Recommending Patents based on Latent Topics

Ralf Krestel
Department of Computer Science
University of California, Irvine
krestel@uci.edu

Padhraic Smyth
Department of Computer Science
University of California, Irvine
smyth@ics.uci.edu

ABSTRACT

The availability of large volumes of granted patents and applications, all publicly available on the Web, enables the use of sophisticated text mining and information retrieval methods to facilitate access and analysis of patents. In this paper we investigate techniques to automatically recommend patents given a query patent. This task is critical for a variety of patent-related analysis problems such as finding relevant citations, research of relevant prior art, and infringement analysis. We investigate the use of latent Dirichlet allocation and Dirichlet multinomial regression to represent patent documents and to compute similarity scores. We compare our methods with state-of-the-art document representations and retrieval techniques and demonstrate the effectiveness of our approach on a collection of US patent publications.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval Models, Selection Process, Search Process

Keywords

Citation Recommendation, Patent Retrieval, Document Ranking, Topic Models, Language Models

1. INTRODUCTION

Millions of patent documents are publicly available electronically. Analyzing these documents can help in our understanding of technological progress, the evolution of new language and terms, and the emergence of new products. Tools for automated patent analysis also have direct benefits to inventors in terms of finding relevant prior work, for companies wishing to patent new products or ideas, and for patent examiners in deciding which patents to grant.

Being able to efficiently and accurately search large patent databases has been a problem for decades. For example in the early 1950's, a mechanized way of finding relevant patents based on punch cards was developed [3]. Since then, information retrieval techniques have been used to provide more assistance in patent search, allowing for searching of large databases in seconds across multiple languages [15]. Systems that support this type of keyword-based search include both commercial systems such as Dialog or LexisNexis[1], as well as systems used in patent offices[2]. Despite this progress, searching patent databases, either by inventors, lawyers, patent examiners, or analysts, still heavily relies on Boolean keyword queries [2] and often requires considerable legal expertise and domain knowledge.

In the context of patent retrieval, different types of search scenarios are relevant depending on the specific tasks that occur at different stages in the life of a patent [2, 5]:

- When writing a patent a key issue for an inventor is the identification of prior art that needs to be referenced—this motivates the development of automated citation recommendation to support the patent writing process.
- Once a patent is written and submitted to a patent office, a detailed search of relevant prior patents is essential for patent office examiners to decide on a submitted patent's novelty, similarity, relevance, and patentability [8, 11].
- The issue of finding relevant prior art becomes important again if the validity of a patent is evaluated at a later stage (e.g., as part of an infringement lawsuit) to decide whether a similar patent already exists.
- More broadly, commercial entities are often interested in patent analysis for strategic competitive reasons and protecting their intellectual property.

Across all of these tasks, finding patents that are similar to the query patent is a crucial component of the problem. For example, accurately recommending relevant citations could save significant time and resources. In contrast to recommending scientific articles, the problem of automatically recommending patents poses some distinct challenges. For example, language usage in patents tends to be more idiosyncratic and less consistent compared to technical language usage in scientific articles [1]. Furthermore, the citation graph for patents conveys different information than in the case of scientific articles. For example, patent citations do not occur within the text of the patent but are provided as a separate list, often consisting of both inventor-generated and examiner-generated sublists.

[1] http://www.dialog.com, http://www.lexisnexis.com
[2] e.g. USPTO http://www.uspto.gov

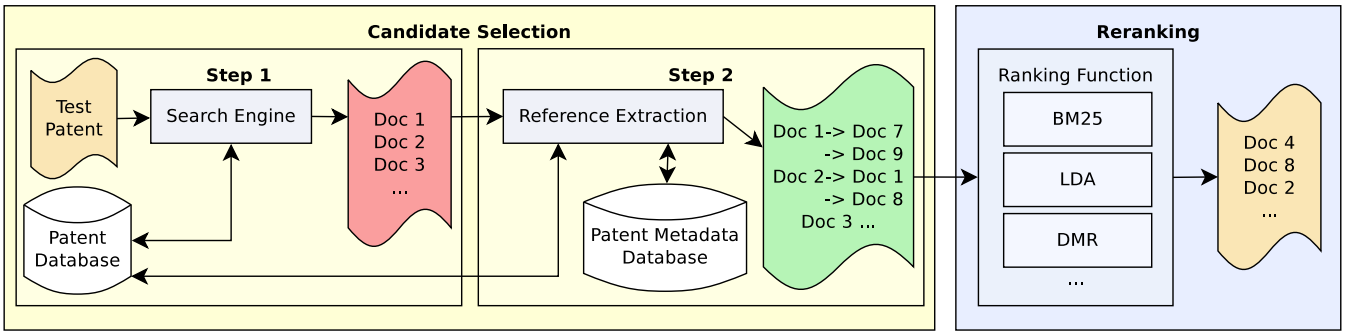

Figure 1: System architecture overview

2. FINDING RELATED PATENTS

We address the problem of finding related or similar patents by investigating a combination of (1) latent Dirichlet allocation (LDA) [4], (2) Dirichlet multinomial regression (DMR) [14], and (3) language models [6]. The combination of topic models and language models, especially when used for document smoothing, has been shown to be beneficial in prior work on document ranking (e.g. [18]). Latent topics can capture concepts and general terms, whereas language models are useful for capturing specific terms and details [10]. In the following we investigate the use of combinations of these models for the specific problem of patent retrieval.

