The simpler the better - Retrieval Model comparison for Prior-Art Search in Patents @ CLEF-IP 2013

Andreas Eiselt and Gabriel Oberreuter

Innovandio S.A., Miguel Claro 195, Santiago, Chile

Abstract. Patentability and novelty search is an essential part of any patent application. It ensures that the idea that should be patented has not already been registered anywhere else in the world. However, this task is complicated by the large number of documents and the fact that they are written in many different languages. In this paper we survey four approaches that will help to automate the task and share the insights we have gained through our participation in the CLEF-IP Workshop 2013.

1 Introduction

The level of innovation is one of the principal measures that determines if an idea can be patented or not. In order to estimate it, an exhaustive search in over 80 million patents from more than 100 patent authorities has to be performed. To date, this is still usually carried out with the help of keyword searches. That this strategy does not work is proven by the fact that 54% of the 2.5 million annual patent applications are rejected. The main problem is that the language used in patents is often subject specific as well as inaccurate and misleading. The reason for this is that the applicants are usually not interested in making their ideas public and therefore try to disguise them as good as possible. Another problem is that patent documents may be written in many different languages, which makes it even harder to find similar ideas.

For our participation in the patentability and novelty search task of the CLEF-IP workshop, which was to automatically find all documents and their respective passages that describe concepts strongly related to those explained in the source document, we explored the applicability of four approaches that will be explained in the following chapter.

2 Task, Corpus and Evaluation

For the CLEF-IP *patentability or novelty search* task all participants were provided with a corpus divided into two sets of patent documents: the first set D_{pat} contains 3.118.088 patent documents (2.680.604 from the EPO and 437.484 from the WIPO) and the second set D_{app} 210 patent documents (belonging to 69 patents). Furthermore a set of 149 topics T was given. Each $t_i \in T$ was defined

as a subset from the claims of one patent document $d_i \in D_{app}$. The basic task was, given a topic t_i, find those passages p_n in $d_{pat} \in D_{pat}$ that are semantically related.

The results were then evaluated on document as well as on passage level. On document level the *Patent Retrieval Evaluation Score* (PRES) [1] was used with a cutoff at 100, while the evaluation on passage level was based on the *mean average generalized precision* (MAgP) [2].

3 Experiments & Results

In order to obtain the most relevant documents and passages from D_{pat} for t_i, the retrieval process was divided into two stages: candidate-retrieval and detailed comparison. During the candidate-retrieval the most relevant documents from D_{pat} were selected and subsequently a detailed comparison was performed to determine the most relevant passages.

In order to reduce the space of possible candidates and improved retrieval quality, we only considered patents from D_{pat}, which shared at least one *International Patent Classification* (IPC) with the patent containing t_i. We furthermore used abstracts, claims and descriptions from all patent family members, since the text contained in t_i was too short.

As the amount of patents in the workshop task, as well as in a real-world scenario is limited, in three of the four approaches the candidate-retrieval was performed by calculating the text similarity between t_i and all possible candidate documents from D_{pat}. The detailed comparison at passage level was executed subsequently between t_i and the top 100 candidates from D_{pat} using the same similarity measure. As text similarity measure, we evaluated 3 approaches which were all based on the Vector-Space Model (VSM) and the cosine similarity: (i) Word uni-grams (ii) Character tri-grams, (iii) Cross-Language Explicit Semantic Analysis (CL-ESA) . The first approach was based on the idea of simply comparing the used vocabulary, ignoring the fact that some documents are written in other languages. This approach was considered the best approximation for a keyword-search, a strategy commonly used by humans to generate a patent search-report. The second approach was based on the findings that two text documents written in different european languages have a strong character N-gram overlap [3]. The third approach is known as Cross-Language Explicit Semantic Analysis (CL-ESA) [4]. It represents a document as a vector of similarities to the documents of a multi-lingual reference corpus. This allows to compare documents on a semantic level, independent of the language in which they are written.

In comparison to the first three approaches, the fourth was based on a heuristic candidate retrieval. Therefore, we generated a set of queries for each source document and executed them on a search engine, which had all documents from D_{pat} indexed. For the top candidate documents we then executed a detailed comparison based again on word uni-grams.

The results of each submitted run, are presented in Table 1. They show, that

Table 1. Results of the submitted runs; *eiselt-cos*: uses word uni-grams; *eiselt-solr* uses heuristic candidate retrieval; *eiselt-c3g* uses character tri-grams; *eiselt-clesa* uses Cross-Language Explicit Semantic Analysis

Run	*Document Level*			*Passage Level*	
	PRES@100	R@100	MAP	MAP(D)	Precision(D)
eiselt.cos	**0.30**	**0.38**	**0.11**	**0.12**	0.08
eiselt.solr	0.26	0.37	0.08	0.11	**0.10**
eiselt.c3g	0.23	0.31	0.10	0.10	0.07
eiselt.clesa	0.21	0.29	0.05	0.10	0.08

the simplest approach (cosine similarity between vectors of word uni-grams) outperforms any other approach. This can be explained by the fact that it guarantees a higher ranking for documents with similar vocabulary. The same documents will get a higher score in case of an intelligent keyword-search as it is typically executed by humans. This also explains the good results of the heuristic candidate retrieval, as it aims to imitate humans behaviour too. That the approach based on character tri-grams did not bring the expected advantage is due to the fact that documents of the same language still share a lot more character n-grams than semantically related documents in different languages. The interpretation of the results obtained using CL-ESA may require further investigation. They show that this approach is, out of the four, the worse approximation for human search behaviour. Nevertheless, they do not reflect necessarily a bad performance. Preliminary investigations of the results have shown that CL-ESA assigned a higher rank to documents which seem to be highly related and which did not appear in the result-set of simple keyword-searches and neither in the patent search report. Hence, in order to obtain a better idea of the result-quality, it would be necessary to manually judge the relatedness of the top-ranked results.

References

1. Magdy, W., Jones, G.J.: PRES: a score metric for evaluating recall-oriented information retrieval applications. In: Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval - SIGIR '10, New York, New York, USA, ACM Press (2010) 611
2. Kamps, J., Pehcevski, J., Kazai, G., Lalmas, M., Robertson, S.: Inex 2007 evaluation measures. In Fuhr, N., Kamps, J., Lalmas, M., Trotman, A., eds.: Focused Access to XML Documents. Volume 4862 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2008) 24–33
3. McNamee, P., Mayfield, J.: Character n-gram tokenization for european language text retrieval. Information Retrieval **7**(1-2) (2004) 73–97
4. Potthast, M., Stein, B., Anderka, M.: A Wikipedia-Based Multilingual Retrieval Model. In Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., White, R., eds.: Advances in Information Retrieval. 30th European Conference on IR Research (ECIR 08). Volume 4956 of Lecture Notes in Computer Science., Berlin Heidelberg New York, Springer (2008) 522–530