

Examiner Assisted Automated Patents Search

Arthi Krishna, Brian Feldman, Joseph Wolf, Greg Gabel, Scott Beliveau and Thomas Beach

United States Patent and Trademark Office, Alexandria, VA, USA
{Arthi.Krishna,Brian.Feldman, Joseph.Wolf, Greg.Gabel, Scott.Beliveau, Thomas.Beach}@uspto.gov

Abstract

One of the most crucial and knowledge-intensive steps of patent examination is the determination of prior art – evidence that the idea claimed by a patent is already known. Automated prior art retrieval algorithms, if effective, can assist expert examiners by identifying literature that would otherwise take substantial research to uncover. Our approach is to build a patent search algorithm which functions as a cognitive assistant to the patent searcher. Contrary to the approach of treating the search algorithm as a black box, all components of the search algorithm are explained, and these components expose controls that can be adjusted by the user. This level of transparency and interactivity of the algorithm not only enables the experts to get the best use of the tool, but also is crucial in gaining the trust of the users. In this paper we discuss the engineering of the cognitive assistant search tool, referred to as Sigma, and the various interactions it affords the users. The tool is currently being piloted to patent examiners in the unit 2427.

Introduction

Searching for prior art is one of the most time-intensive aspects of patent examination because the process of deciding if an already existing patent or publication is a suitable prior art is a complex one. Fully automated prior art retrieval systems are challenged by the technical content of the patents and the subtleties in interpretation of patent laws, which are influenced by recent court decisions. Whereas the human user is confronted by the sheer volume of the literature, which makes thoughtful inspection of candidate results time consuming.

Our approach is to build an automated prior art search system that can assist patent examiners in finding and making decisions on prior art. We have built a search system that not only performs the basic keyword searches, but also has several layers of augmented processing that can be controlled and modified as the search progresses. By designing a user interface that allows the expert to alter the behavior of search algorithm, for instance defining the relative

weights of different sections of the patent (e.g. title, claims, specification and abstract), experts can create strategies of patent retrieval algorithms best suited to examining a particular application. Additionally, the user interface has quick visual cues that provide immediate feedback regarding the quality of the patent search results. For example, visual indicators show whether the input patent belongs to the same “patent family” as the result, thereby helping the user eyeball the quality of the search. By engineering a prior art search algorithm with the sole focus of transparency and control by the users, we aim to achieve both a more accurate tool and also a greater user acceptance of the tool. The user interface design and a preliminary study of the effectiveness of the Sigma tool has been recently published (Krishna et al. 2016). This paper describes the architecture and natural language processing techniques used to create the prior art search algorithm.

Search Algorithm Overview

The core of the search tool is implemented using Apache Solr Lucene (Smiley et al. 2014) open source search system with cloud capability (version 5.4). As will be detailed in the following section, we use UIMA to preform text analysis and this pre-analyzed stream is then stored in Solr Lucene. Lucene uses term frequency inverse document frequency (tf-idf) (Wu et al. 2008) calculations for text relevancy and retrieval. Among Solr’s various built-in query parsers, such as Simple and Field query parsers, the MoreLikeThis query parser, which can take an entire document as the input and retrieve other documents similar to it, is the one we choose for this implementation. The MoreLikeThis parser finds the top unique terms in the input document, and uses these terms to retrieve related documents. A number of customizations are allowed by MoreLikeThis at runtime, such as the number of terms to be searched and the “boost” or importance of

