Deep Learning based Pipeline with Multichannel Inputs for Patent Classification

Mustafa Sofean

FIZ Karlsruhe Leibniz Institute for Information Infrastructure

ARTICLE INFO

Keywords:
Patent Analysis, Neural Network, Deep Learning, Patent Classification

ABSTRACT

Patent document classification as groundwork has been a challenging task with no satisfactory performance for decades. In this work, we introduce a deep learning pipeline for automatic patent classification with multichannel inputs based on LSTM and word vector embeddings. Sophisticated text mining methods are used to extract the most important segments from patent texts, and a domain-specific pre-trained word embeddings model for the patent domain is developed; it was trained on a very large dataset of more than five million patents. A deep neural network model is trained with multichannel inputs namely embeddings of different segments of patent texts, and sparse linear input of different metadata. A series of patent classification experiments are conducted on different patent datasets, and the experimental results indicate that using the segments of patent texts as well as the metadata as multichannel inputs for a deep neural network model, achieves better performance than one input channel.

1. Introduction

Patent documents contain a lot of important valuable knowledge, which can save time for new product development, increase success chance for market, and reduce potential patent infringement. Companies usually use patents as an effective way to protect their intellectual property (IP) and new products' market domination [2]. The World Intellectual Property Organization (WIPO) developed the International Patent Classification (IPC) which is a standard taxonomy. The IPC consists of about 80000 categories that cover the whole range of industrial technologies [7]. There are 8 sections at the highest level of the hierarchy, then 128 classes, 648 subclasses, about 7200 main groups, and about 72000 subgroups at lower levels. Thousands of patent applications arrive every day to patent offices around the world. One of the first tasks of the patent experts in patent offices is to assign manually classification codes to those patents based on their technical contents. However, given the large amount of documents, the patent experts are becoming overwhelmed. Therefore, reliable, fast, and scalable methods are needed to help the professionals in patent classification tasks.

Patent classification is a kind of knowledge management where documents are assigned into predefined categories. The texts of collection of patent documents are abundant, lengthy, written in very technical language, consist of huge vocabulary, and noisy words that reduce the analysis performance in terms of accuracy. We reduce the vocabulary size and noisy words by considering the most important sections of patent texts. Due to the extremely complicated patent language and hierarchical classification scheme, many previous studies focused only on whole texts of patent or some general sections such as title, abstract, detailed description and claims. They did not consider the most important sections like background, technical field, summary, and independent claims that need specific text mining tools to extract. Therefore, efficient services are implemented for semantic structuring and enrichment of the patent texts [16]. The first service is used to structure the description part of patent text into structured segments such as the technical field, background, summary, description of figures, and the embodiments. The second service is able to automatically identify the complete claim hierarchy within patent texts. Word or document embeddings are methods that transform words, paragraphs or whole documents into high-dimensional space. Semantically similar words or documents are close to each other in this representation. Compared to traditional methods, which use very sparse vectors, each word or document can be represented in a single vector with few dimensions. Therefore, in this work we present a domain-specific pre-trained embeddings for the patent domain that can be used for word/phrase similarity or patent analysis such as classification tasks. The performance of patent classification algorithms depends on many factors such as the choice of machine learning algorithms, selection of text features, or preparation of training and test data sets. Most approaches for text classification use the Bag-of-Word (BoW), Term Frequency and Inverse Document Frequency (TF-IDF), or N-grams model to represent the text [6]. Using these approaches for patent classification with traditional machine learning algorithms to assign the IPC labels to patent documents leads to problem of sparsity for the training process and they are not suitable for massive data processing [17].

In this paper, a deep learning pipeline for patent classification is proposed. The proposed pipeline used the important segments of patent texts and metadata as multichannel inputs for a deep neural network. These segments are extracted by sophisticated text mining methods and each word in each segment is represented by an embedding vector by using a pre-trained word embeddings model that was created only on patent texts. The metadata of patent on the

ORCID(s):
[1]https://www.fiz-karlsruhe.de
E-mail addresses: Mustafa.Sofean@fiz-karlsruhe.de

other hand is represented as one-hot vectors. Both embedding and one-hot vectors are used as multichannel inputs for long short-term memory (LSTM) [18] in order to learn a patent classification model.

