IJWR

Semantic Patent Classification Using Stack Generalization of Deep Models

Shahla Nemati*

Department of Computer Engineering, Shahrekord University, Shahrekord, Iran; s.nemati@sku.ac.ir

ABSTRACT

Over the past few years, there has been a significant increase in patent applications, which has resulted in a heavier workload for examination offices in examining and prosecuting these inventions. To adequately perform this legal process, examiners must thoroughly analyze patents by manually identifying the semantic information such as problem description and solutions. The process of manually annotating is both tedious and time-consuming. To solve this issue, we have introduced a deep ensemble model for semantic paragraph-level pattern classification based on the semantic content of patents. Specifically, our proposed model classifies the paragraphs into semantic categories to facilitate the annotation process. The proposed model employs stack generalization as an ensemble method for combining various deep models such as Long Short-Term Memories (LSTM), bidirectional LSTM (BiLSTM), Convolutional Neural Networks (CNN), Gated Recurrent Units (GRU), and the pre-trained BERT model. We compared the proposed model with several baselines and state-of-the-art deep models on the PaSA dataset containing 150000 USPTO patents classified into three classes of 'technical advantages', 'technical problems', and 'other boilerplate text'. The results of extensive experiments show that the proposed model outperforms both traditional and state-of-the-art deep models significantly.

Keywords— Patent semantic analysis, Deep learning, Patent information retrieval, Natural language processing (NLP).

1. Introduction

Intellectual Property (IP) encompasses a wide array of assets spanning from artistic and scientific works to distinctive trademarks and inventions [1]. The primary objective of IP is to encourage and foster the progress of innovative goods by granting creators exclusive economic rights to their masterpieces for a designated duration [2]. A patent is an exclusive legal right granted to an innovation, be it a product or a process, which offers a novel approach or a fresh technical resolution to a particular issue. This intellectual property right safeguards the invention, as described by the World Intellectual Property Organization (WIPO) [1]. Moreover, patents play a crucial role in the dissemination of technology and innovation, serving as a significant resource that showcases technological progress and diversification. As a result, numerous companies seek and officially register patents to safeguard their groundbreaking technologies [2].

Patent documents enhance our understanding of intricate concepts and the fundamental technologies behind various components. Over the years, there has been a significant surge in the count of patents making it challenging for human experts such as examiners and patent attorneys to analyze and manage the documents [3]. This motivates the researchers to automate the labor-intensive and time-consuming patent analysis procedure. Specifically, automating any subtask within the extensive patent analysis process poses a crucial challenge with substantial implications [4]. By doing so, it holds the potential to accelerate the overall analysis procedure. One important sub-task of patent analysis involves the classification of patents, which is usually carried out by experienced patent professionals [3].

Patent classification is either performed using a classification scheme (e.g., IPC, CPC) or is conducted based on the semantic content of the patent [5]. The former assigns one or more classification codes (pre-classification) to the patent based on its content. This is an important task because it will guide the routing of the patent application to the most appropriate sub-department of the patent office for detailed examination [5]. Therefore, this problem has attracted the attention of many researchers in recent years and several machine-learning models have been

http://dx.doi.org/10.22133/ijwr.2024.449332.1210

Citation S. Nemati, "Semantic Patent Classification Using Stack Generalization of Deep Models", *International Journal of Web Research*, vol.7, no.2,pp.1-12, 2024, doi: http://dx.doi.org /10.22133/ijwr.2024.449332.1210.

*Coressponding Author

Article History: Received: 19 December 2023; Revised: 24 February 2024; Accepted: 4 March 2024.

proposed to solve the problem. On the other hand, despite its importance, few studies were addressing the latter (i.e., the semantic patent classification). The main reason is that the patent documents have some characteristics that make their semantic classification more challenging than other texts such as social media or product review texts [6]. Specifically, patent documents are extensive texts that contain a multitude of technical terms and complicated ideas.

Patents usually contain repetitive information making them difficult to read and comprehend [2]. Vital aspects of an invention and its significance are dispersed throughout patent documents. This is because patent holders usually strive to maintain a broad and generalized specification, which serves two purposes. Firstly, it allows them to increase the scope of protection for their invention. Secondly, it relieves them from the obligation of disclosing the specifics of their technology [7]. Consequently, a significant portion of the specification merely reiterates the patent claims and includes general text describing the operation of the invention. Although a patent specification can range from 10 to 30 pages, it contains only a handful of concise passages that elucidate the specific technical impacts of the invention [7]. As a result, patent analysis frequently involves the retrieval of specific text segments from a patent that can provide insight into the underlying inspiration behind the invention being claimed.

Semantic paragraph-level pattern classification aims to address the above-mentioned problems by assigning a semantic label to each patent paragraph based on its content. The content of a patent typically comprises comprehensive explanations concerning key elements of an invention, including its advantages, solutions, challenges, and validations for claimed features. These details are important because the claims that are the most important part of patents are written in legal terminology and are often difficult to understand when read in isolation [2]. Therefore, automatically identifying important parts of patents is of great importance because it can help examiners and patent attorneys to quickly analyze the document. To precisely identify the important paragraphs of a pattern, analyzing the semantic content of the paragraphs is necessary.

Semantic classification holds greater significance and interest as compared to keyword-based classification [7]. The reason behind this is that keywords used in patents may be concise but lack the necessary evidence to comprehend the surrounding context of key arguments. On the other hand, documents provide informative content but can often encompass blended viewpoints [8]. Take, for instance, a single document where distinct paragraphs outline various arguments related to inventions such as advantageous effects, problems, and solutions. Hence, our endeavor in this study is to concentrate solely on identifying crucial arguments specifically at the paragraph level. To address this problem, we proposed an ensemble model in which different deep neural networks classify the paragraphs based on their semantic content. The proposed model combines deep models of different natures such as Long Short-Term Memories (LSTM), Convolutional Neural Networks (CNN), Gated Recurrent Units (GRU), and the pre-trained BERT model.

The rationale behind the selection of the above-mentioned deep models is that we expect each mentioned deep model to focus on a specific kind of underlying semantic in the paragraph. Hence, each model classifies the paragraph based on a specific feature of the texts in the paragraph. This makes the ensemble meaningful and efficient [9]. Specifically, CNN learns to extract features from the patent paragraphs and captures local patterns in the text. LSTM and GRU capture long-range dependencies in the text and can effectively model the context and relationships between words in the patent paragraph. On the other hand, the pre-trained BERT model captures bidirectional context and relationships between words in a patent paragraph. Moreover, BERT learns complex relationships and dependencies in text data, leading to more accurate and context-aware representations of the input text. The main contributions of our study can be summarized as follows:

- We proposed a semantic paragraph-level patent classification model for facilitating the patent annotation process.
- We proposed a novel deep ensemble model for the problem of semantic patent classification.
- We conducted extensive experiments and compared our proposed model with five baseline deep learning and six state-of-the-art deep models for semantic general text classification using a large dataset of USPTO patterns.

