IPC Prediction of Patent Documents Using Neural Network with Attention for Hierarchical Structure

Yuki Hoshino (hoshino.y.ad@m.titech.ac.jp)
Tokyo Institute of Technology
Yoshimasa Utsumi
Rakuten (Japan)
Yoshiro Matsuda
Rakuten (Japan)
Yoshitoshi Tanaka
Tokyo Institute of Technology
Kazuhide Nakata
Tokyo Institute of Technology

Research Article

Keywords: IPC Prediction, Patent Documents , Neural Network, Hierarchical Structure

Posted Date: January 3rd, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1164669/v1

IPC Prediction of Patent Documents Using Neural Network with Attention for Hierarchical Structure

Yuki Hoshino[1*], **Yoshimasa Utsumi**[2], **Yoshiro Matsuda**[3], **Yoshitoshi Tanaka**[4], **and Kazuhide Nakata**[5]

[1,4,5]Tokyo Institute of Technology
[2,3]Rakuten Group, Inc.
[*]hoshino.y.ad@m.titech.ac.jp

ABSTRACT

International patent classifications (IPCs) are assigned to patent documents; however, since the procedure for assigning classifications is manually done by the patent examiner, it takes a lot of time and effort to select some IPCs from about 70,000 IPCs. Hence, some research has been conducted on patent classification with machine learning. However, patent documents are very voluminous, and learning with all the claims (the part describing the content of the patent) as input would run out of the necessary memory. Therefore, most of the existing methods learn by excluding some information, such as using only the first claim as input. In this study, we propose a model that considers the contents of all claims by extracting important information for input. We also propose a new decoder that considers the hierarchical structure of the IPC. Finally, we evaluate the model using an evaluation index that assumes the actual use of IPC selection for patent documents.

1 Introduction

Currently, nearly 3 million patent applications are filed worldwide every year, and all of them are examined manually by patent examiners. This patent examination process takes a very long time and varies from country to country; in many regions, it takes more than six months and, in some countries, it takes more than a year. However, during this examination period, the patent is not registered as a patent and is not granted; thus, improving the efficiency of the examination is an important issue.

Each patent is assigned several international patent classification (IPC) according to its field. Furthermore, each region also provides its own classification expressions, such as cooperative patent classification (CPC) and file index (FI). Although the specific structure of IPC will be introduced in the next chapter, IPC is often used to search for similar patents and existing technologies because it provides a very detailed classification. Currently, patent examiners manually assign IPCs, so automating or semi-automating IPC prediction to assist the examiners will lead to more efficient examination. In addition, because the predicted IPC indicates the field of the patent to be examined, it is possible to assign the examination to a patent examiner who is familiar with the field at the beginning of the examination. Furthermore, because it is useful for searching for similar patents necessary for examination, it may lead to the efficiency of the patent examination itself.

In addition, the classification assigned to patents has a role as a searching key when utilizing patent information. Recently, it has been proposed to utilize patent information so that the analysis result of patent information gives useful guidelines for corporate management, and patent information is effectively utilized for business growth. And, the accuracy of assigned classification affects the value of patent information utilization, and the development of a system that supports patent classification plays an important role in promoting the effective utilization of patent information. For these reasons, assigning IPC automatically or semi-automatically and properly is very valuable.

Here, assigning IPC to patent documents has several features compared with the general natural language classification problem. First, patent documents have a unique structure, and the part that influences the classification is long. The second is that IPCs, which represent fields, have a hierarchical structure. Third, multiple IPCs are assigned to a single patent. Based on these features, in this study, we first compress the information by extracting nouns to handle the long sentence length, and then proposes an encoder to handle it. We also propose a decoder that can consider a hierarchical structure unique to the IPC. By connecting them together, multiple IPC predictions are made. Finally, the effectiveness of the proposed methods was verified by experiments using real data.

2 Feature of patent

In this chapter, the characteristics of patent documents and IPC are introduced.

IPC is generally described as consisting of four hierarchies. The hierarchy is referred to as sections, classes, subclasses, and groups, in order from the top. For example, in "H01F 1/01" H is a section called ELECTRICITY. Next, 01 is the class, and F is the subclass. Finally, "1/01" is a group, but there is actually a hierarchical structure here as well, with the main group before the "/" and the subgroups after. When subgroups are considered, the classification is broken down into the detailed category of "Magnets or magnetic bodies characterised by the magnetic materials therefore of inorganic materials" Based on the above, the purpose of this study is to predict a five-level hierarchy consisting of three levels from sections to subclasses, plus two levels of main groups and subgroups. In this paper, all these classifications from sections to subgroups are collectively expressed as labels. Although there are only eight types of sections, the number of such labels increases as one moves down the hierarchy, and because there are approximately 70,000 subclasses, prediction is more difficult toward the end of the hierarchy.

It is also important to note that multiple labels can be assigned. In general, a patentable technology consists of a combination of technologies from multiple fields, and in such cases, IPCs should be assigned to all fields of the original technology. In 2013, the average number of IPCs granted per patent was approximately 4.16 labels, and some patents were granted more than 100 IPCs.

In a patent document, the contents of the invention are described from various angles and expressions, such as "title of the invention," "abstract," "claims," "detailed description of the invention," and "examples. In particular, the "abstract" and "claims" refer to the technical contents of the patent. From here, we will introduce these in turn.

First, the "abstract" section describes the general contents of the patent. However, the abstract section is not only unstructured, but also does not affect the validity of the patent and may not essentially indicate its contents. In addition, the abstract section may not contain sufficient information to classify IPC. For example, the abstract section of a certain patent is as follows.

> Formulations of anti-VLA-1 antibodies are described.

The IPCs assigned to this patent are C07K 16/28, A61K 39/395, A61K 9/00, A61K 47/18, and A61K 47/26. However, C07K 16/28 represents the contents of "against receptors, cell surface antigens or cell surface determinants," from which we can understand the antibodies, but not the specific contents. Therefore, although the abstract provides a general idea of the content of the patent, it is not sufficient for predicting IPC.

Next, I will explain the claims. The claims are the part that shows the essential contents of the patent and is characterized by the fact that it consists of multiple claims as follows.

1. A compound of Formula (I) [Chemical] or a stereochemically isomeric form thereof, wherein R is hydrogen or fluoro, or an addition salt thereof.

