

Article

DeNERT-KG: Named Entity and Relation Extraction Model Using DQN, Knowledge Graph, and BERT

SungMin Yang, SoYeop Yoo and OkRan Jeong *

Department of Software, Gachon University, Seongnam-si 13120, Korea; ysm0622@gc.gachon.ac.kr (S.Y.); bbusso@gc.gachon.ac.kr (S.Y.)

* Correspondence: orjeong@gachon.ac.kr; Tel.: +82-31-750-5831

Received: 15 July 2020; Accepted: 14 September 2020; Published: 15 September 2020



Abstract: Along with studies on artificial intelligence technology, research is also being carried out actively in the field of natural language processing to understand and process people's language, in other words, natural language. For computers to learn on their own, the skill of understanding natural language is very important. There are a wide variety of tasks involved in the field of natural language processing, but we would like to focus on the named entity registration and relation extraction task, which is considered to be the most important in understanding sentences. We propose DeNERT-KG, a model that can extract subject, object, and relationships, to grasp the meaning inherent in a sentence. Based on the BERT language model and Deep Q-Network, the named entity recognition (NER) model for extracting subject and object is established, and a knowledge graph is applied for relation extraction. Using the DeNERT-KG model, it is possible to extract the subject, type of subject, object, type of object, and relationship from a sentence, and verify this model through experiments.

Keywords: named entity recognition; relation extraction; DQN; BERT; knowledge graph

1. Introduction

With the development of the many technologies that make up artificial intelligence, artificial intelligence is becoming more closely embedded in our lives. Many artificial intelligence technologies are being studied and applied in a range of fields, such as image, video, voice, and text. When technologies such as deep learning and reinforcement learning are applied in various fields, they show similar or better performance than traditional methods, showing their real-world applicability.

In particular, natural language is considered to be one of the most important areas for the future direction of artificial intelligence. For computers to interact more with people and learn on their own, they must be able to process, understand, and create human language. Recently, research on language models has been actively conducted to carry out this natural language processing [1–4].

We generate huge amounts of text through our active Internet activities, and these texts contain a lot of information. Ultimately, for computers to learn on their own and understand human common sense, they must be able to read the natural language used by people and extract the information contained in it.

Figure 1 shows the motivating example of DeNERT-KG, a named entity and relation extraction model based on deep Q-networks [5], BERT [6], and knowledge graph, we proposed. The DeNERT-KG model looks for words corresponding to subject and object in numerous sentences created by people and extracts and shows the relationship between these words. When a sentence is given, it uses the named entity recognition (NER) model DeNERT to find all the NER tags that exist in the sentence. When words tagged with NER are added to existing sentences and applied to the DeNERT-KG model, the relationship between the two words is finally extracted.

Sentences		Results
Steve Jobs and Steve Wozniak co-founded Apple in 1977.	⇒	{ Steve Jobs , PER, Apple , ORG, founded }
James and Lily were married immediately after Hogwarts, sometime around 1978 and 1979.		{ James , PER, Lily , PER, spouse }
Harry Potter, James and Lily's son, was born during the height of the war on 31 July, 1980.		{ Harry Potter , PER, James , PER, children }

Figure 1. A motivating example of DeNERT-KG.

For example, if we apply the DeNERT-KG model to the sentence “Steve Jobs and Steve Wozniak co-founded Apple in 1977”, the NER tagger finds all tagging information. When we find NER tagging information in the same form as {Steve Jobs, PER} and {Apple, ORG}, we add each information to the existing sentence to extract relationship information. The tag PER refers to the person and ORG refers to the organization. Finally, we extract the results with a set of {Steve Jobs, PER, Apple, ORG, found}. The results represent each subject, the type of subject, the type of subject, the type of object, and the relationship between subject and object.

Finding meaningful information in the natural language people use is a very important task in the learning of computers. We would like to propose a model that can extract the subject, the object, and the relationship between the subject and the object from among the information nested within the various sentences. The proposed DeNERT-KG can perform both NER and relationship extraction tasks by utilizing language models, enhanced learning, and knowledge graphs. DeNERT, a model that performs the NER task, uses BERT [6] models and enhanced learning deep Q-networks (DQN) to improve accuracy. The relationship extraction task also utilizes knowledge graph embedding to improve the accuracy of relationship extraction. Finally, using the DeNERT-KG model we propose, NER relationship extraction results can be checked immediately. In other words, the important information contained in a sentence can be easily identified because it can be given in a given sentence, the object, and the relationship.

2. Related Work

Natural language processing (NLP) is an important area of learning human common sense—human language—which is one of the many important technologies that make up artificial intelligence. Every field is important, but this is a particularly important field because, for computers to understand and act like humans, they must be able to understand and process human language, or natural language [1].

The natural language processing field shows significant performance improvements in various tasks, as many studies have been actively conducted recently. The natural language processing field can be considered one of the most important areas in the development of artificial intelligence, as computers are ultimately the areas for understanding and learning human language [3,4].

There are many sub-disciplinary areas in the field of natural language processing research. There are several sub-disciplinary areas such as morpheme analysis, parts tagging, question answering (QA), relationship extraction, emotion analysis, machine translation, speech recognition, speech synthesis, summary, and chatbots. While there are several sub-fields, to grasp the connotations inherent in the text, one must be able to find words within the sentence and extract the relationship between the words. To this end, we would like to explain named entity recognition and relationship extraction in further detail [1–4].

