted the number of documents that contain the phrase and have been classified in this IPC. One can think of it as a document frequency, factored by the IPC classification. For each phrase we then computed the number of unique IPC codes, sum of all IPC frequencies, their variance, median, average and standard deviation (stddev). We also combined the C-Value with one IPC-distribution-value, the C-Value was divided with the sum of the IPC frequency of each given phrase. Our hypothesis for using the IPC for computing the technical significance of phrases (i.e. their termhood) is that phrases occurring in documents with the same IPC codes (having a homogeneous distribution) would more likely refer to domain specific concepts (i.e. MWTs) than phrases occurring in documents with a more heterogeneous distribution of IPC codes (i.e. only being MWUs). As mentioned in 2, the IPC codes are of a semantic nature, and can been seen as a technical language by themselves, into which all documents are translated during the classification process, regardless of their original language [12].

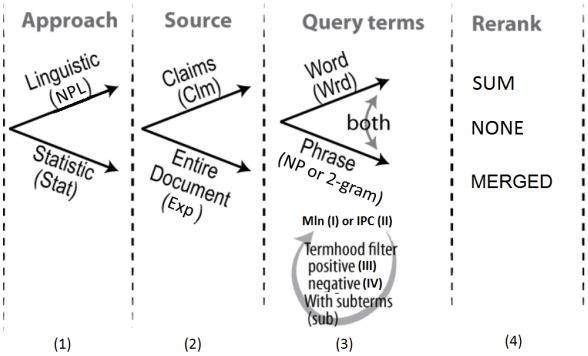

Figure 2: Query formulation Schema

3.4 Query Generation

For all query generation methods we used two different settings for length, 30 search terms (**Shrt**) and 100 search terms (**Lng**). For the combined query generation methods (phrases and word or bigrams and unigram), 15 and, respectively, 50 terms were selected from each group. We first tested to select only 20 and 30 search terms, as reported in [5, 26] on the training set provided by the CLEF-IP 2013 passage retrieval task, but we discovered for most of the methods the performance increased when using the range from 30 to 100 search terms. For all methods we decided to use $log(tf)\cdot idf$ since it was reported as the best term weight method in [5]. As a baseline (**Bsln**) we extracted from each topic the highest ranked top 30 and top 100 unigrams and bigrams based upon their $log(tf)\cdot idf$ values. As we will see, this is a strong baseline, especially when compared with the state-of-the-art reported for CLEF-IP 2013 in [16].

The query generation process is given in four steps, as depicted in Figure 2. The first step (1) consists of selection of the main approach: statistic (**stat**) versus **NLP**. In the NLP method only phrases classified as NPs (**NP**) and content words (**Wrd**) belonging to the categories adjective, noun and verb were selected as search term candidates. The NLP approach consisted of PoS-tagging all sentences with the Stanford PoS-tagger and for NP detection the baseNP Chunker was used [22]. The linguistic analysis of the sentences was automatically post corrected according to the method proposed in [1]. The method is rule-based and corrects errors associated with mislabelled NP sequences. A small direct evaluation of 100 random sampled sentences showed an increase of the F-score from 0.316 without post correction to 0.447 with post correction, but still far from expected performance associated with general text (ranging from 0.66 to 0.91 depending upon NLP applications used). For the stat method we only used stop word filters for unigrams (**1-gram**), while for phrases we extracted bigrams (**2-gram**) within sentences (i.e. no bigram goes across sentence boundaries).

In the second step (2), we deployed two methods: only extracting terms from the claim (**clm**) section (i.e. from the topic itself) or including terms from a larger part of the topic document (hence referred to as [**Exp**]anded). In order

to arrive at expanded terms, cosine similarity values were computed pair-wise between claim sentences and all other paragraphs based upon the phrases and words or bigram and unigram they were composed of, similar to the technique used in [13].