2. The compound according to claim 1 wherein R is fluoro and the compound is a racemic mixture, or an addition salt thereof.

3. The compound according to claim 1 wherein R is fluoro and the compound has an optical rotation &lsqb;&agr;&rsqb;&equals; &minus;14.4&deg; (c&equals;0.3, MeOH, &lgr;&equals;598 nm; 20&deg; C.), or an addition salt thereof.

4. A pharmaceutical composition comprising a therapeutically effective amount of a compound as defined in claim 1 and a pharmaceutically acceptable carrier.

5. A compound as defined in claim 1 for use in the treatment of pulmonary arterial hypertension, pulmonary fibrosis, or irritable bowel syndrome.

Claims can be divided into two main patterns. The first is a claim that is complete by itself and is called an independent claim. In general, the first claim is always an independent claim, which gives you the general idea of the patent. The second is called a dependent claim, which is given in the form of a supplement to the preceding claims, as in the second and following claims above. Here, this example is a relatively short sentence with a small number of claims, and there are some long sentences with more than 100 claims. Therefore, if we want to input all the claims, many models will run out of computer memory because of the large number of inputs. For this reason, previous studies often input the first claim ([1],[2],[3]). However, although dependent claims do not play a central role, they cannot be ignored when predicting IPC. For example, this patent has two IPCs assigned to it, C07D 211/56 and A61K 31/445, which describe the properties of the chemical substance and its properties as a drug, respectively. However, although the chemical substance is written in claim 1, its use as a medicine is not mentioned until claim 4. This means that there is not enough information to predict all IPCs using only claim 1 as the input. Therefore, it is important to effectively extract information from all claims to predict IPC.

3 recent evolution

In the previous chapter, predicting IPC of patent documents can be viewed as a hierarchical multi-label classification problem because the task is to assign multiple labels with a hierarchical structure. Therefore, in this chapter, we introduce existing methods for the hierarchical multi-label classification problem. Next, we introduce HARNN ([3]), which is used for multi-label text classification problems in general, and finally, we explain Patent BERT ([2]), which is an existing study on IPC assignment of patent documents.

3.1 Hierarchical Multi-label Text Classification

As a basic method for hierarchical multi-label classification, a method using an SVM was first proposed in[4]. In this method, the loss function of SVM is changed according to the hierarchy, and the more distant the label, the larger the loss. However, the nature of SVM is that to perform a multi-label classification problem, the binary classification problem must be solved for the number of labels. Therefore, if you want to classify many types of labels, it will take a long time. Another method using decision trees was proposed in[5]. In the same paper, three decision tree methods are proposed, and each is evaluated. However, all of these methods are only used for solving general classification problems and are not suitable for text information. In particular, the fact that text information is ordered and variable in length makes it very difficult to use these basic methods. Therefore, when classifying information at that time, a method called bag of words, which only summarizes the existing words without considering the order of the words, was used, and the accuracy was not very high.

Several studies have used neural networks, such as those on the form of the loss function and those that allow variable length output in the manner of rnn([6],[7]). Among them, the method with the highest prediction accuracy at present is called HARNN ([3]). The encoder of this method is a bidirectional LSTM, and the decoder is a model that introduces a hierarchical structure. Decoders are roughly divided into two types: global and local. The global decoder predicts the labels of all levels at once, while the local decoder predicts the child levels of each level based on the predictions of the parent level. HARNN is currently the state-of-the-art in the task of predicting labels for all hierarchies and is a model with very high prediction accuracy.

3.2 Patent labeling

The attempt to apply machine learning techniques to patent documents begins with[8]. In[8], basic methods such as SVM and k-NN were used to classify patent documents. In addition,[9] showed that other methods, including neural networks, have the potential to be adapted. In[10], classification using Word2Vec was also studied. Deep learning was utilized for the first time in[11], where the prediction accuracy of classification was confirmed using a CNN-based deep learning model.

However,[2] proposed a model based on BERT ([12]), which is widely used as a natural language processing model. BERT is a model that has been pre-trained using a large corpus for the transformer, and a fine-tuned version of BERT is now the state-of-the-art in many fields. Because BERT is a pre-trained model for the encoder only, the decoder part needs to be created separately depending on the task. Patent BERT is very simple, with only one layer of matrix calculation and sigmoid function. This model supports multi-labeling but does not consider the hierarchical structure at all. Here, there is no comparison with general models such as HARNN in this study, and the evaluation metrics are different, so it is not clear which model is better.

4 Proposed method

In this section, we explain the specific method of the proposed patent document classification model. An outline of the proposed classification method is shown in Figure 1. First, nouns and their percentages are extracted from a large amount of textual information contained in the claims of patent documents. Next, the encoder generates the features of the target patent, and finally, the decoder performs the IPC prediction according to the features created by the encoder. In the following sections, we explain noun extraction in 4.1, encoder in 4.2, and decoder in 4.3.

4.1 Noun extraction

First, as mentioned earlier, the claims of patent documents are very long, and sometimes many, so if you try to input the full text, you will run out of RAM and will not be able to do so. In fact, the average number of tokens for the full text of the patent claims was very long (approximately 1200), and some of them were over 10,000 tokens. However, if we look at the claims, we can hypothesize that the words specific to each field are mostly nouns, and in many cases, it is possible to predict IPC from nouns only. In the proposed method, nouns are extracted from all claims using morphological analysis.

Morphological analysis is the process of decomposing a sentence into words and their parts of speech. Morphological analysis has been studied in many languages, and at the same time, many morphological analysis tools have been released. Here, parts of speech vary from language to language, but in general, there are nouns, verbs, adjectives, etc., each of which plays a similar role in the language. Therefore, extracting nouns can produce similar results in any language and is expected to be effective in a variety of languages.