2.1. Named Entity Recognition (NER)

Named entity recognition is the extraction of nouns within a sentence or document that have a unique meaning in the word itself, such as a person, organization, and location, and classifying them based on into which category they fall. Before using machine learning, traditional named entity recognition studies could be broadly divided into rule-based and dictionary-based studies. Most rule-based named entity recognition categorizes datasets based on rules manually defined by

people themselves. Because of the nature of the natural language, this method is mostly irregular and incomplete, and it is highly likely to work well only in certain datasets [7,8].

Dictionary-based named entity recognition categorizes datasets based on collected dictionaries or user-defined dictionaries. Dictionary-based named entity recognition is advantageous for information extraction or retrieval in certain areas where less common frequent terminology is often used. However, there is a disadvantage of having to manually organize dictionaries, and because it is necessary to deal with constantly changing and emerging new words over time, it is costly to manage dictionaries, and there are limitations to processing non-pre-defined words [1,9].

Like many other fields of research, named entity recognition is also actively conducted using machine learning. In particular, research on named entity recognition using deep learning has been active in recent years. Named entity recognition is also carried out using the Bi-LSTM-CRF model [10], which shows meaningful performance in time series data, using supervised learning-based word embedding and non-supervised learning-based word embedding from a large corpus.

A recent study on named entity recognition has made great use of methods using language models pre-learned with the large corpus. Models such as ELMo [11], Open AI GPT [12], and BERT [6] are typical language models. These models use an accession mechanism [13], in which sentences entered at the time the decoder predicts a word are used again in the encoder.

Of the reinforcement learning algorithms, DQN [5,14] is an algorithm that applies deep learning to Google's enhanced learning and is also used for AlphaGo. Ref. [15] proposes a method of performing the NER task by applying reinforcement learning to CoNLL datasets using policy-based active learning. Ref. [16] proposes a method to apply to the Chinese NER task using the bidirectional LSTM-CRF model and DQN together. This model shows high performance, particularly for the news domain dataset.

Various methods have recently been proposed to carry out the NER task, but studies applying reinforcement learning have been very rare. However, applying reinforcement learning in natural language processing has the advantages of obtaining similar or better performance models with fewer datasets and reducing both the amount of computation in learning and inference. Therefore, we utilize DQN and BERT models that improve the performance to perform the NER task model.

2.2. Relation Extraction

Enabling computers to understand human common sense is a very important skill in artificial intelligence research. For humans, language is a kind of mutual protocol in which people naturally learn both socially and culturally but computers need separate learning to understand human common sense. A socially and culturally learned person can easily see sentences and understand the relationships between the words nested in them. Artificial intelligence technology must be able to extract the relationships between words within a sentence to understand and process a person's sentence, or natural language [17,18].

Extracting relationships can be defined differently depending on the field of research. We define the relationship extraction task as extracting the relationship between two words within sentences and documents. Many studies are underway that intend to accomplish this. Mainly in traditional techniques, sentences were created by grammatical dependencies and the relationship among words was extracted based on them [19,20].

Recently, studies have emerged that apply deep learning technology to relationship extraction. PA-LSTM model [21] combines the LSTM (Long Short-Term Memory) model with a form of position-aware attention. This model is improved by applying graph convolutional networks over the pruned dependency tree [22]. Additionally, authors of [23] propose the relation extractors based on extensions of Harris' distributional hypothesis relations and BERT. It outperforms on datasets for relation extraction such as TACRED dataset, a large scale relation extraction dataset [21].

2.3. Joint Task of Extracting Entities and Relationships

In the field of natural language processing, the task of extracting entities from text, and the task of extracting relationships, are performed as different tasks. Recently, a joint task that carries out these two tasks together has also been studied. These point tasks are largely classified into a point method that is extracted at the same time as the pipeline method. The pipeline method is a method of processing tasks in order with the NER task and relation extension task. The point extraction method, on the other hand, achieves the results of the triplet by extracting the entity and relation at the same time [24].

The joint task can be performed depending on existing NLP tools, such as [25], based on RNN like [26], or based on CNN such as [27]. RNN can encode linguistic and syntactic properties in text sequences, and CNN has the advantage of better capturing semantic information in sentences. All of these models extract information about relationships with entities into triplets.

However, many existing studies focus only on the relationship between the entities within the sentence. Of course, the relationship between entities alone can be meaningful enough, but there is a limitation in that these methods cannot capture information about the types of entities in the sentence. Even if they have the same relationship, they have different meanings in substance, depending on which identity is the subject and what type of identity it is.

2.4. DQN, BERT, and Knowledge Graph

Unlike previous studies, we would like to propose a model using DQN, BERT, and knowledge graphs so that we can learn and deduce the subject, type of subject, object, type of object, and relationship at once. There are three techniques to build our model, deep Q-networks, BERT, and knowledge graph.

2.4.1. DQN

The reinforcement learning algorithm is one of the major areas of machine learning and is inspired by behavioral psychology. Within a particular simulation environment, the learning target agent recognizes the current state and learns the sequence of actions among the selectable actions that maximize the rewards. Of the reinforcement learning algorithms, DQN (deep Q-networks) [5,14] developed by Google combines neural networks with reinforcement learning and is well known for being used in the development of the reinforcement learning algorithms used in AlphaGo. DQN solved the problem of training not being converged well due to the continuous state of the simulation environment in existing Q-Learning by approximating Q value using neural networks.

2.4.2. BERT

BERT [6] is a pre-trained language model that was released by Google in 2018. BERT uses a transformer structure. It trains language expressions through unsupervised learning from huge corpus datasets such as Wikipedia and books, and then additionally trains models for specific downstream tasks. This is also called transfer learning, because the parameters are fine-tuned from the pre-trained model. In recent years, BERT has been the first model for most natural language downstream tasks. Therefore, various derivative studies using the BERT model have been actively conducted in various studies such as RoBERTa [28], DistillBERT [29], and ALBERT [30].

