Knowledge-Graph Augmented Word Representations for Named Entity Recognition

Qizhen He,[*] Liang Wu,[*] Yida Yin, Heming Cai
Bilibili, Shanghai, China
{heqizhen, wuliang, yinyida, caiheming}@bilibili.com

Abstract

By modeling the context information, ELMo and BERT have successfully improved the state-of-the-art of word representation, and demonstrated their effectiveness on the Named Entity Recognition task. In this paper, in addition to such context modeling, we propose to encode the prior knowledge of entities from an external knowledge base into the representation, and introduce a Knowledge-Graph Augmented Word Representation or KAWR for named entity recognition. Basically, KAWR provides a kind of knowledge-aware representation for words by 1) encoding entity information from a pre-trained KG embedding model with a new recurrent unit (GERU), and 2) strengthening context modeling from knowledge wise by providing a relation attention scheme based on the entity relations defined in KG. We demonstrate that KAWR, as an augmented version of the existing linguistic word representations, promotes F1 scores on 5 datasets in various domains by +0.46~+2.07. Better generalization is also observed for KAWR on new entities that cannot be found in the training sets.

1. Introduction

Named entity recognition plays an important role in various applications such as search engines and question-answering systems, and has always been studied as a sequence-tagging problem in Nature Language Processing. Recent progress is reported either by constructing more sophisticated classifiers like Bi-LSTM-CRF (Huang et al., 2015) and Bi-LSTM-CNN (Chiu and Nichols, 2016) or by more powerful word representations like ELMo (Peters et al., 2018) and BERT (Devlin et al., 2018). In this paper, we focus on the representation part and propose a more descriptive representation by introducing the information of entities and their relations which are defined in a knowledge base.

The motivation for this paper comes from an intuitive idea: the entities and their relations provided by the knowledge base can provide strong signals to recognize the named entities. In a typical classifier for Named Entity Recognition (NER), such signals can be captured by man-made features. To relieve the pressure on feature engineering and to provide a uniform word representation that is both context-aware and knowledge-aware, we present a knowledge-graph augmented word representation (KAWR). Specifically, the primary contribution of this paper is that 1) we introduce a new kind of word embedding w_i^{entity} based on the related entities in the knowledge graph through a novel Gated Entity-based Recurrent Unit or GERU, 2) based on w_i^{entity}, we design a new attention function that is aware of entity relations, to provide $w_i^{relation}$ which models the relation context of the entities in the sequence, 3) w_i^{entity} and $w_i^{relation}$ then concatenated with the embedding vector $w_i^{language}$ from a standard language model to provide more powerful features for classification. As will be demonstrated in the experiment section, the KAWR can work with the existing context-aware word representations, and provide better results than BERT on five datasets by +0.46~+2.07 increase on F1 score. Noticeably, greater increases can be observed on small training-sets (which is often the case for domain-specific tasks) suggesting less dependency on the training set and better generalization for KAWR.

Although we are not the first to introduce the knowledge information into NLP models (Annervaz et al., 2018; Xin et al., 2018), this is the first time, to our best knowledge, that knowledge information is encoded into word representation and the text context is modeled based on the entity-relations defined in the knowledge base.

The rest of the paper is organized as follows: We first present related work in section 1.1. We then describe the proposed methods and formulations in details in section 2, followed by experiments and results in section 3. We conclude our paper with discussion on future work in section 4.

[*] They contributed equally to this work.

