Exploration of different similarity and modeling techniques in patent information retrieval for patent similarities

Marcus Thuillier
Engineering Department, Northwestern University

https://github.com/MSIA/mtx2701_msia414_2019/tree/Project

Abstract

Prior art search in patent law is a necessary first step when it comes to establish the novelty and innovation of a patent. This challenge is currently query based and software using context clues to identify similar patents are rare. Some of the complexities of finding similarities go beyond just looking for key words however and adopting a context-based approach could show promising results. I experimented with different methods to text mining and similarity measures. The methods attempted are a Tf-Idf vectorizer, Word2Vec, Doc2Vec and LDA models as well as a universal sentence encoder. The similarity measures tried are cosine similarity, word mover's distance and Jensen-Shannon distance. The models are evaluated on the time it takes to train and test them as well as their accuracy performance. Some of simpler models like Tf-Idf vectorizer performed as well, if not better, than some more complex models like Doc2Vec or universal sentence encoders and benefitted from the short train and test times. Further evaluation and testing are needed, but simple context aware methods seem to address the problem appropriately.

1 Introduction

Prior art search for patent lawyers and examiners is a complex and difficult task. Often, the search focuses around some key words or the name of the technology or company. Some software has been developed to try to incorporate context-based queries to get more accurate results during prior art search, but this approach is not widely spread or available. Rather than looking at patent claims or patent titles, it was decided to use the United States Patent and Trademark Office patent descriptions. Those descriptions provide plenty of text and context for each patent and there is hope that this will allow for greater accuracy in information retrieval to identify similar patents. It is left to be decided which method to use to build a model and determine the similarity between those descriptions and a user input, which is what this paper will demonstrate.

2 Related Work

Information retrieval for patents and Intellectual Property documents can be an important source of knowledge for strategic planning purposes. Ernst (2003) establishes the use of this information in different areas of technology management such as competitor monitoring or R&D portfolio management. Developing a reliable methodology to retrieve relevant patent data can provide valuable insights for a company's decision making.

Text retrieval and text analysis performed on patent documents have different objectives. One of them, the so-called "invalidity search" has been explored in Fujii (2007), who found promise in using topic-sensitive citation-based information retrieval methods in finding prior arts related to a patent application and their claims. Another objective, which is the one at interest in this paper, is the "technology survey". The technology survey

is done through querying usually in order to determine the specific technology of the patent. This kind of work is presented in Larkey (1999) who introduces a patent search and classification system. The search uses a tool named "Inquery", a probabilistic information retrieval system based on Bayesian networks and using Tf-Idf for its weighting. The classification is then done using a KNN algorithm. Similar work is done by Kang et al. (2007), who use cluster-based language model to determine the correct International Patent Classification (IPC) codes. The language modeling solution described in this paper views a text query as a random sample from each document and at retrieval, the documents are ranked by query likelihood. The query likelihood is found by treating queries as a sequence of independent terms in a multinomial view, then estimating a unigram language model for each document (simple maximum likelihood) with smoothing to deal with unseen terms. This paper also notes that these techniques do not show substantial improvements over non-cluster language modelling methods since they rely on the assumptions that all relevant documents are within an IPC code, which might not always be the case.

Going into the problem at hand, we also look at Bonino et al. (2010) which outlines some of the techniques used to address the challenges in patent information retrieval. The two main methods described by the paper are Latent Semantic Analysis for related documents retrieval and ontologies based on domain knowledge in semantic searches of patents. Those different ontologies can be document ontologies, modeling the structure or metadata of patent documents, domain ontologies, such as the patent classification ontology or linguistics ontology such as WordNet. These different methods will be further explored when building our methodology. Finally, Gomaa and Fahmy (2013) summarized different methods to determining text similarity. This paper presents popular methods for finding semantic similarities, specifically with corpus-based and knowledge-based algorithms. This paper will attempt to implement some of these algorithms to the topic of patent similarities.