2.4.3. Knowledge Graph

A knowledge graph is a graph of words linking them together and can help a computer learn a person's common-sense more easily. To understand sentences more like humans, we need knowledge graph technology that links words to words. There are a number of studies on knowledge graphs such as WordNet [31], YAGO [32], Probase [33], and ConceptNet [34]. WordNet [31] is a word database for English, and classifies words as nouns, verbs, adjectives, etc., based on the meaning, to classify as synset and link the words to each other in terms of conceptual meaning and relationships. It consists of about 207,000 word-meaning pairs. YAGO [32] is a knowledge graph built on Wikipedia, WordNet,

and GeoNames data and there are more than 10 million objects connected in YAGO. Probase [33] collected data from more than 1.6 billion web pages with knowledge graphs built by Microsoft. It consists of about 2 million objects and more than 200 million pairs. ConceptNet [34] is an open-source knowledge graph that has a large amount of data for 10 key languages, including English and French, and a small amount for about 68 languages, including Korean. More than 8 million objects are linked to each other based on 40 relationships.

3. DeNERT-KG

In this paper, we propose a model that can extract relationships with NER. DQN and BERT are utilized for NER, and ALBERT, a modified version of BERT, and knowledge graph embedding are utilized for relationship extraction. The proposed DeNERT-KG model can extract the type of subject, the object, the type of object, and even the relationship within given sentence.

3.1. DeNERT-KG Model

The DeNERT-KG model extracts results from a given sentence in the form of {subj, sub_type, obj, obj_type, relation}. To extract the relationship between words from sentences, we extract the NER tag of sentence words through the NER model and utilize this result and knowledge graph to finally extract the relationship between words.

Figure 2 shows the overall architecture of the proposed DeNERT-KG. It consists of DeNERT, a NER tagging model based on DQN and BERT, and a KG-RE model that extracts relationships based on knowledge graphs to extract the subject, object, and relationship from a given sentence.

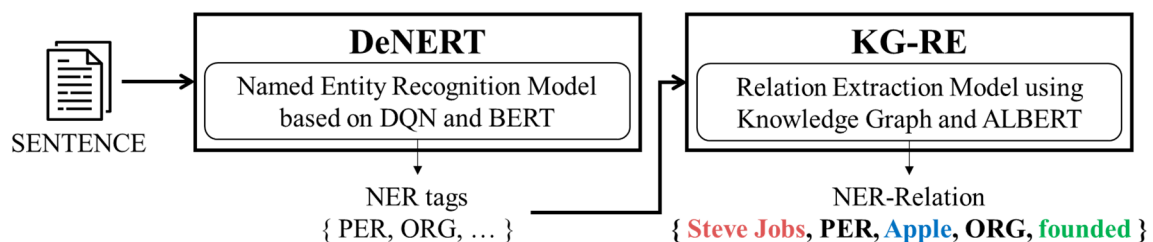


Figure 2. The overall architecture of DeNERT-KG.

3.2. DeNERT

In recent years, most of the studies in the field of named entity recognition have been focused on modeling incomplete and irregular natural language as a vector representation using a large number of corpus datasets and a lot of computing power. These models can be trained by unsupervised learning or semi-supervised learning, but most of them use supervised learning. Supervised learning has the advantage of providing an intuitive and somewhat guaranteed accuracy because it trains from datasets with labeled answers, but it also has the disadvantage of requiring a large number of training datasets and computing power.

To solve these problems, strengthening learning algorithm DQN and language expression model BERT showing high performance in natural language processing introduce how to apply to named entity recognition using two models.

The overall structural diagram of the proposed model, DeNERT, is shown in Figure 3. It is divided into the language model BERT model part, which is in charge of preprocessing input token and vectorizing the tokenized word, and the DQN agent model part, which trains the named entity classification problem by receiving these vectorized sentences. Named entity recognition is a classification problem that determines whether each word is an entity name; or, if it is, it classifies it into a named entity category.

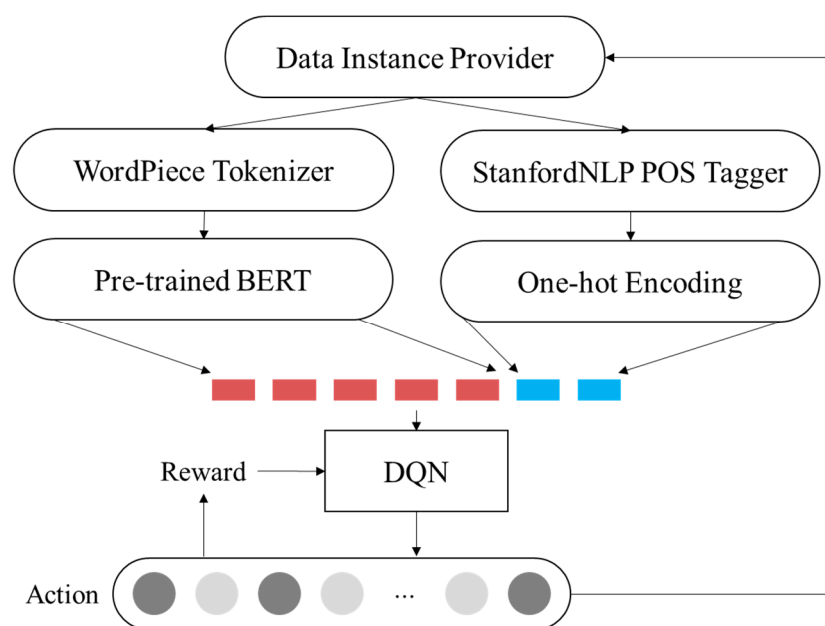


Figure 3. Architecture of DeNERT.