1.1 Related Work

Language representation, which is the basis for many natural language processing problems, has been a hot research area for decades. Bags-Of-Words model was introduced in the last century and has dominated the text classification literature for a long time. The drawbacks are that the BOW model ignores the sequence of the words and discards contextual information. Language models based on n-gram are then proposed to model the probability distribution over sequences of words. However, due to the sparsity of the word combinations, it is difficult for n-gram to model long sequences. On the other hand, embeddings give language representation another form, where words or phrases are mapped into vectors in a continuous vector space with a much lower dimension (compared to the BOW vector space with one dimension per word). Various embedding techniques have been proposed (Mikolov et al., 2013; Omer et al., 2014; Rémi et al., 2014). This kind of continuous and low-dimensional vector representation then greatly boosts neural network to capture high level linguistic characteristics. For example, RNN models are used to 1) give the probability distribution on each word in the sentence as a general language model (Sundermeyer et al., 2012), 2) encode the entire sentence by the hidden states (Liu et al., 2016); CNN models have also been proposed to capture the contextual information through convolution and pooling (Conneau et al., 2017). Despite their success, word embeddings proposed then are context-independent, which means each word holds the same embedding vector across all the appearances in the text corpus. Therefore, the downstreaming task should have a sophisticated network architecture that needs to be carefully designed to capture contextual information. To overcome this shortage, several contextual-aware representations based on either RNN or self-attention (Lin et al., 2017) were proposed in 2018. ELMo (Peters et al., 2018) sums up several bi-LSTM layers' output vectors, while BERT (Devlin et al. 2018) introduces Masked Language Model together with stacked transformers. These models significantly increase the capability of the word representations so that the task-specific architectures can be simplified greatly to achieve the same or even better results.

However, such representation models capture essentially statistical language behavior, which is dependent to a large extent on the training corpus. So additional information that goes beyond datasets could be leveraged for better performance, especially in the case of domain-specific tasks where the training samples are often insufficient. Strubell et al. (2018) propose syntactically informed self-attention by introducing the pre-trained syntactic model into self-attention encoders. They then demonstrate the effectiveness of the proposed model on the Semantic Role Labeling (SRL) task.

Knowledge Graph, which models world knowledge in the form of entities and facts, is another emerging research area in natural language processing, making its contributions to various applications from search engines to question-answering systems. Typically, world knowledge is represented in the form of fact triplets (subject entity, relation, object entity), denoted by (h, r, t). Various KG embedding techniques have been reported to encode the entities and relations into numerical representations since TransE (Bordes et al., 2013). While some techniques focus on graph structure encoding (Bordes et al., 2013; Xiao et al., 2015; Ji et al., 2015; Shi and Weninger, 2017), others are trying to learn entity/relation embeddings along with their semantic information (Xie et al., 2016; Zhong et al., 2015). Soon these graph embeddings are used to benefit some NLP tasks. Annervaz et al. (2018) proposed a knowledge graph augmented neural networks where entity and relation vectors are retrieved by and then concatenated with the context vector (which encodes the entire input text by LSTM) for text classification. Xin et al. (2018) proposed a new entity typing model by introducing knowledge attention which is formulated as function of the entity to be typed.

2. The Proposed Model

The basic idea of our proposed model is to encode the entity information into the word embeddings. These embeddings can be further used to strengthen context modelling by taking entity-relations into account, and provide a more powerful word representation that is not only context-aware but also knowledge-aware.

2.1 Word Embeddings Based on Entity

We introduce here a new word embedding technique which is based on the entities in knowledge graph. Given an N-token sequence $(t_1, t_2, \ldots, t_N)$ and a knowledge graph G (that contains entities, relations and fact triplets), the aim is to generate an output vector w_i^{entity} for each token t_i based on the entities in G.

Each token t_i in the sequence can be related to an entity set $\mathbb{E}_i = \{\mathbb{E}_{i,j}\}$, where $\mathbb{E}_{i,j}$ is a related entity in the knowledge graph G, with t_i being the entity name or part of the entity name. $\mathbb{E}_{i,j}$ can be represented by a vector $e_{i,j}$ in a pre-trained embedding model.

Intuitively, w_i^{entity} should be close to the embedding vector of the most probable (according to the context) entity. For example, considering the token "United" in the sentence "Donald Trump is the president of the United States of America", we get (through an Entity Retrieving operation) a set of entities related to "United" like "United States of America", "United Nations" and "United Airlines" from the knowledge base. Ideally the output vector w_i^{entity} for word “United” in the above sentence should be close to the embedding vector of the entity “United States of America” due to the context “Donald Trump” and “States of America”.