3 Dataset

The data for this project is acquired through the PatentsView database[1] of the United States Patent and Trademark Office. The database consists of information on published patent applications from 2001 until today and granted patents from 1976 until today. The API is publicly available and was created through the Office of Chief Economist in the US Patent and Trademark Office. The dataset is over 50 gigabytes and about 6 million rows where each row is a unique patent with a patent description. According to execution time considerations described in the results, a sample of 100,000 patents is taken in which the patent descriptions have lengths anywhere from 30 words to 2423 words. The patent descriptions will be altered in a process which will be described in the method part of this paper.

4 Method

This paper attempts to approach the problem with several methods[2], which will be described in this part. The first task comes in formatting the data correctly. The text data for patent descriptions is preprocessed. Any non-alphanumeric characters are filtered out, as well as any extra whitespace. All characters are lowered. An additional step is taken to optimize the run-time of the models. The first 5 sentences of each description are selected, and the rest of the description is discarded. Most of the details on the patent are in these sentences, with the rest the description addressing figures and more minute details. After this step, the words are tokenized, and all stop words are removed using the NLTK python library. Any description less than 30 words is discarded to aid the models. Finally, the words are lemmatized to group together different inflections of a word to be analyzed as a single item with context. Now the texts are ready for analysis.

The first method is a vector space approach, where Tf-Idf vectorization is combined with

[1] PatentsView Website

[2] Text Similarity Approaches

cosine similarity to identify similar patents[3]. Once the patent descriptions are fed in, we use the Tf-Idf vectorizer function from the python library sklearn and try it out with both unigram and unigram/bigram, before using cosine similarity to determine the most similar patents.

A second method will use cosine similarity as well but create a gensim doc2vec model for comparison of patents. The Doc2Vec[4] model will be tried and presented in the results section. After the model is built, cosine similarity is used to get the most similar patent to the input description.

Although we expect doc2vec to perform better as we are dealing with full descriptions of patents and not just sentences, we wanted to try a word2vec[5] model as well. This time, the similarity metric will be the word mover distance[6], the "minimum traveling distance between documents", which will need to be minimized. After the model is build and returns WMD for each description, we sort by smallest value to get the most similar patent.

Another method tried will use Latent Dirichlet Allocation[7] to find patent similarities. To compute the similarity after training the LDA model, the Jensen-Shannon distance will be used[8], with the smaller distance corresponding to the "closest documents statistically".

Finally, the last method will be created with the use of sentence encoder of tensorflow-hub[9]. After loading encoders from the hub, the embeddings for the patent descriptions are retrieved and similarity between input descriptions and the embeddings is again determined by cosine similarity.

5 Results

Now that we've established the five methods to be undertaken, we will test them and try to determine the best one. Two aspects of the model will be tested. The first will be the time required to both train and test the function. We assume we are able to save the model after training, which will be timed, and that it can be retrieved and used with a test patent description, a process that will also be timed. The second aspect of the model tested will be its accuracy, which is defined as such: we have with the help of a paraphrasing tool changed the descriptions for some of the patents in the database. We want to see if the models are able to print out the original patent with the modified input as one of the top similar patents. A further set of testing is considered to be done with a patent professional who would be able to determine the actual similarity between input and outputs and will be pursued in the future.

	method 1 Tf-Idf & Cos	method 2 doc2vec & Cos	method 3 word2vec & WMD	method 4 LDA & JSD	method 5 Encoder & Cos
10,000 docs	0.02	0.34	1.71	1.42	0.47
100,000 docs	0.25	3.16	17.87	17.41	2.89
250,000 docs	0.82	12.13	66.09	61.03	Runs out of memory
500,000 docs	1.64	23.00	Runtime error	Runtime error	Runs out of memory