Therefore, to apply reinforcement learning to this classification problem, it is necessary to configure the training environment in a form that reinforcement learning agents can train. To construct a proper training environment, we need to define some important elements used in reinforcement learning.

First, if an input sentence is given, the same sentence is input into two models, a BERT model and a POS (Part-Of-Speech) tagger. In the BERT model, the input sentences are tokenized using the WordPiece Tokenizer and then input into the pre-trained BERT model. Each word that is input is converted into each word representation vector by the BERT model. As the POS Tagger model, the POS tagger model of Stanford NLP is used to tag the POS of each word, and each tagged tag is one-hot encoded into a 34-dimensional vector. Thereafter, each vector output from the BERT model and the POS tagger model is concatenated and input into the Deep Q-Network reinforcement learning model.

In the DQN model, if a sentence that has been preprocessed is given in the form of a 34-dimensional vector; it solves the problem of classifying which named entity category each token belongs to in order from the beginning of the sentence. If the agent has classified each word correctly, the agent will receive a positive reward. If a word is misclassified, the agent will receive a negative reward. The process of classifying all tokens in a sentence becomes one epoch of the reinforcement learning model, and the process is repeated continuously while converging on the optimal policy.

Named entity recognition is a traditional classification task that determines whether each word is a named entity or not. To apply the reinforcement learning model to this classification task, it is necessary to configure the reinforcement learning agent in a form that can be trained. Therefore, the important elements used in reinforcement learning should be redefined to suit the named entity recognition problem. The following describes three key elements of the reinforcement learning model defined in this paper: state, action, and reward.

3.2.1. State

The state represents the environment observed by the agent. As the environment changes, the agent trains the policy regarding which action to choose next. The reinforcement learning model receives a high-dimensional vector representation of each word extracted from the BERT model. We use a neural network that is a non-linear function approximator for Q functions to infer each probability value for which action to choose based on state. In other words, the state space of the reinforcement learning model is continuous. We use two models for experiments—BERT-base and BERT-large—which are 768

and 1024 in size, respectively. Since a POS-tagged one-hot vector has 34 dimensions, the input layer of the DQN model has 802 and 1058-dimensional vector plus 34 dimensions, respectively.

3.2.2. Action

The selectable action space is discrete, and the action space that the agent can select corresponds to the number of categories of the named entity to be finally classified. In our proposed model, action space is defined based on the named entity category of the datasets. For example, the CoNLL dataset [35] is classified into five categories: person, location, organization, miscellaneous, and outside. The token tagging has a Begin (B-) prefix to indicate the start of each named entity and an Inside (I-) prefix to indicate the middle or continuation of the named entity. The number of actions the agent can select is B-PER, I-PER, B-LOC, I-LOC, B-ORG, I-ORG, B-MISC, I-MISC, and Outside 9 dimensions.

3.2.3. Reward

The reward is a feature of reinforcement learning that is most noticeably different from other machine learning methods such as supervised learning and unsupervised learning. Depending on how the reward function is designed, the agent can train and converge quickly to the optimal policy, which directly affects the training time. The reward of the DeNERT model is trained by receiving a positive reward (+1) if the agent correctly classifies the named entity category for each token, and a negative reward (−1) if it misclassifies. It is the simplest and most widely used approach in many reinforcement learning models, and it is still a powerful one.

3.3. Knowledge-Graph-Based Relation Extraction Model

Recently, there have been many studies that aim to help computers understand human sentences as humans do. However, there is still a limit to developing human cognitive abilities in computers. To understand sentences more like humans, we need knowledge graph technology that links words to words. Various studies on knowledge graphs such as WordNet, YAGO, Probase, and ConceptNet are underway.

Knowledge graphs link words to the relationship between words, so using them can improve the performance of the relationship extraction model. We create knowledge graph embedding using ConceptNet, an open-source knowledge graph that has more than 8 million objects connected based on more than 40 relationships. First, we collect all data from ConceptNet to create knowledge graph embedding. ConceptNet has a lot of words (entities), and a lot of information based on the relationships between each entity. We use each entity; the relationship the entity has; and SurfaceText, which contains contextual information about the relationship between the entities.

As shown in Figure 4, SurfaceText is used as a context sentence, and the information between two entities is a graph, so it goes through a separate processing process. The graph is eventually a link between the two entities, which can be expressed in the form of {head, relation, tail}. The “head” is the entity that the link points to when the two entities are connected based on the relationship—in other words, the head of the arrow. Conversely, “tail” means the tail of the arrow. The context embedding is obtained through the BERT encoder after going through the tokenization process. The corresponding head, relation, and tail of the graph are concatenated using [SEP], which is the special token of BERT, and then applied to the BERT encoder. To obtain knowledge graph embedding that contains both context and relationship information within the graph, each acquired embedding value is concatenated.

Knowledge graph embedding is used for the final relationship information extraction for a given sentence. Based on the NER information found through the DeNERT model in a given sentence, we extract the relationship between the subject and the object. At this time, we can conduct self-attention by concatenating the knowledge graph embedding.

Figure 5 shows the learning style of the model, which extracts the relationship by adding the NER information extracted through the DeNERT model to the sentence. [CLS] and [SEP] are separate tokens used in the ALBERT model, which means the beginning and end of arming, respectively.

Tok 1, Tok 2, Tok n stands for each token after tokenization of input for learning in the ALBERT model. SUBJ also means the word that is given within the input sentence, that is, the word that is the subject of the relationship, and OBJ is the object, that is, the word that is the object of the relationship. In addition, NER tag information extracted through the DeNERT model is added as subj_type and obj_type, respectively. In ALBERT, a bidirectional pre-trained language model output becomes subj, obj, and relation.