To achieve this, we design a recurrent neural network unit called Gated Entity-based Recurrent Unit (GERU), as illustrated in Figure 1. In GERU, where each token t_i is associated with an input vector m_i, we use RNN to model the context of t_i, and generate w_i^{entity} accordingly through an attention scheme over the related entities.

A typical attention scheme can be defined as a function to map a query and a set of key-value pairs to an output

$$attention(q, K, V) = softmax(f(q, K))V \quad (1)$$

where the query vector is represented by q, vectors of keys and values are packed into matrices K and V respectively, and f defines a distance measure function between q and K.

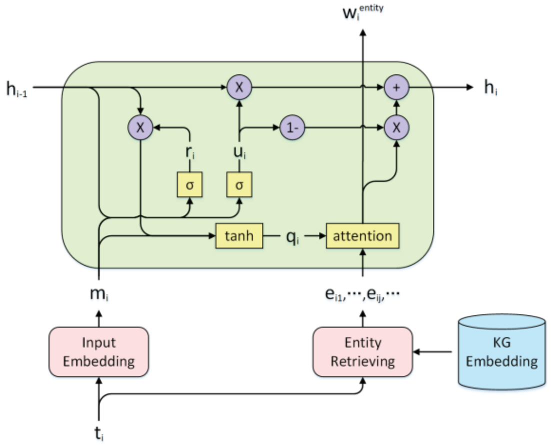

Figure 1 Gated Entity-based Recurrent Unit.

In the proposed GERU, the forward context of t_i, denoted by $\vec{q}_i$, is formulated as a function of the previous hidden status $\vec{h}_{i-1}$ and the current input m_i with a reset gate $\vec{r}_i$. $\vec{q}_i$ (with a projection matrix $\overrightarrow{W}^Q$) is then regarded as the query vector q in the attention settings, where the keys K and values V are based on E_i, which is a collection of the related entity embeddings $\{e_{i,j}\}$ from the pre-trained KG embedding model. The formulations are as follows:

$$\vec{q}_i = \tanh\left(\overrightarrow{W}_q * \left[\vec{r}_i * \vec{h}_{i-1}, m_i\right] + \vec{b}_q\right) \quad (2)$$

$$\overrightarrow{w}_i^{entity} = attention(\vec{q}_i \overrightarrow{W}^Q, E_i \overrightarrow{W}^K, E_i) \quad (3)$$

$$\vec{r}_i = \sigma(\overrightarrow{W}_r * \left[\vec{h}_{i-1}, m_i\right] + \vec{b}_r) \quad (4)$$

where $\overrightarrow{W}^Q$ and $\overrightarrow{W}^K$ are projection matrices, $\overrightarrow{W}_q$, $\overrightarrow{W}_r$, $\vec{b}_q$, $\vec{b}_r$ are parameters, $\vec{r}_i$ is the reset gate, σ is the sigmoid function, and the distance measure in the attention function is defined as

$$f(q, K) = qK^T \quad (5)$$

We take $\overrightarrow{w}_i^{entity}$ as an entity-based word representation for t_i with forward context. And $\overrightarrow{w}_i^{entity}$ is also used to update the hidden state $\vec{h}_i$

$$\vec{h}_i = \vec{u}_i * \vec{h}_{i-1} + (1 - \vec{u}_i) * \overrightarrow{w}_i^{entity} \quad (6)$$

where $\vec{u}_i$is the update gate and defined as

$$\vec{u}_i = \sigma\left(\overrightarrow{W}_u * \left[\vec{h}_{i-1}, m_i\right] + \vec{b}_u\right). \quad (7)$$

The differences between GERU and a standard Gated Recurrent Unit (GRU) (Cho et al., 2014) are that: 1) the hidden status $\vec{h}_i$ is updated by using $\vec{q}_i$ in GRU, but we use $\overrightarrow{w}_i^{entity}$ to update $\vec{h}_i$ in GERU in order to strengthen the signals directly from the related entities; 2) the output of GRU is usually given by projections on the hidden status $\vec{h}_i$, but the output of GERU ($\overrightarrow{w}_i^{entity}$) is given by an attention over the related entities with regard to $\vec{q}_i$ as the query vector.

