IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 5 July 2023, accepted 21 August 2023, date of publication 25 August 2023, date of current version 1 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3308588

RESEARCH ARTICLE

Simple and Effective Way to Disambiguate and Standardize Patent Applicants Using an Attention Mechanism With Data Augmentation

JUHYUN LEE[1], **SANGSUNG PARK**[2], **AND JUNSEOK LEE**[3]
[1]Institute of Engineering Research, Korea University, Seoul 02841, Republic of Korea
[2]Department of Data Science, Cheongju University, Cheongju-si 28503, Republic of Korea
[3]College of AI Convergence Engineering, Kangnam University, Youngin-si 16979, Republic of Korea

Corresponding authors: Sangsung Park (hanyul@cju.ac.kr) and Junseok Lee (jxli12@kangnam.ac.kr)

This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant NRF-2022R1I1A1A01069422, and in part by NRF funded by the Korean Government [Ministry of Science and ICT (MSIT)] under Grant RS-2023-00247410.

ABSTRACT Innovation in artificial intelligence and data science has sparked evolutions across numerous industries. Some companies are focusing on developing novel technologies to seize a rapidly evolving market, while others are exploring new business models to keep pace. The former and latter are typically referred to as first movers and fast followers in the technology market and identifying them can offer insights into technology market trends. Patent analysis is a good approach to exploring first movers and fast followers. However, patent applicants are classified into different patterns based on the structure or type of a company, making it challenging to disambiguate and standardize patent applicants. Therefore, this study proposes a method to disambiguate and standardize patent applicants. We present a simple and effective data augmentation approach that can help understand patent applicant patterns. The proposed approach trains on the augmented data via the attention mechanism. Our experiments provide empirical evidence for the performance of the proposed method, which accurately classifies 96.6% of the augmented data. Moreover, statistical hypothesis testing validates that the output of the proposed method is consistent with the ground truth.

INDEX TERMS Attention mechanism, data augmentation, named entity recognition, patent applicants.

I. INTRODUCTION

The global technology market has entered a new era of innovation, sparked by DeepMind's AlphaGo and OpenAI's ChatGPT. As the achievements of scientific advancements are continually revealed, there is an escalating demand for highly specialized research [1]. Companies are allocating substantial budgets to research and development (R&D) to seize new markets, and their efforts have led to the development of various technologies. Along this path, they are classified as either first movers or fast followers. First movers focus on novel endeavors to seize markets, whereas fast followers focus on pursuing business opportunities after them. Consequently, companies aspire to spearhead the technology market or identify their competitors, and patents are an excellent means to meet their needs. A patent, a legal document drafted to assert rights over a technology, helps identify (i) the technology market leaders and (ii) their competitors.

Companies can identify first movers and fast followers through patent analysis. A patent includes details about the applicant and the inventor of the technology. In many cases, the applicant is the company to which the inventor belongs. Therefore, even with a simple patent landscape, information about applicants can provide a wealth of insights into the technology market. However, ambiguous and non-standardized names of patent applicants make it challenging to obtain these insights. This paper proposes a simple and effective solution to this problem. That is, our goal is to disambiguate and standardize patent applicants.

The associate editor coordinating the review of this manuscript and approving it for publication was Agostino Forestiero.

Thoma et al. [2] highlighted the difficulty of standardizing rights holders like inventors or applicants for patents because their titles may vary depending on the entity identifiers of applicants or how the patent office drafts them. Nevertheless, previous studies have proposed various methods for the disambiguation and standardization of inventors and applicants. For example, Raffo and Lhuillery [3] suggested a name standardization method for patent inventor retrieval. Zhang et al. [4] introduced a patent standardization method for enhancing technology productivity at a microscale. Pezzoni et al. [5] developed an inventor matching process using natural language processing (NLP) techniques like parsing. Morrison et al. [6] proposed a patent disambiguation method based on inventors' geolocation data, as patent documents include the addresses of inventors and applicants. However, since previous studies employed rule-based approaches, they have limitations when applied to data with new patterns.

As interest in patent analysis has increased, extensive research has employed machine learning-based approaches. For instance, Li et al. [7] provided an overview of disambiguation using co-ownership and collaborative variables, visualizing inventors in geolocation data using networks. Ventura et al. [8] merged inventor disambiguation algorithms proposed in prior studies and enhanced standardization performance. Moreover, Kim et al. [9] aimed to maximize the effectiveness of disambiguation by using an ensemble model and clustering analysis, proposing a novel function to enhance scalability. Yin et al. [10] emphasized two key factors to consider for patent owner disambiguation: synonyms and homonyms. Synonyms and homonyms represent variations of single names and names different owners share, respectively [11], [12]. The researchers particularly focused on addressing the issues caused by synonyms and homonyms in China.

Onishi et al. [13] introduced a manual approach using addresses and websites to disambiguate and standardize patent applicants. Additionally, Neuhäusler et al. [14] proposed a procedure for cleaning patent applicant names, calculating their similarity, and matching them [14]. Their manual procedure is appropriate to disambiguate and standardize patent applicants, but due to the use of conventional wisdom, this approach has difficulty handling cases that deviate from formalized procedures. To the best of our knowledge, few studies have examined these limitations in depth. To reduce these research gaps, we suggest a data augmentation and attention mechanism-based method.

The main contributions of our paper are summarized as follows:

- This paper proposes a method for the disambiguation and standardization of patent applicants, as this can help identify companies of technology market leaders and followers.
- We introduce a simple and effective data augmentation process to improve the performance of the proposed method. This paper presents and statistically verifies a research hypothesis and validates the effect of data augmentation.
- The attention mechanism helps the proposed method to focus on meaningful aspects of patent applicants. Our goal is to increase the performance of disambiguation and standardization for patent applicants by training the model using augmented data and the attention mechanism.

The rest of the paper is structured as follows. Section II provides the background related to our method. Section III explains the proposed method, and Section IV describes the experiment. Finally, Section V discusses several limitations and offers suggestions for future work.