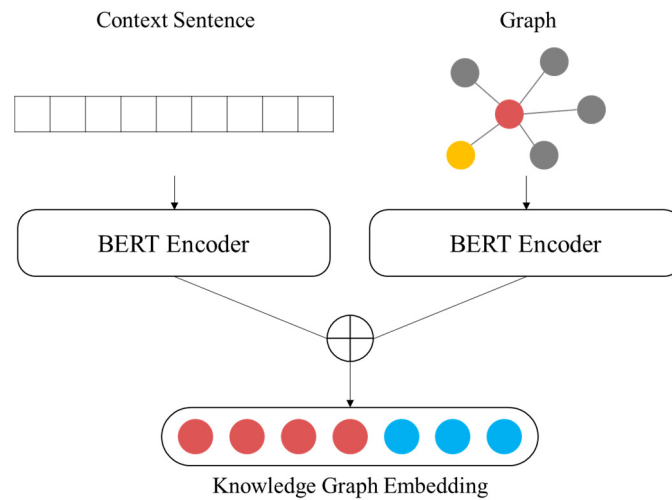


Figure 4. Knowledge graph embedding layer.

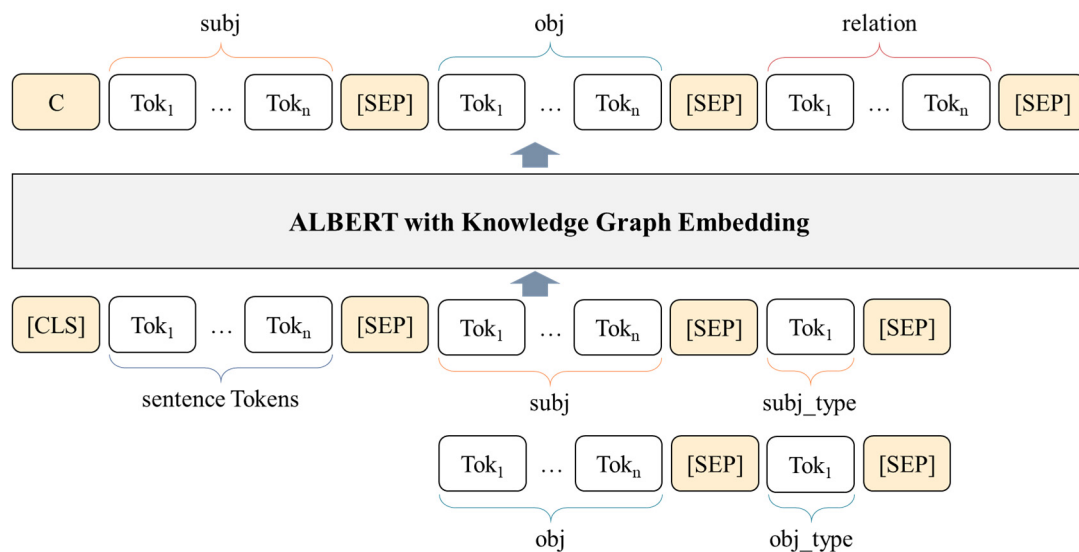


Figure 5. Input and output of the relation extraction model.

4. Experiments

4.1. Dataset

4.1.1. NER Dataset

The accuracy of the DeNERT model is important because the DeNERT-KG model goes through the NER model and then applies the relationship extraction model based on it. To verify the accuracy of DeNERT, we used CoNLL 2003 [35], EWNERTC [36], and W-NUT 2007 [37] datasets in the experiment.

Table 1 shows the statistics of the datasets we used. CoNLL 2003 is a traditional English dataset for named entity recognition. It is classified according to four categories: PER, LOC, ORG, and MISC. Each category refers to person, location, organization, and miscellaneous, respectively. It consists of

about 15,000 training sets, about 3500 Dev sets, and 3700 test sets. As with CoNLL 2003, the EWNERTC dataset is classified into person, location, organization, and miscellaneous; however, it does not provide the distributed data set for train/dev/test set, so we divide data by 80% for train, 10% for dev, and 10% for test set. On the other hand, the W-NUT 2017 dataset is classified into six categories: person, location, group, creative-work, product, and corporation.

Table 1. Statistics of named entity recognition (NER) datasets.

Dataset	Category	Train	Dev	Test
CoNLL 2003	PER	6600	1842	1617
	LOC	7140	1837	1668
	ORG	6321	1341	1661
	MISC	3438	922	702
EWNERTC	People	283,188	35,399	35,398
	Location	237,054	29,632	29,632
	Organization	60,752	7594	7594
	Misc	778,781	97,348	97,348
W-NUT 2007	Person	995	587	560
	Location	793	107	244
	Group	414	64	235
	Creative-Work	346	238	360
	Product	345	208	253
	Corporation	267	46	88

Because the structure of the model proposed and reinforced learning can be used several times for agent learning per dataset record, the actual learning involves masking the name of the object, and when the actual sentence is entered into the model, the completed sentence is randomly sampled from each list of object names and used as input.

For example, if the sentence “Nader Jokhadar had given Syria the lead with a well-struck header in the seventh minute” is given as input, the actual model learning will replace the masked part of the object name with another word in the same category and use it for multiple learning. People, names, etc., can be replaced with these different words for learning because they do not lose their meaning even if they are replaced by other words in the same category.

4.1.2. Relation Extraction Dataset

For a knowledge graph-based relationship extraction model, we utilize ConceptNet data and TACRED data sets. ConceptNet is a dataset that is used to embed knowledge graphs, and only English data out of the ConceptNet 5.6 dataset is used for embedding learning.