Similarly, we can get $\overleftarrow{w}_i^{entity}$ based on the backward context through a backward GERU:

$$\overleftarrow{q}_i = \tanh\left(\overleftarrow{W}_q * \left[\overleftarrow{r}_i * \overleftarrow{h}_{i-1}, m_i\right] + \overleftarrow{b}_q\right) \quad (8)$$

$$\overleftarrow{w}_i^{entity} = attention(\overleftarrow{q}_i \overleftarrow{W}^Q, E_i \overleftarrow{W}^K, E_i) \quad (9)$$

$$\overleftarrow{h}_i = \overleftarrow{u}_i * \overleftarrow{h}_{i-1} + (1 - \overleftarrow{u}_i) * \overleftarrow{w}_i^{entity} \quad (10)$$

$$\overleftarrow{r}_i = \sigma(\overleftarrow{W}_r * \left[\overleftarrow{h}_{i-1}, m_i\right] + \overleftarrow{b}_r) \quad (11)$$

$$\overleftarrow{u}_i = \sigma(\overleftarrow{W}_u * \left[\overleftarrow{h}_{i-1}, m_i\right] + \overleftarrow{b}_u). \quad (12)$$

Our proposed word embedding $w_i^{entity}$is then given by

$$w_i^{entity} = \frac{\left(\overrightarrow{w}_i^{entity} + \overleftarrow{w}_i^{entity}\right)}{2} \quad (13)$$

The input vector m_i, together with the matrices $\overrightarrow{W}_q, \vec{b}_q, \overrightarrow{W}_r, \vec{b}_r, \overrightarrow{W}_u, \vec{b}_u, \overrightarrow{W}^K, \overrightarrow{W}^Q, \overleftarrow{W}_q, \overleftarrow{b}_q, \overleftarrow{W}_r, \overleftarrow{b}_r, \overleftarrow{W}_u, \overleftarrow{b}_u, \overleftarrow{W}^K$, $\overleftarrow{W}^Q$ are model parameters to be learnt. We can also pre-train these parameters by feeding w_i^{entity} into a down-streaming classification network.

2.2 Attention Based on Entity Relationship

Given the entity-based word embeddings described in 2.1, we introduce a new attention function to model the relation context of a word in the sentence. Given the sentence $(t_1, t_2, \ldots, t_N)$, the relation context of the word t_i refers to the relations (from the knowledge wise) between t_i and other words in the sentence. The basic idea is that when we are considering the context of a word t_i, we need to pay special attention to those words that may have a relation with t_i in the knowledge graph.

For a typical KG embedding model, the entities and relations are encoded into vectors as e and r respectively, and the following equation holds when the fact triplet $(\mathbb{E}_h, \mathbb{R}_k, \mathbb{E}_t)$ exists in KG

$$e_h - r_k \approx e_t \qquad (14)$$

where e_h , e_t and r_k are the embedding vectors of the subject entity $\mathbb{E}_h$, the object entity $\mathbb{E}_t$, and the relation $\mathbb{R}_k$ respectively.

Considering q as the query entity and K as packed entities in the sequence, we then define a new attention function $attention^{rel}$ over the entities in the sentence based on their relations with the query entity:

$$attention^{rel}(q, K, V) = softmax(Rel(q, K))V \qquad (15)$$

where $Rel(q, K)$ provides a relation measure (instead of the distance measure in the typical attention function) between query vector q and key matrix K as follows

$$r_i = q - k_i \qquad (16)$$

$$\hat{r}_i = r_i * R^T \qquad (17)$$

$$Rel(q, k_i) = \hat{r}_i * W_R \qquad (18)$$

where k_i is the i-th entity vector in K with the dimension $1 * m$ and R is an $n * m$ matrix which holds n vectors of the pre-defined relations, given m the embedding size of both entities and relations in KG. r_i is regarded as the relation vector between q and k_i, and $\hat{r}_i$ is an $1 * n$ vector, measuring the similarities between r_i and the predefined n relations. W_R is an $1 * n$ projection vector, where the j-th element suggests how much attention we need to pay to k_i if the j-th pre-defined relation holds between q and k_i. If we treat $\hat{r}_i$ as the unnormalized probability distribution over the n relations, $Rel(q, k_i)$ can then be considered as the expected attention we need to pay to k_i with regard to q.