The remaining of this paper is organized as follows. Section 2 outlines the related works. Section 3 provides our methods for extracting the important segments of patent texts. Section 4 describes our word embeddings model. Section 5 describes the architecture of a deep learning pipeline for patent classification. Section 6 presents our experiments and results. Finally, Section 7 gives the conclusion and outlines the future research.

2. RELATED WORK

Various methods involving machine learning and text mining have been proposed in order to extract value from patent information [1]. The most recent change in the field of machine learning has been the widespread use of neural networks e.g. Deep Learning (DL) for resolving complex tasks. Word embeddings are a well-known method based on DL, where words are represented as a low dimensional real-valued vector enabling convenient and real world usage of neural networks for text analysis and natural language processing (NLP). The work in [1] provides an overview of previous research of text mining and visualization based techniques for patent analysis and a taxonomy that classifies these approaches. Patent classification approaches are mainly based on the IPC taxonomy. These approaches focused on a small dataset of labels, and the documents in the dataset were related to a general domain such as CLEF-IP dataset [8]. There are many thousands of classes across the IPC schema, which make challenges for a classification task. The work [14] has written an excellent survey on standard machine learning methods for patent text classification task applied for the International Patent Classification (IPC) hierarchy and challenges in it, and most of these works used either bag-of-word or TF-IDF as features. Recently, deep learning approaches for patent classification have been proposed. The work [3] outlines a general deep learning approach for patent classification based on sparse auto-encoders and deep belief networks. However, their proposal is limited to a theoretical approach and lacks practical experiments. The work [15] automatically classified the patents based on word embeddings and long-short term memory units (LSTMs) in a neural network. Hu et al. [13] 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. Most of related deep learning methods were used only one input for the deep learning model, and they did not provide any source code for the implementation. The datasets they used are related to general domains. This paper focuses on providing a deep learning pipeline which is free available for the interests to apply patent classification tasks (multi-class, and multi-label) on multichannel inputs namely patent text sections and patent metadata. We also provide a word and phrase embeddings model for paten domain. In addition, the dataset that we used in our experiments is related to specific domains such as the information technology and life science domains.

3. Extraction of Semantic Sections of patent

A patent document often contains dozens of fields that can be grouped into two categories: First; structured fields (metadata), which are uniform in semantics and format such as patent number, inventor/s, and citation/s, filing date, issued date and assignee/s. Second; textual fields, which consist title, abstract, claims, and the detailed description of the invention. The detailed description texts of the invention include many different sections, namely the technical field, background, summary, embodiments, the description of figures and drawings of the invention. Our significant and scalable segmentation methods are developed for structuring the patent texts into pre-defined sections [20]. The description of the patent text is segmented into semantic sections by using a hybrid text mining techniques such as machine learning, rule-based algorithm, and heuristics (learning-by-problem-solving pattern). Particularly, our segmentation process was used to classify each patent paragraph in description text of patent into one of several pre-defined semantic sections. The segmentation methods start with extracting a description text from the patent document, and checks if the text is structured by headlines. Then the text is divided into paragraphs, a pre-processing step will take place to remove undesired tokens and apply stemming, a rule-based algorithm will be used to identify the headers, and a machine learning model is used to predict a right segment for each header. For the patent that do not have complete structured segments or do not have any heading at all, heuristic methods will be used. The final step is, identify boundaries of each segment and their related text content. The evaluation was conducted on three different dataset. The first dataset consists of 100 documents which are randomly selected from European patents. The second dataset consists of 50 patents and randomly selected from German patents. The third dataset is randomly extracted from Japanese and Chinese patents, and consists of 50 patents. The performance of our segmentation methods achieved up to 94% of accuracy. This tool is used to extract the most important segments which are technical field, background, and summary of invention. Another text mining service called Claim Structure Recognition (CSR) was developed to automatically identify the complete claim hierarchy within patent claim texts [9]. The service is based on the Information Processing Framework (UIMA) achieved up to 93% of accuracy in an expert-based evaluation. We only used this tool for extracting the independent claim. Both services are integrated into a scalable patent mining and analytics framework built on top of big-data architecture and a scientific workflow system for allowing the user to efficiently annotate, analyze and interact with patent data on a large-scale via visual interaction [16].

