World Patent Information 61 (2020) 101965

Contents lists available at ScienceDirect

World Patent Information

journal homepage: http://www.elsevier.com/locate/worpatin

Patent classification by fine-tuning BERT language model

Jieh-Sheng Lee *, Jieh Hsiang

Department of Computer Science and Information Engineering, National Taiwan University, Taiwan

A B S T R A C T

In this work we focus on fine-tuning a pre-trained BERT model and applying it to patent classification. When applied to large datasets of over two million patents, our approach outperforms the state of the art by an approach using CNN with word embeddings. Besides, we focus on patent claims without other parts in patent documents. Our contributions include: (1) a new state-of-the-art result based on pre-trained BERT model and fine-tuning for patent classification, (2) a large dataset USPTO-3M at the CPC subclass level with SQL statements that can be used by future researchers, (3) showing that patent claims alone are sufficient to achieve state-of-the-art results for classification task, in contrast to conventional wisdom.

1. Introduction

Patent classification is a multi-label classification task. It is challenging because the number of labels can be large, e.g. more than 630 at subclass level. We see this task from two aspects. From the perspective of Deep Learning, pre-training an unsupervised language model on large corpus and fine-tuning the model on downstream tasks have resulted in several state-of-the-art performances recently. Such pre-training models include ELMo (Embeddings from Language Models) [1], ULMFiT (Universal Language Model with Fine-tuning) [2], OpenAI GPT (Generative Pre-Training) [3], BERT (Bidirectional Encoder Representations from Transformers) [4] and OpenAI GPT-2 [5]. Among them, BERT is the most suitable for experiments if having the availability of source code and pre-trained models considered. Therefore, we set a goal to know how well BERT can perform on patent classification after fine-tuning.

From the perspective of patent research, it is time to have a new baseline with a large dataset based on the CPC (Cooperative Patent Classification) system. In general, Deep Learning outperforms other methods when the size of dataset is large. In the past, the sizes of datasets for patent research vary widely. Such variation made comparison difficult. Inference is also less valuable because sometimes the datasets were outdated. In this work, we prepared a new dataset based on the CPC with more than three million US patents. Patent researchers can leverage the dataset or our approach to cover more tasks, since the entry barriers for data, algorithm and computation are all much lower than before.

The CPC system and the IPC (International Patent Classification) system are two of the most commonly used classification systems. The CPC is a more specific and detailed version of the IPC system. On January 1, 2013, the CPC system came into force and, with which, the United States Patent and Trademark Office (USPTO) replaced its original system. A growing number of national patent offices have decided to follow the CPC. The CPC patent documentation coverage is expected to significantly improve relative to the IPC coverage in the coming years [6]. Therefore, it is foreseeable that the importance of the CPC system is likely to increase for various patent tasks in the future. However, most of the papers in the field were based on the IPC because of the CLEF-IP competition [7]. The CLEF-IP competition in 2011 was based on the IPC at the subclass level. The dataset consisted of patents filed between 1978 and 2009. Key performances were evaluated with precision at top 1, precision at top 5, recall at top 5, F1 at top 5 and other metrics. It is not clear to us why recall at top 1 and F1 at top 1 were omitted. This is critical for our work, because a precision value could be very high at the cost of a very low recall. Therefore, precision at top 1 alone might not be a fair number to compare if recall or F1 is not provided. In this work, our F1 score at top 1 is the best performance of patent classification. We use it to benchmark with the best F1 scores in other works.

Moreover, our datasets are based on patent claims. The importance of patent claims was underappreciated in the past. When drafting a new patent application, it is a common practice for patent practitioners to draft the patent claims first. The rest of the patent document could be derived or extended from the claims. In patent law, the claims define the scope of the "metes and bounds" of the patented invention. It is a 'bedrock principle' of patent law that 'the claims of a patent define the invention to which the patentee is entitled the right to exclude' [8]. One reason to use patent claims mainly is for our downstream task of patent claim generation in the future. To our knowledge, our work is the first to focus on patent claims and claims only, instead of using claims as

* Corresponding author. Computer Science and Information Engineering, No. 1, Sec. 4, Roosevelt Rd, Taipei, 10617, Taiwan.
E-mail address: d04922013@ntu.edu.tw (J.-S. Lee).

https://doi.org/10.1016/j.wpi.2020.101965
Received 29 June 2019; Received in revised form 15 March 2020; Accepted 30 April 2020
Available online 1 June 2020

supplementary data in the past. To keep our model simpler, we use only the first claim of each patent and leave the benefit of other independent and dependent claims to future research.