In the third step (3), the selection process of the unigram and bigram or words and NPs is conducted. As seen in Figure 2 NPs and bigrams were also exposed to four different filters before being ranked according to their $log(tf) \cdot idf$ value. The main filter Termhood (**Trm**), which can either be a manually established filter (**Mln**) (I) or an automatic established filter (II) based upon the ML method described in Section 3.3 (hence referred to **IPC**). Each filter can be defined as either being positive list (**Pstv**) (III) i.e. a MWT or a negative list (**Ngtv**) (IV) i.e. not a MWT. Furthermore, there were several instances, where only a part of a larger NP would be considered a technical term (e.g. *a water vapor* in *a water vapor permeability of 7000*, *writing tool* in *a conventional writing tool of this kind*). In order to be able to select or disregard a sub-phrase of a larger NP, we also deployed a skip-gram method (**Sub**) for the Termhood filters. We allowed a window of 3 additional words to the length of each phrase e.g."glucose syrup"∼5 should be found in a range of five words.

In the fourth step (4), we deployed two re-ranking methods, **SUM** and **MERGED**. The **SUM** method sums up all passage similarity values from a retrieved document. The method is based upon the assumption that relevant documents will have more relevant passages compared to non-relevant documents, therefore a higher common similarity score. Passages belonging to a specific retrieved document are thereafter listed internally among each other in order of to their similarity values. The **MERGED** method first extracts the position for a retrieved document given by the document index, and thereafter consider the similarity value given by passage index for each passage belonging to a specific retrieved document. In the case of **MERGED**, if no passages were retrieved for a document retrieved by the document index, this document will obviously not appear in the merged list, as there is no passage to be added at that position. In our experiments we will also mention the **NONE** re-ranking, denoting that no re-ranking was applied to the list of retrieved passages.

4. RESULTS

4.1 Learning Termhoodness

Table 1 displays the smaller ML experiment (637 terms) on how to learn phrases' Termhoodness, and the various sets of features that were tested. The best performance in terms of correctly classified instances as MWTs is given when we exclude C-value and IPC: C-value, which was unexpected. However, as a single feature they achieved, respectively, 66% and 71% correctly classified MWTs. We also observed that the syntax feature influenced the outcome: by only using the syntax feature the method correctly classified 67% to 68% as MWTs. In our initial experiments, we observed that with the IPC-distribution-values, DF and MI we can minimise the decrease in classification performance, otherwise observed when removing C-values. We can thereby conclude - yes it is possible to learn Termhoodness for phrases without using C-value computation.

On the larger sample set (4,400 terms) we used the features: phrase length, DF, MI, IPC-distribution-values. We run the experiment with and without IPC: stddev - the version without it was slightly better than the version with it: F1 – 0.851, accuracy 0.794 (without), F1 0.845 accuracy - 0.790 (with). We did, however, observe that if we added the syntax feature we could even improve the performance further.

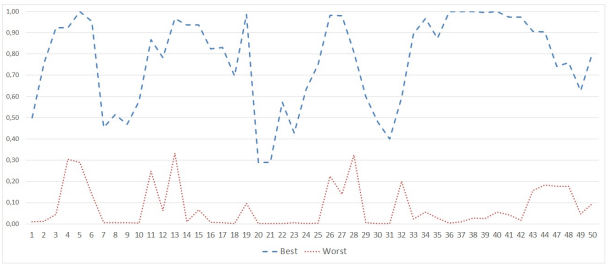

Figure 3: PRES performance per Topic

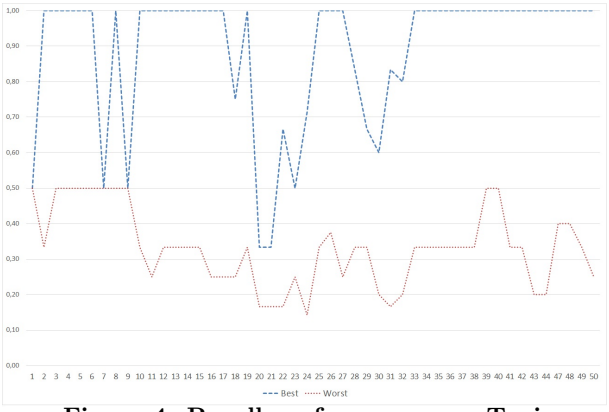

Figure 4: Recall performance per Topic

4.2 Experiment on the CLEF-IP 2013 patent passage retrieval task

In Section 2 we gave a thorough report of the importance of using domain knowledge in terms of meta-data such as the IPC classification schema and acknowledging the linguistic characteristics of the patent discourse. We have examined several different aspects of automatic query generation for each topic. In total, we generated 220 (44 baseline, 88 stat, 88 NLP) different runs for each IR model (for a total of 660 runs). Consequently, for each topic we generated 220 versions and all in all we executed 11,000 (220*50) search queries for each IR model. As we will not be able to present them all, we have selected some to make specific observations on the various components of our automatic query generation methods. Table 2 explains the runs present in subsequent tables in this section.