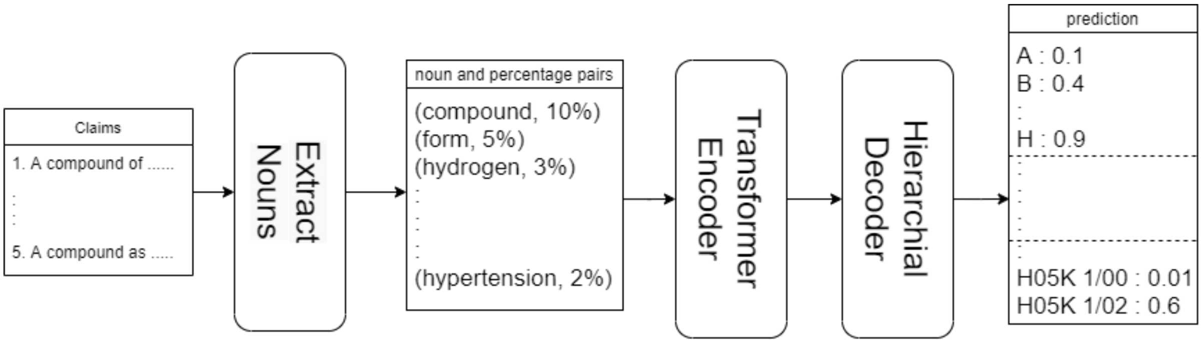

Figure 1. Overview of the proposed model

When only nouns are extracted, a sequence of nouns is obtained, but because the order of the nouns is considered to be of little significance in determining the patent content, we decided to count the number of times without considering the order information. However, the number of occurrences is considered to be significantly affected by sentence length. For example, the meaning of ten nouns out of a thousand nouns would be different from that of ten nouns out of twenty nouns. Therefore, we calculated the percentage of occurrences by dividing by the total number of noun occurrences in the actual input. As a result, we were able to keep the average number of tokens to approximately 140, and because the memory requirement during training increases in the order of the square of the number of tokens in the model of the proposed method, we were able to reduce the memory requirement to approximately 1.4%.

4.2 Encoder

In this section, we explain the central computational structures of our model, attention, and transformer, in order. Finally, we describe the specific encoder model that extracts the information of nouns and their proportions using the proposed method.

4.2.1 Attention

The attention mechanism ([13]) is a general term for a method that calculates the importance of variable-length input by inner products or compresses input by importance, as shown in the following equation.

$$\boldsymbol{\alpha}(q,K) = \mathrm{Softmax}(qK^T) \quad (1)$$
$$\mathrm{Attention}(q,K,V) = \boldsymbol{\alpha(q,K)}V \quad (2)$$

Here, q,K,V are called query, key, and value, respectively, and key and value often contain the same value. In Equation 1, we first take the inner product of q and each column vector of K. The closer the directions of the two vectors are, the larger the value of the inner product, so the value of the inner product can be regarded as a measure of the closeness or relevance of the elements. Next, by applying the softmax function to the elements, we put them in the $(0,1)$ interval and normalize the sum to 1 so that it can be regarded as a proportion. Therefore, $\boldsymbol{\alpha}(q,K)$ can be regarded as a decomposition of q and K according to their degree of association. Next, in the second equation, we take the product of $\boldsymbol{\alpha}(q,K)$ and V. Because α can be regarded as a proportion, we can regard it as taking a weighted average of V. Therefore, the above attention can be interpreted as a method of compressing values considering the relevance from the viewpoint of the query.

Multi-head attention ([14]), a method that applies the attention mechanism and is widely used in the field of deep learning, especially in natural language processing, is shown in Figure 2. Here, scaled dot-product attention (SDPA) is represented as in Equation 3, and compared to the attention mechanism described above, there are two changes. First, the query q is matrixed, compressing the information from multiple perspectives and producing each result at once. Second, scaling by dividing by $\sqrt{d_k}$ encourages the output of a size independent of the number of viewpoints.

$$\mathrm{SDPA}(Q,K,V) = \mathrm{softmax}(\frac{\mathrm{QK^T}}{\sqrt{\mathrm{d_k}}})\mathrm{V} \quad (3)$$

Multi-head attention (MHA), as shown in Fig. 2, measures the relevance of various perspectives by performing Scaled Dot-Product Attention multiple times on the input in small dimensions, and compresses the information using different compression rates, thus The range of expression is expanding. In particular, self-dot attention, which uses the same value for all Q,K,V, is very powerful and is used not only in natural language processing but also in image recognition because it can be calculated for inputs of arbitrary length and can measure the relevance of inputs to each other. However, it requires considerable computation time and memory because it has to calculate the square of the length of the input.

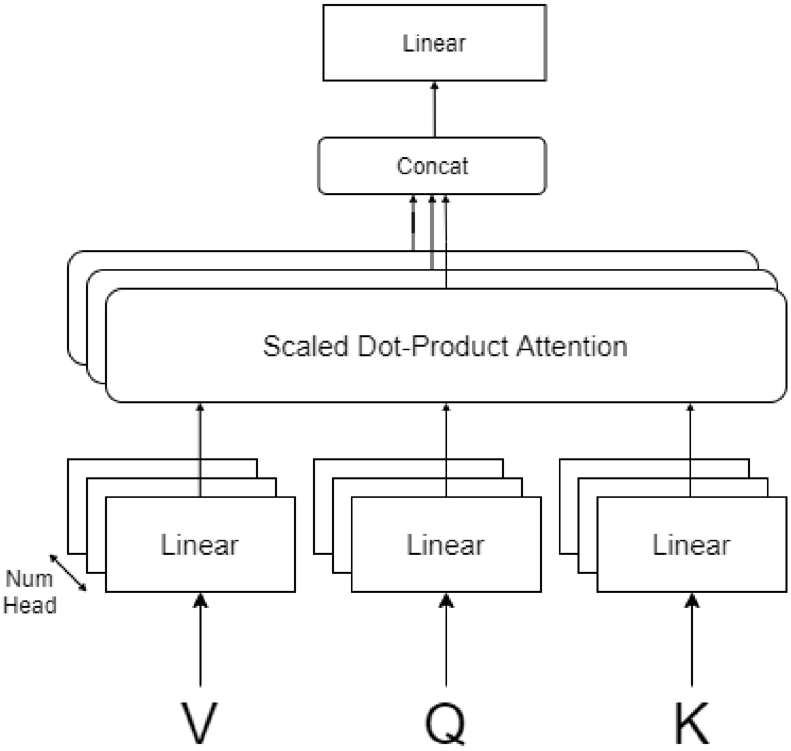

Figure 2. Structure of Multi Head Attention

The transformer is an iterative method that uses self-dot attention and simple dense layers introduced in the previous chapter. The unit structure is expressed as in equation (4), where the normalization is expressed as Norm.

$$\mathrm{Trans}(Q,K,V) = \quad \mathrm{Norm}(Q+\mathrm{relu}(W\mathrm{Norm}(\mathrm{MHA}(Q,K,V)))) \tag{4}$$

When used in natural language processing, the input words are converted into word vectors in the embedding layer and then converted into a sequence of vectors. In addition, relative (or absolute) positional information was added to each word using positional encoding. Then, by repeatedly using the unit structure of the transformer, it is possible to learn all inputs, including their relationships.