II. BACKGROUND

NLP is used in various industries, including document summarization, machine translation, and question-answering. Recent natural language understanding (NLU) advances have enabled machines to comprehend the context and semantic features of language. With the development of NLP and NLU, we can easily retrieve database query results [15], [16], [17]. Additionally, chatbots respond to questions and reduce customer discomfort [18], [19], [20]. To allow machines to comprehend natural language, numerous methodologies have been proposed. Among these, the recurrent neural network (RNN) is a significant contribution to the advancement of NLU. Hopfield [21] proposed an architecture that learns sequences of tokens in texts. Suppose the weight to obtain the output of input x_t at time step t is w_{xh}, and the weight flowing from hidden state h_{t-1} to h_t at time step $t-1$ is w_{hh}. Then, the hidden state h_t at time step t is as follows in (1):

$$h_t = tanh\left(w_{hh} \cdot h_{t-1} + w_{xh} \cdot x_t + b_h\right) \tag{1}$$

where b_h and $tanh$, respectively, represent the bias and the hyperbolic tangent, the activation function.

Fig. 1(a) shows the recurrent mechanism of the RNN. The recurrent mechanism was well suited for extracting natural language features, but long-term dependencies limited the performance of RNNs in comprehending natural language. RNNs sequentially receive natural language and process it using the recurrent mechanism. As texts grew longer, the vanishing gradients problem of neural network was discovered. Hochreiter and Schmidhuber [22] introduced the long short-term memory (LSTM) algorithm to resolve this issue. Using the forget gate, input gate, and output gate, the LSTM selects (i) information to be forgotten from the previous time step and (ii) information to be remembered from the current time step. The forget gate f_t at time step t is given by (2):

$$f_t = \sigma\left(w_{xf} \cdot x_t + w_{hf} \cdot h_{t-1} + b_f\right) \tag{2}$$

where $w_{\cdot f}$ and b_f, respectively, represent weight and bias connected to f_t and σ indicates a sigmoid function.

The forget gate f_t decides which information to forget, and the input gate selects which information to remember. In (3) and (4), i_t indicates the amount of information to be updated,

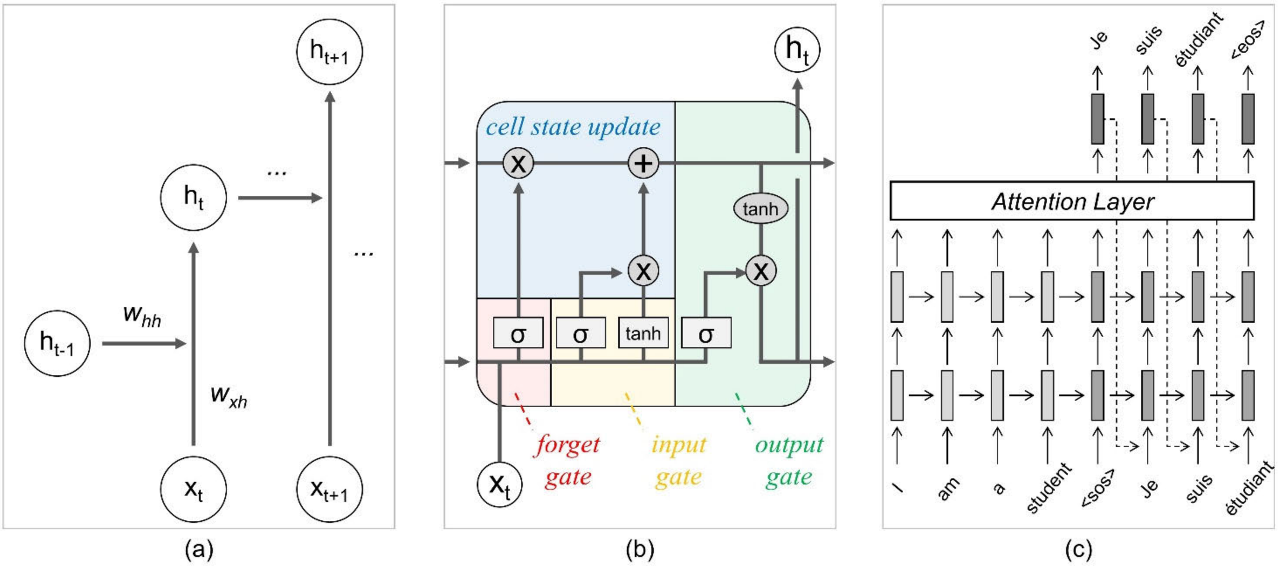


FIGURE 1. Architecture for NLU. (a) RNN. (b) LSTM. (c) attention mechanism-based LSTM.

while $\tilde{c}_t$ represents the information that needs to be updated.

$$i_t = \sigma\left(w_{xi} \cdot x_t + w_{hi} \cdot h_{t-1} + b_i\right) \tag{3}$$

$$\tilde{c}_t = tanh\left(w_{xc} \cdot x_t + w_{hc} \cdot h_{t-1} + b_c\right) \tag{4}$$

Now, the cell state updates its weights based on the information obtained via the forget and input gates. In (5), c_t represents the information updated at time step t.

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{5}$$

where the operator $\odot$ denotes the Hadamard product [23].

The output gate then decides what information to output, which is used to update weights in the next time step. Thus, (6) and (7) represent output information o_t and weight h_t derived from the output gate, respectively.

$$o_t = \sigma\left(w_{xo} \cdot x_t + w_{ho} \cdot h_{t-1} + b_o\right) \tag{6}$$

$$h_t = o_t \odot tanh\left(c_t\right) \tag{7}$$

The forget gate, input gate, and output gate of the LSTM are illustrated in Fig. 1(b). The LSTM outperforms the RNN in most NLU tasks. The sequence-to-sequence (Seq2Seq) architecture boosts the NLU task performance of the RNN and LSTM. The Seq2Seq architecture comprises an encoder and a decoder. Typically, the encoder extracts features from the dataset, and the decoder uses these extracted features to generate data. In machine translation, the encoder converts the input sentence into a vector and the decoder returns the translated sentence.

The RNN architecture, composed of Seq2Seq, employs a context vector containing information about the input sentence. However, the context vector is limited in that it loses information when the sentence becomes lengthy. Vaswani et al. [24] proposed transformers with attention mechanisms to address such limitations of the RNN-based Seq2Seq architecture. Fig. 1(c) shows Seq2eq employing the attention mechanism. Consider the translation of "I am a student" into French. The translated sentence is "je suis étudiant." The Seq2Seq decoder receives the <sos> token that indicates the beginning of the sentence and outputs "je," corresponding to "I." This is where the attention mechanism comes in, helping the model to focus on "I" when it is returning "je."

When performing NLU tasks, scaled dot-product attention instructs Seq2Seq on which tokens to focus on. Scaled dot-product attention is simple. Suppose the encoder's hidden state at time step t is k_t and the decoder's hidden state in the decoder is q_t. Then, the attention score for the t-th hidden state in the encoder is as given in (8):

$$\frac{q_t^T \cdot k_t}{\sqrt{d}} \tag{8}$$

where q_t^T and d represent transposed q_t and its size, respectively.