In Table 3 we show our best runs for each evaluation metric and compare it with the best official runs of the participant of CLEF-IP 2013 patent passage retrieval task and with the baseline. For PRES and MAP at document level, one or more of our query generation methods are statistically significant to the official runs of the CLEF-IP 2013 task. As the number of experiments is very large, statistical significance is performed using the recently proposed method by Carterette [4], in order to avoid the possibility that one test will incorrectly show a significant result. For each metric, we first performed an ANOVA to test the omnibus null hypothesis that all the runs are equal. This was rejected for MAP

Table 1: Experiment with different features to learn MWUs Termhood significance

Features	Feature combination																															
syntax	x	x	x	x	x	x	x	x	x			x	x	x	x	x	x		x	x		x	x	x							
syntax_freq	x	x	x	x	x		x	x	x			x	x	x	x				x	x												
phrase_lenght	x	x	x	x	x		x	x	x			x	x	x	x				x			x										
C-value	x	x	x										x	x	x	x					x											
DF:phrase	x	x		x	x		x	x				x	x	x		x	x	x														
probability (MI)	x	x		x	x		x	x	x			x	x			x	x	x					x		x							
IPC:CValue	x	x	x					x		x											x											x
IPC:sum	x	x	x	x	x	x	x	x	x	x	x																					
IPC:count	x	x	x	x	x	x	x	x	x	x	x																					
IPC:mean	x	x	x	x	x	x	x	x	x	x	x																					
IPC:median	x	x	x	x	x	x	x	x	x	x	x																					
IPC:variance	x	x	x	x	x	x	x	x	x	x	x																					
IPC:stddev	x				x	x				x	x																					
Correctly Classified %	77	77	77	78	76	71	77	77	78	77	71	77	76	77	75	75	69	70	76	67	71	73	68	68	65	68	66	71				

Table 2: Reference of query formulation runs mentioned in subsequent tables. We have three baselines (b1-b3), two versions of the state-of-the-art (g1,g2), 13 runs using linguistic information (nlp1-nlp13) and 9 using statistical information (s1-s9)

Run	Approach	Source	Query Terms	Length	IR Model	ReRank
b1	baseline	document	1-gram	100	LMJM	MERGED
b2	baseline	document	1-gram, 2-gram	100	BM25	MERGED
b3	baseline	document	2-gram	100	LMJM	MERGED
g1	Luo and Yang [16]	document	Wrd,hyphened MWUs	N/A	BM25	N/A
g2	Luo and Yang [16]	document	Wrd, hyphened MWUs	N/A	BM25	N/A
nlp1	NLP	Expanded	Wrd, NP:Trm-Pstv-Sub-IPC	100	LMJM	MERGED
nlp2	NLP	Expanded	Wrd	100	Vector	MERGED
nlp3	NLP	Expanded	Wrd	100	LMJM	MERGED
nlp4	NLP	Claim	Wrd,NP:Trm-Pstv-Sub-IPC	100	LMJM	MERGED
nlp5	NLP	Claim	Wrd	30	BM25	MERGED
nlp6	NLP	Expanded	Wrd,NP:Trm-Pstv-Sub-Mln	100	LMJM	MERGED
nlp7	NLP	Expanded	Wrd,NP	100	LMJM	MERGED
nlp8	NLP	Expanded	Wrd,NP:Trm-Pstv-IPC	100	LMJM	MERGED
nlp9	NLP	Expanded	Wrd,NP:Trm-Ngtv-Sub-Mln	100	LMJM	MERGED
nlp10	NLP	Expanded	Wrd,NP:Trm-Pstv-Sub-IPC	100	LMJM	MERGED
nlp11	NLP	Expanded	Wrd,NP:Trm-Pstv-Sub-IPC	30	LMJM	MERGED
nlp12	NLP	Expanded	Wrd	30	LMJM	MERGED
nlp13	NLP	Claim	Wrd	30	LMJM	MERGED
s1	stat	Expanded	1-gram, 2-gram	100	LMJM	MERGED
s2	stat	Claim	1-gram	100	LMJM	MERGED
s3	stat	Expanded	1-gram	100	Vector	MERGED
s4	stat	Expanded	1-gram,2-gram:Trm-Pstv-IPC	100	LMJM	MERGED
s5	stat	Expanded	1-gram,2-gram:Trm-Pstv-Sub-Mln	100	LMJM	MERGED
s6	stat	Expanded	1-gram,2-gram:Trm-Pstv-Mln	100	LMJM	MERGED
s7	stat	Expanded	2-gram:Trm-Ngtv-Sub-IPC	100	BM25	MERGED
s8	stat	Expanded	1-gram	100	BM25	MERGED
s9	stat	Expanded	1-gram,2-gram:Trm-Pstv-Sub-Mln	100	BM25	MERGED