The rest of the paper is structured as follows: section 2 presents a brief overview of existing research on the patent classification problem. Our proposed ensemble patent classification model is presented in section 3. We demonstrate the results of applying our model to the semantic classification of ptt enss’ prr agrpphs nn scciion 4. ii nll yy, seiii on 5 concludes our findings with some future directions in semantic patent classification.

2. Literature Review

Recent research has provided insights into patent classification from various perspectives. Some studies concentrate on how patent text is represented

[10], [11] or extracting various features from the patent text [12], [13]. In contrast, other studies prioritize developing the most efficient classification algorithm [5], [14]. In the following subsections, we review traditional machine learning (ML), deep learning (DL) models, and ensemble models used in patent-related studies. Finally, we present distinctions between the proposed models and the existing studies. We also mention a brief overview of the novelties of the current study.

2.1. Traditional ML Models

Traditional methods have depended on classical machine learning techniques and text representation based on Bag-Of-Words (BoW). These models have constraints in capturing the semantic and contextual nuances of the text, as they are only able to capture lexical information. These initial studies on patent classification utilized basic Natural Language Processing (NLP) methods and feature engineering approaches to prepare texts before inputting them into classical ML classifiers. For example, Fall and colleagues [15] conducted stop words elimination, stemming, and term selection based on information gain, and subsequently inputted the modified texts into Naïve Bayes (NB), Support Vector Machine (SVM), and K-Nearest Neighbor (KNN) classifiers. In a similar study, Tikk et al. [16] implemented stop word removal, stemming, dimensionality reduction, and elimination of infrequent terms before feeding the processed data into a neural network called HITEC. More recently, Yun and Geum [3] proposed a topic modeling approach utilizing the SVM algorithm for automatic patent classification. Similarly, Chikkamath et al. [7] utilized ML algorithms to automate the highlighting process of patent documents.

Text representation has also been investigated in several studies. Li and Shawe-Taylor [17] utilized TF-IDF vector representation to represent patents and employed Kernel Canonical Correlation Analysis (KCCA) and SVM for prediction. Khattak and Gerhard [18] emphasized the attributes of low-frequency terms for patent categorization. They employed three-term weighting methods (i.e., TF-IDF, BM25, and SMART) along with five distinct machine-learning approaches. Wu et al. [19] combined SVM with a Hybrid Genetic Algorithm (HGA) to enhance the classifier's effectiveness. Zhang [20] suggested an interactive approach to patent classification that involves combining multiple classifiers and active learning. They utilized TF-IDF for patent representation and applied PCA for dimensionality reduction. Similarly, Seneviratne and colleagues [21] introduced a patent classification algorithm utilizing the KNN method. More detailed review of ML algorithm for patent classification may be found in [3] and [7].

2.2. Deep Learning Models

In 2017, researchers began experimenting with new deep-learning techniques for processing text in the context of automated patent classification. One study by Grawe et al. [22] involved removing stop words from the text and converting it into a meaningful representation using Word2Vec, which was then inputted into an LSTM model. Similarly, Xiao et al. [23] also utilized Word2Vec embeddings and LSTM to classify patents. Several advancements in patent classification using deep learning techniques began with the introduction of CNNs. For example, Li et al. [24] developed an algorithm called DeepPatent that combines word vector representation and a CNN model to classify patents. Similarly, Zhu et al. [25] utilized word embedding to segment and vectorize data, followed by a symmetric hierarchical CNN called PAC-HCNN, which outperformed traditional RNNs in patent classification. Additionally, Abdelgawad et al. [14] compared various neural network models and found that CNNs are a suitable option for patent classification.

More recently, transformer-based pre-trained models have attracted great attention from NLP researchers and have been also used in patent classification tasks. For example, Roodsari et al. [26] proposed utilizing fine-tuned transformer-based pre-trained models, such as BERT and XLNet for multi-level patent classification. Henriques et al. [1] compared traditional ML models with typical deep models and some pre-trained transformer-based models. Shajalal et al. proposed a novel explainable deep framework for patent classification using several deep models [27]. Zhang et al. proposed a reliable multi-view deep model for patent classification using the evidence theory [8]. Chikkamath et al. [2] fine-tuned transfer learning models for sentence-level patent classification.

Different from the above-mentioned deep methods, Li et al., [28] proposed a new feature extraction mechanism which combines deep and classical ML models. This method used Word2Vec vectorize the input data and combined the obtained vectors with those obtained using the embedding layer of the deep models. They reported an improvement over CNN, LSTM, and Bert models on a Chinese patent dataset. In another study, Suzgun et al. [29], created HUPD dataset containing domain-specific textual data and well-structured bibliographic metadata. They also introduced the problem of patent acceptance prediction problem and applied traditional ML, CNN, and some transformer-based models on the HUPD dataset. Finally, Yoshikawa and Krestel [30], proposed a novel approach that uses Large Language Models (LLMs) to create summaries of patent textual fields. They reported that models trained on AI-generated summaries of claims and detailed descriptions perform substantially better than models trained on the original patent text. More detailed review of deep models for patent classification may be found in [2] and [10], [11]. Also, a discussion on the use of deep learning and, especially, LLMs can be found in [31].

2.3. Ensemble models

Recently, some researchers proposed ensemble and fusion models that combine multiple classifiers to obtain higher accuracy for patent classification tasks. As an early study, Mathiassen et al. [32] tried several ensemble methods on different ML models to improve the overall patent classification accuracy. Benites et al. [33] proposed an ensemble method using the SVM model trained on different feature sets for the classification at the upper levels of the IPC hierarchy. More recently, Kamateri et al. [4] proposed a new ensemble framework named EPCF to capture and represent the diverse features needed for classifiers of the ensemble model for patent classification. They argued that their proposed framework not only is useful for patent classification tasks but also can be used in any research domain.

Another study that utilized the concept of ensemble models in patent classification was proposed by Chikkamath et al. [5]. Specifically, they proposed an ensemble model consisting of three classifiers each trained on a different part of the patent text; the first classifier trained on the title and the abstract of the patent, the second on the description, and the third on the claims part. They utilized the same deep-learning classifiers for these three parts and combined the probabilities returned by the classifiers using an averaging method. They showed that the ensemble of the GRU models outperformed the other models significantly. More detailed review of ensemble models for patent classification may be found in [4].

2.4. Difference from existing studies

Our study involves the classification of patents by analyzing the meaning of their text. This process involves categorizing them into multiple classes, which distinguishes our approach from previous studies such as the works of Henriques et al. [1] and Roodsari et al. [26] that used hierarchical classification methods to organize patents into predetermined categories. Because of this difference, it's not feasible to make a direct comparison of our results with theirs since the problems being addressed are distinct. Our study shares similarities with the work of Chikkamath et al. [7], who used a finely-tuned transfer learning technique at the sentence level for multi-class patent classification. However, our study differs in that we operate at the paragraph level, which makes it unfeasible to compare our findings with theirs. Additionally, while their focus was on developing a Chrome extension for highlighting different sections within paragraphs, our objective is to classify patent paragraphs based on their semantic content.