The problem we address is as follows: given a granted patent q, return a ranked list of patents and patent applications $R=\{d_1, d_2, \ldots, d_n\}$, all published before the application date of q, ordered by the degree of similarity to q. The highest ranked patents d_i are considered to be most similar to q.

System Architecture.

Figure 1 shows the overview of our system's architecture. In a manner similar to that of Mase et al. [13], we use a two-stage approach; first generating a list of candidates and then using different ranking functions to determine the most similar patents.

The first stage consists of two steps:

1. To find the candidates we use a search engine to determine the top-K most similar patents to the test patent based on tf-idf.
2. We use the top N terms (with N=30)of the summary field of the test document to construct a query to retrieve the top-k most similar patents and patent applications (as suggested by Xue and Croft [17]). To overcome some limitations of this tf-idf based retrieval we extend the list of retrieved documents by extracting citations within the top-K returned documents. If the citation is another USPTO patent document then we add the cited document to our candidate set for ranking.

In the second stage, after the generation of the candidate set in Step 2, we re-rank the candidate patent documents using combinations of language models and topic models. To provide an example of the number of patents returned in a candidate set, when using K=500 in step 1, an average candidate set produced from step 2 consists of around 2,700 patents.

Baselines.

We compare our algorithms with two baseline algorithms: (1) BM25 for long queries [12] and (2) language models (LM) [6]. To compute similarity scores for the test patent relative to each candidate we treat the test document as the query q and the candidates from our two-step candidate selection process as our document collection.

For BM25 the similarity score for each candidate d with test document q is computed as:

$$\text{BM25}_d = \sum_{w \in q} \log[\frac{N_c}{df_w}] \cdot \frac{(k_1+1)tf_{wd}}{k_1((1-b)+b\times L)+tf_{wd}} \cdot \frac{(k_3+1)tf_{wq}}{k_3+tf_{wq}}$$

where N_c is the number of documents in the collection, df_w is the document frequency of term w, and L is the length of document d divided by the average document length for all documents in the collection. tf_{wd} is the term frequency of term w in document d. The tuning parameters k_1,k_3, and b are set to 1.5,1.5, and 0.75 respectively, as suggested in [12].

For LM we use a Dirichlet smoothed language model [19]:

$$\text{LM}_d = P_{lm}(q|d) = \prod_{w \in Q} P_{lm}(w|d)$$

$$P_{lm}(w|d) = \frac{N_d}{N_d+\delta} P_{ml}(w|d) + (1 - \frac{N_d}{N_d+\delta}) P_{ml}(w|c)$$

$P_{ml}(w|d)$ is the maximum likelihood estimate of word w in document d, and $P_{ml}(w|c)$ is the maximum likelihood estimate of word w in the entire collection c. N_d is the number of tokens in document d and δ=500.

2.1 LDA and Language Models

We follow Wei and Croft [16], who proposed an LDA-based document model for ad-hoc retrieval where they used topics to smooth a language model representation. To find related documents for a given patent, we compute a similarity score defined as the probability of a query patent q given document d:

$$\text{LDA}_d = P_{lda}(q|d) = \prod_{w \in q} P_{lda}(w|d)$$

$$P_{lda}(w|d) = \sum_{z=1}^{N_z} P(w|z, \hat{\phi}) P(z|\hat{\theta}, d)$$

where z is a latent topic and $\hat{\theta}$ and $\hat{\phi}$ are posterior estimates of θ and ϕ—the inferred topic and word distributions. We use collapsed Gibbs sampling [9] to infer Θ and Φ given the observed words, the model, and the priors.

The number of topics N_z, as well as α and β, are set beforehand. We achieved best results when choosing N_z to be $\sqrt{N_c}$, the square root of the number of documents in the collection. We also used $50/N_z$ for α and $200/N_v$ for β, with N_v being the vocabulary size.

