Automated Generation of Timestamped Patent Abstracts at Scale to Outsmart Patent-Trolls

Felix Hamborg, Moustafa Elmaghraby, Corinna Breitinger, Bela Gipp

Department of Computer and Information Science
University of Konstanz, Germany
[first.lastname]@uni-konstanz.de

Abstract. The fundamental idea of patent systems is to protect inventors who have invested resources during the development of their invention. Patent trolls abuse these systems by filing obvious patents with significantly less cost and usually without the intention to produce or offer the invention. Instead, patent trolls sue other companies that allegedly violate their obvious patents. We propose a method that challenges patent trolls by generating large amounts of obvious patent abstracts automatically. In contrast to prior art, our approach generates abstracts for any patent category and achieves high diversity in content and structure of the resulting abstracts. Furthermore, we timestamp the generated abstracts using a decentralized timestamping service so that users can prove that a generated abstract existed at a certain point in time. In a survey, we found that the quality of the generated abstracts, using criteria defined by the European Patent Office, was 6% higher compared to prior art.

Keywords: Natural Language Generation, Timestamping.

1 Introduction

A patent grants the inventor the right to prevent other parties from producing, using, importing, or selling an invention without approval. *Non-practicing entities* (NPE) – commonly known as *patent trolls* – use patents as a means for profit. Instead of researching to advance products or methods, NPEs buy patents from other companies or file *obvious patents* [1] that are worded in such a way that many use cases or approaches are covered by the patent. NPEs then use such patents to litigate alleged infringements. Usually, NPEs threaten other companies with a costly lawsuit unless the company agrees to pay a settlement or a licensing fee. Companies threatened with lawsuits often choose to settle with a NPE, even if they did not (intentionally) infringe, because patent litigation is extremely expensive. Median attorneys' fees range from $0.3m to $12.5m per lawsuit [2]. The actions of NPEs can drain companies' resources [3] in their attempt to defend themselves against the NPE's litigations. In some cases, these processes can amount to millions of dollars [2,3]. Such acts have harmed companies of all sizes, ranging from startups to huge corporations [4].

Our main research question is whether an automated approach can successfully contribute towards preventing NPEs from pursuing their damaging behavior of filing obvious patents. This motivates our goal of implementing a system that automatically

generates obvious patent abstracts at scale. Such abstracts must additionally be syntactically and grammatically sound. While a patent consists of multiple components, such as a classification into categories, figures, and so-called claims that define the limits of what is protected by the patent, we choose to generate patent abstracts, because they represent the summarized explanation of the invention.

In Section 2, we provide an overview of state-of-the-art systems capable of generating patents and their used techniques. In Section 3, we describe our abstract generation method. In Section 4, we evaluate the performance of our approach in a survey using criteria defined by the European Patent Office (EPO) [5].

2 Previous work

Existing approaches generate grammatically accurate patent abstracts given a suitable learning dataset. However, we identified the following set of shortcomings for existing solutions: (1) *high specialization*, patents can typically only be generated for a single category [6], (2) *non-diverse,* sentence structure features no variation [7], (3) *accessibility*, existing solutions are not open source or not free of charge [8], (4) *timestamping*, no secure mechanism is provided for later proving the time of existence for generated patents [6]–[8], and (5) *nonsensical semantics* [6].

All Prior Art (APA) uses an approach [6] that generates patent abstracts using an algorithm merging different existing abstracts together and creating new obvious patent abstracts. These abstracts are then published under the creative common license, which shall prevent filing similar obvious ideas as patents. The generated patents feature no trusted timestamp that could verify their precedence. Also, the abstracts are not matched against later-filed patents. The generated texts are syntactically correct, but the quality of the semantics is lacking, which makes them nonsensical. *Cloem* is a company [8] that creates variants of patent claims, called *cloems*. The generated claims can be published to keep potential competitors from attempting to file similar patent claims. This is achieved through multiple specialized parsers for patent claims. In addition, Cloem uses proprietary dictionaries created with the aid of *Wordnet*, *Wikipedia*, and data derived from the analysis of 70m patents. The details about the algorithms are undisclosed. Cloem timestamps and publishes the generated patents on their website. We found the semantic quality of the generated patents to be higher compared to APA, however, it is also a paid service. *Transform any text into a patent application* is an open source method [7] that transforms a given text into a patent application. The idea is to find common grammatical structures in patent applications, and to then extract sentences containing similar structures from the input texts. This is done by analyzing the sequence of part-of-speech (POS) tags of patents then searching for the most similar sequences in the input text. The system produces titles, abstracts, and descriptions with correct grammar. However, the system only accepts text with specific POS structure, otherwise it cannot generate a patent. Also, all generated patents are structurally similar.