3. Data and Model

In this section, we will give an outline of the dataset that we used in our study and describe the model that we developed for this research. The dataset was chosen from the United States Patent and Trademark Office (USPTO) and structured with the necessary labels.

Before explaining the dataset, we should emphasize that patent text data and general texts (such as books, articles, or websites) have several distinctions. For example, patents usually contain repetitive information making them difficult to read and comprehend. Another feature of patent data is the uses of specialized technical jargon, legal language, and domain-specific vocabulary. This results in a higher density of unique terms and abbreviations that may not be present in general texts. Moreover, patent data primarily focuses on inventions and innovations, detailing their functionality, novelty, and applications. The patent data typically revolves around technical descriptions, usage scenarios, and legal claims. Also, patent texts often feature longer texts, especially in the claims and detailed descriptions. Patents can be dense with information and may contain long, complex sentences filled with legal and technical detail.

3.1. Dataset

In order to evaluate the effectiveness of the proposed approach and compare it with established methods, we conducted a study using the PaSA dataset described in [7]. The PaSA dataset is specifically created to assess and improve explainable AI models in the realm of patent analysis. The paragraphs in PaSA contain vital key arguments that are important for thorough patent reviews and aid in the critical evaluation of an invention's limits. This dataset consists of 150,000 patterns from the USPTO, which are divided into three categories: 'technical advantages' (positive), 'technical problems' (negative), and 'other boilerplate text' (neutral). The patents included in the dataset are from the years 2010 to 2020. Three sample patent paragraphs are shown in Table 1. Also, you can find more information about the dataset in Table 2.

Table 1 shows the comparison of average and maximum word counts within the "neutral" class, which seem to be higher compared to the other two classes. To visually represent how patents are distributed based on their lengths across the three classes, we have displayed patents with lengths up to the average (approximately 300 words) in Figure 1. Additionally, Figure 2 displays word clouds that represent the most frequently used words within each of the three classes.

Table 1. Sample patent paragraphs from the PaSA dataset

Patent No.	Paragraph texts	Class
US09855987B2	However, in the main frames disclosed in the above Patent Literature, a portion of the main frames which is in the vicinity of the head pipe is required to prevent interference with a handle. For this reason, the vehicle body frame is not designed flexibly	Negative
US09857886B2	A head-mounted display comprising a display and a detector. The detector is configured to detect a direction of at least one of the head-mounted display and a line of vision. The display is configured to display an output image of an application, wherein the application is selected based on the direction being outside a range of a front direction.	Neutral
US09854852B2	The present invention can enhance the shaping capability of the breast side areas.	Positive

Table 2. Specification of the PaSA dataset.

Class	*# of records*	*Avg # of words*	*Max # of words*
Neutral	50000	577.80	7240
Positive	50000	124.83	6697
Negative	50000	193.46	4669
Total	150000	298.69	7240

3.2. Proposed Model

In our research, we have utilized the concept of meta-learning. It involves training a set of level-0 classifiers with the available data to make predictions. These predictions are then employed to train a meta-classifier, which ultimately makes the final prediction. Our primary objective is to create an advanced composite classifier $C*$ by combining n level-0 classifiers, $c_1, c_2, \dots, c_n$ and a meta-learner m. Figure 3 illustrates the overall framework of our proposed model. In Figure 3, we can see that the initial predictions are given by the level-0 classifiers. Then, these predictions are used to train the level-1 classifier. In our research, we have used the XGBoost classifier, which is a sophisticated technique, at the level-1 stage.

As shown in Figure 4, this technique operates by creating separate training trees using the original training data and new dataset splits, where samples are weighted according to their prediction error. The process involves three specific steps that are repeated until convergence is achieved.

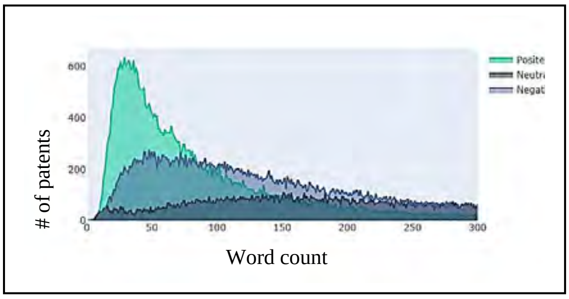


Figure. 1. Distribution of patents based on their lengths in the three classes.

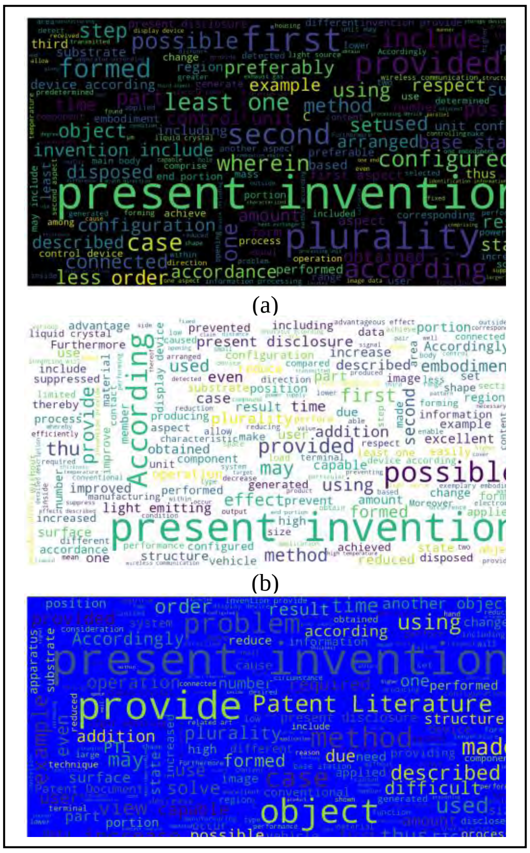


Figure. 2. Word clouds of three classes, (a) the positive, (b) the neutral, and (c) the negative class.

- An initial model m_0 is defined to predict the target variable y. Based on the predictions made by m_0, a residual $(y – m_0)$ is formed.
- A new model h_i is trained on the residuals from the previous step.
- m_{i-1} and h_i are combined to form a boosted version of m_{i-1} called m_i.

Steps 2 and 3 are repeated until the residuals are minimized to their maximum extent. The resulting predictions from the trained trees are then combined to create the final prediction for the proposed model.