We combine topic and language models by using the latent topics to smooth the language models [16]. The similarity score between the test patent and document d can then be computed as:

$$\text{LM-LDA}_d = P_{lmlda}(q|d) = \prod_{w \in Q} P_{lmlda}(w|d)$$

$$P_{lmlda}(w|d) = \gamma(P_{lm}(w|d)) + (1-\gamma)P_{lda}(w|d)$$

$P_{lm}(w|d)$ is the smoothed language model and $P_{lda}(w|d)$ can be computed using the inferred topic and word distributions as defined earlier. γ is set to 0.3 as suggested in [16] (see Section 3.3 for a discussion of how this parameter influences the retrieval results).

2.2 Dirichlet Multinomial Regression for Patent Retrieval

Mimno and McCallum [14] proposed Dirichlet multinomial regression (DMR), which extends the LDA model to allow the topics to be conditioned on arbitrary features. We can use DMR to model different topical content across different patent sections (abstract, claims, etc.).

The language used in a patent document can vary significantly depending on the section. Different sections can be written by different individuals, e.g. the details of an invention by engineers and the claims by an attorney. By treating sections individually we hope to gain more coherent topics without losing the overall topical context of the patent document as a whole.

In the DMR model the document-topic prior α is a function of the observed document features encoded in a vector x. A 1 in x indicates the presence of a feature in document d and a 0 indicates its absence.

Here we split the patent text into its four sections: *title+abstract*, *claims*, *summary*, and *details*. We model each section as an individual document with two types of metadata information: patent id and section id, where the ids are presented as binary features in the DMR model. The similarity score is then computed as:

$$\text{DMR}_d = P_{dmr}(q|d) = \prod_{w \in q} P_{dmr}(w|d)$$

$$P_{dmr}(w|d) = \sum_{s=1}^{4} \frac{N_s}{N_d} P_{dmr}(w|s)$$

$$P_{dmr}(w|s) = \sum_{z=1}^{N_z} P(w|z, \hat{\phi}) P(z|\hat{\theta}, s)$$

To get $P_{dmr}(w|d)$ we combine the topic models for the different sections s. N_s is the number of words in section s; and N_d is the number of words in the whole patent d. The topic models computed in this way can be used to smooth the language models in a manner analogous to the LDA models:

$$\text{LM-DMR}_d = P_{lmdmr}(q|d) = \prod_{w \in Q} P_{lmdmr}(w|d)$$

$$P_{lmdmr}(w|d) = \gamma(P_{lm}(w|d)) + (1-\gamma)P_{dmr}(w|d)$$

We use the same setting as for LM-LDA with γ=0.3.

3. EXPERIMENTS

To evaluate our proposed methods, we use each algorithm to rank patents in terms of similarity to a query patent q, where the set that is ranked is (a) restricted to patents or applications published prior to the date of publication of q, and (b) restricted to the candidates generated by the two-stage candidate set generation process described earlier. We use the citations in a patent q as a proxy for ground truth in terms of determining a set of patents that are relevant to q. This strategy was also used in [17] and in the NTCIR[3] workshop series.

There are a number of drawbacks to using existing citations in a patent as a measure to evaluate patent similarity algorithms. For example, it is quite common in practice that truly relevant patents are not cited, potentially deliberately. In addition, if there exists a published patent application and a granted patent for the same application, only one or the other might have been cited. However, since these limitations affect all of the algorithms, it is reasonable to argue that a patent's citations provide a useful surrogate ground truth by which to compare and evaluate algorithms in the context of patent retrieval.

3.1 Data

For testing, we randomly picked 100 granted utility patents published on November 13th, 2012 by the USPTO[4]. For each of these 100 test patents we selected the top-K=500 patents found by the search engine as candidates (see step one in Figure 1). We expanded these candidate sets by adding the references as described in step two in Figure 1. This resulted in approximately 275,000 patents and patent applications for the 100 test patents in total. For each test patent, the average number of references to US patents published after 1975 was 25.7, of which 9.9 on average were added by the patent examiner. In addition each test patent had on average 4.5 references to non-US patents, old US patents, or other non-patent literature. An average of 18.7 references were present in the citation candidate set whose average size per test patent was 2,772.

3.2 Evaluation Methodology

The US patents cited in the test patents act as our ground truth set. We discard references to non-US patents and non-patent literature. For computation of precision and recall we also do not include references to US patents that are not among the candidates and thus can not be retrieved and ranked by our algorithms.

For each test patent q we generate a ranking of patents to be cited using all the candidates for that patent. We use mean average precision (MAP) [12] to compare the rankings R_i generated by different methods for all test patents $q \in Q$:

$$MAP(Q) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} \text{Precision}(R_{jk})$$

where R_{jk} is the set of ranked results from the top of the list down to item k in the list, and the set of relevant items is $\{i_1 \ldots i_{m_j}\}$. If no relevant document is retrieved, precision is taken to be 0.