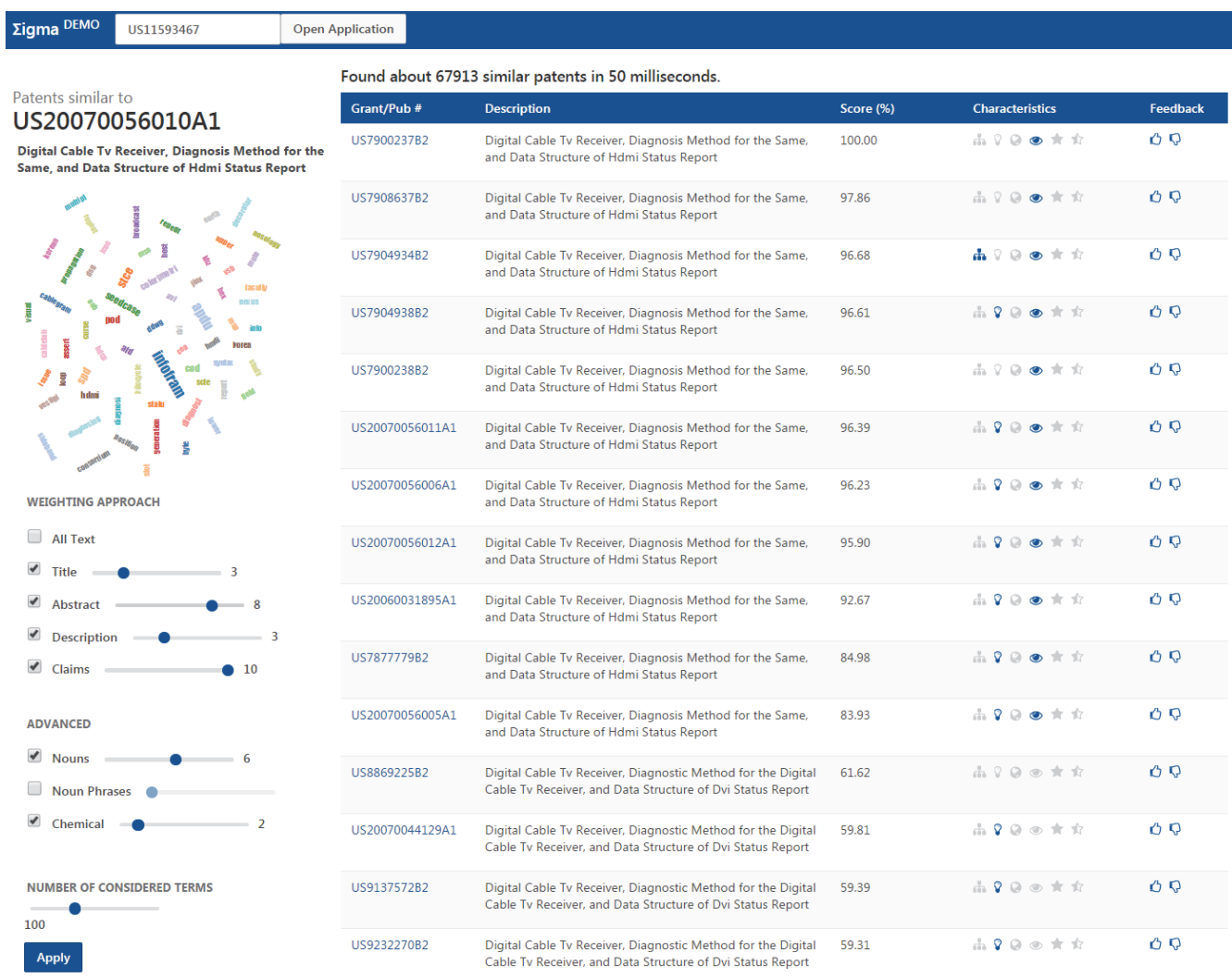


Figure 1. Sigma Prior Art Search Tool

Solr fields (e.g. abstract - 10). We heavily exploit the customization features to build a tool that the user can use to modify and fine tune.

UIMA Processing

Though the core of the search system tool is implemented using Apache Solr Lucene, most of the patent specific logic is done in the preprocessing stage leveraging the Apache UIMA framework (Ferrucci 2004) in conjunction with several open source tools. The UIMA pipelines were defined in Java using uimaFIT 2.2.0 (Ogren 2009). This affords us programmatic/dynamic instantiation of UIMA components, instead of manually building UIMA xml descriptor files. Using JCas, the Java Cover Classes based Common Analysis System (CAS) API, several layers of UIMA annotations are built as described in this section.

Input Patent Xml

The patent text of granted United States patents and pre-grant published patent applications are available in the public corpus (<http://patents.reedtech.com>). There is considerable variation in the xml formats of these documents published over the years. As a part of this effort, code developed to parse several patent xml variations into Java objects can be found on <http://github.com/USPTO/PatentPublicData>.