and PRES with ($p<0.05$), meaning that at least two runs are significantly different. We then performed post-hoc pairwise two-sided t-tests using the single-step method to adjust the p-values for multiple comparisons. The results shown in Table 3 indicate for each cell the runs to which it is statistically significantly different by their run ID as upper-script. As we can see, while results are visibly different, the relatively low number of topics in this track, compounded by the rigour of the test, results in few clear cases of improvement.

The overall best result for PRES and recall is achieved when phrases (`Sub` i.e. skip-gram method) defined as MTWs and words are extracted from a larger part of the topic document (Expanded) see `nlp1`. The results confirm the finding in [6, 18] that phrases used in patent retrieval are in fact improving recall oriented measurements more rather than precision oriented measurements (see Section 2.4). For the more precision oriented measurements, the `stat` method was preferred over the `NLP` method as seen in `s1` and `s2`. We also observe that phrases (bigram) when combined with unigrams also achieved a better MAP value on the document

Table 3: Comparison with other CLEF-IP participant runs for CLEF-IP 2013 (g1, g2) and with the baseline (b1, b2). Bold indicates best performer in each column. Superscript indicates lower runs to which a statistically significant difference can be shown.

Run	PRES	Recall	MAP	MAP(P)	Prec(P)
nlp1	$\mathbf{0.544}^{g2,g1,s2}$	**0.631**	$0.285^{g1,g2,s2}$	0.112	0.218
s1	0.492	0.574	$\mathbf{0.300}^{g1,g2,s2}$	0.114	0.208
s2	0.444	0.560	0.187	**0.146**	**0.282**
b1	$0.536^{s2,g2,g1}$	0.622	0.226	0.132	0.229
b2	0.488	0.569	0.257^{s2}	0.111	0.166
g1	0.433	0.540	0.191	0.132	0.213
g2	0.432	0.540	0.190	0.132	0.214

Table 4: Best runs for each for the three IR models tested, according to each metric

	Run	PRES	Recall	MAP	MAP(P)	Prec(P)
best LMJM	nlp1	**0.544**	**0.631**	0.285	0.112	0.218
	s2	0.444	0.560	0.187	**0.146**	**0.283**
	s1	0.492	0.574	**0.300**	0.114	0.208
best Vector	nlp2	**0.500**	0.604	**0.236**	**0.128**	**0.238**
	s3	0.490	**0.611**	0.208	0.120	0.232
best BM25	s9	0.507	0.576	**0.291**	0.108	0.195
	nlp5	**0.521**	**0.607**	0.247	**0.143**	**0.226**

level, when a more restrictive search term selection method `stat` is applied compared to baseline `b2`. This observation supports the claim made in [26] that phrases do influence MAP positively.