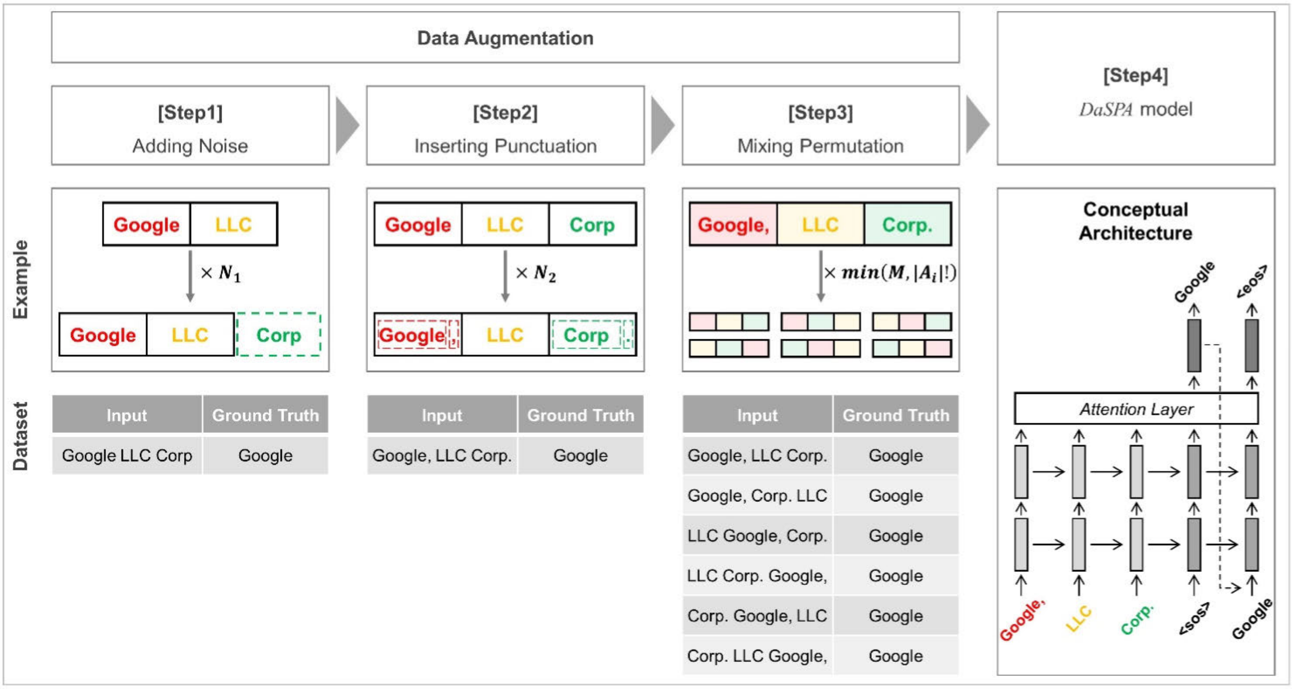
The attention mechanism improves the limitations of RNN-based Seq2Seq models with a simple dot-product. This study employs a scaled dot-product attention-based LSTM for a simple and effective approach.

III. PROPOSED METHOD

This paper suggests the disambiguation and standardization of patent applicants (*DaSPA*) model. The proposed method leverages patent applicant features for simple and effective data augmentation. In addition, it uses the attention mechanism to disambiguate and standardize patent applicants.

A. FLOWCHART OF PROPOSED METHOD

Fig. 2 shows a flowchart of the proposed method. The goal of the first phase is to increase the number of patent applicants.



Input	Ground Truth
Google LLC Corp	Google

Input	Ground Truth
Google, LLC Corp.	Google

Input	Ground Truth
Google, LLC Corp.	Google
Google, Corp. LLC	Google
LLC Google, Corp.	Google
LLC Corp. Google,	Google
Corp. Google, LLC	Google
Corp. LLC Google,	Google

FIGURE 2. Flowchart of proposed method.

Data augmentation enables the *DaSPA* model to pay attention to the ground truth of patent applicants. Data augmentation is carried out by adding noise, inserting punctuation, and then mixing permutations.

In Step 1, the proposed method adds noise to patent applicant names, which we define as words not present in the ground truth, like "Corp" or "LLC." In Step 2, punctuation, like periods or commas, is inserted to make the augmented data as similar as possible to the actual patent applicant names. Step 3 involves mixing permutations of patent applicant names to increase the diversity of the dataset.

Lastly, the *DaSPA* model is trained on the augmented data, with the augmented data and ground truth serving as input and output, respectively. Using the augmented data as input, the *DaSPA* model outputs the ground truth, starting from the <sos> token, which indicates the beginning of the sentence. The *DaSPA* model continues to generate text until it outputs the <eos> token, which represents the end of the sentence.

B. STEP1: DATA AUGMENTATION – ADDING NOISE

The proposed method adds noise, inserts punctuation, and returns disambiguated and standardized patent applicants in a simple and effective manner using random permutation. The noise in our augmented data is the words of patent applicants' entity identifiers [25]. For example, the noise "Corp" represents a company. "Inc" and "Ltd" are used for patent applicants that are either stock companies or limited liability companies. Table 1 lists the noises used for the patent applicant disambiguation and standardization tasks.

TABLE 1. List of noises used for patent applicant disambiguation and standardization tasks.

Noise	Description
FOUNDATION	Represents an institution established with assets such as donations
GROUP	Refers to the parent company of many affiliates
INC	As a word meaning a corporation, 'CO', 'CORP', 'CORPORATION', and 'LTD' are used in the US and UK. Japan and Germany use 'Kabushiki gaisha' and 'Aktiengesellschaft (AG)' respectively
IND	Abbreviation for industry
INVEST	Refers to a company involved in investment
LLC	Represents the abbreviation of limited liability company. In Germany it is referred to as Gesellschaft Mit Beschränkter Haftung (GMBH)
SE	Societas Europaea, which is a public company registered under European Union corporate law

We added noise to ensure the proposed method focused on the patent applicant disambiguation and standardization tasks. The pseudo code for the first stage of data augmentation, adding noise, is specified in Algorithm 1.