The TACRED dataset is used not only to learn the relationship extraction model but also to verify the proposed DeNERT-KG model. TACRED data can be obtained through the Linguistic Data Consortium (LSD), consisting of a total of 106,264 sentences. It is separated into train, dev, and test set and contains 68,124; 22,631; and 15,509 sentences, respectively. A relationship extraction dataset made from Coppers is used in the TAC Knowledge Base Population Challenge, containing a total of 42 relationships, including “no_relation”.

Table 2 shows the relationship information of TACRED. It has a total of 42 types of relationship information related to people and institutions, including “no_relation” that has no relationship. The TACRED dataset is used as a very important dataset for learning and validating the model we propose because it has tagged information as well as relationship information.

Table 2. Relations in TACRED dataset.

Relations	
no_relation	per:children
org:alternate_names	per:cities_of_residence
org:city_of_headquarters	per:city_of_birth
org:country_of_headquarters	per:city_of_death
org:dissolved	per:countries_of_residence
org:founded	per:country_of_birth
org:founded_by	per:country_of_death
org:member_of	per:date_of_birth
org:members	per:date_of_death
org:number_of_employees/members	per:employee_of
org:parents	per:origin
org:political/religious_affiliation	per:other_family
org:shareholders	per:parents
org:stateorprovince_of_headquarters	per:religion
org:subsidiaries	per:schools_attended
org:top_members/employees	per:siblings
org:website	per:spouse
per:age	per:stateorprovince_of_birth
per:alternate_names	per:stateorprovince_of_death
per:cause_of_death	per:stateorprovinces_of_residence
per:charges	per:title

4.2. Experiment Setup

The DQN model was implemented using Tensorflow.js (tf.js), a JavaScript-based machine learning library. Model learning uses the browser's WebGL GPU acceleration technology to learn with built-in GPU. The BERT model was implemented through the TensorFlow library. The relationship extraction added DeNERT-KG model also uses Google colab's TPU environment. The model is implemented in PyTorch. Table 3 shows more detail about the configuration of the server.

Table 3. Configuration of the server.

Type	Detail Configuration
CPU	Intel Xeon E5-2620 v3@2.40GHz
OS	Ubuntu 16.04
Memory	32 GB
HDD	3.6 TB
GPU	EVGA GeForce GTX 1080

4.3. Results

4.3.1. Result for NER Task

CoNLL 2003, EWNERTC, and W-NUT 2017 datasets are utilized to verify the performance of the DeNERT model that performs NER task. We use F1 score as the metric of the experiments.

Table 4 is an experiment result of EWNERTC dataset. The proposed DeNERT model is applied to Deep Q-network by conforming the BERT model and POS embedding. We conducted an experiment on EWNERTC datasets to verify the performance improvement when applying POS tagging and different sizes of BERT models used in DeNERT model. All models are the results of Deep Q-Network application. First of all, we conducted an experiment on the size of the BERT model. We also experimented with models with POS embedding on each model. As a result, the DeNERT model we propose shows the highest F1-score.

Table 4. Experiment result on EWNERTC dataset.

Model	Precision	Recall	F1
BERT-Base	77.52	70.58	73.89
BERT-Base + POS	78.82	70.84	74.62
BERT-Large	80.25	73.89	76.94
BERT-Large + POS (DeNERT)	81.97	74.13	77.85

Table 5 is the result of a comparative experiment with existing models to verify the model we propose. CoNLL 2003 and W-NUT 2017 datasets are used to conduct experiments. Results of the CoNLL 2003 dataset show the highest F1-score 93.45 when the proposed model is used. In W-NUT 2017 dataset, our model shows the second score 45.78. The W-NUT 2007 data show relatively lower results than the CoNLL dataset because many noise data exist.

Table 5. Comparison result on CoNLL 2003 and W-NUT 2017 dataset.

Dataset	Model	F1-Score
CoNLL 2003	Akbik et al., 2019 [38]	93.18
	Brian et al., 2020 [39]	91.47
	Peters et al., 2017 [40]	91.93
	DeNERT	93.45
W-NUT 2017	Pius et al., 2017 [41]	41.06
	Gustavo et al., 2019 [37]	41.86
	Akbik et al., 2019 [38]	49.59
	Brian et al., 2020 [39]	40.59
	DeNERT	45.78

4.3.2. Result for Relation Extraction Task

DeNERT-KG, which we propose, extracts subject and object through NER, and finally uses a knowledge graph to extract the relationship between subject and object. To verify the model of DeNERT-KG from which the relationship was finally extracted, we utilized the TACRED dataset containing the relationship information in the experiment.

First, an experiment was conducted to verify the performance improvement when using knowledge graph embedding to extract relationship information. Table 6 is the result of conducting experiments on various variants of BERT and BERT and adding knowledge graph embedding. We are proposing ways to apply knowledge graph embedding to extract relationship information. As shown in Table 6, it can be seen that better performance was achieved when applying knowledge graph embedding was applied, in comparison to simply applying a language model. Finally, we were able to get the best results when we used the DeNERT-KG model with ALBERT and knowledge graph embedding in DeNERT.

Table 6. Experiment result on TACRED dataset.

Model	Precision	Recall	F1-Score
BERT-only	66.4	66.8	66.6
BERT + KG	72.6	68.3	70.4
Electra-base + KG	48.2	43.6	45.8
ALBERT + KG (DeNERT-KG)	71.8	73.1	72.4

Table 7 is a comparative test of the same TACRED dataset with other existing studies. The DeNERT-KG we propose shows the highest F1-score, as we can see in the table. Although the proposed model did not achieve higher performance improvements than the existing one, the existing models

extract only relevant information from the sentences. We can see the meaningful result because we can extract the subject and the object through the DeNERT model, extract the relationship between the two, and finally identify up to {subject, subject_type, object, object_type, and relation}.