Let's consider the sequence $(t_1, t_2, \dots, t_N)$ with a fact triplet $(\mathbb{E}_h, \mathbb{R}_k, \mathbb{E}_t)$, where the subject entity $\mathbb{E}_h$ is represented by $(t_p, t_{p+1}, \dots, t_{p+l1})$, and the object entity $\mathbb{E}_t$ is represented by $(t_q, t_{q+1}, \dots, t_{q+l2})$. Since the word embeddings $\{w_i^{entity}\}$ for $\mathbb{E}_h$ given by GERU are close to the entity embedding e_h, and $\{w_j^{entity}\}$ for $\mathbb{E}_t$ are close to e_t, Equation (14) probably holds for w_i^{entity} and w_j^{entity}:

$$w_i^{entity} - r_k \approx w_j^{entity} \qquad (19)$$

for $i = p \dots p + l1$ and $j = q \dots q + l2$.

Then $attention^{rel}$ can be applied upon words through w_i^{entity}. In this way, we provide one more embedding vector $w_i^{relation}$ for token t_i to model the relation context in the sequence $(t_1, t_2, \dots, t_N)$ as Equation (20)

$$w_i^{relation} = attention^{rel}(w_i^{entity}, X^E, X^L W^{Vr}) \qquad (20)$$

where w_i^{entity} is the output of GERU, X^E is the packed $\{w_i^{entity}\}$ for all the tokens, X^L is the packed $\{w_i^{language}\}$ which is the output from a typical language model, W^{Vr} is a projection matrix.

In the above formulations, R is a collection of all the relation vectors provided by a pre-trained KG embeddings, W_R and W^{Vr} are task-dependent parameters to be learnt.

2.3 KG-Augmented Word Representation

Given the N-token sequence $(t_1, t_2, \dots, t_N)$, we can get the embedding representation $w_i^{language}$ for t_i from the language aspect, through typical language models such as Self-Attentions (Lin et al., 2017) or Recurrent Neural Networks (Peters et al., 2018).

As described in previous sections, $w_i^{relation}$ and w_i^{entity} (the importance of which are evaluated through an ablation study in the experiment section) are proposed to encode the related knowledge information. We then concatenate the above 3 embedding vectors to get a new representation vector w_i^{KAWR} (as Equation (21)), which we call KG-Augmented Word Representation or KAWR as illustrated in Figure 2.

$$w_i^{KAWR} = (w_i^{language}, w_i^{relation}, w_i^{entity}) \qquad (21)$$

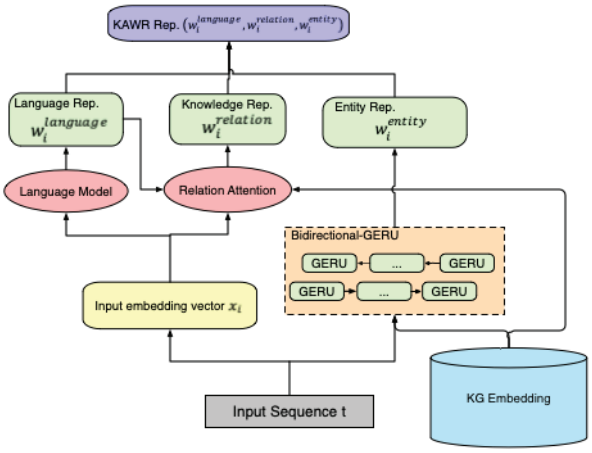
KAWR Rep. $(w_i^{language}, w_i^{relation}, w_i^{entity})$
Language Rep. $w_i^{language}$
Knowledge Rep. $w_i^{relation}$
Entity Rep. w_i^{entity}
Language Model
Relation Attention
Bidirectional-GERU
GERU
...
GERU
GERU
...
GERU
Input embedding vector x_i
Input Sequence t
KG Embedding

Figure 2 Model Architecture of KAWR.