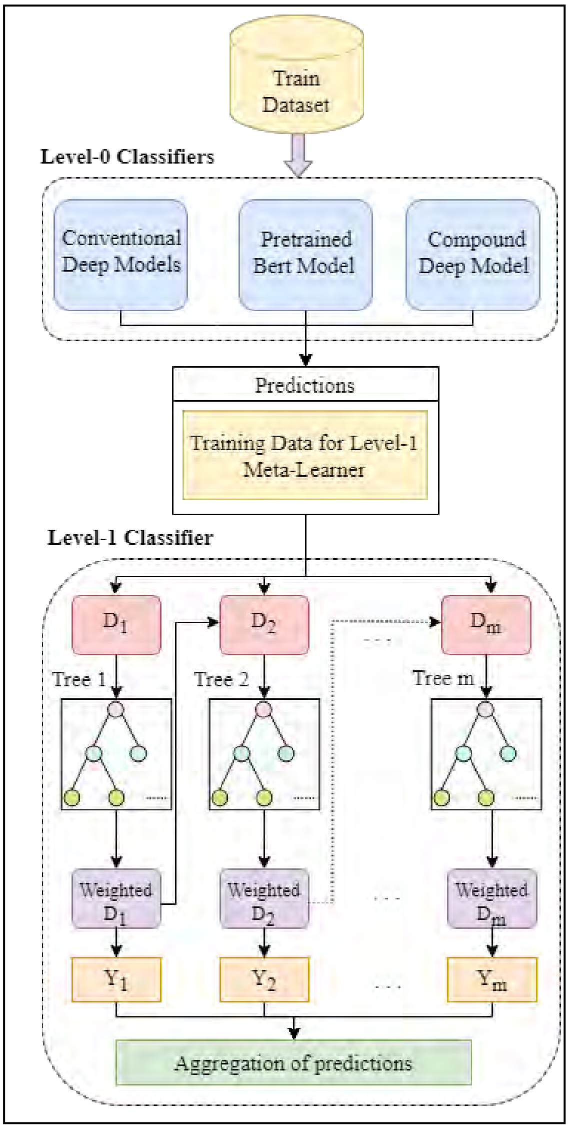


Figure. 3. Overall structure of the proposed model.

In the proposed model, we have used a combination of three different kinds of deep learning models as follows:

- Conventional deep models: This part includes five traditional deep neural networks from the categories of CNNs and RNNs. The structure of these models can be seen in Figure 4.
- Pre-trained transformer-based model: The BERT model is an encoder that uses transformers and has multiple layers [34]. The model has three embedding modules and 12 transformer layers, where each layer has a dense layer and an attention layer. In our research, we used the BERT model

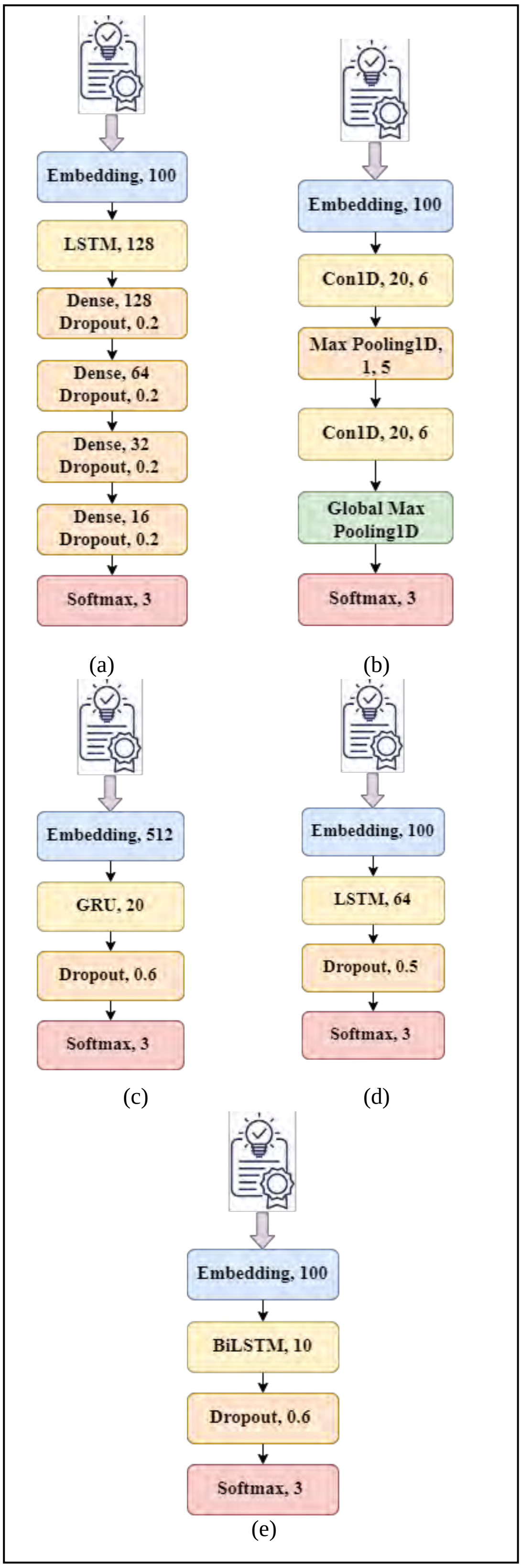


Figure. 4. Architectures of five conventional deep learning models used as level-0 classifiers in the proposed model. LSTM-1 (a), CNN (b), GRU (c), LSTM-2 (d), and BiLSTM (e).

implementation provided by Huggingface[1]. The architecture of this model is illustrated in Figure 5.

- Compound model: A compound model was developed by combining a CNN layer to extract local features and a bidirectional LSTM model to capture context in both directions within paragraphs. An attention layer was then utilized to emphasize the crucial components of the paragraph. The classification results were generated using a dense layer and a softmax layer. The architecture of this model is illustrated in Figure 6.

In our research, we have included the above-mentioned deep learning models because they can identify unique text features and use different methods to make precise classifications. This variety is important for effective utilization of meta-learning and stacking models [35]. We used the ensemble idea to improve performance, robustness, and generalization compared to using a single deep learning model. This can be particularly beneficial because different deep learning models can capture different aspects of the data and learn different representations. Therefore, their ensemble can reduce the likelihood of systematic errors. Moreover, the ensemble model tends to smooth out overfitting since errors made by one model might be compensated by correct predictions from others. Also, the ensemble can mitigate sensitivity to noise in the data or outliers. We have created a model that uses a modified version of stack generalization technique, and you can see it illustrated in Algorithm 1.

We have shown a correlation matrix in Figure 7 to demonstrate the diversity of level-0 models and hherr prddttt oons on the tttt datttttt . The lbbll GGold” indicates the target column or the gold standard in the dataset. The Bert model demonstrates the highest correlation with the Gold standard, and the five models exhibit different degrees of correlation with the Gold standard, as evident from the visual representation.

The main novelties of the proposed model are as follows:

1. The proposed model is the first deep ensemble model for paragraph-level semantic patent classification. Previous models either use traditional ML models or exploit single deep learning models.