Table 7. Comparison result on TACRED dataset.

Model	Precision	Recall	F1-Score
Matching-the-Blanks [23]	-	-	71.5
C-GCN + PA-LSTM [22]	71.3	65.4	68.2
PA-LSTM [21]	65.7	64.5	65.1
Logistic Regression [22]	73.5	49.9	59.4
ALBERT + KG (DeNERT-KG)	71.8	73.1	72.4

5. Conclusions

In this paper, we propose a DeNERT-KG model that can extract the subject, object, and relationships from a sentence. The ability to understand and process people's language is very important for computers to learn on their own. To understand a person's language, in particular, one must be able to find the subject and object underlying the sentence and find the relationship between the two. To carry out these tasks, we use the DeNERT model with Deep Q-Network to extract the subject, object, and type of each word. It also applies the extracted information to the relationship extraction model applied with knowledge graph embedding so that the object can finally be given, and so the relationship information can be extracted.

When inputting sentences, DeNERT-KG, which can extract subject, object, and relationship, is a very important model for use in various fields. However, as it is still a model that goes through the steps of extracting relationships with NER, it can be given through one model, creating a high-performance model that can extract triples of entities and relationships.

Already, many studies are being actively conducted in the field of NER and relationship extraction. However, studies to perform both NER and relationship extraction tasks at the same time are relatively limited due to the complexity of the task. We not only propose a model that can perform two tasks at once but also show the possibility of using DQN in the NER task and the possibility of the knowledge graph in relation extraction task. Additionally, designing and building a model that can extract quadruple information at once, not only subject, object, and relationships, but also the emotion between subject and object, is expected to make an important contribution to the NLP task.

Still, there are some challenges such as handling multiple entities. If the proposed model has multiple entities in the sentence, each entity can be identified as a separate entity and extract multiple relationships, but there is a limitation that cannot be extracted to equivalent entities in the same relationship. When the list of multiple entities is extracted, it can be resolved to some extent by adding an automatic postprocessing process. In future work, we could create a robust model for multiple entities. Additionally, there is a challenge to overcome the limitation of the dependency on the dataset. Only extracting the relationship of the proposed model is possible, especially for the relationship specified in the dataset. To overcome this limitation, we could utilize techniques such as distant supervision to enable the extraction of unlabeled relationships.

Author Contributions: Conceptualization, S.Y. (SungMin Yang), S.Y. (SoYeop Yoo), and O.J.; methodology, S.Y. (SungMin Yang) and S.Y. (SoYeop Yoo); software, S.Y. (SungMin Yang) and S.Y. (SoYeop Yoo); validation, S.Y. (SungMin Yang) and S.Y. (SoYeop Yoo); formal analysis, S.Y. (SungMin Yang) and S.Y. (SoYeop Yoo); investigation, O.J.; resources, O.J.; data curation, S.Y. (SungMin Yang) and S.Y. (SoYeop Yoo); writing—original draft preparation, S.Y. (SungMin Yang) and S.Y. (SoYeop Yoo); writing—review and editing, O.J.; visualization, S.Y. (SungMin Yang) and S.Y. (SoYeop Yoo); supervision, O.J.; project administration, O.J.; funding acquisition, O.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Basic Science Research Program through the NRF(National Research Foundation of Korea), and the MSIT(Ministry of Science and ICT), Korea, under the National Program for

Excellence in SW supervised by the IITP(Institute for Information and Communications Technology Promotion) (Nos. NRF2019R1A2C1008412, 2015-0-00932).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chowdhary, K.R. Natural Language Processing. In *Fundamentals of Artificial Intelligence*; Springer: New Delhi, India, 2020; pp. 603–649.
2. Hirschberg, J.; Manning, C.D. Advances in natural language processing. *Science* **2015**, *349*, 261–266. [[CrossRef](#)] [[PubMed](#)]
3. Otter, D.W.; Medina, J.R.; Kalita, J.K. A Survey of the Usages of Deep Learning for Natural Language Processing. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, 1–21. [[CrossRef](#)] [[PubMed](#)]
4. Young, T.; Hazarika, D.; Poria, S.; Cambria, E. Recent Trends in Deep Learning Based Natural Language Processing. *IEEE Comput. Intell. Mag.* **2018**, *13*, 55–75. [[CrossRef](#)]
5. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [[CrossRef](#)] [[PubMed](#)]
6. 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, Minneapolis, MN, USA, 2–7 June 2019; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA, 2019; Volume 1, pp. 4171–4186.
7. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of the Advances in Neural Information Processing Systems 26, Lake Tahoe, NV, USA, 5–10 December 2013; Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q., Eds.; Curran Associates, Inc.: San Diego, CA, USA, 2013; pp. 3111–3119.
8. Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; Dyer, C. Neural Architectures for Named Entity Recognition. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; Association for Computational Linguistics: Stroudsburg, PA, USA, 2016; pp. 260–270.
9. Wilks, Y. Natural Language Processing. *Commun. ACM* **1996**, *39*, 60–62. [[CrossRef](#)]
10. Huang, Z.; Xu, W.; Yu, K. Bidirectional LSTM-CRF Models for Sequence Tagging. *arXiv* **2015**, arXiv:1508.01991.
11. Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep Contextualized Word Representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LA, USA, 1–6 June 2018; Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; Volume 1, pp. 2227–2237.
12. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Improving Language Understanding by Generative Pre-Training. *OpenAI* **2018**, 1–10, work in progress.
13. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 2017, pp. 5999–6009.
14. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing Atari with Deep Reinforcement Learning. *arXiv* **2013**, arXiv:1312.5602.
15. Fang, M.; Li, Y.; Cohn, T. Learning how to active learn: A deep reinforcement learning approach. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA, 2017; pp. 595–605.
16. Yang, Y.; Chen, W.; Li, Z.; He, Z.; Zhang, M. Distantly Supervised NER with Partial Annotation Learning and Reinforcement Learning. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–26 August 2018; pp. 2159–2169.
17. Li, P.; Mao, K.; Yang, X.; Li, Q. Improving relation extraction with knowledge-attention. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Hong Kong, China, 3–7 November 2020; pp. 229–239.