OpenNLP Parser

We use OpenNLP (Manning et al. 2014) parser trained with the Penn Tree bank (Marcus, Santorini, and Marcinkiewicz 1993) to find sentence boundaries. The sentences are then chunked and parts of speech are annotated, as shown in the code snippet. The parts of speech are used in other more

complex annotators, such as the acronym annotator (described later in the paper), as well as being exposed to the users for search, e.g. Nouns and Chemicals.

```
Path posModel = resourceDir.resolve("opennlp/en-parser-chunking.bin");

AnalysisEngineDescription pos = AnalysisEngineFactory.createEngineDescription(Parser.class, UimaUtil.SENTENCE_TYPE_PARAMETER, "gov.uspto.uima.type.Sentence",

UimaUtil.TOKEN_TYPE_PARAMETER, "gov.uspto.uima.type.Token", Parser.PARSE_TYPE_PARAMETER, "gov.uspto.uima.type.Parse", Parser.TYPE_FEATURE_PARAMETER, "pos", Parser.CHILDREN_FEATURE_PARAMETER, "child", UimaUtil.PROBABILITY_FEATURE_PARAMETER, "probability");

ExternalResourceFactory.createDependencyAndBind(pos, UimaUtil.MODEL_PARAMETER, ParserModelResourceImpl.class, "file:" + posModel);
```

Synonym

Several factors, such as new inventive concepts and the purposeful introduction of “abstract vocabulary” (Verberne and Oostdijk 2010), result in patents containing a large number of novel terms. Examiners tend to overcome the challenge of finding relevant literature by knowing synonyms that are specific to their particular field of expertise. Synonyms of nouns found in the patent text are retrieved using APIs pro-

vided by WordNet (Fellbaum 1998) and Wiktionary; the resulting synonym set is stored in Solr fields corresponding to the patent section in which the noun was found, e.g. abstract_nouns.

Computer: {data processor, computing device, computing machine, electronic computer, information processing system}

VOD (video on demand): {video, movie, demand, content, stream, media}

Based on the search queries submitted by examiners to the internal search system, we have recently been able to predict synonyms of words through conditional probability of co-occurrence in queries. Also, by limiting our calculations to only those search queries generated by examiners within a specific technical field, the synonyms have been more accurate. We have also added a feature by which users can input their own set of synonyms.

Chemical

Open-Source Chemistry Analysis Routines, OSCAR4 (Jesop et al. 2011), software is a tool kit that is used to recognize named entities of chemicals. OSCAR4 API is invoked through the UIMA framework, and chemical formula and structure are annotated. The standard InChI (Stein, Heller, and Tchekhovskoi 2003) chemical notation is added to a newly created Solr field, chemicals, for each document. For example, a patent with either water or H₂O gets the InChI value 1S/H₂O/h1H₂ added to the chemical field.