KAWR can work with different kinds of language models. If $w_i^{language}$ is generated by a simple context-independent language model such as word vectors (Mikolov et al., 2013), we simply set $w_i^{language} = x_i$, where x_i is the input embedding vector that can be pre-trained on large text corpus. If $w_i^{language}$ is based on context-aware representation, such as ELMo or BERT, we can put relation attention not only over the input sequence $\{x_i\}$, but also over the output sequences of the middle layers, to provide multiple attention heads, which are then concatenated together as $w_i^{relation}$.

2.4 Model Training

As mentioned in sections 2.1 and 2.2, the entity representation w_i^{entity} can be pre-trained. And the language representation $w_i^{language}$ can either be pre-trained (in case we use

BERT or simple word vectors), or be trained along with the down streaming task (in case we use ELMo). The parameters W_R and W^{Vr} in knowledge representation $w_i^{relation}$ are task-dependent, and need to be trained along with the down streaming task.

To pre-train the parameters in w_i^{entity}, we propose a multi-task learning process, where 3 tagging tasks (i.e. Named Entity Recognition, Chunking, POS-tagging) are trained by sharing the parameters in w_i^{entity}. The network structure is illustrated in Figure 3. The classifier for each task is a one-layer fully-connected network with softmax over the classes. The classification layers of the tasks are completely independent to each other without any connections.

The training process follows the protocol proposed in (Søgaard and Goldberg, 2016). Each time, we randomly choose a task, followed by a mini-batch of training samples. Then we predict the task label, compute loss with respect to the true label, and update the model parameters accordingly.

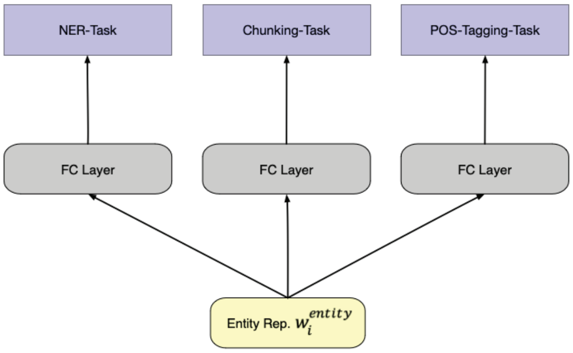

Figure 3 Multi-Task Learning Network for pre-training.

3. Experiments and Results

In our experiments, we build KAWR based on the pre-trained BERT and a KG embedding model, and compare KAWR with the original BERT on 5 datasets covering various domains.

3.1 Pre-trained Knowledge Graph Embeddings

Wikidata, hosted by Wikimeda Foundation, is a free and open knowledge database that can be edited collaboratively and used by anyone under a public domain license. With growing data donations such as migrating of Freebase by Google, Wikidata now covers tens of millions of entities with descriptions and statements in various domains, and is becoming an important source to wiki-pages, such as Wikipedia, Wikivoyage and Wikiquotes. On the other hand, despite the graph embedding techniques mentioned in section 1.1, training embedding models on such a big graph remains challenging work. To solve this problem, Facebook recently proposed a distributed system which is called PyTorch-BigGraph(PBG), for learning embeddings of extremely big graphs (Lerer et al., 2019). They also provide a pre-trained model (which is trained on Wikidata by using PBG) for download.

The model contains 78 million entities, 4131 types of relations, and 364,672,248 fact triplets extracted from the statement. To reduce the computational complexity of our experiments, we filter those entities and relations which are irrelevant to our training data, and keep the remaining 2.4 million entities with 2.3 thousand relations and 9,563,453 fact triplets during our experiments. Since the filtered entities are irrelevant and thus will never be retrieved, this filtering pre-process does not affect the results.