Table 1 – Training Time in minutes for all methods

method 1	method 2	method 3	method 4	method 5
0.047	0.055	17.34	2.49	0.049

Table 2 – Testing Time in minutes with 100,000 documents with 1 test description for all methods

	method 1 Unigram	method 1 Unigram and Bigram	method 2	method 4	method 5 (n=45)
Top similarity	74%	80%	82%	0%	69%
Top 3	84%	90%	86%	0%	78%
Top 5	88%	90%	86%	0%	80%
Top 10	90%	94%	90%	0%	82%
Top 50	96%	96%	92%	0%	93%
Top 500	100%	100%	96%	2%	98%

Table 3 – Model Performance (on 100,000 documents) (n=50) for different methods

[3] Tf-Idf and Cosine Similarity
[4] Doc2Vec
[5] Word Mover Distance
[6] Word2Vec and WMD
[7] LDA
[8] LDA and JS Distance
[9] Sentence Encoder

In Table 1, all of the methods seem to scale linearly in training. This favors method 1 which is the shortest to train. Only methods 1 and 2 were able to handle upwards of 250,000 documents, while methods 3 and 4 run for a very long time and method 5 uses up the memory. We will therefore sample 100,000 random documents for testing of the methods.

In table 2, it appears the length of computing word mover distance is what makes method 3 so slow, while the computing of Jensen-Shannon distance makes method 4 slower than others as well. Computing the cosine similarity is almost instantaneous. It could be considered in the future to switch out WMD and JS distance for other similarity measures for methods 3 and 4, to improve runtime and be able to compare performance on bigger datasets.

In table 3, the third method was not evaluated with the test set due to resource limitations. Method 4 also ran into memory issues with runtime, but the results did not look promising. Therefore we focused on methods 1, 2 and 5. If a model returned the corresponding patent number first in similarity, it is counted as "top similarity", and so on for top 3, 5, 10, 50 and 500. With this testing approach, although the second method returned the top similarity the most often, a simple Tf-Idf method using unigram/bigram looked to be comprehensively the most promising.

6 Discussion

Different methods to address the problem of context-based information retrieval in patents data were presented. A dataset with extensive description of the patents is used to train models with the intention of returning similar patents from queries. Five main modelling methods are used in the paper, as well as three different measures of similarity. Two options suffered in their scalability because they used similarity measures that calculate distances between vectors. Using Word2Vec and LDA did not seem to be "wrong" methods per se, but to be scalable and useful with such a dataset, other similarity measures would need to be used instead of WMD and JS Distance. The three other methods used cosine similarity as their similarity measure. All three methods produced fairly similar results considering the sample size and conditions of testing. As there is no established measure of accuracy for similarity problems, it was decided to use paraphrased text to see if the algorithm could find the original patent in a database of 100,000 documents. The unigram/bigram Tf-Idf model performed the best in this setting, returning the corresponding patent to the input description 90 percent of the time as a top three similar patent. However, this paper should only be considered a first step in evaluating these methods. Both Doc2Vec and Sentence Encoder also showed promise, although they suffered from longer training times. The results shown in this paper highlight that context-based information retrieval is possible when looking at patents. Although there is no definitive answer as to what method is the best, this paper has shown that even fast and simple methods such as Tf-Idf and cosine similarity are efficient. It is likely that more complicated methods might be more effective in identifying context-based similarities overall, but this would have to be determined by a patent professional. Thus, it is important that the work started here be further developed with the current dataset and methods. The results of continued work on this could end with a solution outperforming the existing simple word query-based patent prior art research that patent lawyers and examiners have to use.

References

Atsushi Fujii. 2007. *Enhancing Patent Retrieval by Citation Analysis*.

Dario Bonino, Alberto Ciaramella and Fulvio Corno. 2010. *Review of the state-of-the-art in patent information and forthcoming evolutions in intelligent patent informatics*.

Holger Ernst. 2003. *Patent information for strategic technology management*.

In-Su Kang, Seung-Hoon Na, Jungi Kim and Jong-Hyeok Lee. 2007. *Cluster-based Patent Retrieval*.

Leah S. Larkey. 1999. *A Patent Search and Classification System*.

Wael H. Gomaa and Aly A. Fahmy. 2013. *A Survey of Text Similarity Approaches*.