However, set transformer[15] is the transformer minus positional encoding. into only does it mention self-dot attention, but it is also a method of compressing variable length input to a fixed length by using a fixed length (often 1, in particular) for the query. This model was shown to be effective for several tasks in the original study.

4.2.2 Proposed encoder

Because the input of the model is a set of nouns and their percentages, the encoder is based on the set transformer. However, because the percentage of nouns is considered as input here, it is necessary to embed the percentage information. Here, word embedding can be regarded as a matrix calculation of the one-hot encoding vector and the matrix of the embedding vector, as shown in the upper part of Figure 3. If we extend this to consider the percentage information, we can convert the original 1 of one-hot encoding into percentage, as shown in the bottom row. In this case, the output of the matrix calculation is the embedding vector multiplied by the percentage. Therefore, the calculation of the percentage encoding layer is $r_i\boldsymbol{e_i}$, where $\boldsymbol{e_i}$ is the embedding vector of the ith element, and r_i is the percentage.

Based on the above, the overall picture of the encoder is as follows. The forward propagation calculation of the encoder including percentage encoding and transformer is as shown in Equations (5)–(7). The figure can be represented as Figure 4, where the meaning of the word and its percentage can be considered by first converting the word to an embedding vector and then multiplying by the percentage. Next, the words are input to the transformer encoder, which includes an attention mechanism so that it can learn the interaction between the words. For example, to assign IPC, which means manufacturing of hats, it is considered meaningful that the two words "hat" and "manufacturing" are together. In such a case, it is necessary to learn the relationship between the two words, and the attention mechanism of the transformer plays an important role in learning the relationship.

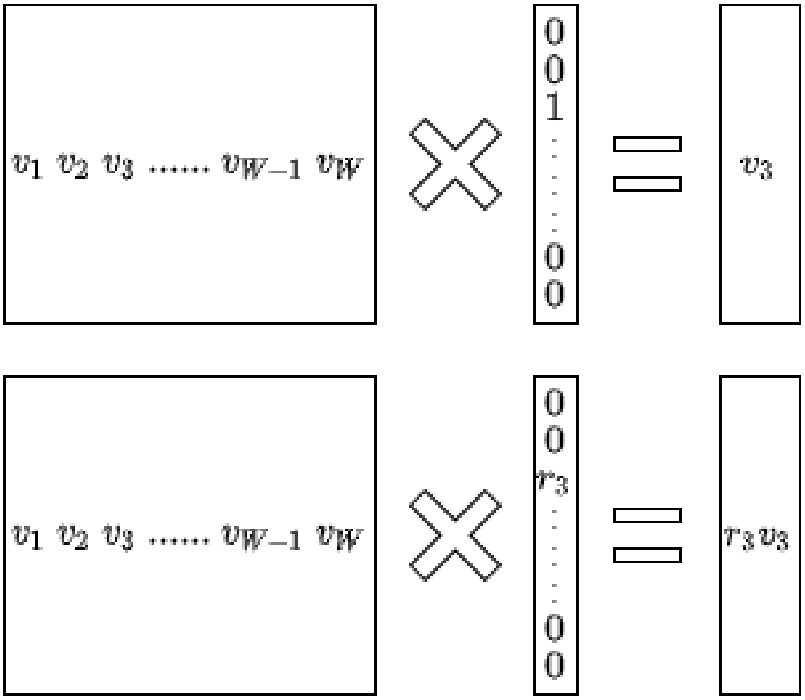
v_1 v_2 v_3 ...... v_{W-1} v_W
0
0
1
0
0
v_3
v_1 v_2 v_3 ...... v_{W-1} v_W
0
0
r_3
0
0
$r_3 v_3$

Figure 3. Interpretation of ratio encoding

$$\boldsymbol{e_i} = \text{Embedd}(x_i) \tag{5}$$

$$\boldsymbol{h_i^0} = r_i\boldsymbol{e_i} \tag{6}$$

$$H_{i+1} = \text{Trans}(H_i, H_i, H_i) \tag{7}$$

4.3 Decoder

As mentioned earlier, IPC has a hierarchy, and by utilizing this information, it is expected to improve accuracy. The following decoder is proposed to utilize this information.

First, it is assumed that the IPC is very different for each field. For example, in the field of chemistry, the names of chemicals may be considered, and in the field of mechanics, it may be important to know what the purpose is. Therefore, it is expected that IPC prediction should be processed in a way that is specific to each field, and a decoder with a hierarchical structure is proposed according to the IPC to be targeted.

The outline of the proposed decoder is shown in Figure 5. The UNIT part in Figure 5 is the unit processing of the decoder, and each unit predicts the labeling of each child node. First, the shallowest layer performs the same processing and predicts the section. In the next level, there are as many units as the number of sections, and the class within each section is predicted. Hence, each node performs a unique process within each hierarchy, and the computation can be adapted to each field.

Next, I will explain Attention. Attention was originally thought to be the most expressive method to be implemented in each unit. However, the advantage of attention is that it can be computed using all the outputs of the encoder without compression, but this requires memory and computation time. However, the number of units is as many as the number of nodes of IPCs to be predicted, and even if the number of IPCs to be predicted is limited to approximately one thousand, it must be performed several thousand times. Therefore, the attention mechanism is considered to be global because the GPU memory would be insufficient if it is implemented in all units. As mentioned earlier, attention is a compression method that calculates the importance according to the query, so the query is important. In this case, the query is a linear transformation of the output of the previous level by a trainable matrix and vector. As shown in Fig. 6, the column vectors of the matrices in the linear transformation can be regarded as vectors corresponding to each prediction target in the previous hierarchy and are considered to be weighted sums of the outputs of the parent nodes. In other words, the larger the prediction result of the parent node, the more it reflects the vector. Here, because the output of the parent node is obtained by the sigmoid function, the query tends to have a large value when it is predicted that many labels will be given in the previous level. This is in line with the intuition