2. The proposed model employs three kinds of learning models. Traditional deep learning models, transfer learning-based models, and a new combined model. The use of these models in the proposed ensemble improves the diversity and variances of predictions. This, in turn, improves the performance of the overall ensemble.

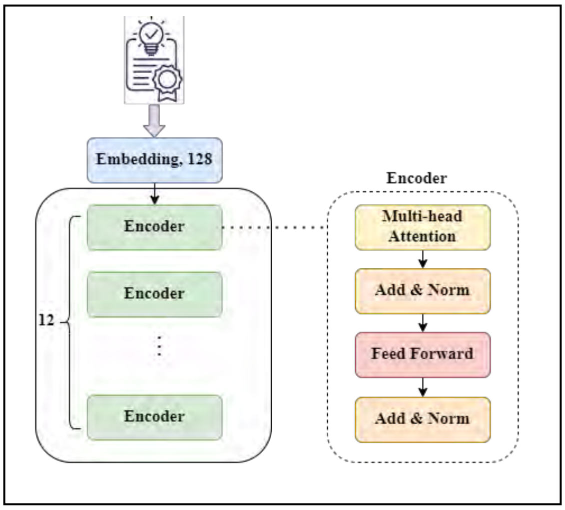


Figure. 5. Architectures of the BERT model used as one of level-0 classifiers in the proposed model.

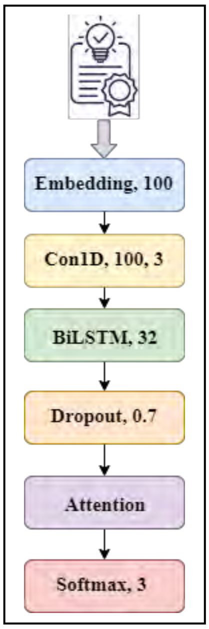


Figure. 6. Architectures of the compound model used as one of level-0 classifiers in the proposed model.

3. The proposed model uses the attention mechanism in its compound deep model at the level-0 layer to allow the model to focus on specific parts of the input text that are most relevant to the patent

[1] https://huggingface.co/transformers/v2.10.0/model_doc/bert.html

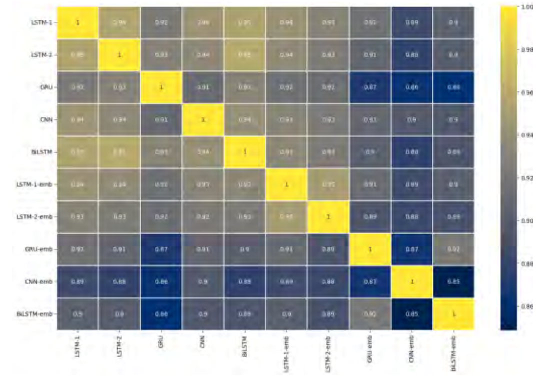


Figure. 7. Correlation between the outputs of level-0 classifiers.

paragraph type. It also enables the model to consider the entire input at once, making it easier to capture dependencies in long patent paragraph.

4. The proposed model extracts dynamic and context-sensitive representations of words via the compound model at the level-0 layer.
5. The proposed model uses a weighting tree algorithm (i.e. XGBoost) as the level-1 layer for better generalization and improved performance. It also tries to refine not-very-confident predictions made by level-0 classifiers by learning from the patterns in the errors made by the classifiers and improving the final predictions.

4. Experiments and Results

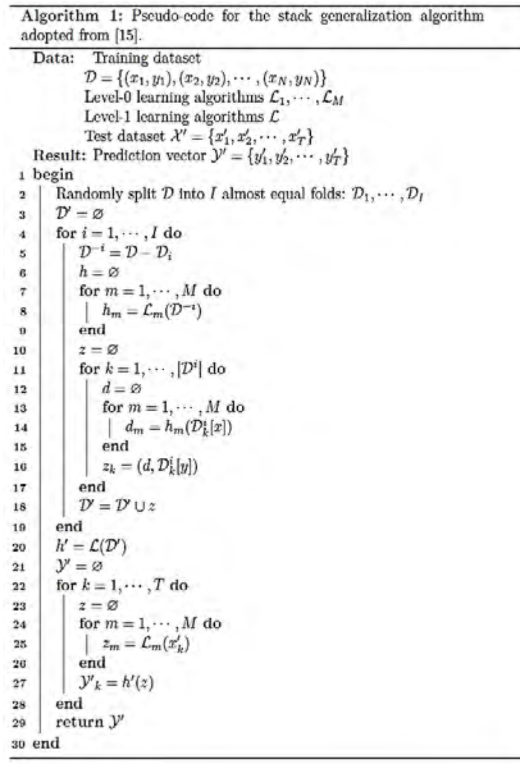
4.1. Experimental Settings

The research procedures were carried out using the Python programming language on the Google Collaboratory platform. This Linux-based environment eliminates the need for high-end hardware to develop machine learning models, as it provides a cloud-based processing capacity. The hardware specifications provided by this environment are detailed below:

- GPU: 1x Tesla K80, 12GB GDDR5 VRAM
- CPU: 1x Single Core Hyper-Threaded Xeon Processors @2.3Ghz, 45MB Cache
- RAM: 12.6 GB
- Disk: 68.3 GB

4.2. Compared Baselines

In order to demonstrate the effectiveness of our proposed approach, we utilized six state-of-the-art deep-learning techniques that have been used for semantic text classification:



```
Algorithm 1: Pseudo-code for the stack generalization algorithm adopted from [15].
Data:   Training dataset
        $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$
        Level-0 learning algorithms $\mathcal{L}_1, \cdots, \mathcal{L}_M$
        Level-1 learning algorithms $\mathcal{L}$
        Test dataset $\mathcal{X}' = \{x'_1, x'_2, \cdots, x'_T\}$
Result: Prediction vector $\mathcal{Y}' = \{y'_1, y'_2, \cdots, y'_T\}$
begin
    Randomly split $\mathcal{D}$ into $I$ almost equal folds: $\mathcal{D}_1, \cdots, \mathcal{D}_I$
    $\mathcal{D}' = \varnothing$
    for $i = 1, \cdots, I$ do
        $\mathcal{D}^{-i} = \mathcal{D} - \mathcal{D}_i$
        $h = \varnothing$
        for $m = 1, \cdots, M$ do
            $h_m = \mathcal{L}_m(\mathcal{D}^{-i})$
        end
        $z = \varnothing$
        for $k = 1, \cdots, |\mathcal{D}^i|$ do
            $d = \varnothing$
            for $m = 1, \cdots, M$ do
                $d_m = h_m(\mathcal{D}^i_k[x])$
            end
            $z_k = (d, \mathcal{D}^i_k[y])$
        end
        $\mathcal{D}' = \mathcal{D}' \cup z$
    end
    $h' = \mathcal{L}(\mathcal{D}')$
    $\mathcal{Y}' = \varnothing$
    for $k = 1, \cdots, T$ do
        $z = \varnothing$
        for $m = 1, \cdots, M$ do
            $z_m = \mathcal{L}_m(x'_k)$
        end
        $\mathcal{Y}'_k = h'(z)$
    end
    return $\mathcal{Y}'$
end
```