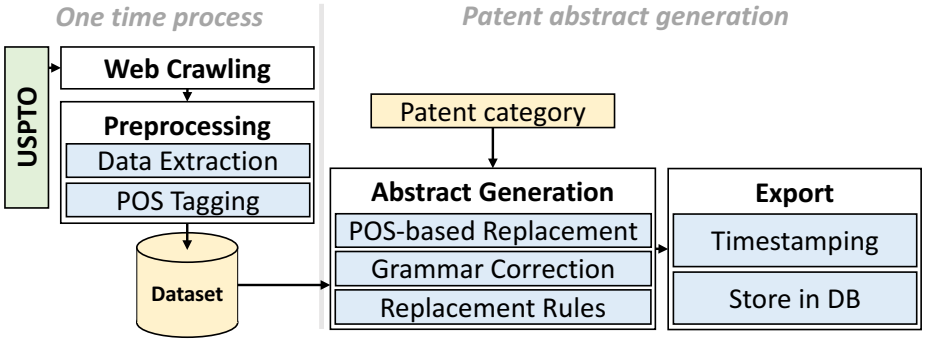
The currently available approaches generate grammatically correct patent abstracts but suffer from practical limitations: they are fine-tuned to a single category, not free of charge, and the generated abstracts are of poor language quality, or poor semantics.

Hence, we identify the following requirements to improve the state-of-the-art in generating patent abstracts. First, the approach must be *generic*, i.e., the workflow should be able to generate patents for any given category, instead of being tailored to only one category. Second, the approach must generate *grammatically and semantically correct* patents of sufficiently high quality. Third, the resulting patents must be *unique*, i.e., the patent abstracts must be sufficiently different from their sources.

3 Patent abstract generation

We describe our method for patent abstract generation following the workflow shown in Figure 1. The *patent abstract generation process* starts with the user requesting a *patent category.* The second task, *abstract generation,* generates a patent abstract for the requested category. Finally, the system *exports* the generated abstracts to a database and timestamps them. We describe the process in more detail later in this section.

Fig. 1. Overall workflow



In a one-time or regularly repeating process the system performs *web crawling* to gather patents from a patent office, which are later used to generate new patents. We utilize patents filed with the USPTO [9] because their database contains over 2.3m patents, which can be crawled at no cost. The next task is *preprocessing* the patents. For each patent's URL, we *extract* the title, abstract, category, publishing date, and inventors from the HTML data. We perform *POS-tagging* using Stanford CoreNLP.

The abstract generation consists of three subtasks: first, *POS-based replacement* to generate new abstracts. Second, *grammar correction,* and third, *further replacement rules* to improve the language quality of the texts. Our method randomly selects one patent abstract, called *template abstract*, of the requested category from our dataset. All other patents in the dataset belonging to the same category are called *patent candidates.*

The POS-based replacement task replaces all nouns and verbs of the template abstract, hereafter called *tokens*, with nouns and verbs from the patent candidates. Specifically, for each token in the template abstract that shall be replaced, we determine the token's relative frequency within the patent candidates. We then replace the token from the template abstract with a token retrieved from the patent candidates that has the same or most similar relative frequency. This way, we improve the semantic soundness of

the resulting abstract, since such tokens are more likely interchangeable. If there are multiple candidate tokens with the same frequency, we sample one randomly.

The grammar correction task fixes the tense of the replacing verbs and the plurality of nouns. We use *SimpleNLG*, which is an natural language generation (NLG) library that comes with a default lexicon covering many commonly used English words [10]. However, our experiments with medical patents showed that the default lexicon is insufficient to cover the wide range of nouns and verbs used in medical patents. Thus, we additionally use the Specialist Lexicon [11], which covers general English terms and medical terminology. We adjust the tense of the replacing verb to the tense of the replaced verb and do the equivalent for noun plurality using devised rules.