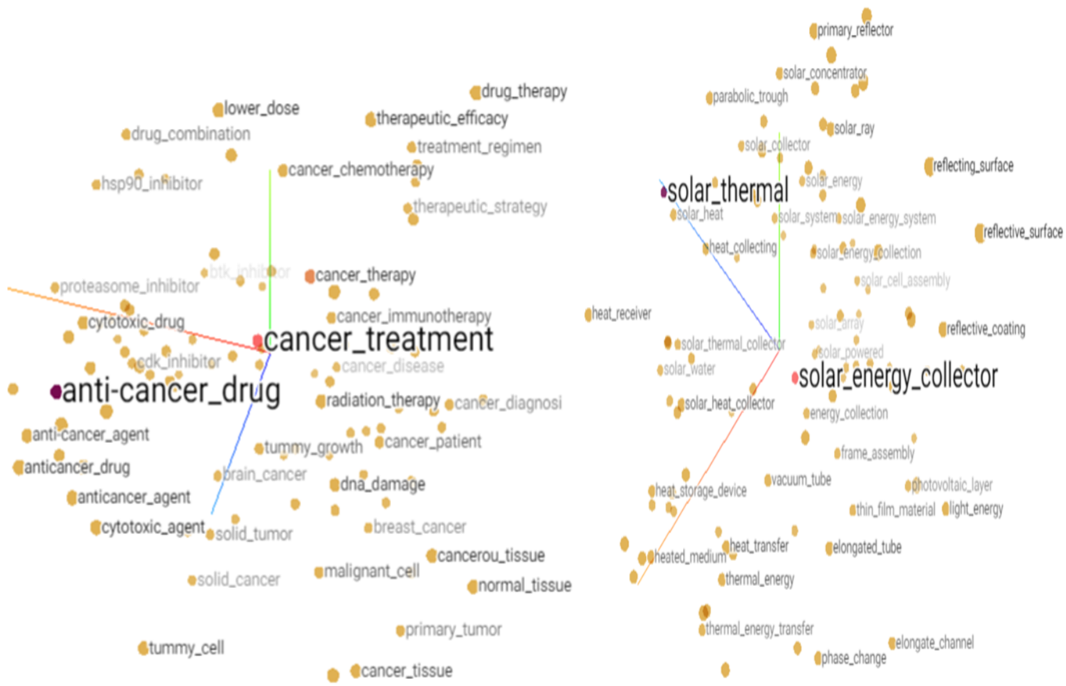

Figure 1: "solar energy collector" space (left) and "cancer treatment" space (right)

4. Word/Phrase Embedding

Word embeddings represent words as dense vectors in a vector space. Pre-trained on a large number of tokens, relations of these representations in a vector space are making a machine learning system understand word context and semantics. We go beyond a single word embedding to do a better task of modelling complicated concepts. For instance there is no individual usage of the term "oil" refers to the concept "vehicle oil", or "vegetable oil", because they have different meaning. Therefore, we create our embedding model not only on a single term but also with phrases. Natural Language Processing (NLP) processes are performed on the provided texts to automatically extract the most significant word and phrases. We observed that the most key phrases in patents are noun phrases. Therefore, a large-scale, efficient, distributed NLP pipeline is built to extract the significant noun terms and phrases from each patent document. The NLP pipeline is a sequence of stages, each stage performs a process on the text, such as: Sentence Detection (splitting the text into sentences); Tokenization (terms identified by spaces between them); Part-of-Speech tagging (labeling each term in a sentence with its appropriate part of speech such as noun, verb, adjective); Shallow Syntactic Parsing or Chunking (splitting the text of sentence into smaller chunks by noun phrase); Stopword Removal (removing the standard stopwords list and customized stopwords list that are general, very frequent and non-significant words such as invention, embodiment, claims, background, disclosure, etc.); Lemmatization (converting only all plural noun terms to singular ones); and Pruning (ignoring terms that have very low or very high frequency). Custom ngram model is used for the long noun phrases, and additional rule-based tasks are applied to filter out undesired words and phrases. For terms contain more single words than one word; we linked them via underscores and treated them as single word, e.g., digital_rights_management and quantum_cryptography, so that they are represented as single vectors. We implemented our NLP pipeline by using spaCy[2], and a common word2vec [21] is used for training the model using the Skip-Gram, a frequency threshold of 5, vector size 100, and window 10. The model is trained on more than five million patent documents (Titles and Abstracts). Figure 1 shows a dimensional projection of the word embeddings space for two phrases "solar energy collector" and "cancer treatment", respectively.