- CRNN [36]: This approach treats each sentence as an individual unit, where a CNN is applied to the input word vectors in each region. Then, a max pooling technique is used to decrease the size of the local features. Next, an LSTM layer is utilized to capture more complex relationships, and a linear decoder is applied to make predictions.
- IWV [37]: This model comprises three convolution layers, a max pooling layer, and a fully connected layer stacked sequentially for sentiment polarity classification.
- SS-BED [38]: This model utilizes two parallel LSTM layers that are trained on distinct word embedding matrices to analyze both semantic and sentiment features. The LSTM layers' outputs are then fed into a fully connected network, which consists of a single hidden layer, to predict outcomes.
- ARC [39]: This model uses a single-layer bidirectional GRU to process the word vectors. The output of the GRU is then fed into an attention layer to highlight important aspects of the input. The results of the attention mechanism are then passed through a CNN layer, followed by a max-pooling layer, and finally a fully connected

layer to produce the final output.

- AC-BiLSTM [40]: This model uses a one-dimensional CNN layer with CNNs of varying filter sizes to extract features at a localized level. The output from the CNN layer is then passed through a bidirectional LSTM layer, followed by an attention mechanism. Lastly, the model's output layer includes a dropout layer and a softmax layer for further refinement.
- ABCDM [41]: ADBCM is a technique that uses a combination of two bidirectional LSTM and GRU layers to capture contextual information efficiently from previous and upcoming contexts. It can consider the sequential flow of information in both forward and backward directions. The technique also incorporates an attention mechanism within the bidirectional layers, which helps to highlight specific words based on their importance. Additionally, ADBCM uses convolution and pooling mechanisms to simplify the features and extract localized features more effectively.

4.3. Evaluation Criteria

To evaluate the effectiveness of models, we employed *precision* i.. *recall* “ρ), *accuracy*, and *F1* evaluation criteria in the experiments [41] as shown in Equations (1)-(4).

$$\pi = \frac{TP}{TP+FP} \tag{1}$$

$$\rho = \frac{TP}{TP+FN} \tag{2}$$

$$F1 = \frac{2\times\pi\times\rho}{(\pi+\rho)} \tag{3}$$

$$accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{4}$$

where TP, TN, FP, and FN are true positive, true negative, false positive, and false negative, respectively.

4.4. Results

In this section, we will begin by comparing the proposed model with the baseline models that act as level-0 classifiers within the proposed model, as shown in Figure 4. After that, we will contrast the proposed model with advanced models discussed in section 4.2. Finally, we will demonstrate the impact of integrating pre-trained Glove embeddings into the models. In the experiments, we used Glove 100d of the Gensim library.

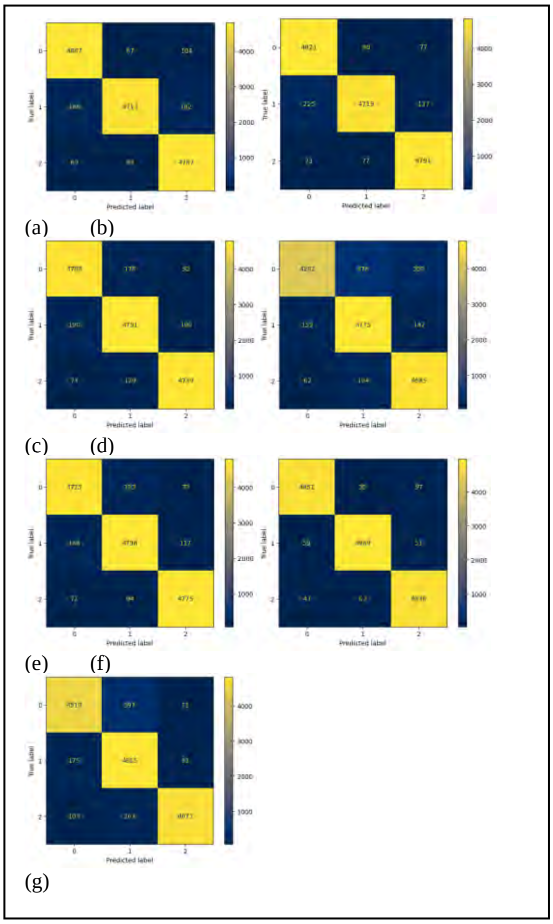
The comparison of level-0 classifiers within the proposed model is presented in Table 3. The table clearly shows that the transformer-based pre-trained BERT model outperforms all other models significantly. This superiority can be attributed to the complex nature of the model, its method of extracting features, and the attention mechanism it employs to focus on crucial parts of sentences. Following closely behind is the BiLSTM model, which incorporates two LSTM layers to capture contextual information from both past and future words in a sentence.

We have provided a detailed analysis of the performance of the base classifiers in Figure 8, through their confusion matrices. The results show that predicting class 1, which is also known as the positive class, was the most challenging for the classifiers. This was evidenced by the high rate of false negatives across all classifiers except for the BERT model. Additionally, the classifiers had difficulty in distinguishing between the neutral and positive classes, where the true class was positive but the classifiers incorrectly labeled it as neutral. This similarity in the text patterns between these two classes as compared to the negative class was highlighted.

In Table 4, we compared the state-of-the-art techniques outlined in section 4.2 with our proposed model. The results reveal that our proposed model outperforms all other models by a significant margin. Interestingly, models such as ACBiLSTM, SS-BED, and ABCDM, which incorporate bidirectional LSTM layers in their architecture, also demonstrated improved performance. This underscores the importance of considering sentence context in the context of the patent dataset. Essentially, the inclusion of bidirectional LSTM layers proved beneficial given the dataset's inclusion of lengthy patent-related paragraphs.

To assess the impact of using pre-trained embeddings in the input layers of deep models, we conducted a comparison of model performances with and without Glove pre-trained embedding in Figure 9.

It is evident from the illustration that using the pre-trained Glove embedding layer, in most cases, results in a decrease in the accuracy of the model. This finding is different from what was reported in some previous research studies like [32] and [35]. The difference in the textual content between our current study and the previous studies could be a possible explanation for this discrepancy. Our study deals with lengthy patent paragraphs that have vastly different contexts when compared to the brief tweets or user comments analyzed in [32] and [35]. In such



(a) (b)

(c) (d)

(e) (f)

(g)

Figure. 8. Comparison of the confusion matrices of the level-0 classifiers. LSTM-1 (a), LSTM-2 (b), GRU (c), CNN (d), BiLSTM (e), BERT (f), and Compound (g).