2. Related work

We highlight the most relevant works in recent years. Li et al. [9] proposed DeepPatent as a deep learning algorithm based on CNN (Convolutional Neural Network) and word vector embedding. They evaluated the algorithm on the CLEF-IP dataset, compared it with other algorithms in the CLEF-IP competition and claimed a precision of 83.98%, which outperformed all other algorithms. DeepPatent was further tested on USPTO-2M, a newly contributed dataset having 2,000, 147 US utility patents in 637 categories at the IPC subclass level after data cleaning. DeepPatent achieved a precision of 73.88% at top 1, with no F1 score disclosed. Further experiments by the authors using the same dataset showed that DeepPatent outperformed Random Forest, Decision Tree, BP Networks and Naive Bayes. The best F1 score is less than 43% at top 5. In this work, we use DeepPatent as the baseline to benchmark. We also assumed that the methods above benchmarked with DeepPatent are unlikely to perform better if the dataset is larger than USPTO-2M.

The idea of fine-tuning a pre-trained language model for patent classification was proposed in the Australasian Language Technology Association Workshop 2018 [10]. The task is to classify Australian patents at the IPC section level (8 labels). The dataset has 75,250 patents (60/40 as training/testing data split). Hepburn [11] used SVM and ULMFiT to achieve the best results in the student category. ULMFiT is a transfer learning technique, and the fine-tuning idea is similar to fine-tuning a pre-trained BERT model. The major difference between Hepburn's work and ours is the pre-trained model itself. Another difference in this work is the size of the dataset. Our dataset has over three million patents, while the dataset for the workshop has only 75,250. Our F1 score is 80.98% at the CPC section level. The F1 score is 78.4% at the IPC section level in Hepburn's work. Since these two datasets are significantly different is size, we skip further benchmarking and focus on larger datasets.

It is noted that most of patent classification tasks were done on IPC in the past. Tran and Kavuluru [12] claimed being the first to reinitiate the patent classification task under the new CPC coding scheme. They used logistic regression as a base classifier and exploited extra data or method, such as the hierarchical taxonomy of CPC, the citation records of a test patent, various label ranking and cut-off methods. By experimenting on 436,993 U S. patents (2010–2011 & 70/30 as training/testing data split) at the subclasses level, their best method achieved 69.89% in micro-F1 score. In this work, we will also skip benchmarking with their results since we target a larger dataset without ad-hoc feature engineering. Feature engineering is difficult to scale up in general.

By aiming at CPC and a larger dataset, we also skip any benchmark with the CLEF-IP results. It is uncertain whether fitting a big model like BERT to smaller datasets may make any sense. Conversely, it should be more fruitful for other algorithms to benchmark with a larger CPC dataset in the future. Nevertheless, some of the recent works based on the legacy CLEF-IP are still noteworthy, for knowing the highest F1 score in the past. For example, comparing with fastText, Yadrintsev et al. [13] claimed that KNN is a viable alternative to traditional text classifiers. Their dataset has 699,000 patents (70/30 as training/testing data split). Their best result achieved 71.02% in micro-F1 score at the IPC subclass level.

Lim and Kwon [14] showed 87.2% precision when using titles, abstracts, claims, technical fields, and backgrounds of patents. However, no respective recall or F1 score was disclosed. A fair comparison is therefore not feasible. Their dataset has 564,793 Korean patents at the IPC subclass level. Their method combined multinomial naive Bayes with other tricks such as a Korean language morphological analyzer, using 1860 stopwords removed and TF-ICF (a variation of the well-known TF-IDF).

Hu et al. [15] showed that a hierarchical feature extraction model can capture both local features as well as global semantics. An n-gram feature extractor based on CNN was designed to extract local features. A bidirectional long–short-term memory (BiLSTM) neural network model was proposed to capture sequential correlations from higher-level representations. The training, validation and test datasets contain 72,532, 18,133, and 2679 mechanical patents from the CLEF-IP dataset. The number of labels is 96 for mechanical patents only. The hierarchical model outperformed other models using CNN, LSTM or BiLSTM alone. Their best F1 is 63.97% at top 1. Back to the CLEF-IP competition itself, Verberne and D'hondt [16] reached their best F1-value 70.59% in a series of classification experiments with the Linguistic Classification System (LCS). The training dataset has 905,458 patents and the testing has only 1000 patents.