Suppose the i-th applicant among N patent applicants is A_i and that its j-th token is $A_i^{(j)}$ ($j = 1, 2, \ldots, k_i$). For example, if A_i is <Google LLC>, then <Google> and <LLC> are, respectively, $A_i^{(1)}$ and $A_i^{(2)}$. $|A_i|$, the number of tokens in A_i, is $k_i = 2$. In this case, the ground truth of <Google LLC> is <Google>.

```
Algorithm 1 Adding_Noise
Input: A list of patent applicants, $\mathcal{A}$
       Number of iterations, $N_1$
       Noise set, $S_{noise}$
Output: A list of patent applicants with added noises
1:  for $A_i$ in $\mathcal{A}$ do
2:    for 1 to $N_1$ do
3:      Adding spaces at the beginning and end of $A_i$
4:      Randomly select a number ($n_1$) from 1 to $|A_i|$
5:      Randomly select one noise from the $S_{noise}$
6:      Adding noise to the $n_1$-th space
7:    end for
8:  end for
```

Next, Algorithm 1 adds spaces before and after A_i (Line 3). Then, there are $(k_i + 1)$ number of spaces in the patent applicant with $|A_i|$ equal to k_i. Algorithm 1 then iteratively adds noise to A_i by N_1 times (Lines 2–7). In the first iteration, the proposed method adds noise to the n_1-th space, randomly selected between 1 and $(k_i + 1)$. Now, $|A_i|$ is equal to $(k_i + 1)$. In the next iteration, noise is added to a position randomly selected between 1 and $(k_i + 2)$. The value of $\left|A'_i\right|$ of A'_i that went through N_1 iterations is $(k_i + N_1)$. For example, when N_1 is 1, the output of Algorithm 1 for <Google LLC> is <Google LLC Corp>.

C. STEP2: DATA AUGMENTATION – INSERTING PUNCTUATION

The proposed method adds punctuation to noise-added patent applicant names because we want the model to pay attention to the names of patent applicants, not noises. Data augmentation, therefore, adds punctuation so that the noises added in Step 1 appear more realistic. Punctuation includes periods and commas used to describe the company type, such as in <Co., Ltd.>.

```
Algorithm 2 Inserting_Punctuation
Input: A list of patent applicants, $\mathcal{A}$
       Number of iterations, $N_2$
       Punctuation set, $S_{punc}$
Output: A list of patent applicants with inserted
        punctuations
1:  for $A_i$ in $\mathcal{A}$ do
2:    for 1 to $N_2$ do
3:      Adding spaces at the beginning and end of $A_i$
4:      Randomly select a number ($n_2$) from 2 to $|A_i|$
5:      Randomly select one punctuation from the $S_{punc}$
6:      Inserting punctuation after the ($n_2 - 1$)-th token
7:    end for
8:  end for
```

Algorithm 2 is the pseudo code for adding punctuation. We insert N_2 number of punctuation marks to noise-added patent applicant name A'_i, whose length with spaces added at its beginning and end is $(k_i + N_1 + 1)$. Then, Algorithm 2 selects a natural number n_2 at random between 2 and $(k_i + N_1 + 1)$ (Line 4). Algorithm 2 subsequently adds randomly selected punctuation after the $(n_2 - 1)$-th token (Lines 5-6). The length of A'_i does not change after N_2 iterations $\left(\left|A'_i\right| = |A''_i|\right)$. The total number of possible permutations of a token of length N is equal to (9).

$$N! = \prod\nolimits_{X=1}^{N} X \tag{9}$$

where X is a natural number between 1 and N.

Algorithm 2 selects a natural number n_2 from 2 to $(k_i + N_1 + 1)$, not from 1, and adds punctuation after the $(n_2 - 1)$-th token because punctuation usually comes after a token. Therefore, we set the minimum value of n_2 to 2.

$$(k_i + N_1)! < (k_i + N_1 + N_2)! \; when \; N_2 > 0 \tag{10}$$

Adding punctuation following a token is relevant to the third stage of data augmentation. This stage expands noises and punctuation-added patent applicant names through permutations. Suppose the punctuation added in the second stage is an independent token; then the length of the output from the second stage is $(k_i + N_1 + N_2)$. In contrast, the length of the output A''_i from Algorithm 2 is $(k_i + N_1)$. Due to (10), the number of permutations considered in the Algorithm 2 approach is always smaller than the other. Therefore, the proposed method can reduce unnecessary computations for data augmentation. For example, when N_2 is 2, the output from Algorithm 2 for <Google LLC Corp> is <Google, LLC Corp.>.

D. STEP3: DATA AUGMENTATION – MIXING PERMUTATION

The final stage in data augmentation is mixing permutations, which involves obtaining permutations of tokens from patent applicant names processed in the first and second stages and mixing them to augment the data. Suppose the output after the first and second steps is <Google, LLC Corp.>. There are three tokens in this case: <Google,>, <LLC>, and <Corp.>. The permutations of <Google, LLC Corp.> are <Google, LLC Corp.>, <Google, Corp. LLC>, <LLC Google, Corp.>, <LLC Corp. Google,>, <Corp. Google, LLC>, and <Corp. LLC Google,>. We can get 3! permutation outputs from this case.

Mixing permutations expands a single patent applicant's name into multiple forms, the number of which depends on the token numbers. To avoid having an excessive number of permutations, we use the hyperparameter M, which controls the number of permutations. Moreover, the proposed method examines the min $(M, |A_i|!)$ number of permutations. $|A_i|!$ represents all possible permutations of A_i, and mixing permutations increases the number of data instances. Thus, (11) indicates how many data instances exist for the original N data instances after mixing permutations.

$$\sum\nolimits_{i=1}^{N} min\,(M, |A_i|!) \tag{11}$$

Algorithm 3 is the pseudo code for the third stage of data augmentation. Algorithm 3 returns patent applicant permutations with a minimum of M and $|A_i|!$. For example, if M is 3, Algorithm 3 returns <Google, LLC Corp.>, <Google, Corp. LLC>, and <LLC Google, Corp.> for <Google, LLC Corp.>.

Algorithm 3 Mixing_Permutation

```
Input: A list of patent applicants, $\mathcal{A}$
       Maximum number of iterations, $M$
Output: A list of permutated patent applicants
1: for $A_i$ in $\mathcal{A}$ do
2:     for $m$ to min $(M, |A_i|!)$ do
3:         while $m \leq$ min $(M, |A_i|!)$ then
4:             Select the case of the $m$ -th permutation
5:         end while
6:     end for
7: end for
```

Lines 3-5 in Algorithm 3 augment patent applicants names with permutations, which increases the diversity of patent applicant titles with noises and punctuation. Thereby, data augmentation helps the proposed method pay more attention to the names of patent applicants.