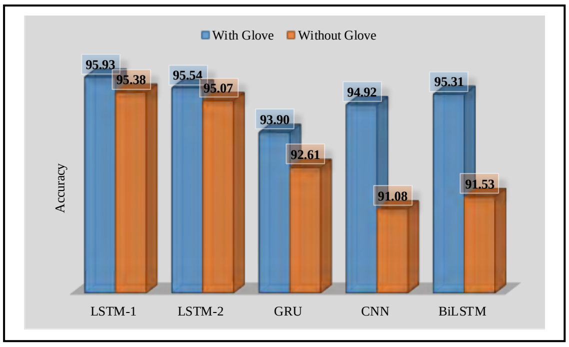


Figure. 9. Comparison of the accuracy of the level-0 in the proposed method, with- and without using the pre-trained Glove embedding.

cases, learning new embedding from the training dataset from the training dataset improves the accuracy, as shown in Figure 9.

5. Conclusions

Table 3. Comparison of the proposed method and the level-0 base classifiers.

method	*accuracy*	*precision*	*recall*	*F1*
LSTM-1	95.93	95.96	95.93	95.94
LSTM-2	95.54	95.56	95.56	95.54
GRU	93.90	94.16	93.94	93.89
CNN	94.92	94.93	94.92	94.93
BiLSTM	95.30	95.31	95.32	95.31
BERT	97.70	97.70	97.70	97.70
Compound	93.33	93.47	93.32	93.36
Proposed	**98.03**	**98.03**	**98.02**	**98.02**

Table 4. Comparison of the proposed method and the state-of-the-art methods.

method	*accuracy*	*precision*	*recall*	*F1*
ARC	95.52	95.73	95.51	95.53
CRNN	94.66	94.99	94.72	94.66
ABCDM	96.32	96.33	96.33	96.32
SS-BED	96.16	96.23	96.17	96.18
ACBiLSTM	96.33	96.47	96.32	96.36
IWV	93.66	93.70	94.71	94.65
Proposed	**98.03**	**98.03**	**98.02**	**98.02**

In this study, we have presented an innovative deep learning approach for semantic patent classification, which involves assigning patents to multiple semantic categories. This is different from previous research that focused on hierarchical classification into pre-defined groups. Our model combines various deep learning techniques, including LSTM, CNN, GRU, Bi-directional LSTM, and the pre-trained BERT model, in a stacked generalization approach. We conducted extensive experiments on a substantial dataset of 150000 USPTO patents, demonstrating that our model outperformed both baseline deep learning models and state-of-the-art approaches in multi-class semantic classification of textual data. Despite its contributions, this study is subject to some limitations. First, we employed some conventional simple deep models as level-0 classifiers in the proposed ensemble model. This may reduce the accuracy of the ensemble model. Second, we used stack generalization as a meta-learning process to improve the performance of the model using the XGBoost as the level-1 classifier. Other classifiers and aggregation schemas may improve the performance of the proposed model.

Our model can classify patents beyond the conventional three-class semantic classification, extending to other semantic categories within patents. Additionally, our model can be used as a useful filtering tool for downstream applications, such as paragraph highlighting, which is essential in patent analysis. Moving forward, we plan to evaluate the effectiveness of our model on similar patent text challenges and explore additional ensemble deep learning models to improve the efficiency of semantic patent classification tasks.

Declarations

Funding

This research did not receive any grant from funding agencies in the public, commercial, or non-profit sectors.

Authors' contributions

S. Nemati designed the experiments, analyzed the data, interpreted the results, and wrote the manuscript.

Conflict of interest

The authors declare that no conflicts of interest exist.

Acknowledgements

This work has been financially supported by the research deputy of Shahrekord University. The grant number was 2GRD6M44264.

References

[1] R. Hen5ique,, A. . erreira,1. nd M. Ca33lli, “A 9 . e Caee of1 Patent Classification Using Deep Learning with Transfer .. arniWges *Journal of Data and Information Science*, vol. 7, no. 3, pp. 49–70, 2022.

[2] R. Chikkamath, R. F. Ali, C. Hewel, and M. Endres, “xxplainable AetificialeIerBlligences for HiWhesghtingfaen SearehecBin Paaeacr cxfeeT2023.

[3] eeeYun aet t oMeu,, “Autoaa tee i sdiii conesi(F) Fpatents: A topic oo al cone F) oroach,” *Computers \& Industrial Engineering*, vol. 147, p. 106636, 2020.

[4] .. Kattt eri, M. Salapp aii s, and K. Diaaa ntara,, “An ensebb le framework for patent claiii fication,” *World Patent Information*, vol. 75, p. 102233, 2023.

[5] E. Kamateri, V. Stamatis, K. Diamantaras, and M. Salampasi,, “Autottt ed Single-Label Patent Classification uiing nnsemble Claiii fier”””” in *2022 14th International Conference on Machine Learning and Computing (ICMLC)*, 2022, pp. 324–330.

[6] J.-S. Lee and .. Hiiang, “Patent claiii fication by fine-tuning BRRT language model,” *World Patent Information*, vol. 61, p. 101965, 2020.

[7] R. Chikkamath, V. R. Parmar, C. Hewel, and M. Endres, “Patent Sentiee nt Analysis to Highlight Patent Paragraph,, ” *arXiv preprint arXiv:2111.09741*, 2021.

[8] .. Zhang, W. Liu, Y. Chen, and X. Yue, “Reliable multi-view deep patent claiii fication,” *Mathematics*, vol. 10, no. 23, p. 4545, 2022.

[9] M. E. Basiri, S. Nemati, M. Abdar, S. Asadi, and U. R. Acharrya, “A novel fuii on-based deep learning model for sentiment analysis of COVID-19 tweet””””*Knowledge-Based Systems*, vol. 228, p. 107242, 2021.

[10] .. Recch and R. d restel, “D. main-specific word embeddings for patent classification,” *Data Technologies and Applications*, vol. 53, no. 1, pp. 108–122, 2019.

[11] M. Sofean, “Deep learning baeed pipeline with uu ltichannel inputs for patent claiii fication,” *World Patent Information*, vol. 66, p. 102060, 2021.

[12] .. Hu, S. ii , .. Hu, and G. Yang, “A hierarchical feature extraction model for multi-label mechanical patent classif.cation,” *Sustainability*, vol. 10, no. 1, p. 219, 2018.

[13] .. Bai, I. Shi,, and S. Park, “MXXN: Multi-Stage Extraction Network for Patent Docuee nt C aiii f”cation,” *Applied Sciences*, vol. 10, no. 18, p. 6229, 2020.

[14] L. Abdelgawad, P. Kluegl, E. Genc, S. Falkner, and F. Hutter, “Optimizfng neural networks for patent classification,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2019, pp. 688–703.