3. Data

Most of the past patent datasets were from the CELF-IP or patent offices. We found it easier to leverage the Google Patents Public Datasets [17] on BigQuery released in 2017. A dataset based on SQL lowers the entry barrier of data preparation significantly. We deem SQL statements as a better way than sharing conventional datasets for two reasons: (1) Separation of concerns. If a dataset contains pre-processing or post-processing already, it could be harder for other researchers to reuse when needing different manipulations. (2) Clarity and flexibility. An SQL statement is precise and easy to revise for different criteria. The SQL statement for our training dataset is listed in Appendix A.

Our new dataset is called USPTO-3M (3,050,615 patents). Based on the SQL statements, it would be easy for other researchers to cover all patents if computing resource for training is not a constraint. When benchmarking with the DeepPatent, we use its dataset USPTO-2M when possible. When benchmarking different settings based on PatentBERT, we use our new USPTO-3M dataset.

The USPTO-2M dataset is the largest dataset contributed by DeepPatent. It is the basis for the authors to conclude that CNN outperforms the other four machine learning techniques. Unfortunately, we found 1739 records incorrectly having no IPC label in the training data of the USPTO-2M dataset. Therefore, it is not rigorously possible for us to benchmark with the same training data. However, since the erroneous data occupies less than 0.09% of the whole training dataset (1,950,247 patents), it should be reasonable to assume that the deterioration of DeepPatent's performance is minor if any. By removing the erroneous data, we train PatentBERT with the remaining 1,948,508 patents from the USPTO-2M dataset. No such erroneous issue exists in the test data of the USPTO-2M dataset. Therefore, benchmarking PatentBERT with DeepPatent based on such minor discrepancy should be still a fair comparison.

4. Method & experimental setup

In this work, we leverage the released BERT-Base pre-trained model (Uncased: 12-layer, 768-hidden, 12-heads, 110 M parameters) [18]. We leave other models such as the BERT-Large (340 M parameters) to the future because the BERT-Base is already sufficient to outperform DeepPatent.

Our implementation follows the fine-tuning example released in the BERT project. For multi-label purpose, we use sigmoid cross entropy with logits function to replace the original softmax function which is suitable for one-hot classification only. We intentionally keep the code change as minimal as possible so as to make the BERT test a vanilla baseline for future experiments to compare against. All hyperparameters remain as default values. The number of training epochs is 3. During our experiments, we also observed that it might be sufficient for the max_seq_length to be shorter if having fewer labels, e.g. 9 labels at CPC section level. We leave testing different hyperparameters at different

Table 1
PatentBERT vs DeepPatent.

	Method	Patent Data	Train	Test	F1 (%)	Precision (%)	Recall (%)	EVAL
1	**DeepPatent**	IPC + Title + Abstract	2006–2014-A	2015-A	–	73.88	–	Top 1
2	**DeepPatent**	IPC + Title + Abstract	2006–2014-A	2015-A	< **43**	–	–	Top 5
3	**PatentBERT**	IPC + Title + Abstract	2006–2014-A*	2015-A	**65.87**	**81.75**	55.16	Top 1
4	**PatentBERT**	IPC + Title + Abstract	2006–2014-A*	2015-A	**44.76**	30.31	85.52	Top 5

(1) 2006–2014-A: 1,950,247 patents from USPTO-2M for DeepPatent.
(2) 2006–2014-A*: 1,948,508 patents from USPTO-2M for DeepPatent (0.09% discrepancy in training data after removing erroneous data).
(3) 2015-A: 49,900 patents in 2015 from the USPTO-2M dataset contributed by DeepPatent.

Table 2
CPC vs IPC for PatentBERT.

	Method	Patent Data	Train	Test	F1 (%)	Precision (%)	Recall (%)	EVAL
1	PatentBERT	**IPC** + Claim	2006–2014-B	2015-B	63.74	79.14	53.36	Top 1
2	PatentBERT	**CPC** + Claim	2006–2014-B	2015-B	**66.83**	84.26	55.38	Top 1
3	PatentBERT	CPC + Claim	2006–2014-B	**2015-C**	**66.80**	84.24	55.35	Top 1
4	PatentBERT	CPC + Claim	**2000~2014**	2015-C	**66.71**	84.95	54.92	Top 1

(1) IPC subclass level: 632 labels. CPC subclass level: 656 labels.
(2) 2006–2014-B: 1,933,105 patents from USPTO-3M for PatentBERT.
(3) 2015-B: 49,670 of the 298,559 patents in 2015 from USPTO-3M in this paper.
(4) 2015-C: 150,000 of the 298,559 patents in 2015 from USPTO-3M in this paper.
(5) 2000–2014: 2,900,615 patents from USPTO-3M for PatentBERT. USPTO-3M: 3,050,615 patents, our new dataset with SQL statements (2000–2015, from Google Patents Public Datasets on BigQuery).