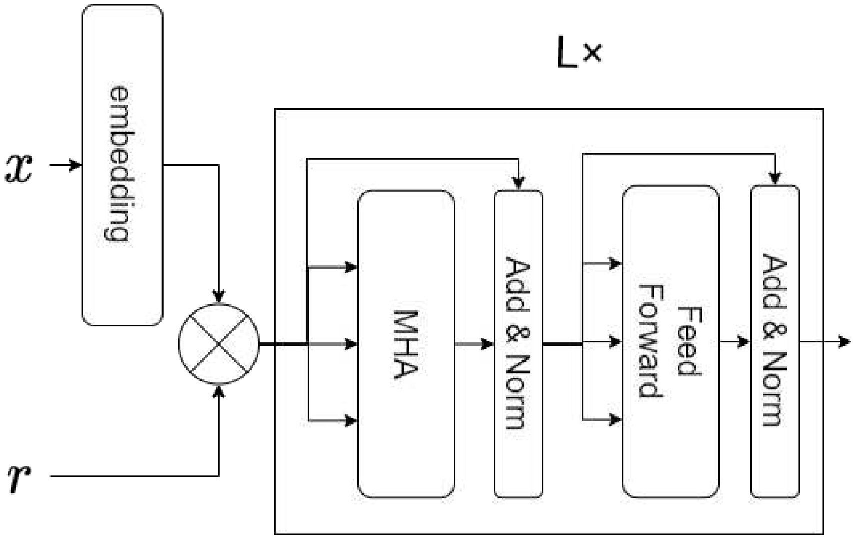

Figure 4. structure of encoder

that the more labels a node has, the more information it has. This makes it possible to take global attention into account the predicted values of the previous level.

From the above, it is necessary for the decoder unit to be a model that receives input from the parent node and the context vector by attention, and outputs two outputs: one to the child nodes and the other to the entire model (predicted value). However, as mentioned earlier, the number of nodes will be very large, so we cannot introduce a very complicated model. The unit structure is shown in Figure 7.

In addition, the dimensions of the output to the intermediate layer and to the child nodes should be varied according to the complexity of the work to be done in that model. For example, if only two IPCs are predicted by the descendant nodes of a node, the dimension of the middle layer is considered to be small. However, if there are more than hundred target IPCs, they should be embedded in a larger dimension to make them more expressive. Therefore, the dimensions of the middle layer are set according to the number of IPCs targeted by the descendant nodes. However, the output of the middle layer of the first node is given as a hyperparameter, and it is adjusted to be one-dimensional when there is only one target IPC. In the preliminary experiments, we compared two ways of changing the dimensions of the middle layer—linear and logarithmic—and adopted the logarithmic one. As a result, the number of IPCs owned by node n is $l(n)$, the number of IPCs to be predicted by the model is L, and the feature dimension of the encoder is D.

$$\dim h_r^n = 1 + \lceil (D-1)\log_L l(n) \rceil$$

Based on the above, the forward propagation calculation of the entire decoder is expressed as follows, assuming that the output of the encoder is h_{enc} and $P(n)$ is the parent node of node n.

$$\begin{aligned}
c_1 =& \quad \mathrm{MHA}(q_1, h_{enc}, h_{enc}) && (8)\\
h_1^n =& \quad \tanh({W_1}_1^n c_1 + {b_1}_1^n) && (9)\\
y_1^n =& \quad \sigma({W_2}_1^n h_1^n + {b_2}_1^n) && (10)\\
q_r =& \quad V_{r-1} y_{r-1} + b_{r-1} \qquad (r \geq 2) && (11)\\
c_r =& \quad \mathrm{MHA}(Q_r, h_{enc}, h_{enc}) \qquad (r \geq 2) && (12)\\
h_r^n =& \quad \tanh({W_1}_r^n h_{r-1}^{P(n)} + {b_1}_r^n) \qquad (r \geq 2) && (13)\\
y_r^n =& \quad \sigma({W_2}_r^n (h_r^n \oplus c_r) + {b_2}_r^n) \qquad (r \geq 2) && (14)
\end{aligned}$$

5 experiment

To evaluate the effectiveness of the proposed method and its practicality, we experimented on the task of predicting IPC using real data against which existing methods were compared.