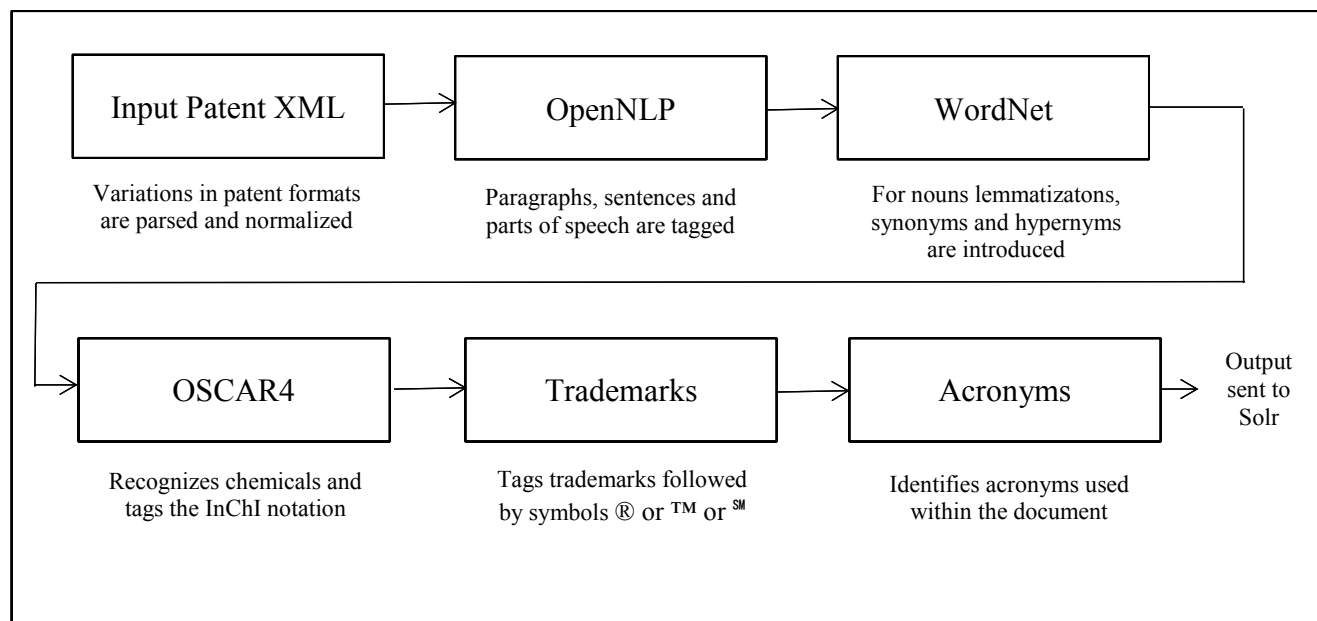


Figure 2. UIMA Processing Pipeline

Trademark

Nouns or sequences of nouns with a trailing trademark symbol (® or TM or SM) are annotated and introduced as values in a designated trademark Solr field. These fields can then be compared for similarity. Some examples of trademarks extracted include blu-ray, firewire, dolby and pentium.

Acronym

Technical documents often introduce acronyms that are heavily reused within the document. We have created an acronym expansion finding script that identifies acronyms and the corresponding expansions within the document. Some examples include: ep_g - electronic programming guide and http - hypertext transfer protocol. Every time the acronym occurs in the text, it is reinforced with the expansion so that similar documents can be retrieved with more accuracy.

User Interactions

As a cognitive assistance tool, Sigma’s approach to enable and work with the users is twofold. Firstly, Sigma exposes the underlying controls of the search algorithm in order to provide transparency and control of the algorithms to the end users. This section details the primary methods 1. Boosts and 2. Interesting Terms by which user are empowered to control the algorithms. The second principle of machine and users sharing feedback will be discussed in the following section.

Solr/Lucene Field Boosts

Solr/Lucene MoreLikeThis algorithm affords the flexibility of passing boost values for select Solr fields through the payload option. Taking advantage of this run time feature, we expose a number of core Solr fields to the user. As shown in Figure 3, the different sections of the patents—abstract, claims, title and description—are options that can be checked or unchecked, and the sliders next to them control the weights. The flexibility of choosing the patent sections to compare is a powerful tool, as for example the claim-to-claim comparison of patents can provide insights into double patenting, a phenomenon where the same invention is attempted to be patented more than once by the applicant.



Figure 3. User Controls for Alternating Weights of Patent Sections

Using the same mechanism, we also expose patent specific processing done through UIMA. Nouns, synonyms, noun phrases, trademarks and chemicals are some of the NLP generated fields that can be controlled similarly by the users.

Interesting Terms


Select words from the patent application are used for matching with the corpus of patents available. As we have seen in the previous sections the weights influence the ranking of these words. Another feature offered by the more like this algorithm is the ability to expose and alter these words.

The words chosen and their corresponding weights are displayed as a word cloud at the left margin of the interface. The user are able to choose the number of top words to consider, and we limit it to the 25-500 range in order to avoid overloading the system. Furthermore using a filter query, we are able to refine the initial search results to explicitly exclude any of the words the user wants to avoid. We allow the user to input a limited list of synonyms, these synonyms are expanded at query time.