5. Deep Learning based Pipeline Architecture

In this section we describe our pipeline architecture that uses deep neural network for large-scale patent classification by combining vector representations of multiple inputs of patent information and a well-designed Long Short-Term Memory (LSTM) network. Our deep neural network model is inspired by Google's work [4] for Wide-and-Deep models. Firstly, we extracted the most important segments of patent texts which are title, abstract, technical field, background, summary, and the independent claim. For texts of each segment, a tokenization process is used for breaking the text into individual words, and the sequence length of each segment is set according to the maximum length of each. The deep learning architecture has two components: deep, and wide. It feed-forward neural networks with embeddings of each segment, and uses them as deep layers for deep neural

[2] https://spacy.io/

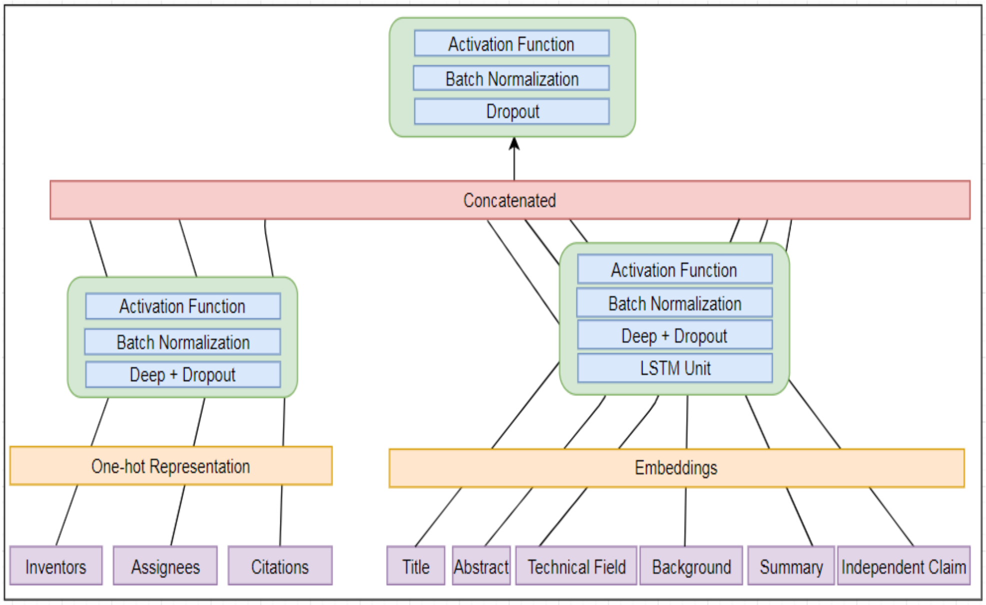

Figure 2: The Deep Learning Pipeline Architecture

network model, and the patent metadata on the other hand is used as a wide part for the model.

Specifically, the architecture is described as follows: for the wide components of the model, we used one-hot representation for patent metadata features (such as inventors, citations, and assignees), these one-hot vectors are fed into separate sub-networks, and at the end they are represented as deep networks. The right side of Figure 2 shows the architecture of wide layers since the multi-sparse inputs of patent metadata are feed into separate sub-networks. For the deep components of the model, deep layers are created for the most important patent text segments. These are sequential input to a Long Short-Term Memory (LSTM) network that takes the embeddings as inputs. The left side of the Figure 2 shows the architecture of deep layers since we used a pre-trained word embeddings model (section 4) to encode each segment texts into vectors, and then we feed them into LSTM layers. To avoid network overfitting and help network stability, additional layers are added for each input channel, dropout layer is used to drop out 30% of input in order to prevent neural networks from overfitting [11], and Batch normalization layer is used to normalize the input layer by adjusting and scaling the activations [12]. The exponential linear unit (ELU) is used as activation function [5].