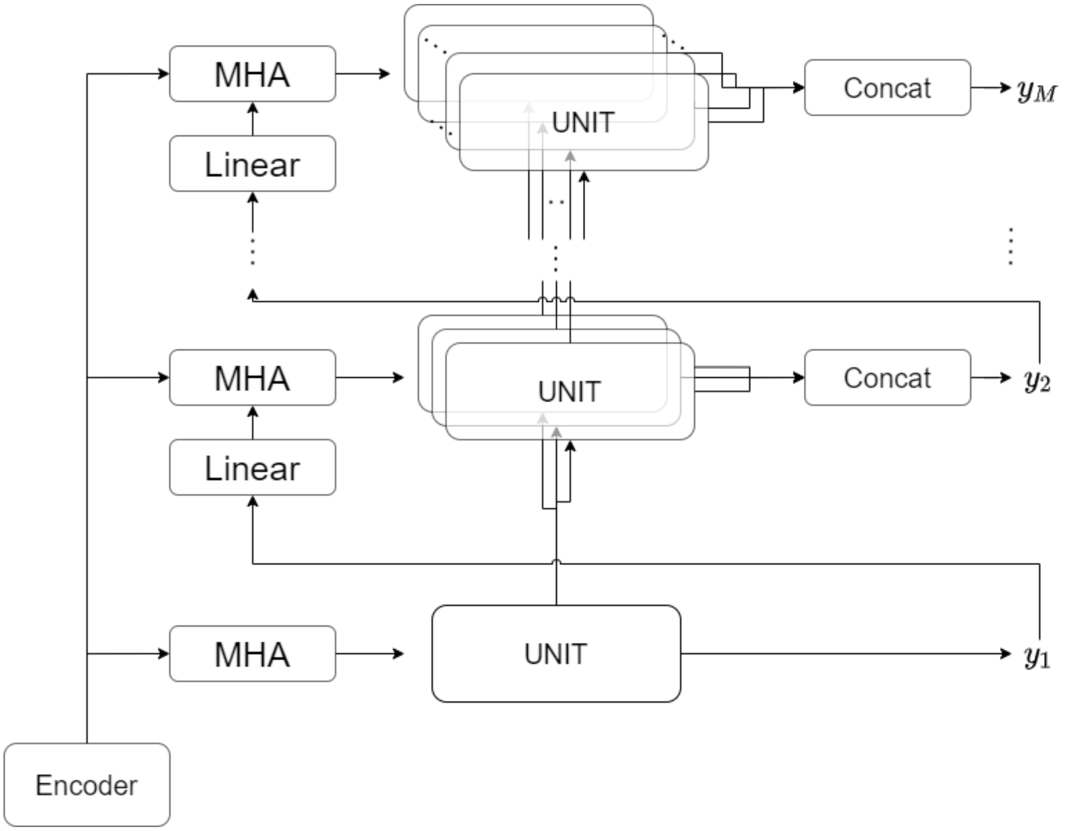

Figure 5. Structure of decoder

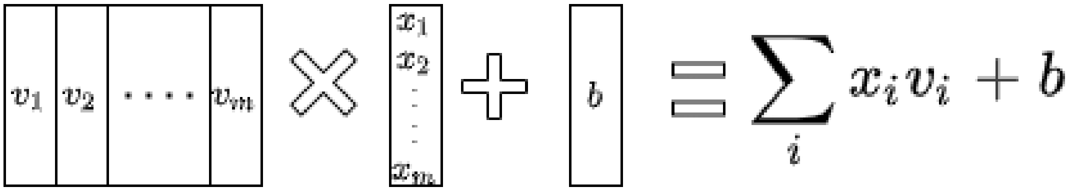

Figure 6. image to make a query

5.1 experimental setup

Real data registered as U.S. patents were used for the numerical experiments. For the training data, it was assumed that the application was filed between 2010 and 2012 and registered by 2020. For the test data, we assumed that the applications were filed during the following year (2013) and registered by 2020. As a result, the total number of training data was 733,154, of which 10% were used as validation data. In contrast, the number of test data points was 261,622.

First, as a preprocessing of the data, the Python nltk module[16] was used for noun extraction, and only the 512 nouns with the highest number of occurrences in each sentence were used as inputs. The IPCs to be predicted were those that appeared in more than 100 cases in the training data (4092 labels in total). Although some patent documents without any IPCs appeared in the training data, we did not exclude them because we thought they could not be excluded in practice. For the hyperparameters of the model, the encoder dimensionality was 256, the depth was 4, the dropout rate was 0.2, and the regularization weight was 10^{-5}. For the embedding layer, the trained word2vec model[17] compressed using PCA was used as the initial value, and the embedding layer was frozen for the first 10 epochs and trained using Adam. The first 10 epochs were trained by freezing the embedding layer using an Adam. After that, the freeze was removed, and all layers were trained using Adam for 5 epochs. The implementation was done using the Tensorflow module in Python.

As a baseline, we compared with HARNN, which has the highest prediction accuracy for general hierarchical multi-label classification, and Patent BERT, which has the highest prediction accuracy for patent document classification. The model size of the HARNN was set to 256 dimensions along with the proposed model, and as in the proposed model, the first claim was

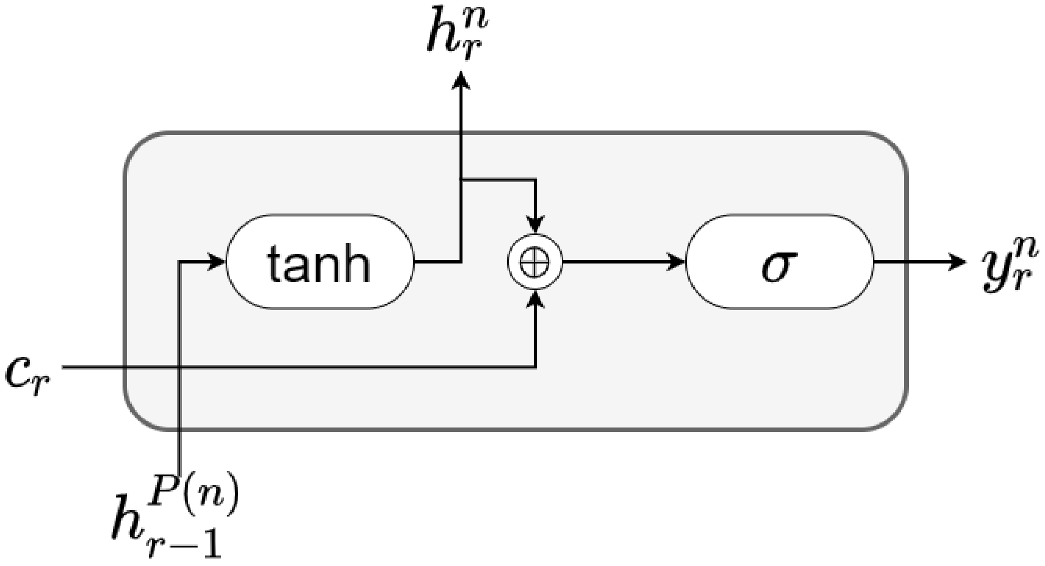

Figure 7. Unit of the decoder

used as the input. The word2vec model was used as the initial value for the embedding layer. The model size of the HARNN is 256 dimensions along with the proposed model, and the word2vec model is used as the initial value of the embedding layer, as in the proposed model. A 12-layer, 768-hidden, 12-heads model[18] was used as the BERT of Patent Bert.

5.2 evaluation metrics

In previous studies, various evaluation metrics have been used owing to their different objectives. In[3], the evaluation metrics were precision, recall, and F-measure for all the prediction labels at all levels, because it is necessary to hit all the prediction labels equally. However, Patent BERT uses the prediction accuracy of the single label that it is most confident in predicting, which is not considered practical. Therefore, in this study, we provided a practical evaluation metric.

In predicting the IPC of a patent document, only the terminal labels are important. For example, even if the main group is correct but the subgroup is wrong, it is considered to be an erroneous assignment and meaningless. Therefore, we first consider only whether the terminal subgroups are correct. Here, the question is how to evaluate the prediction accuracy of the terminal labels. The multi-label classification problem is considered as a multiple binary classification problem. Then, we evaluated the model using the evaluation index of binary classification problems.