Figure 4. Interesting Terms of Patent US11061715 Visualized As Tag Cloud

User Feedback

Providing quick feedback of the quality of the search results is the key to guiding the prior art search process in the right direction. Patent examiners have a few heuristics that help them determine if two patents are related. Each result has a number of indicators, corresponding to these heuristics, which visually indicate how the result is related to the searched patent application. If a patent is listed as being in the same family as the patent application,  indicator is colored in. Patent documents within a family tends to be the

most related to each other with a high degree of similarity. The CPC (Cooperative Patent Classification) and USPC (United States Patent Classification) indicators are shown active if the patent application shares at least one CPC or USPC class respectively with the patent application. Any overlap in the patents cited by the result and the patents cited by the patent application results in the reference indicator being on. Similar patent application cases tend to get assigned to examiners in the same department or art unit. The shared art unit indicator reflects if the input patent application and the result have been assigned to examiners in the same art unit. The patent applicants provide references of patents they consider relevant to the case during the filing of the application, if any of those are in the results they are indicated by the half star. For patent applications that have already been processed, references cited by examiners during the prosecution process are also available and indicated by the star icon. Users are able to indicate if a result is appropriate using the thumbs up indicator; these results are treated as examiner cited references in evaluating the quality of the search results.

Performance

A preliminary study of this tool (Krishna et al. 2016) showed that different variations in the search algorithms were optimal for the seven different technical areas that were tested, giving us confidence that no one setting of the parameters is optimal across all technical areas.

The Sigma tool has been stood up on one Amazon Web Services (AWS) m3 server with 2 CPU, 3.75 GB RAM and 32 GB attached storage. A test corpus of 100500 patent documents with 60300 granted patents and 40200 pre-grant publications was used to run preliminary performance tests. The search usually responded in under a second with an average of 253ms for a random sample of 100 test cases. The service calls to determine status of user feedback icons (patent family etc.) take substantially longer and the page is typically rendered in 3-5 seconds.

Conclusions and Future Work

Designing a patent search system that acts as a cognitive assistant to the patent examiner enables us to leverage the nuanced skill developed by patent examiners. Currently we are working closely with examiners in unit 2427 to learn other strategies for finding prior art. We plan to expand the test group to 4 units by the end of this year. Based on our interactions, we are working on features requested by the examiners, such as performing searches for only a subset of the patent's ideas, termed as 103 searches.

Acknowledgements

We would like to thank David Chiles, Michael Henderson, Star Ying and Brigit Baron for their support.

References

- Krishna, A., Feldman, B., Wolf, J., Gabel, G., Beliveau, S., Beach, T. 2016. User Interface for Customizing Patents Search: An Exploratory Study. *HCI* (26) 2016: 264-269
- Smiley, D., Pugh, E., Parisa, K., Mitchell, M. 2014. *Apache Solr 4 Enterprise Search Server* (1st ed.). Packt Publishing.
- Wu, H. C., Luk, R. W. P., Wong, K. F., Kwok, K. L. 2008. Interpreting TF-IDF term weights as making relevance decisions. *ACM Transactions on Information Systems*.
- Verberne, D'hondt, Oostdijk, Koster. 2010. Quantifying the challenges in parsing patent claims. *Proceedings AsPIRe*
- Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press
- Ferrucci D, Lally A. 2004. UIMA: an architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*. 10(3-4):327-48.
- Manning, Christopher D., Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. *In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55-60.
- Marcus, Mitchell; Santorini, Beatrice; and Marcinkiewicz, Maryann. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*. 19(2).
- Ogren P, Bethard S. 2009. Building Test Suites for UIMA Components. *In Proceedings of the Workshop on Software Engineering, Testing, and Quality Assurance for Natural Language Processing (SETQA-NLP 2009)*. Association for Computational Linguistics, Boulder, Colorado: 1-4.
- Jessop D, Adams S, Willighagen E, Hawizy L, Murray-Rust P. 2011. OSCAR4: a flexible architecture for chemical text-mining. *J Cheminform* 3:41.
- Stephen E. Stein, Stephen R. Heller, and Dmitrii Tchekhovskoi. 2003. An Open Standard for Chemical Structure Representation: The IUPAC Chemical Identifier, *In Proceedings of the 2003 International Chemical Information Conference* (Nimes), Infonortics, pp. 131-143.