[3] http://research.nii.ac.jp/ntcir/index-en.html

[4] USPTO patents are publicly available at http://www.google.com/googlebooks/uspto-patents.html

Table 1: Citation Recommendation Results

	Mean Average Precision			
	Total	+/-	Examiner	Others
BM25	0.062	-51.2%	0.048	0.029
LM	0.127	—	0.116	0.066
LDA	0.134	+5.5%	0.106	0.079
DMR	0.143	+12.6%	0.117	0.085
LM-DMR	0.164	+29.1%	0.135	0.090
LM-LDA	0.177	+39.4%	0.147	0.106

3.3 Results and Analysis

The main results from our experiments are summarized in Table 1. We report MAP for the total number of citations and show in the third column the performance difference as a percentage relative to the language model . It is clear from these results that the combination of language model representations with topic model representations (either LDA or DMR) outperforms the individual methods, with LM-LDA yielding the best results. The last two columns in Table 1 show MAP scores considering only the references added by the patent examiner or by others (usually the patent author), respectively. Note that LM on its own performs as well or better than LDA and DMR on their own, for references cited by the examiner, whereas this is not true for references cited by others. (The combination of LM with LDA or DMR is still best overall on both sets). This suggests that references added by examiners tend to contain language that more closely matches the language in the original patent, relative to the references added by non-examiners. This could be due to the fact that patent examiners make more extensive use of keyword-based search, compared to non-examiners, when determining which prior patents should be cited.

4. DISCUSSION AND CONCLUSIONS

In this paper we investigated the application of language modeling and topic modeling to the problem of patent retrieval given a query patent. We conducted experiments on a subset of the USPTO patent corpus using patent citations as ground truth. We found that language models provide systematic improvements in terms of precision and recall over simpler methods such as BM25. Furthermore, our experiments showed that the combination of topic modeling and language modeling provides further significant improvements in performance over either alone. Additional improvements could potentially be gained by incorporating meta-information such as the category information for a patent or citation graphs [7].

5. ACKNOWLEDGEMENT

Supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20155. The U.S. government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. government.

6. REFERENCES

[1] K. H. Atkinson. Toward a more rational patent search paradigm. In *Proceedings of the 1st ACMWorkshop on Patent Information Retrieval*, pages 37–40, 2008.

[2] L. Azzopardi, W. Vanderbauwhede, and H. Joho. Search system requirements of patent analysts. In *SIGIR'10*, pages 775–776. ACM, 2010.

[3] M. Bailey, B. Lanham, and J. Leibowitz. Mechanized searching in the U.S. Patent Office. *Journal of the Patent Office Society*, 35(7):566–587, 1953.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, January 2003.

[5] D. Bonino, A. Ciaramella, and F. Corno. Review of the state-of-the-art in patent information and forthcoming evolutions in intelligent patent informatics. *World Patent Inf.*, 32(1):30–38, 2010.

[6] W. B. Croft and J. Lafferty. *Language Modeling for Information Retrieval*. Kluwer Academic Publishers, Norwell, MA, USA, 2003.

[7] A. Fujii. Enhancing patent retrieval by citation analysis. In *SIGIR'07*, pages 793–794. ACM, 2007.

[8] S. Fujita. Technology survey and invalidity search: A comparative study of different tasks for japanese patent document retrieval. *Inf. Process. Manage.*, 43(5):1154–1172, September 2007.

[9] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proc Natl Acad Sci U S A*, 101 Suppl 1:5228–5235, April 2004.

[10] R. Krestel and P. Fankhauser. Personalized topic-based tag recommendation. *Neurocomputing*, 76(1):61–70, 2012.

[11] R. J. Mann and M. Underweiser. A new look at patent quality: Relating patent prosecution to validity. *Journal of Empirical Legal Studies*, 9(1):1–32, 2012.

[12] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK, July 2008.

[13] H. Mase, T. Matsubayashi, Y. Ogawa, M. Iwayama, and T. Oshio. Proposal of two-stage patent retrieval method considering the claim structure. *TALIP*, 4(2):190–206, June 2005.

[14] D. M. Mimno and A. McCallum. Topic models conditioned on arbitrary features with dirichlet-multinomial regression. In *UAI'08*, pages 411–418. AUAI Press, 2008.

[15] F. Saad and A. Nürnberger. Overview of prior-art cross-lingual information retrieval approaches. *World Patent Inf.*, 34(4):304–314, December.

[16] X. Wei and W. B. Croft. Lda-based document models for ad-hoc retrieval. In *SIGIR'06*, pages 178–185. ACM, 2006.

[17] X. Xue and W. B. Croft. Automatic query generation for patent search. In *CIKM'09*, pages 2037–2040. ACM, 2009.

[18] X. Yi and J. Allan. A comparative study of utilizing topic models for information retrieval. In *ECIR'09*, pages 29–41. Springer-Verlag, 2009.

[19] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *Trans. Inf. Syst.*, 22(2):179–214, 2004.