E. STEP4: DASPA MODEL

This study proposes a *DaSPA* model based on an attention mechanism appropriate for patent applicant disambiguation and standardization tasks through data augmentation. Algorithm 4 is the pseudo code for *DaSPA* model. The *DaSPA* model is trained on data augmented by adding noise, inserting punctuation, and mixing permutations with the original data.

Algorithm 4 DaSPA

```
Input: A list of patent applicants, $\mathcal{A}$
       A list of disambiguated and standardized
       patent applicants, $\mathcal{Y}$
       Number of iterations, $N_1$ and $N_2$
       Maximum number of iterations, $M$
       Noise set, $S_{noise}$
       Punctuation set, $S_{punc}$
Output: DaSPA model based on attention mechanism
1: for $(A_i$ in $\mathcal{A})$ and $(y_i$ in $\mathcal{Y})$ do
2:     $A_i^{'} = Adding\_Noise\ (A_i, N_1, S_{noise})$
3:     $A_i^{''} = Inserting\_Punctuation\ \left(A_i^{'}, N_2, S_{punc}\right)$
4:     $A_i^{'''} = Mixing\_Permutation\ \left(A_i^{''}, M\right)$
5:     for $a_i$ in $A_i^{'''}$ do
6:         $DaSPA = Attention\ (a_i, y_i)$
7:     end for
8: end for
```

Lines 2–4 of Algorithm 4 explain data augmentation. The input and output of *DaSPA* are augmented data and manually disambiguated and standardized data, respectively. *DaSPA* is an attention mechanism-based language model. After training is complete, *DaSPA* can be divided into an encoder and a decoder. The *DaSPA* decoder is used for the disambiguation and standardization of patent applicants.

Hypothesis. *DaSPA* is a model suitable for the disambiguation and standardization of patent applicants.

$$\begin{aligned} &H_0 : notH_1 \\ &H_1 : DaSPA\ improves\ the\ performance\ of \\ &\quad disambiguation\ and\ standardization \\ &\quad of\ patent\ applicants \end{aligned} \tag{12}$$

We expect the *DaSPA* model to transform patent applicant names into ones similar to the ground truth. The hypothesis of (12) is to validate that the *DaSPA* model is appropriate for the disambiguation and standardization of patent applicants.

IV. EXPERIMENTS

This study aims to evaluate the performance of the *DaSPA* model for patent applicant disambiguation and standardization tasks. This paper introduces the data and model architecture used in this study and describes the training and performance evaluation of the *DaSPA* model.

A. EXPERIMENTAL SETUP

We used two datasets for the experiment. The first dataset (D^{Raw}) consisted of 1,016 patent documents and included 439 unique applicants. The second dataset (D^{FK}) was a list of global companies, collected from (i) the "Fortune 500" list published by Fortune in 2022 and (ii) companies listed on the KOSPI (Korea Composite Stock Price Index).

TABLE 2. Descriptions of datasets used in experiments.

Dataset	Training	Test	Total	# of unique samples
D^{Raw}	826	190	1,016	439
D^{Aug}	121,327	30,173	151,500	439
D^{Aug+FK}	175,606	43,694	219,300	954

Table 2 presents the number of samples contained in three datasets: D^{Raw}, D^{Aug}, which is D^{Raw} augmented using the proposed method, and D^{Aug+FK}, which is a combination of $D^{Aug}$and D^{FK}. During the data augmentation process, the values of N_1, N_2, and M were all set to 5. D^{Aug+FK} contained approximately twice as many unique samples as D^{Aug}.

Table 3 lists the noise set (S_{noise}) used in the study. The punctuation set (S_{punc}) used in the study included AG, CO, CORP, CORPORATION, FOUNDATION, GMBH, GROUP, INC, IND, INVEST, Kabushiki gaisha, LLC, LTD, and SE (see Table 1). As our goal was to propose a simple and effective *DaSPA* model by adding minimal noise and punctuation, we used only periods, commas, hyphens, and slashes for noises added to patent applicant names.

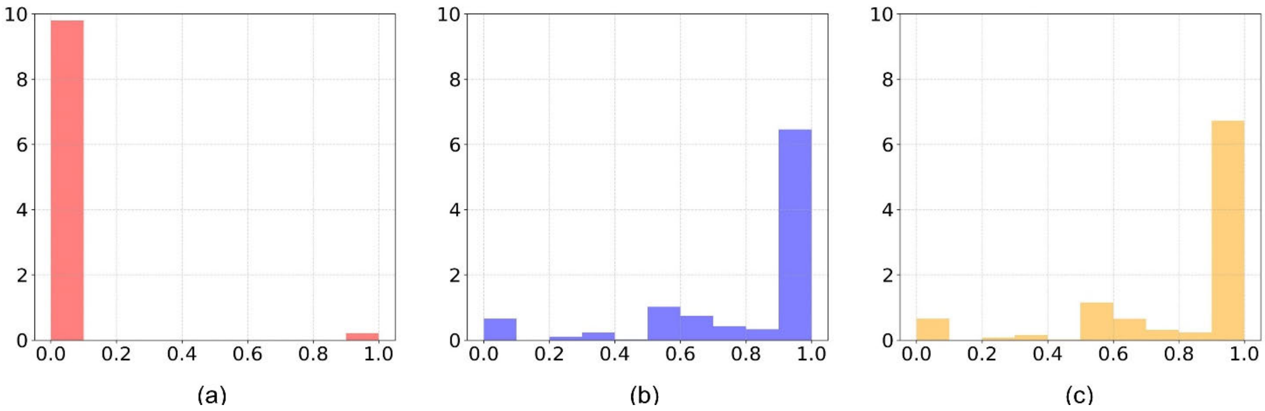


FIGURE 3. Comparison of results of proposed method and MUP. (a) DaSPA_v0. (b) DaSPA_v1. (c) DaSPA_v2.

TABLE 3. List of punctuation used for experiments.