Finally, we concatenated nine components which are text-based LSTM layers (title LSTM, abstract LSTM, technical field LSTM, background LSTM, summary LSTM, and independent claim LSTM), and metadata-based LSTM layers (inventors, assignees, and citations) into a final set of deep layers with dropout, batch normalization, and softmax activation function for multi-class and sigmoid for multi-label classification task [10]. In addition, we use categorical cross-entropy, and binary cross-entropy as a loss functions for a multi-class and multi-label classification tasks, respectively. The top of the Figure 2 shows the final layers of the network. since we concatenated all layers together with the dense features, resulting in a dense vector of approximately 512 dimensions. Our pipeline now is suitable for multi-class and multi-label classification tasks, and one could also easily use it and easily set the hyperparameters.

6. Experiments and Results

In this section, we introduce our datasets and experimental results. First, we describe the extraction and preparation the classification datasets that we used in this work. The implementation of the pipeline is described, and a series of experiments are conducted on the datasets for the evaluation and run on NVIDIA Tesla GPU. The evaluation metrics we used in this work are Accuracy, Precision, Recall, and the weighted harmonic mean of precision and recall (F1) scores.

6.1. Dataset Generation

In this work, we used two datasets for the experiments: ***First;*** (Dataset_1) is a domain-specific patent dataset. Based on the content of patents, the patents that are related to the same domain are much more similar to each other; however, apply a text mining task like a classification is a bit hard to the patent documents that are in the same domain. We established the training, and test datasets that are domain-specific datasets and related to information technology domain. The datasets are extracted from the European Patent Office (EPO) and the World Intellectual Property Organization (WIPO) by using our paten search engine system called STN[3]. We extracted all patents that contain the title, abstract, detailed description, claims, and at least one IPC label. The total number of extracted records in the datasets is about

[3] `https://www.stn.org`

Table 1
Evaluation Results (Dataset_1). (TI:title, AB:abstract, TECHF: Technical Fields, BACK: Background, SUMM: Summary, IND_CLAM: Independent Claim, INVs: Inventors, and PAs: Patent Assignees)

Input	Accuracy	Precision	Recall	F1-score
All texts of segments as one channel input	62%	50%	51%	50%
TI, AB, TECHF , BACK, SUMM, and IND_CLAM as multichannel inputs	66%	56%	61%	58%
TI, AB, TECHF, BACK, SUMM, IND_CLAM, INVs, and PAs as multichannel inputs	**67%**	**58%**	**66%**	**62%**
TI and TECHF as multichannel inputs	61%	50%	46%	48%
TI and TECHF, INVs, and PAs as multichannel inputs	62%	53%	52%	52%

430,000 patents filed between 1978 and 2016. Each patent document contains a title, patent number, issued date, patent type, and list of citations, classification codes, a list of inventors, a list of assignees, abstract, claims, and a detailed description. ***Second;*** (Dataset_2) is related to information technology domain and all sub domains of life science domain. This dataset is extracted in the same extraction processes of Dataset_1, and the total number of extracted documents was 1,915,308 records.

We used our segmentation tools [20] [9] to extract the most important sections (technical field, background, and summary of invention) from the text of detailed description part, and independent claim from texts of claims, respectively. All segments and metadata are separately extracted into csv files.

6.2. Implementation

All components of the deep learning pipeline are run in Jupyter[4] iPython notebooks, including loading the data, the pre-processing tasks, and load pre-trained word embeddings model and prepare the embedding matrix for each patent text segment. The core of our pipeline is implemented by Keras[5] which is a deep learning library, a high-level neural networks API, written in Python, and transparent use of a Graphics Processing Unit (GPU). We have open-sourced our implementation and it is available on the GitHub[6]. The word embeddings model is available on Kaggle[7]. The datasets we have created in this work are so large and available on request.