For all measurements, the Language Model with Jelinek-Mercer smoothing (LMJM) achieved the best results. When comparing PRES performance between different IR models, we see that LMJM favors methods using a combination of phrases (MWT) and words (Table 4). For PRES, recall and for MAP on document level the model (LMJM) prefers unigram and phrases (bigram). Meanwhile, LMJM favors unigrams extracted with the `stat` method for precision and MAP on passage level. The Vector model favors query generations methods consisting of only words from either the claims section or a larger part of the topic document using either the `NLP` or `stat`. The BM25 model favors short queries composed of only words using the `NLP` method. It is interesting that the baseline method using only words (`b1`) achieved a recall value very close to the best method using both words and MWTs (`nlp1`). This finding indicates that some important words are lost when deploying more restrictive search term selection methods (stat or NLP). The post ranking method MERGED, i.e. merging results from a document index and a passage index, outperforms other post ranking methods (SUM and NONE). For instance, changing post rank to NONE for the query generation method `nlp1` would decrease the results significantly on the document level, as seen in Table 5. For the post ranking, we see that the merged list not only improves the passage retrieval but also the document retrieval compared to only using the result from the document index.

Finally, in Table 6 we examine different query generation methods related to our questions. The table lists a series of comparison points (e.g. what is the difference between **NLP** and **stat**?). As there are many parameters, for each such question we show the best performing run having each feature, and the matching run where all other parameters are the same and only the observed one is changed. In general therefore, we will have four runs for each comparison. In some cases, the two best runs identified already match each other in all other parameters, and therefore only they will be shown. "Best" is taken with respect to PRES.

We can observe for instance that when assembling a search query, the recall oriented measurements achieve better results when a larger part of the topic document is used as in the Exp method. When combining phrases and words the `NLP` method is preferred over the statistic, meanwhile when using only MWT (including all different filter of Trm) the statistic method is favored over the pure linguistic methods. This observation confirms the findings reported in [6], i.e. using bigram but applying a linguistic filter to select the final query terms will improve the overall performance (see Section 2.4). Considering the length of the search query, longer queries are preferred over shorter which contradict what was reported in [26, 5].

In Figures 3 and 4 we show the PRES and recall values in terms of the best and the worst performance for each topic. We have excluded those query generation methods, which generated zero since at least for one method, each topic would not retrieve any relevant passages. For recall we can see that 3 topics (PSG-1, PSG-7 and PSG-9) are quite stable in terms of methods, they will retrieve the same number of relevant passages regardless of the query generation method. However, the PRES values differ significantly; PSG-1 goes from 0.5 to 0.01, PSG-7 goes from 0.46 to 0.01, PSG-9 goes from 0.47 to 0.01. For at least one query generation method per topic we would be able to retrieve all relevant passages for 26 topics i.e. recall at 1, meanwhile for PRES we only managed to achieve the maximum score for 7 topics.

5. CONCLUSION

We have presented a set of query generation methods, many of which outperform state-of-the-art methods for patent

Table 5: Comparison with between post-ranking methods for the nlp1

Method	PRES	Recall	MAP	MAP(P)	Prec(P)
nlp1 MERGED	**0.544**	**0.631**	**0.285**	0.112	0.218
nlp1 - document index only	0.442	0.531	0.172	N/A	N/A
nlp1, NONE	0.429	0.534	0.184	0.112	0.218