CPC levels to the future.

5. Results

In the following, we show that PatentBERT outperforms DeepPatent and CPC is better than IPC for PatentBERT.

5.1. PatentBERT vs DeepPatent

As shown in Table 1, the precision at top 1 is 81.75% for PatentBERT in row 3 and 73.88% for DeepPatent in row 1. The F1 at top 5 is 44.76% for PatentBERT in row 4 and less than 43% for DeepPatent in row 1. The best F1 score is 65.87% at top 1 for PatentBERT in row 3, while the F1 score at top 1 for DeepPatent is unknown. It is noted that a precision score could be high when its respective recall score is low, or vice versa. The F1 score can become lower when its precision score goes higher. Therefore, it is more common to use the F1 score as a harmonic mean and a fair metric to take both precision and recall into consideration.

We estimate the F1 score for DeepPatent based on visual inspection. Li et al. [9] showed that DeepPatent outperforms four other machine learning techniques, in terms of the F1 score at top 5 on the USPTO-2M dataset, in Fig. 7 of their paper. No precise F1 score was provided, unfortunately. In the Fig. 10, the authors also show different F1 scores at top 5 corresponding to various window sizes of CNN. Based on visual inspection, we found that the best achievable F1 at top 5 score is less than 43% too. Both of the figures match the number 43% we estimated. No F1 score at top 1 was provided by Li et al. [9]. Therefore, our benchmarking focuses on comparing the F1 at top 5 scores in Table 1.

5.2. CPC vs IPC for PatentBERT

For other tasks in our project, we use patent claims instead of title and abstract in our experiments here. As shown by row 1 and 2 in Table 2, CPC is better than IPC at subclass level for PatentBERT models. The F1 score is 63.74% for IPC and 66.83% for CPC respectively. In row 3, we test the model with more data (2015-B dataset) and the F1 score is as stable as 66.80%. In row 4, we train the model with more data (2000–2014) and the F1 score remains stable too (66.71%). It is noted that, in Table 1, the F1 score at top 1 (65.87%) is significantly higher than the F1 score at top 5 (44.76%) for PatentBERT. In Table 2, the F1 score at top 1 for CPC is slightly better than the score for IPC. Therefore, for our downstream tasks in the future, we think: (1) the F1 score at top 1 is a convenient metric to measure both of the precision and the recall for PatentBERT, unless one of them is more important than the other (such as prior art search which needs a much higher recall with less concern on precision), (2) switching from IPC to CPC should be the right direction.

6. Conclusion

Patents might be an ideal data source for humans to solve *artificial innovation* in the long run. However, patent classification as the groundwork has been a challenging task with no satisfactory performance for decades. In this paper, we present a new state-of-the-art approach based on fine-tuning a pre-trained BERT model and it outperforms DeepPatent. Our results also show that using patent claims alone is sufficient to achieve state-of-the-art results for the classification tasks. Most important of all, the recent success of the two-stage framework (pre-training & fine-tuning) in Deep Learning is promising for patent researchers to explore more in the future. Patent classification in this work is just an example.

CRediT authorship contribution statement

Jieh-Sheng Lee: Conceptualization, Methodology, Software, Data curation, Writing - original draft. **Jieh Hsiang:** Supervision, Writing - review & editing.

Acknowledgements

The research reported in this manuscript has been funded by the Ministry of Science and Technology (MOST) in Taiwan (Project: 106-2221-E-002-207-MY2).

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.wpi.2020.101965.

References

[1] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word Representations, in: Proc. 2018 Conf. North American Chapter Assoc. Comput. Linguist. Hum. Lang. Technol., 1, Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 2227–2237, https://doi.org/10.18653/v1/N18-1202.