In binary classification problems, there are generally four patterns: the number predicted to be 1 and is 1 (True Positive; TP), the number predicted to be 1 and is 0 (False Positive; FP), the number predicted to be 0 and is 1 (False Negative; FN), and the number predicted to be 0 and is 0 (True Negative; TN).

		actual	
		1	0
predicted	1	TP	FP
	0	FN	TN

Based on these, the precision, recall, and F-measure were obtained as follows, respectively.

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{TP}{TP+FN}$$

$$\text{F} - \text{Measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

These evaluation indices are affected by the threshold for the probability of granting a prediction. Therefore, we used the AUC as a method to evaluate only the model.

In addition, although the evaluation indices have been used to achieve full automation, we assume a situation in which patent examiners refer to them when granting IPCs and view the system as one that recommends those judged to have a high probability of being granted. For this purpose, recall Recall@N was introduced. This is to obtain recall when the top N of the probability of granting is predicted to be granted. This assumes that the patent examiner is recommended N IPCs that are expected to be granted, and the examiner selects from among them to grant. With this evaluation index, it is possible to evaluate how many correct IPCs are included in the N recommendations.

5.3 results

First, we conducted a simple experiment to determine whether it was appropriate to extract nouns. Specifically, because it would be time-consuming to use all labels for prediction, the prediction accuracy was compared for each part of speech by limiting the target labels to only sections. We compared the prediction accuracies for each part of the speech. For the encoder, we used the proposed encoder, and for the decoder, we used a simple encoder that consists only of a single matrix calculation and a sigmoid function. The results are shown in Table 1.

Table 1. 8 label classification with each part of speech

part of speech	Precision	Recall	F-measure	AUC
Noun	**0.805**	**0.805**	**0.805**	**0.972**
Verb	0.685	0.691	0.688	0.935
Adjective	0.726	0.727	0.726	0.949

The results show that the model of nouns predict labels best. Here, the accuracy of adjectives is higher than that of verbs, but this is probably due to the fact that there are approximately 400,000 adjectives in the training data, while there are approximately 50,000 verbs. This result indicates that nouns contain more information for technical content classification than other parts of speech.

In the next experiment, we tested the prediction accuracy of the proposed model for each level of the hierarchy. The results are presented in Table 2. It can be seen that the prediction accuracy of the section is high, and the prediction accuracy decreases as we move down the hierarchy. In particular, the prediction accuracy of the main group and the subgroups decreased, but the reason may be that the decoding features were too small. It is important to identify the cause of this problem and to handle it to perform IPC prediction with high accuracy.

Table 2. Evaluation by level

rank	Number of labels	Precision	Recall	F-measure
section	8	0.795	0.794	0.795
class	94	0.671	0.676	0.673
subclass	305	0.565	0.562	0.563
maingroup	1263	0.390	0.392	0.391
subgroup	4087	0.243	0.263	0.252

Next, to compare the prediction accuracy for each field, we calculated the prediction accuracy for each section. The results are shown in Table 3. This result shows that the accuracies of A and G are high, while the accuracies of B, D, and E are low.

Table 3. Evaluation by section

section	Number of subgroups	Average number of labeled patents	Precision	Recall	F-measure
A(Human Necessities)	649	324.8	0.263	0.319	0.288
B(Performing Operations, Transporting)	534	231.9	0.236	0.151	0.184
C(Chemistry, Metallurgy)	482	248.7	0.260	0.220	0.238
D(Textiles, Paper)	8	122.3	0.218	0.148	0.177
E(Fixed Constructions)	75	178.0	0.170	0.138	0.152
F(Mechanical Engineering, Lighting, Heating, Weapons)	282	212.7	0.195	0.226	0.210
G(Physics)	861	521.3	0.281	0.277	0.279
H(Electricity)	1196	408.0	0.232	0.240	0.236

This result is generally higher for those with a larger number of data points per label. Therefore, there are many labels that have not reached a sufficient amount of data, and it is expected that more accurate predictions can be made if more data are available.

Next, we trained a model to predict all labels and compared it with existing methods. The results of each evaluation index are listed in Table 4. The results show that the proposed method significantly improves all evaluation metrics compared to the existing methods. Therefore, it can be said that the proposed method is more effective than the existing method patent BERT. In addition, when we compare patent BERT with HARNN, the AUC is not very different, but there is a large difference in F-measure, indicating that the effect of pretraining is significant. While the accuracy of the proposed model is relatively higher than that of the two existing methods, the F-measure is approximately 0.25 in absolute terms. This means that the proposed model is not yet at a practical level when it is used as a tool to fully automate the IPC assignment.

Table 4. Comparison with existing methods

model	Precision	Recall	F-measure	AUC
ours	**0.243**	**0.263**	**0.252**	**0.925**
HARNN	0.128	0.126	0.124	0.812
patent BERT	0.194	0.194	0.195	0.806

Finally, to analyze the usefulness of the proposed model when used as a recommendation tool for patent examiners, Recall@N for the proposed method is shown in Figure 8. This shows that if we look at 30 labels, we can cover 70% of the labels, and if we look at 300 labels, we can cover 95% of the labels. Although it is impractical to look at 300 labels, it may be worthwhile to use this method to look at the top 30 labels to prevent IPC from being overlooked.