Punctuation	Description	Punctuation	Description
.	Period used to end a sentence	.,	Period and comma
,	Comma used in sentences	,.	Comma and period
..	Two periods	-	Hyphen used to join words
,,	Two commas	/	Slash, a type of punctuation mark

B. EXPERIMENTAL RESULTS

We compare the performance of the *DaSPA* model trained on three different datasets. *DaSPA_v0*, *DaSPA_v1*, and *DaSPA_v2* are models trained on D^{Raw}, D^{Aug}, and D^{Aug+FK}, respectively. Accuracy, macro precision, macro recall, and macro F-1 score are the indicators of the models' performance (see Appendix A). The model performance results are compared in Table 4. *DaSPA_v0*, trained on D^{Raw}, performed poorly. However, *DaSPA_v1*, trained with augmented data, performed significantly better, accurately classifying 96.6% of the 439 labels. *DaSPA_v2* had slightly lower performance than *DaSPA_v1*. However, *DaSPA_v2* covers twice as many labels as *DaSPA_v1*, making it difficult to conclude that *DaSPA_v2* has lower performance than *DaSPA_v1*.

TABLE 4. Comparison of proposed model performance.

Model	Accuracy	Macro Precision	Macro Recall	Macro F1-score
DaSPA_v0	0.000	0.000	0.000	0.000
DaSPA_v1	**0.966**	**0.961**	**0.949**	**0.950**
DaSPA_v2	0.940	0.948	0.869	0.889

Fig. 3 is the histogram of the modified unigram precision (MUP), which shows how similar the patent applicant names and ground truth are (see Appendix A). Fig. 3(a) is the MUP of *DaSPA_v0*. *DaSPA_v0*'s MUP is skewed to the left. Fig. 3(b) and (c) show the MUP of *DaSPA_v1* and *DaSPA_v2*, respectively. The MUPs of both models were enhanced over *DaSPA_v0*. In particular, their average MUPs are greater than 0.8. Therefore, *DaSPA_v1* and *DaSPA_v2* are suitable for the disambiguation and standardization of patent applicant tasks.

Now, we present statistical evidence to demonstrate that the proposed method is suitable for the tasks of disambiguation and standardization for patent applicants. Table 5 shows the results of the statistical hypothesis testing. Avg and Std in the table represent the mean and standard deviation of MUP. We used Levene's test, T-test, and Wilcoxon Rank-Sum test for our statistical testing. Firstly, Levene's test statistically verifies whether the variances of MUP among *DaSPA* models are equivalent. Next, the T-test validates the mean differences of MUP based on variance homogeneity. Lastly, the Wilcoxon Rank-Sum test shows the nonparametric statistics for the T-test. The experimental significance level was set at 0.05. Appendix B provides detailed procedures for Levene's test, T-test, and Wilcoxon Rank-Sum test used to support the hypothesis.

The MUP average of *DaSPA_v0* was very poor at 0.021. In contrast, the MUPs of *DaSPA_v1* and *DaSPA_v2* trained with augmented data significantly improved to 0.815 and 0.824, respectively. In Levene's test, the variances of *DaSPA_v0* and *DaSPA_v1* differed. While *DaSPA_v1* and *DaSPA_v2* did not have identical variances, they were statistically more similar than *DaSPA_v0*, as Levene's test statistics decreased by 15.250, from 57.999 to 52.623. In the T-test, the MUPs of *DaSPA_v1* and *DaSPA_v2* were statistically significantly higher than the MUP of *DaSPA_v0*. Surprisingly, *DaSPA_v2* had a higher MUP than *DaSPA_v1*. *DaSPA_v1* accurately classified 96.6% of the 439 labels (see Table 4), while *DaSPA_v2* accurately classified 94% of the 954 labels. When accuracies are compared, the performance of *DaSPA_v2* is lower than that of *DaSPA_v1*, but hypothetical testing demonstrates that *DaSPA_v2* has improved

TABLE 5. Results of statistical hypothesis testing.

Dataset	Avg	Std	Levene test		T-test		Wilcoxon Rank-Sum test	
			DaSPA_v1[a]	*DaSPA_v2*	*DaSPA_v1*	*DaSPA_v2*	*DaSPA_v1*	*DaSPA_v2*
DaSPA_v0	0.021	0.144	57.999[b] <0.001[c]	52.623 <0.001	-75.062 <0.001	-76.191 <0.001	-21.585 <0.001	-21.621 <0.001
DaSPA_v1	0.815	0.296	-	15.250 <0.001	-	-3.901 <0.001	-	-5.039 <0.001
DaSPA_v2	0.824	0.294	-	-	-	-	-	-

[a] Represents Levene's test results of *DaSPA_v0* and *DaSPA_v1*. [b] Represents Levene's test statistics. [c] Levene's test p-value.

performance compared to the baseline. The study (i) showed that the proposed method is suitable for the patent applicants' disambiguation and standardization tasks, (ii) contributed to improving the model with data augmentation, and (iii) provided empirical evidence that additional training with lists from Fortune and KOSPI increases the degree of agreement with the ground truth.

V. CONCLUSION

A. DISCUSSION AND IMPLICATIONS

Companies are investing large amounts to lead the technology market or keep up with their competitors. Patent analysis is one of the most prevalent business strategies. Companies also actively use the patent system to secure exclusive rights to technologies, and patents contain a wealth of technology market information.

This study aimed to disambiguate and standardize patent applicant names because patent analysis can provide (i) trends in technology development over time and (ii) various information, like leading applicants of a particular technology. However, traditional patent analysis has required a lot of time and resources to obtain disambiguated and standardized patent applicant names, leaving many researchers to rely on manual labor. Fortunately, previous research has proposed alternative approaches employing (i) addresses or websites of applicants, (ii) manual procedures for disambiguating and standardizing patent applicant names, and (iii) a combination of existing methods. However, these approaches are limited in that they cannot handle cases deviating from standardized patterns. Therefore, a novel approach beyond rule-based approaches has been in demand.

This paper proposed a simple and effective *DaSPA* model that employs data augmentation and an attention mechanism. The process involved adding noise, inserting punctuation, and mixing permutations to train *DaSPA* on augmented data with various patterns of patent applicant names. The attention mechanism helped *DaSPA* focus on making patent applicants similar to the ground truth.

We performed experiments to demonstrate the suitability of the proposed method for disambiguating and standardizing patent applicant names and the effectiveness of data augmentation. As a result, data augmentation helped the proposed method learn varying patterns of patent applicant names. We suggested an effective *DaSPA* model with simple data augmentation. Therefore, we expect that the proposed method can contribute to identifying technology market-leading companies and competitors that need to be pursued.