[2] J. Howard, S. Ruder, Universal Language model fine-tuning for text classification, in: Proc. 56th Annu. Meet. Assoc. Comput. Linguist. (Volume 1 Long Pap.), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 328–339. https://www.aclweb.org/anthology/P18-1031.

[3] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving Language Understanding by Generative Pre-training, 2018, unpublished manuscript, https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf.

[4] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional Transformers for language understanding, in: Proc. 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol., vol. 1, NAACL-HLT 2019, Minneapolis, MN, USA, 2019, pp. 4171–4186. June 2-7, 2019, https://aclweb.org/anthology/papers/N/N19/N19-1423/.

[5] A. Radrof, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language Models Are Unsupervised Multitask Learners, 2018.

[6] B. Degroote, P. Held, Analysis of the patent documentation coverage of the CPC in comparison with the IPC with a focus on Asian documentation, World Patent Inf. 54 (2018) S78–S84, https://doi.org/10.1016/j.wpi.2017.10.001.

[7] CLEF-IP (n.d.), http://ifs.tuwien.ac.at/~clef-ip/.

[8] Phillips v. AWH Corp., 415 F.3d 1303, 1312 (Fed. Cir. 2005) (en banc), https://casetext.com/case/phillips-v-awh-corp-3.

[9] S. Li, J. Hu, Y. Cui, J. Hu, DeepPatent: patent classification with convolutional neural networks and word embedding, Scientometrics 117 (2018) 721–744, https://doi.org/10.1007/s11192-018-2905-5.

[10] Australasian Language Technology association workshop 2018, http://www.alta.asn.au/events/sharedtask2018/.

[11] J. Hepburn, Universal Language model fine-tuning for patent classification, Proc Australas Lang Technol Assoc Work (2018) 93–96, 2018.

[12] T. Tran, R. Kavuluru, Supervised Approaches to Assign Cooperative Patent Classification, vol. 10682, CPC) Codes to Patents, 2017, pp. 22–34, https://doi.org/10.1007/978-3-319-71928-3.

[13] V. Yadrintsev, A. Bakarov, R. Suvorov, I. Sochenkov, Fast and accurate patent classification in search engines, J. Phys. Conf. Ser. 1117 (1) (2018) 012004.

[14] S. Lim, Y. Kwon, IPC Multi-Label Classification Based on the Field Functionality of Patent Documents, vol. 10086, 2016, pp. 677–691, https://doi.org/10.1007/978-3-319-49586-6.

[15] J. Hu, S. Li, J. Hu, G. Yang, A hierarchical feature extraction model for multi-label mechanical patent classification, Sustainability 10 (2018) 219, https://doi.org/10.3390/su10010219.

[16] S. Verberne, E. D'hondt, N. Oostdijk, C. Koster, Quantifying the challenges in parsing patent claims, Proc 1st Int Work Adv Pat Inf Retr 1609 (2010) 14–21.

[17] Google, Google Patents public datasets on BigQuery, (n.d.). https://console.cloud.google.com/bigquery?p=patents-public-data.

[18] Released BERT pre-trained models, (n.d.). https://github.com/google-research/bert.

Jieh-Sheng Lee: is currently a PhD candidate in the Department of Computer Science and Information Engineering at National Taiwan University and also an in-house patent attorney at Novatek Microelectronics Corp. His research focuses on applying Deep Learning to patents, particularly patent text generation based on latest NLP techniques. While in university, his team won the regional Championship of the ACM International Collegiate Programming Contest in Taiwan. His master thesis in Computer Science and Information Engineering won the First Prize of the Acer Dragon Thesis Award. The other master thesis in the Institute of Technology Law at National Chiao Tung University won the same First Prize again.

Jieh Hsiang: is a Distinguished Professor in the Department of Computer Science and Information Engineering at the National Taiwan University (NTU) and a Research Fellow of the Institute of Information Systems of the Academia Sinica. He is also the Director of the Research Center of Digital Humanities of NTU. He was the University Librarian of NTU between 2002 and 2008, and was a full professor in Computer Science at the State University of New York at Stony Brook before returning to Taiwan in 1993. He has authored over 150 research papers and several books, and has received a number of research awards, particularly one Test-of-Time Award from IEEE, two Outstanding Research Awards from the National Science Council of Taiwan, and one Academia/Industry Collaboration Award from the Ministry of Education. His service as the University Librarian also resulted in two Distinguished Service Awards, one from the Taiwanese Library Association and one from the National Taiwan University.