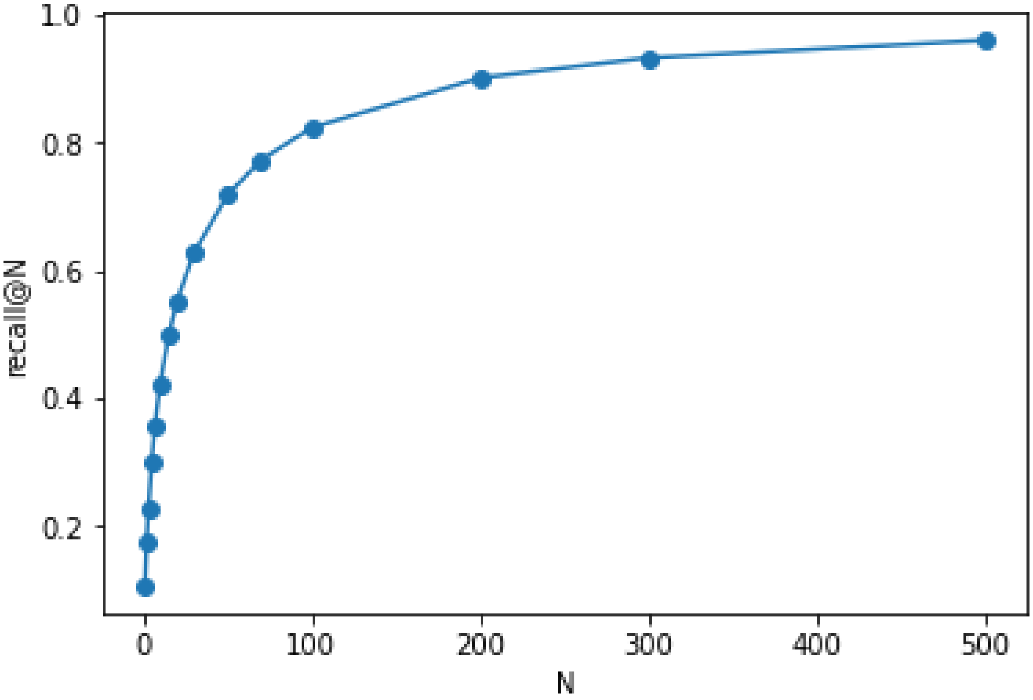

Figure 8. result of recall@N

6 conclusion and future work

In this paper, we propose a method for IPC assignment of patent documents. The first is to extract nouns and their percentages from all claims, and the second is a decoder to process locally specific information while taking global attention to handle the hierarchical structure of IPC. The second is a decoder that processes locally specific information while taking global attention to support the IPC hierarchy. We also used Recall@N to test the model when it was used to recommend labels that were likely to be assigned.

There are three issues that need to be addressed in the future. The first is to improve the feature extraction method. To reduce the amount of memory used in the model, we extracted nouns and their percentages as input, but this is a kind of compromise. It is true that nouns are the parts of speech that affect prediction more than other parts of speech, but information compression methods other than noun extraction should also be considered. For example, when extracting nouns, it is possible that by extracting which other nouns they were used with, it will be possible to infer what technologies were combined to create the new technology. There is also the possibility of further improving the accuracy of the feature extraction by utilizing other information such as the dependency relations of sentences and claims. Therefore, various feature extraction methods should be compared.

The next step is to examine what caused the various differences in prediction accuracy obtained in the experiment. In this study, we have found that the difference in prediction accuracy per hierarchy in the experiment is different between subclasses and above and main groups and below, but we do not know the cause of this difference. In addition, there is a large difference in prediction accuracy between sections, which cannot be explained by the number of labeled data alone. Therefore, when we are able to analyze the factors that determine these accuracies, it will be possible to use them as a reference in considering the feature extraction method mentioned earlier, and this is a point that should be considered in the future.

The final was to validate the prediction against a larger number of target labels. In the experiments conducted in this study, the prediction was performed for 4092 labels. However, there are approximately 70,000 IPCs in the real world, and there is a possibility of memory shortage and loss of accuracy if predictions are made on this scale. Therefore, it is necessary to verify whether learning is possible when the number of labels is increased, as well as the prediction accuracy. If it is not possible to learn, it may be necessary to divide the data into separate parts and predict them in some way.

References

1. Li, S., Hu, J., Cui, Y. & Hu, J. Deeppatent: patent classification with convolutional neural networks and word embedding. *Scientometrics* **117**, 721–744 (2018).
2. Lee, J.-S. & Hsiang, J. Patent classification by fine-tuning bert language model. *World Pat. Inf.* **61**, 101965 (2020).
3. Huang, W. *et al.* Hierarchical multi-label text classification: An attention-based recurrent network approach. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 1051–1060 (2019).
4. Rousu, J., Saunders, C., Szedmak, S. & Shawe-Taylor, J. Kernel-based learning of hierarchical multilabel classification models. *J. Mach. Learn. Res.* **7**, 1601–1626 (2006).
5. Vens, C., Struyf, J., Schietgat, L., Džeroski, S. & Blockeel, H. Decision trees for hierarchical multi-label classification. *Mach. learning* **73**, 185 (2008).
6. Nam, J., Kim, J., Loza Mencía, E., Gurevych, I. & Fürnkranz, J. Large-scale multi-label text classification — revisiting neural networks. In Calders, T., Esposito, F., Hüllermeier, E. & Meo, R. (eds.) *Machine Learning and Knowledge Discovery in Databases*, 437–452 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2014).
7. Wehrmann, J., Cerri, R. & Barros, R. Hierarchical multi-label classification networks. In Dy, J. & Krause, A. (eds.) *Proceedings of the 35th International Conference on Machine Learning*, vol. 80 of *Proceedings of Machine Learning Research*, 5075–5084 (PMLR, 2018).
8. Fall, C. J., Törcsvári, A., Benzineb, K. & Karetka, G. Automated categorization in the international patent classification. *SIGIR Forum* **37**, 10–25, DOI: 10.1145/945546.945547 (2003).
9. Benzineb, K. & Guyot, J. *Automated Patent Classification*, 239–261 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2011).
10. Grawe, M. F., Martins, C. A. & Bonfante, A. G. Automated patent classification using word embedding. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 408–411, DOI: 10.1109/ICMLA.2017.0-127 (2017).
11. Li, S., Hu, J., Cui, Y. & Hu, J. Deeppatent: patent classification with convolutional neural networks and word embedding. *Scientometrics* **117**, 721–744, DOI: 10.1007/s11192-018-2905-5 (2018).
12. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186, DOI: 10.18653/v1/N19-1423 (Association for Computational Linguistics, Minneapolis, Minnesota, 2019).
13. Cheng, J., Dong, L. & Lapata, M. Long short-term memory-networks for machine reading. *CoRR* **abs/1601.06733** (2016). 1601.06733.

14. Vaswani, A. *et al.* Attention is all you need. *arXiv preprint arXiv:1706.03762* (2017).

15. Lee, J. *et al.* Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning*, 3744–3753 (PMLR, 2019).

16. Loper, E. & Bird, S. Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics* (2002).

17. Yamada, I. *et al.* Wikipedia2Vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from Wikipedia. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 23–30 (Association for Computational Linguistics, 2020).

18. Devlin, J., Chang, M., Lee, K. & Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR* **abs/1810.04805** (2018). 1810.04805.