6.3. Experimental Results

For our experiments, we classify patents into related subclass level of IPC, and we used four evaluation measures namely accuracy, precision, recall, and F1 . The first experiment in our work was done on the domain-specific dataset (Dataset_1) which is randomly split into 80% and 20%, as training and test datasets, respectively. For multi-class classification, 43 unique main IPC classes (subclass level) were extracted from the dataset and each class has at least more than 500 documents. 31 unique classes were considered for multi-label classification task since we only considered the classes that appear at least 3000 times in the IPC field. In this task, we used our deep learning model to predict top 1 classification label (the subclass of main IPC label) for each patent document. Simple pre-processing tasks such as tokenization, stopword removal, lemmatization, and converting letters into lower case are performed on each text section of each patent document. Additional simple pre-processing tasks are performed on metadata of patent like inventors and assignees. For the hyper-parameters, the training epoch was fixed to 20, the training batch size was set as 500, and the number of input words for each text segment was set to the maximum length of each segment, and finally, a fully-connected layer with appropriate activation function was connected to related categories from the IPC labels matrix.

We conducted a series of patent classification experiments on our dataset, and we also studied how the full text, different parts of a patent information, and their combination affect the classification performance. This allows us to choose the right text segment or metadata from the patent information to better represent a patent document for the classification task. All models are trained on GPU, the experimental results for the multi-class classification task are selected as best performance from the whole training epochs, and are shown in the table 1. Firstly; we used the full text of most important sections in patent (title, abstract, technical fields, background, summary, and independent claim) as one input to the deep learning model, we obtained approximate accuracy, precision, recall, and F1 scores of 62%, 50%, 51%, and 50%, respectively.

Then, we used all text segments of patent document as multichannel inputs to the deep network model, the text of each section is converted by using a pre-trained word embeddings model into a matrix (text-based LSTM layer), apply dropout, batch normalization, and ELU activation func-

[4] http://jupyter.org
[5] https://github.com/keras-team/keras
[6] https://github.com/sofean-mso/DeepL4Patent
[7] https://www.kaggle.com/darshmso/w2vec-patent-domain

Table 2
Evaluation Results (Dataset_2). (TI:title, AB:abstract, TECHF: Technical Fields, BACK: Background, SUMM: Summary, IND_CLAM: Independent Claim, INVs: Inventors, and PAs: Patent Assignees)

Input	Accuracy	Precision	Recall	F1-score
All texts of segments as one channel input	67%	84%	61%	71%
TI, AB, TECHF , BACK, SUMM, and IND_CLAM as multichannel inputs	**74%**	**92%**	**63%**	**75%**
TI and TECHF as multichannel inputs	66%	83%	59%	69%
TI and TECHF, INVs, and PAs as multichannel inputs	68%	85%	61%	71%