We apply further replacement rules to improve the language quality. We found that almost all abstracts start with a sentence containing a type-defining noun, such as "[*Techniques, methods,* an *apparatus*] [are, is] disclosed for […]". We observed that replacing the first noun with another noun decreases the semantic quality of the generated patent abstract, so we chose not to replace the first noun since it fits best to the patent template abstract. As we will show in Section 4, this functionality is one reason why our approach achieves better semantic quality compared to the reviewed approaches. Also, we do not replace auxiliary verbs in the first sentence, since they accompany the main verb and are not category-specific. To ensure semantic soundness our method always replaces words that occur multiple times with the same word.

Finally, our method timestamps the abstract using *OriginStamp* [12], which is a trusted timestamping service that runs on the Bitcoin blockchain. Trusted timestamping is the process of keeping a tamper-proof and permanent record of the creation time of documents. OriginStamp allows its users to prove that their timestamped data existed at a certain point in time in a certain state by submitting a SHA256 hash of the data to the service. Users can then retrieve and verify the timestamps that have been committed to the blockchain. Timestamping is a key component of our project, since it is the means for proving the time of existence of the generated patent abstracts.

4 Evaluation and discussion

We conducted a survey to evaluate our method using the criteria for patent applications defined by the EPO [5] and common NLG criteria [13]. Therefore, we randomly sampled three abstract from patents filed at the USPTO in January 2017, and three abstracts each generated by our method or APA, respectively. All abstracts belonged to the category *data processing systems*, since APA only generates abstracts in this category. We asked the participants, ten computer science students aged between 20 and 30, to first read an introduction that explained the evaluation criteria. Participants were not told that they were rating abstracts from different sources and that some of the abstracts were automatically generated. The experiment was not time constrained. Participants were shown one abstract at a time and asked to rate each criterion on a Likert scale from one (lowest quality) to six (highest). The NLG criteria were *readability* (Read)*, accuracy* (Acc)*,* and *usefulness* (Use)*.* The EPO criteria were *inventiveness* (Inv), i.e., the

degree of invention*, application* (App), i.e., whether the invention can be applied industrially*, novelty* (Nov)*,* i.e., whether the idea is new, and *inventive step* (InvS), i.e., how non-obvious the idea is. The setup does not allow a realistic assessment of novelty, inventiveness, and inventive step, since a comprehensive study of prior art would be required. However, we were still interested in these criteria to get insights on how inventive the abstracts appeared to the participants.

Table 1 shows that our method outperforms APA by 0.16 (6%) in the average total. The average score was also higher in all criteria except for readability. The average readability score shows that the readability of APA patent abstracts (3.07) are slightly higher than the ones generated by our system (2.93), with a margin of 0.14. As expected, the quality of real patent abstracts was rated higher than that of both generation methods, specifically by 0.56 (20%) better than our method.

Table 1. Mean scores per source and rate criterion on a Likert scale (1 is lowest quality, 6 is best). The variance is shown in brackets. Bold numbers indicate the better performing method.

Source	*Read*	*Acc*	*Use*	*Inv*	*App*	*Nov*	*InvS*	*Avg.*
APA	**3.07** (1.05)	2.40 (0.04)	2.73 (0.49)	2.67 (0.25)	2.60 (0.48)	2.53 (0.25)	2.33 (0.21)	2.62 (0.40)
Own method	2.93 (0.57)	**2.73** (0.21)	**2.93** (0.09)	**2.87** (0.09)	**2.87** (0.09)	**2.60** (0.04)	**2.53** (0.05)	**2.78** (0.17)
USPTO	4.35 (0.01)	2.95 (0.17)	3.35 (0.09)	3.30 (0.44)	3.50 (0.28)	2.95 (0.38)	2.95 (0.25)	3.34 (0.23)

To evaluate the consistency of the abstracts across all criteria, we also calculated the variance of the scores given by study participants. Our system showed more consistent performance than APA for readability, usefulness, inventiveness, application, novelty, and inventive step. The variance was particularly good for usefulness (0.09) and application (0.09). However, the accuracy (0.21) is worse than that of APA (0.04).