B. LIMITATIONS AND FUTURE RESEARCH

This paper measured the empirical performance of the suggested model on 1,016 patent documents. The aim of the first experiment was to classify 439 applicants involved in 1,016 documents without data augmentation. In the second experiment, we measured the performance of the model trained on augmented datasets. In the last experiment, we measured the performance of the model trained on augmented datasets and lists from Fortune and KOSPI in addition. *DaSPA_v0*, used in the first experiment, performed very poorly. In contrast, *DaSPA_v1* and *DaSPA_v2*, used in the second and third experiments, had high accuracies of 0.966 and 0.940, respectively. In the statistical test with *DaSPA_v1*, *DaSPA_v2* had statistics of -3.901 and -5.039 in the T-test and Wilcoxon Rank-Sum test, respectively, with p-values less than 0.001. Therefore, we concluded that *DaSPA_v2* returns outputs most consistent with the ground truth among the three models.

However, the proposed method has the following limitations:

- The proposed method does not consider relations between patent applicants. For example, mergers and acquisitions (M&A) of corporations frequently occur in the automobile industry. Therefore, the *DaSPA* model needs to account for changes in technology ownership before and after an M&A.
- The proposed method disregards company subsidiaries. For instance, Samsung Display and Samsung Electronics are subsidiaries of the Samsung Group. When analyzing the patents of the Samsung group, Samsung Display and Samsung Electronics need to be differentiated. However, the current *DaSPA* model converts both subsidiaries to <Samsung>.

Future studies need to address the limitations of the proposed method. Additionally, advanced approaches could incorporate specific information, such as the addresses of patent applicants, to overcome difficulties caused by synonyms and homonyms of patent applicants.

APPENDIX A
DESCRIPTION OF PERFORMANCE MEASURES

Suppose a dataset is classified into L number of labels. True Positive (TP_l) and True Negative (TN_l) occur when the l-th predicted label matches the actual label, whether it is Positive or Negative. Then, False Positive (FP_l) refers to the frequency with which an object with a Negative label is misclassified as Positive. Lastly, False Negative (FN_l) occurs when an object with a Positive label is mistakenly classified as Negative. To calculate the precision, recall, and F1-score for the l-th label, we use (A1).

$$\begin{aligned} Precision_l &= \frac{TP_l}{TP_l + FP_l} \\ Recall_l &= \frac{TP_l}{TP_l + FN_l} \\ F1 - score_l &= \frac{2 \times Precision_l \times Recall_l}{Precision_l + Recall_l} \end{aligned} \tag{A1}$$

In a multi-class classification with L labels, macro precision, macro recall, and macro F1-score are calculated as in (A2).

$$\begin{aligned} Macro\ Precision &= \frac{1}{L} \times \sum\nolimits_{l=1}^{L} Precision_l \\ Macro\ Recall &= \frac{1}{L} \times \sum\nolimits_{l=1}^{L} Recall_l \\ Macro\ F1 - score &= \frac{1}{L} \times \sum\nolimits_{l=1}^{L} F1 - score_l \end{aligned} \tag{A2}$$

Macro precision, macro recall, and macro F1-score are decimal numbers between 0 and 1, where the performance of the multi-class classification improves as the three measures get closer to 1.

Unigram Precision, one of the bilingual evaluation understudies (BLEU), measures the similarity between two texts. Suppose that the text to be tested is T_t and the ground truth is T_g. Unigram Precision then calculates the similarity based on the frequency of T_t appearing in T_g. Unigram Precision calculated by (A3).

$$\frac{The\ number\ of\ T_t\ words\ which\ occur\ in\ T_g}{Total\ frequency\ of\ words\ in\ the\ T_t} \tag{A3}$$

MUP is a more advanced version of Unigram Precision. Count T_g refers to the number of times a specific word appears in T_g. The numerator of MUP is whichever is smaller between Count T_g and the frequency calculated using Unigram Precision (Count). MUP calculated by (A4).

$$\frac{\sum_{T\ t} min\ (Count,\ CountT_g)}{Total\ frequency\ of\ words\ in\ the\ T_t} \tag{A4}$$

MUP is a measure of the textual similarity of two texts, expressed as a decimal between 0 and 1, where the similarity of the two texts increases as the MUP value gets closer to 1.

APPENDIX B
DESCRIPTION OF HYPOTHESIS TESTING

Levene's test is used to verify if the variances of two or more groups are equivalent. Its hypothesis is presented in (B1).

$$\begin{aligned} &H_0 : notH_1 \\ &H_1 : Variances\ are\ not\ equal \end{aligned} \tag{B1}$$

The T-test is used to confirm the difference in means between two groups. Its method depends on whether the variances of the two groups are equal. Thus, (B2) represents the hypothesis of the T-test used in this study.

$$\begin{aligned} &H_0 : \mu_1 \geq \mu_2 \\ &H_1 : \mu_1 < \mu_2 \end{aligned} \tag{B2}$$

Table 5 shows that the T-test statistic for *DaSPA_v0* and *DaSPA_v1* is -75.062. μ_1 and μ_2 represent the average MUPs of *DaSPA_v0* and *DaSPA_v1*, respectively. A negative T-test statistic indicates that the average MUP of *DaSPA_v1* is greater. Unlike the T-test, the Wilcoxon Rank-Sum test uses the median to compare the central tendencies of two groups. Therefore, if the statistical distribution assumptions are not met, we can examine the nonparametric test results.

REFERENCES

[1] J. Ruscio, F. Seaman, C. D'Oriano, E. Stremlo, and K. Mahalchik, "Measuring scholarly impact using modern citation-based indices," *Meas., Interdiscipl. Res. Perspective*, vol. 10, no. 3, pp. 123–146, Jul. 2012, doi: 10.1080/15366367.2012.711147.

[2] G. Thoma, K. Motohashi, and J. Suzuki, "Consolidating firm portfolios of patents across different offices. A comparison of sectoral distribution of patenting activities in Europe and Japan," *IAM Discuss. Paper Ser.*, vol. 19, p. 11, Jan. 2010.

[3] J. Raffo and S. Lhuillery, "How to play the 'names game': Patent retrieval comparing different heuristics," *Res. Policy*, vol. 38, no. 10, pp. 1617–1627, Dec. 2009, doi: 10.1016/j.respol.2009.08.001.

[4] G. Zhang, J. Guan, and X. Liu, "The impact of small world on patent productivity in China," *Scientometrics*, vol. 98, no. 2, pp. 945–960, Feb. 2014, doi: 10.1007/s11192-013-1142-1.

[5] M. Pezzoni, F. Lissoni, and G. Tarasconi, "How to kill inventors: Testing the Massacrator? Algorithm for inventor disambiguation," *Scientometrics*, vol. 101, pp. 477–504, Jul. 2014, doi: 10.1007/s11192-014-1375-7.