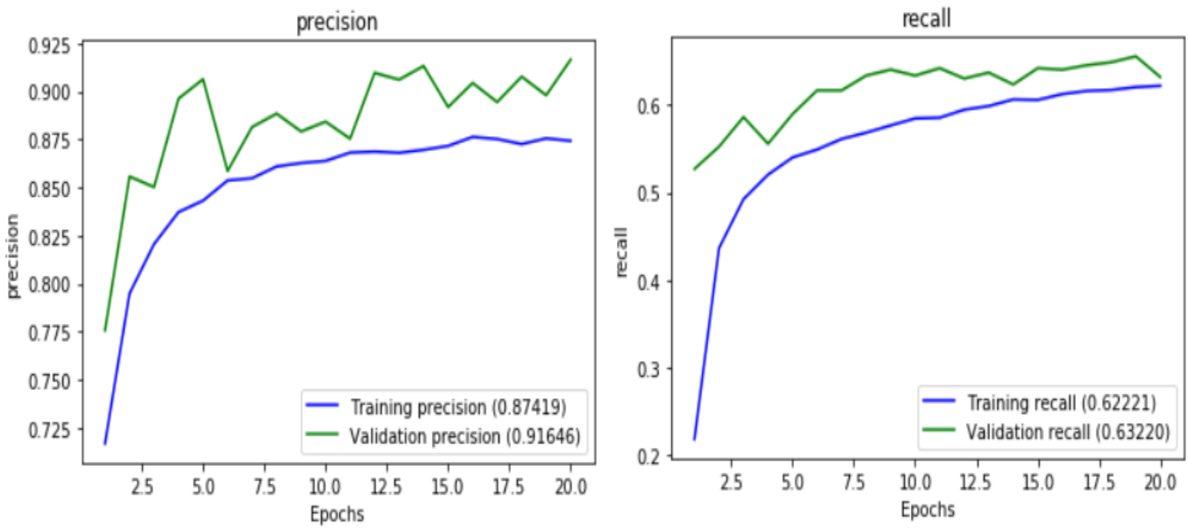

Figure 3: Precision and recall scores of the model

tion and then we concatenated all matrixes into a final layer with dropout, batch normalization, and softmax activation function. The results can be seen that the best performance we obtained approximate accuracy, precision, recall, and F1 scores of 66%, 56%, 61% and 58%, respectively. This indicates that using the segments of patent text as multichannel inputs improved the performance by 4%, 6%, 10%, and 8% in terms of all evaluation criteria, respectively. We go in depth to evaluate our approach on patent metadata information (inventors and assignees), as well as texts of each patent section as a multichannel inputs to the deep neural network model, apply dropout, batch normalization, and ELU activation function and then we concatenated all matrixes into a final layer with dropout, batch normalization, and softmax activation function. The best performance we obtained is 67%, 58%, 66%, and 62% for accuracy, precision, recall, and F1, respectively. Comparing to the first experiment, we found that using metadata and text segments of patent as multichannel inputs for the model improved the performance of all evaluation criteria by 5%, 8%, 15%, and 12%, respectively. We also conducted another experiment using two input channels, the title, and the technical field which is a short paragraph/s, with three or four sentences or could be more, which display the technological areas of the described invention. The best performance achieved by the model was 61%, 50%, 46%, and 48% for the four evaluation criteria, as well as when we added the inventors and assignees as additional input channels, and the performance is a bit improved to 62%, 53%, 52%, and 52%, respectively. Additionally, we conducted multi-label patent classification experiments, the joint set of labels/classes are represented with a binary indicator matrix, and the best performance was 45% in term of F1 for the most top five labels returned by the model.

The second experiment is conducted on the second dataset (Dataset_2) that is related to more than one domain. One of the common issues found in this dataset is imbalanced classes issue that usually reflects an unequal distribution of classes within a dataset. Therefore, a simple undersampling technique is used on the dataset where some of the documents are randomly deleted from the majority class in order to match the numbers with the minority class. For multi-class classification, 22 unique main IPC classes (subclass level) were extracted from the dataset and each class has 25,000 documents. With the same scenario in the experiment that is applied on the Dataset_1, a series of patent classification experiments are conducted on the dataset and the evaluation results are shown in the table 2. The best performance we obtained is 74%, 92%, 63%, and 75% for accuracy, precision, recall, and F1, respectively. This again indicates that using the segments of patent text as multichannel inputs improved the performance of patent classification in terms of all evaluation criteria. Furthermore, Figure 3 illustrates the precision and recall scores achieved by the classification model within 20 epochs. In this experiment, we did not test the metadata of patent such as assignees and inventors which was difficult to scale up.

Finally, according to our experimental results, we summarize all results in the tables 1 and 2 , and we found that when using the segments of patent texts with the metadata as multichannel inputs to the LSTM model, the model can achieve a better performance than a single input. Comparing our results with DeepPatent [19], which is the most relevant work in recent years, the performance of DeepPatent is 83.50% with precision and based only on 4000 characters from the description section. Our performance achieved 92% with precision, and based on the most important segments of patent texts and metadata of patents that are used as multi-channel inputs for deep network model. In the future, we can apply our approach on the same dataset that has been used in the previous works and compare the results.