Through manually testing random samples of the generated patents, we observed that the semantics quality of our generated patent abstracts could vary widely. This depended on the length of the generated abstract. We also noticed a limited amount of grammar mistakes occurring for specialized scientific or rarely occurring words. We deduce that the main cause is the accuracy of the POS tagger. The diversity can be improved by using more patent sources beyond the USPTO.

5 Conclusion and future work

We proposed a method that generates patent abstracts to address the problem of non-practicing entities (NPEs) filing obvious patents. Our system introduces four main improvements to the current state-of-the-art: first, our system can generate abstracts for any patent category. Second, the method performs trusted timestamping so that users can prove that a generated abstract existed at a certain point in time. Third, the generated abstracts score better overall than APA as to criteria for patent applications as defined by the European Patent Office. Fourth, the abstracts are also better than APA

according to criteria for natural language generation. We believe that the proposed system is a first step towards limiting the high cost of NPEs abusing the patent system.

Future improvements to our proposed system include publishing the generated abstracts on a publicly available archive. Then, we will devise and implement a search engine that captures obvious patent abstracts by measuring their similarity to previously generated and published abstracts. We plan to measure the similarity between two abstracts using semantic similarity measures [14]. Finally, the system should inform the authors of detected obvious patents. We also plan to further investigate how we can improve the semantic quality of the generated abstracts.

References

1. T. Fischer and J. Henkel, "Patent trolls on markets for technology – An empirical analysis of NPEs' patent acquisitions," *Res. Policy*, vol. 41, no. 9, pp. 1519–1533, 2012.
2. C. Barry and R. Arad, "2016 Patent Litigation Study: Are we at an inflection point?," 2016.
3. J. E. Bessen, M. J. Meurer, and J. L. Ford, "The Private and Social Costs of Patent Trolls," *SSRN Electron. J.*, 2011.
4. J. Muellin, "Famous patent 'troll's' lawsuit against Google booted out of East Texas," 2017. [Online]. Available: https://arstechnica.com/tech-policy/2017/02/famous-patent-trolls-lawsuit-against-google-booted-out-of-east-texas/. [Accessed: 06-May-2017].
5. European Patent Office, "Guidelines for Examination in the European Patent Office," 2016. [Online]. Available: http://www.epo.org/law-practice/legal-texts/html/guidelines/e/g_i_1.htm. [Accessed: 15-May-2017].
6. A. Reben, "All Prior Art – Algorithmically generated prior art." [Online]. Available: http://allpriorart.com/. [Accessed: 01-Jan-2017].
7. S. Lavigne, "Transform any text into a patent application." [Online]. Available: http://lav.io/2014/05/transform-any-text-into-a-patent-application/.
8. Cloem S.A.S.U., "Cloem - reinventing creativity," 2017. [Online]. Available: https://www.cloem.com/. [Accessed: 06-May-2017].
9. United States Patent and Trademark Office, "Patents." [Online]. Available: https://www.uspto.gov/patent. [Accessed: 15-May-2017].
10. A. Gatt and E. Reiter, "SimpleNLG: a realisation engine for practical applications," *Proceedings of the 12th European Workshop on Natural Language Generation*. Association for Computational Linguistics, pp. 90–93, 2009.
11. A. Browne, A. McCray, and S. Srinivasan, "The specialist lexicon," *Natl. Libr. Med. Tech. Reports*, pp. 18–21, 2000.
12. B. Gipp, N. Meuschke, and A. Gernandt, "Decentralized Trusted Timestamping using the Crypto Currency Bitcoin," *iConference 2015*, pp. 1–6, 2015.
13. E. Reiter, "Task-based evaluation of nlg systems: Control vs real-world context," *Proc. UCNLG+Eval Lang. Gener. Eval. Work.*, pp. 28–32, 2011.
14. F. Hamborg, N. Meuschke, A. Aizawa, and B. Gipp, "Identification and Analysis of Media Bias in News Articles," in *Proceedings of the 15th International Symposium of Information Science*, 2017.