18. Hong, Y.; Liu, Y.; Yang, S.; Zhang, K.; Wen, A.; Hu, J. Improving graph convolutional networks based on relation-aware attention for end-to-end relation extraction. *IEEE Access* **2020**, *8*, 51315–51323. [[CrossRef](#)]
19. Wang, Q.; Mao, Z.; Wang, B.; Guo, L. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 2724–2743. [[CrossRef](#)]
20. Kumar, S. A Survey of Deep Learning Methods for Relation Extraction. *arXiv* **2017**, arXiv:1705.03645.
21. Zhang, Y.; Zhong, V.; Chen, D.; Angeli, G.; Manning, C.D. Position-aware Attention and Supervised Data Improve Slot Filling. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; Association for Computational Linguistics: Stroudsburg, PA, USA, 2017; pp. 35–45.
22. Zhang, Y.; Qi, P.; Manning, C.D. Graph Convolution over Pruned Dependency Trees Improves Relation Extraction. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; pp. 2205–2215.
23. Soares, L.B.; FitzGerald, N.; Ling, J.; Kwiatkowski, T. Matching the blanks: Distributional similarity for relation learning. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA, 2020; pp. 2895–2905.
24. Pang, Y.; Liu, J.; Liu, L.; Yu, Z.; Zhang, K. A Deep Neural Network Model for Joint Entity and Relation Extraction. *IEEE Access* **2019**, *7*, 179143–179150. [[CrossRef](#)]
25. Miwa, M.; Sasaki, Y. Modeling Joint Entity and Relation Extraction with Table Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; Association for Computational Linguistics: Stroudsburg, PA, USA, 2014; pp. 1858–1869.
26. Gupta, P.; Schütze, H.; Andrassy, B. Table filling multi-task recurrent neural network for joint entity and relation extraction. In Proceedings of the 26th International Conference on Computational Linguistics, Osaka, Japan, 11–16 December 2016; Technical Papers, pp. 2537–2547.
27. Adel, H.; Schütze, H. Global Normalization of Convolutional Neural Networks for Joint Entity and Relation Classification. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; Association for Computational Linguistics: Stroudsburg, PA, USA, 2017; pp. 1723–1729.
28. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv:1907.11692.
29. Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv* **2019**, arXiv:1910.01108.
30. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *arXiv* **2019**, arXiv:1909.11942.
31. Miller, G.A. WordNet: A lexical database for English. *Commun. ACM* **1995**, *38*, 39–41. [[CrossRef](#)]
32. Suchanek, F.M.; Kasneci, G.; Weikum, G. Yago. In Proceedings of the 16th international conference on World Wide Web, Banff, AB, Canada, 8–12 May 2007; ACM Press: New York, NY, USA, 2007; p. 697.
33. Wu, W.; Li, H.; Wang, H.; Zhu, K.Q. Probase. In Proceedings of the 2012 international conference on Management of Data, Scottsdale, AZ, USA, 20–25 May 2012; ACM Press: New York, NY, USA, 2012; p. 481.
34. Liu, H.; Singh, P. ConceptNet—A Practical Commonsense Reasoning Tool-Kit. *BT Technol. J.* **2004**, *22*, 211–226. [[CrossRef](#)]
35. Sang, E.F.T.K.; De Meulder, F. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. *arXiv* **2003**, arXiv:cs/0306050.
36. Sahin, H.B.; Tirkaz, C.; Yildiz, E.; Eren, M.T.; Sonmez, O. Automatically Annotated Turkish Corpus for Named Entity Recognition and Text Categorization Using Large-Scale Gazetteers. *arXiv* **2017**, arXiv:1702.02363.
37. Aguilar, G.; Maharjan, S.; López Monroy, A.P.; Solorio, T. A Multi-task Approach for Named Entity Recognition in Social Media Data. In Proceedings of the 3rd Workshop on Noisy User-generated Text, Copenhagen, Denmark, 7 September 2017; Association for Computational Linguistics: Stroudsburg, PA, USA, 2017; pp. 148–153.
38. Akbik, A.; Bergmann, T.; Vollgraf, R. Pooled contextualized embeddings for named entity recognition. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 724–728.

39. Lester, B.; Pressel, D.; Hemmeter, A.; Choudhury, S.R. Computationally Efficient NER Taggers with Combined Embeddings and Constrained Decoding. *arXiv* **2020**, arXiv:2001.01167.
40. Peters, M.E.; Ammar, W.; Bhagavatula, C.; Power, R. Semi-supervised sequence tagging with bidirectional language models. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA, 2017; Volume 1, pp. 1756–1765.
41. von Däniken, P.; Cieliebak, M. Transfer Learning and Sentence Level Features for Named Entity Recognition on Tweets. In Proceedings of the 3rd Workshop on Noisy User-generated Text (W-NUT), Copenhagen, Denmark, 7 September 2017; Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; Volume 3, pp. 166–171.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).