Table 6: Comparison of different Query formulations

Run		PRES	Recall	MAP	MAP(P)	Prec(P)
Expanded versus Claim						
best Expanded	nlp1	0.544	0.631	0.285	0.112	0.218
matching Clms	nlp4	0.423	0.510	0.190	0.084	0.171
best Clms	nlp13	0.463	0.574	0.179	0.144	0.270
matching Expanded	nlp12	0.532	0.616	0.249	0.132	0.229
Word & NP versus Word & MWT (exact phrase)						
Best/Matching WrdNP	nlp7	0.499	0.578	0.265	0.103	0.207
Best/Matching WrdTrm	nlp8	0.521	0.614	0.279	0.091	0.186
1 & 2-gram versus 1-gram & MWT (exact phrase)						
Best/Matching stat ngrams	s1	0.492	0.574	**0.300**	0.114	0.208
Best/Matching stat 1grmTrm	s4	0.542	0.628	0.292	0.105	0.204
NLP versus statistic						
best stat	s4	0.542	0.628	0.292	0.105	0.204
matching NLP	nlp8	0.521	0.614	0.278	0.091	0.186
best/matching NLP	nlp1	0.544	0.631	0.285	0.112	0.218
best/matching stat	s5	0.526	0.623	0.278	0.1117	0.207
Ngtv-MWT list versus Pstv-MWT list						
best NgtvTrm	nlp9	0.517	0.596	0.272	0.111	0.221
matching PstvTrm	nlp6	0.533	0.621	0.268	0.116	0.218
best PstvTrm	nlp1	0.544	0.631	0.285	0.112	0.218
matching NgtvTrm	nlp10	0.508	0.590	0.263	0.111	0.216
Mln-MTW list versus IPC-MTW list						
best IPC	nlp1	0.544	0.631	0.285	0.112	0.218
matching Mln	nlp6	0.533	0.621	0.268	0.112	0.218
best Mln	s6	0.541	0.613	0.292	0.106	0.212
matching IPC	s4	0.542	0.628	0.292	0.105	0.204
Word (any word type category) versus Phrases (any phrase type category)						
best Phrs	s7	0.438	0.510	0.215	0.063	0.141
matching Wrd	s8	0.454	0.531	0.212	0.108	0.219
best Wrd	b1	0.536	0.622	0.226	0.132	0.229
matching Phrs	b3	0.431	0.538	0.166	0.082	0.163
30 terms versus 100 terms						
best 100	nlp1	0.544	0.631	0.285	0.112	0.218
matching 30	nlp11	0.484	0.592	0.201	0.098	0.204
best 30	nlp12	0.532	0.616	0.249	0.132	0.230
matching 100	nlp3	0.503	0.591	0.222	0.128	0.261

passage retrieval. We addressed automatic query generation using statistical and linguistic information in combination with domain knowledge. We tested and confirmed several observations made in previous literature. For instance, that phrases, especially NPs, will have a positive effect on the performance of patent text mining applications. We have shown that, for patent text retrieval, the phrase methods need to be combined with words in order to avoid data sparseness. The IPC classification meta-data is a very useful resource for the patent text mining applications. In this paper we used IPC in combination with linguistic and statistical information in order to detect the termhood values of phrases. The key insights of our work are:

- On the matter of *Queries based on statistics versus linguistic methods*: for words, the statistical methods are preferred since the linguistic filter on words is too aggressive (as reported in Section 2.4). For phrases, statistical methods are also preferred, but when combining words and phrases the NLP methods are preferred, (Contrib. 2, Section 1).

- On the matter of *Queries established only from claims versus from the entire document (Exp)*: To only extract terms from the claims section is preferred for precision related evaluation metrics on passage level (Contrib. 3, Section 1). For recall related evalua-

tion metrics on document level the query formulation should use the entire document.

- By combining statistical, linguistic and domain information (IPC) it is possible to identify phrases' termhoodness without computing the C-value (Contrib. 1, Section 1).
- By merging lists from a document index and a passage index we achieved an overall improvement in retrieval performance, regardless of evaluation metrics. The passage index removed noise from the document list, while the document index improved the ranking position for the passages (Contrib. 4, Section 1).

To conclude, given the rhetorical structure of patent documents, as well as the richness and diversity in vocabulary, in order to improve performance of automatic query generation, the genre characteristic needs to be incorporated in the methods. In the future, we will explore our term extraction method on a larger set of documents and also adapt it to other domain specific genres, which have rich metadata information. We will also try to predict which query method to use for a specific topic in order to achieve optimal performance (as shown in Figures 3 and 4). So finally, to answer the question *When is the time ripe for NLP for patent passage retrieval?*, our answer would be using NLP in an IR setting is still too time consuming, in order to become a mainstream method. However, by using linguistic analyses to better understand a text genre is very useful in order to adapt statistic methods, as in the case for learning phrases' Termhoodness. All query search topics and runs are available on the web page[2], in order to contribute to the reproducibility goal within the scientific field of patent retrieval.

6. ACKNOWLEDGMENT

This work has been partly supported by the Self-Optimizer project (FFG 852624) in the EUROSTARS programme, funded by EUREKA, the BMWFW and the European Union.

7. REFERENCES

[1] L. Andersson, M. Lupu, J. Palotti, F. Piroi, A. Hanbury, and A. Rauber. Insight to Hyponymy Lexical Relation Extraction in the Patent Genre Versus Other Text Genres. In *Proc. of IPaMinKONVENS*, 2014.

[2] P. Anick, M. Verhagen, and J. Pustejovsky. Identification of Multiword Expressions in the brWaC. In *Proc. of LREC*, 2014.