7. Conclusion

In this work, we introduced a deep learning based pipeline for large-scale patent classification. Different parts of patent information are used as multichannel inputs for a Long Short-Term Memory (LSTM) that takes the both vectors (embeddings and one-hot) in order to learn a patent classification model. Finally, in order to evaluate how our model performs in patent classification, we performed a series of patent classification experiments on different patent datasets, and the experimental results indicated that using the segments of patent texts as well as the metadata as multichannel inputs for a deep neural network model, achieve a good performance. Our future directions of research will continue to add different deep neural networks such as Convolutional Neural Network (CNN), and compare it with the current results.

References

[1] Abbas, A., 2015. A literature review on the state-of-the-art in patent analysis. World Patent Information 37, 3–13.

[2] Amy TrappeyaCharles V.Trappeyb, Chun-YiWua, C.W., 2012. A patent quality analysis for innovative technology and product development. Advanced Engineering Informatics 26, 26–34.

[3] Bing Xia, B.L., Lv, X., 2016. Research on patent document classification based on deep learning. International Conference on Artificial Intelligence and Industrial Engineering (AIIE) .

[4] Cheng H-T, Koc L, H.J.S.H., 2016. Wide deep learning for recommender systems. ArXiv e-prints. http://adsabs.harvard.edu/abs/2016arXiv160607792C .

[5] Clevert D-A, Unterthiner T, H.S., 2015. Fast and accurate deep network learning by exponential linear units (elus). arxiv: https://arxiv.org/abs/1511.07289 .

[6] Dhondt E, Verberne S, K.B., 2013. Text representations for patent classification. Computational Linguistics 39, 755–775.

[7] Domonkos Tikk, Gyoergy BiroeJae, D.Y., 2005. Experiment with a hierarchical text categorization method on wipo patent collections. Applied Research in Uncertainty Modelling and Analysis, International Series in Intelligent Technologies 20.

[8] Florina Piroi, M.L., Allan Hanbury, V.Z., 2011. Retrieval in the intellectual property domain. In Proceedings of the CLEF 2011 Labs and Workshop , 19–22.

[9] Hackl-Sommer, Rene; Schwantner, M., 2015. Patent claim structure recognition. Archives of Data Science .

[10] Han, Jun; Morag, C., 1995. The influence of the sigmoid function parameters on the speed of backpropagation learning. Natural to Artificial Neural Computation. Lecture Notes in Computer Science. 930, 195–201.

[11] Hinton GE, Srivastava N, S.R., 2012. Improving neural networks by preventing co-adaptation of feature detectors. arxiv: https://arxiv.org/pdf/1207.0580 .

[12] Ioffe, S., Szegedy, C., 2015. Batch normalization: accelerating deep network training by reducing internal covariate shift. arxiv: https://arxiv.org/abs/1502.03167 .

[13] Jie Hu, Shaobo Li, J.H., GuanciYang, 2018. A hierarchical feature extraction model for multi-label mechanical patent classification. Sustainability , 10(1):219.

[14] Juan Carlos Gomez, M.F.M., 2014. A survey of automated hierarchical classification of patents. Springer International Publishing .

[15] Mattyws F. Grawe, C.A.M., Bonfante, A.G., 2017. Automated patent classification using word embedding. 16th IEEE International Conference on Machine Learning and Applications (ICMLA) .

[16] Mustafa Sofean, H.A., Alrifai, A., 2018. A workflow-based large-scale patent mining and analytics framework. Proceedings of 24th International Conference on Information and Software Technologies (ICIST); .

[17] Najafabadi, Villanustre, K.S.M., 2015. Deep learning applications and challenges in big data analytics. Journal of Big Data .

[18] Sepp Hochreiter, J.S., 1997. Long short-term memory. Neural Comput 9, 1735–1780.

[19] Shaobo Li, Jie Hu, Y.C., Hu, J., 2018. Deeppatent: patent classification with convolutional neural networks and word embedding. Scientometrics 117.

[20] Sofean, M., 2017. Automatic segmentation of big data of patent texts. International Conference on Big Data Analytics and Knowledge Discovery. DaWaK , 343–351.

[21] Tomas Mikolov, Ilya Sutskever, K.C.G.C., Dean, J., 2013. Distributed representations of words and phrases and their compositionality. 27th Annual Conference on Neural Information Processing Systems (NIPS) .