[15] C. J. Fall, A. Törcsvári, K. Benzineb, and G. Karetka, “Automated categorization in the international patent classification,” in *Acm Sigir Forum*, 2003, pp. 10–25.

[16] D. ii kk, G. Biró, and A. Törcvvári, “A hierarchical online classifier for patent categorization,” in *Emerging technologies of text mining: Techniques and applications*, IGI Global, 2008, pp. 244–267.

[17] Y. Li and J. Shawe-Taylor, “Advanced learning algorithss for cross-language patent retrieval and clasii fication,” *Information processing \& management*, vol. 43, no. 5, pp. 1183–1199, 2007.

[18] A. S. Khattak an. G. Heyer, “Significance of low frequent terss in patent claiii fication uiing IPC hierarchy,” in *11th International Conference on Innovative Internet Community Systems (I2CS 2011)*, 2011.

[19] C.-H. Wu, Y. Ken, and –. Huang, “Patent classific tio, system using a new hybrid genetic algorithm support vector ccc hine,” *Applied Soft Computing*, vol. 10, no. 4, pp. 1164–1177, 2010.

[20] X. hhang, “Interactive patent claiii fication based on uu lti-classifier fuii on and active learning,” *Neurocomputing*, vol. 127, pp. 200–205, 2014.

[21] D. Seneviratne, S. Geva, G. Zuccon, G. Ferraro, T. Chappell, and M. Meireles, “A iignature approach to patent classification,” in *Information Retrieval Technology: 11th Asia Information Retrieval Societies Conference, AIRS 2015, Brisbane, QLD, Australia, December 2-4, 2015. Proceedings 11*, 2015, pp. 413–419.

[22] M. F. Grawe, C. A. Martins, and A. G. Bonfante, “Automated patent classification using word ebb edding,” in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2017, pp. 408–411.

[23] .. Xiao, G. Wang, andoY. vuo, “Reaaarch . n patent .ext classi,ica.ion baeed on word2vec and MMMM”tin *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, 2018, pp. 71i 74.

[24] S. Li, .. Hu, Y. Cui, and .. Hu, “DeepPatent: patent classification with convolutional neural networks and word ebb edding,” *Scientometrics*, vol. 117, pp. 721–744, 2018.

[25] H. Zhu, C. He, Y. Fang, B. Ge, M. Xing, and W. Xiao, “Pat nt nutoaa tic classification baeed on rymmecric hierarchical convolution neural network,” *Symmetry*, vol. 12, no. 2, p. 186, 2020.

[26] A. Haghighian Roudsari, J. Afshar, W. Lee, and S. Lee, “PatentNet: multt-label classification of patent documents uiing deep learning baeed language underttanding,” *Scientometrics*, pp. 1–25, 2022.

[27] M. Shajalal, S. Denef, M. R. Karim, A. Boden, and G. Steven,, “Unveiling Black-Boxes: Explainable Deep eearning Models for Patent Classification,” in *World Conference on Explainable Artificial Intelligence*, 2023, pp. 457–474.

[28] M. Suzgun, L. Melas-Kyriazi, S. Sarkar, S. D. Kominers, and S. Shieber, “The harvard uppto patent dataeet: A large-scale, well-structured, and multi-purpose corpus of patent application,, ” *Adv Neural Inf Process Syst*, vol. 36, 2024.

[29] R. ii , W. Yu, Q. Huang, and Y. Liu, “Patent Text Classification based on Deep Learning and Vocabulary Network,” *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 1, 2023.

[30] N. Yoshikawa and R. K.ettel, “Ipp ro. in. P.tent Classification using AI-Generated Summaries,” 2024.

[31] E. Kamateri, M. Salampasis, and E. Perez-Molina, “Will AI oolve the patent classification proble?? ,” *World Patent Information*, vol. 78, p. 102294, 2024.

[32] H. Mathiassen and D. Ortiz-Arroyo, “Autottt ic categorization of patent applications using classifier combination””” in *Intelligent Data Engineering and Automated Learning--IDEAL 2006: 7th International Conference, Burgos, Spain, September 20-23, 2006. Proceedings 7*, 2006, pp. 1039–1047.

[33] F. Benite,, S. Malmaii, and M. Zapp ieri, “Classifying patent applicationi ii th enmmble ee thod,,” *arXiv preprint arXiv:1811.04695*, 2018.

[34] J. Devlin, M.-W. Chang, K. Lee, and K. oo utanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

[35] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria, and U. R. Acharya, “ABCDM: An attention-based bidirectional CNN-RNN deep oo del for eentiee nt analysi,,” *Future Generation Computer Systems*, vol. 115, pp. 279–294, 2021.

[36] J. Wang, L.-C. Yu, K. R. aa i, and X. hh ang, “Diee nsional sentiment analysis using a regional CNN-MMMMmodel,” in *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)*, 2016, pp. 225–230.

[37] S. M. Rezaeinia, R. Rahmani, A. Ghodsi, and H. Veisi, “Sentiment analysis based on improved pre-trained word ebb edding,,” *Expert Systems with Applications*, vol. 117, pp. 139–147, 2019.

[38] A. Chatterjee, U. Gupta, M. K. Chinnakotla, R. Srikanth, M. Gall. y. and P. Agrawal. “Undertt anding emotions in text uii ng deep learning and big data,” *Computers in Human Behavior*, vol. 93, pp. 309–317, 2019.

[39] S. Wen and .. Li, “Recurppnt convolutional neural network with attention for twitter and yelp sentiment classification: ARC model dor eentim. nt claiii fication,” in *Proceedings of the 2018 International Conference on Algorithms, Computing and Artificial Intelligence*, 2018, pp. 1–7.

[40] G. ii u and .. Guo, “Bidirectional MMMMwith attention ccc hanimmand convolutional layer for text claiii fication,” *Neurocomputing*, vol. 337, pp. 325–338, 2019.

[41] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria, and U. R. Acharya, “ABCDM: An attention-based bidirectional CNN-RNN deep oo del for eentiee nt analysi,,” *Future Generation Computer Systems*, vol. 115, pp. 279–294, 2021.

Shahla Nemati was born in Shiraz, Iran, in 1982. She received the B.S. degree in hardware engineering from Shiraz University, Shiraz, in 2005, the M.S. degree from the Isfahan University of Technology, Isfahan, Iran, in 2008, and the Ph.D. degree in computer engineering from Isfahan University, Isfahan, in 2016. Since 2017, she has been an Assistant Professor with the Computer Engineering Department, Shahrekord University, Shahrekord, Iran. She has written several articles in the fields of data fusion, emotion recognition, affective computing, and audio processing. Her research interests include data fusion, affective computing, and data mining.

- Email: s.nemati@sku.ac.ir
- ORCID: 0000-0003-2906-5871
- Web of Science Researcher ID: AAA-3341-2019
- Scopus Author ID: 24512475100
- Homepage: https://www.sku.ac.ir/~snemati#