[3] A. Burga, J. Codina, G. Ferraro, H. Saggion, and L. Wanner. The challenge of syntactic dependency parsing adaptation for the patent domain. In *ESSLLI-13*, 2013.

[4] B. A. Carterette. Multiple Testing in Statistical Analysis of Systems-based Information Retrieval Experiments. *ACM Trans. Inf. Syst.*, 30(1), 2012.

[5] S. Cetintas and L. Si. Effective query generation and postprocessing strategies for prior art patent search. *J. AM. Soc Info. Tec.*, 2012.

[6] E. Dhondt. *Cracking the Patent using phrasal representations to aid patent classification*. PhD thesis, Radboud University Nijmegen, Netherlands, 2014.

[2]http://www.ifs.tuwien.ac.at/~clef-ip/2013/claims-to-passage

[7] G. Ferraro. *Towards deep content extraction from specialized discourse: the case of verbal relations in patent claims*. PhD thesis, Universitat Pompeu Fabra, 2012.

[8] M. A. Finlayson and N. Kulkarni. Detecting multi-word expressions improves word sense disambiguation. In *Proc. of MWE*, 2011.

[9] K. Frantzi, S. Ananiadou, and H. Mima. Automatic recognition of multi-word terms:. the c-value/nc-value method. *Internat. Journal on Digital Libraries*, 2000.

[10] C. G. Harris, R. Arens, and P. Srinivasan. Using classification code hierarchies for patent prior art searches. In *Current Challenges in Patent Information Retrieval*. Springer, 2011.

[11] D. Hunt, L. Nguyen, and M. Rodgers. *Patent Searching: Tools & Techniques*. Wiley, 2007.

[12] I.-S. Kang, S.-H. Na, J. Kim, and J.-H. Lee. Cluster-based patent retrieval. *Info. Processing Management*, 2007.

[13] K. Konishi, A. Kitauchi, and T. Takaki. Invalidity Patent Search System of NTT DATA. In *Proc. of NTCIR-4*, 2004.

[14] L. S. Larkey. A patent search and classification system. In *Proc. of DL*, 1999.

[15] M. Lease. Natural language processing for information retrieval: The time is ripe (again). In *PIKM '07*, 2007.

[16] J. Luo and H. Yang. Query Formulation for Prior Art Search-Georgetown University at CLEF-IP 2013. In *Proc. of CLEF*, 2013.

[17] W. Magdy and G. J. Jones. Pres: A score metric for evaluating recall-oriented information retrieval applications. In *Proc. of SIGIR*, 2010.

[18] P. Mahdabi, S. Gerani, J. X. Huang, and F. Crestani. Leveraging conceptual lexicon: Query disambiguation using proximity information for patent retrieval. In *Proc of SIGIR*, 2013.

[19] H. Nanba, H. Kamaya, T. Takezawa, M. Okumura, A. Shinmori, and H. Tanigawa. Automatic translation of scholarly terms into patent terms. In *In Proc. of the 2nd Pair workshop*, 2009.

[20] F. Piroi, M. Lupu, and A. Hanbury. Passage Retrieval Starting from Patent Claims A CLEF-IP 2013 Task Overview. In *CLEF 2013*, 2013.

[21] J. Pustejovsky, P. Anick, and S. Bergler. Lexical semantic techniques for corpus analysis. *Computational Linguistics*, 1993.

[22] L. A. Ramshaw and M. P. Marcus. Text chunking using transformation-based learning. In *Natural Language Processing Using Very Large Corpora*, 1999.

[23] E. SanJuan, J. Dowdall, F. Ibekwe-SanJuan, and F. Rinaldi. A symbolic approach to automatic multiword term structuring. *Comput. Speech Lang.*, 2005.

[24] J. Turmo, A. Ageno, and N. Català. Adaptive information extraction. *ACM Comput. Surv.*, 2006.

[25] S. Verberne, E. D'hondt, N. Oostdijk, and C. Koster. Quantifying the challenges in parsing patent claims. In *Proc. of AsPIRe*, 2010.

[26] X. Xue and W. B. Croft. Automatic query generation for patent search. In *Proc. of CIKM*, 2009.