[6] G. Morrison, M. Riccaboni, and F. Pammolli, "Disambiguation of patent inventors and assignees using high-resolution geolocation data," *Sci. Data*, vol. 4, no. 1, pp. 1–21, May 2017, doi: 10.1038/sdata.2017.64.

[7] G.-C. Li, R. Lai, A. D'Amour, D. M. Doolin, Y. Sun, V. I. Torvik, A. Z. Yu, and L. Fleming, "Disambiguation and co-authorship networks of the U.S. patent inventor database (1975–2010)," *Res. Policy*, vol. 43, no. 6, pp. 941–955, Jul. 2014, doi: 10.1016/j.respol.2014.01.012.

[8] S. L. Ventura, R. Nugent, and E. R. H. Fuchs, "Seeing the non-stars: (Some) sources of bias in past disambiguation approaches and a new public tool leveraging labeled records," *Res. Policy*, vol. 44, no. 9, pp. 1672–1701, Nov. 2015, doi: 10.1016/j.respol.2014.12.010.

[9] K. Kim, M. Khabsa, and C. L. Giles, "Inventor name disambiguation for a patent database using a random forest and DBSCAN," in *Proc. IEEE/ACM Joint Conf. Digit. Libraries (JCDL)*, Jun. 2016, pp. 269–270, doi: 10.1145/2910896.2925465.

[10] D. Yin, K. Motohashi, and J. Dang, "Large-scale name disambiguation of Chinese patent inventors (1985–2016)," *Scientometrics*, vol. 122, no. 2, pp. 765–790, Feb. 2020, doi: 10.1007/s11192-019-03310-w.

[11] G. Louppe, H. T. Al-Natsheh, M. Susik, and E. J. Maguire, "Ethnicity sensitive author disambiguation using semi-supervised learning," in *Proc. 7th Int. Conf. Knowl. Eng. Semantic Web*, Prague, Czech Republic, Sep. 2016, pp. 228–272, doi: 10.1007/978-3-319-45880-9_21.

[12] H. Han, C. Yao, Y. Fu, Y. Yu, Y. Zhang, and S. Xu, ''Semantic fingerprints-based author name disambiguation in Chinese documents,'' *Scientometrics*, vol. 111, no. 3, pp. 1879–1896, Mar. 2017, doi: 10.1007/s11192-017-2338-6.

[13] K. Onishi, Y. Nishimura, N. Tsukada, I. Yamauchi, T. Shimbo, M. Kani, and K. Nakamura, ''Standardization and accuracy of Japanese patent applicant names,'' Inst. Innov. Policy Res., Tech. Rep. 2012-001, Sep. 2012, doi: 10.2139/ssrn.2147190.

[14] P. Neuhäusler, R. Frietsch, C. Mund, and V. Eckl, ''Identifying the technology profiles of R&D performing firms—A matching of R&D and patent data,'' *Int. J. Innov. Technol. Manage.*, vol. 14, no. 1, Feb. 2017, Art. no. 1740003, doi: 10.1142/S021987701740003X.

[15] F. Da, G. Kou, and Y. Peng, ''Deep learning based dual encoder retrieval model for citation recommendation,'' *Technol. Forecasting Social Change*, vol. 177, Apr. 2022, Art. no. 121545, doi: 10.1016/j.techfore.2022.121545.

[16] X. Gao, Z. Zhang, T. Mu, X. Zhang, C. Cui, and M. Wang, ''Self-attention driven adversarial similarity learning network,'' *Pattern Recognit.*, vol. 105, Sep. 2020, Art. no. 107331, doi: 10.1016/j.patcog.2020.107331.

[17] A. Gordo, F. Radenovic, and T. Berg, ''Attention-based query expansion learning,'' in *Proc. 16th Eur. Conf. Comput. Vis.*, Glasgow, U.K., Aug. 2020, pp. 172–188, doi: 10.1007/978-3-030-58536-5.

[18] R. Etezadi and M. Shamsfard, ''The state of the art in open domain complex question answering: A survey,'' *Appl. Intell.*, vol. 53, no. 4, pp. 4124–4144, Jun. 2023, doi: 10.1007/s10489-022-03732-9.

[19] J. Kim, S. Chung, S. Moon, and S. Chi, ''Feasibility study of a BERT-based question answering chatbot for information retrieval from construction specifications,'' in *Proc. IEEE Int. Conf. Ind. Eng. Eng. Manage. (IEEM)*, Dec. 2022, pp. 0970–0974, doi: 10.1109/IEEM55944.2022.9989625.

[20] H. A. Pandya and B. S. Bhatt, ''Question answering survey: Directions, challenges, datasets, evaluation matrices,'' 2021, *arXiv:2112.03572*.

[21] J. J. Hopfield, ''Neural networks and physical systems with emergent collective computational abilities,'' *Proc. Nat. Acad. Sci. USA*, vol. 79, no. 8, pp. 2554–2558, 1982.

[22] S. Hochreiter and J. Schmidhuber, ''Long short-term memory,'' *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[23] R. A. Horn, ''The Hadamard product,'' in *Proc. Symposia Appl. Math.*, 1990, pp. 87–169, doi: 10.1090/PSAPM/040/1059485.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, and I. Polosukhin, ''Attention is all you need,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[25] A. Belz, A. Graddy-Reed, F. Shweta, A. Giga, and S. M. Murali, ''Deterministic bibliometric disambiguation challenges in company names,'' in *Proc. IEEE 17th Int. Conf. Semantic Comput. (ICSC)*, Feb. 2023, pp. 239–243, doi: 10.1109/ICSC56153.2023.00047.

JUHYUN LEE received the Ph.D. degree in industrial and management engineering from Korea University, Republic of Korea. He is currently a Research Professor with the Institute of Engineering Research, Korea University. His research interests include developing machine learning algorithms for unstructured data, such as text, signal, and image, and applying them to solve problems, such as predictive modeling, document classification, and sentiment analysis.

SANGSUNG PARK received the Ph.D. degree in industrial engineering from Korea University. He is currently an Assistant Professor with the Department of Data Science, Cheongju University, Republic of Korea. His research interests include patent big data analysis, data mining and machine learning, technology management, and evaluation that combines various industrial engineering theories.

JUNSEOK LEE received the Ph.D. degree in industrial and management engineering from Korea University. He is currently an Assistant Professor with the College of AI Convergence Engineering, Kangnam University, Republic of Kora. His research interests include developing a machine learning algorithm for detecting abnormal manufacturing, such as fault classification, and applying the machine learning algorithm for technology management.