3.2 Experimental Setup

Given the input token sequence $(t_1, t_2, \dots, t_N)$, and the corresponding output sequence of BERT $(w_1^{BERT}, w_2^{BERT}, \dots, w_N^{BERT})$, we build our KAWR as the Equation(21) where

$$w_i^{language} = w_i^{BERT} \tag{22}$$

We then feed w_i^{KAWR} and w_i^{BERT} into down-streaming classification layers on the target tasks for fine-tuning.

All experiments were carried on PowerEdge C4130 with Tesla P40 GPU with 20 GB of memory. The models are trained using AdamWeightDecayOptimizer which is based on stochastic gradient descent, and the hyper parameters are listed in Table 1. The training process are implemented using TensorFlow. And all the numbers reported in the following section are averaged over 5 random restarts.

Hyper-Parameter	Value
Batch-size	16
Learning rate	2e-5
KG-embedding Dimension	200
Word embedding Dimension	768
GERU hidden status Dimension	200

Table 1 Hyper-parameters we used in our experiments.

3.3 Results and Analysis

We compare KAWR and BERT with two different classification layers. One is a simple classification layer (an FC layer) on each token, where the predictions are made not conditioned on the surrounding predictions (i.e., no LSTM or CRF layers). And another is a Bi-LSTM-CRF network (which is reported as the state-of-the-art technique on the tagging tasks (Huang et al, 2015)).

We report F1 scores with precisions and recalls of all the 4 models (BERT-FC, KAWR-FC, BERT-Bi-LSTM-CRF, KAWR-Bi-LSTM-CRF) on 5 different data sets covering various domains, such as newswire (CoNLL2003), molecular biology (Genia), biomedical (NCBI), financial (SEC) and the noisy user-generated text (WNUT16). The specifications of the data sets are listed in Table 2. And the results are presented in Table 3.

KAWR outperforms the original BERT on all of the experiments, by +0.46~+2.07 with the FC classifier, and by +0.21~+2.86 with the Bi-LSTM-CRF classifier. Interestingly, KAWR-FC can even give comparable results with BERT-Bi-LSTM-CRF in most cases. We also notice that in the case of SEC and WNUT16 where training sizes are small, KAWR wins by a larger margin, +1.97 and +2.07 respectively, which suggests KAWR relies less on the training sizes and holds better generalization than BERT due to the introduction of knowledge information. For example, the improvements on SEC are mainly reported on "LOC (F1 66.67 vs 63.41) and "ORG"(F1 47.17 vs 42.59) thanks to the entities defined in knowledge graph.

Dataset	No. of classes	Training size	Testing size
CoNLL2003	4	14041	3453
Genia	5	18546	3856
NCBI	4	6102	1014
SEC	4	1169	306
WNUT16	10	1900	254

Table 2 Dataset Specifications.

We did a T-test on F1 scores of 10 random restarts for BERT-FC and KAWR-FC on CoNLL2003, and got p-value 0.023 which suggests the significance. Learning curves of different models are illustrated in Figure 4, where we can see KAWR performs consistently better than BERT (especially in the early stage when the training set is small).

A number of good cases of KAWR that fail in BERT are listed in Table 4, where we can see that by introducing the entities and relations, KAWR can make the right predictions in those cases where linguistic context is insufficient. In the first case (from soccer news in dataset CoNLL2003), the baseline model recognizes the word "ARSENAL" as a PER entity (probably due to the linguistic context "SAVE") and our model gives the right prediction as an ORG entity (probably due to the knowledge triplet: ("Vieria", "Arsenal", "member of sports team")).

In case 2 (from the user-generated twitter text WNUT16) where the text is too short to provide enough linguistic contextual information, BERT cannot recognize the entity "Marvin Gaye" while KAWR successfully predicts the entity as Person because w_i^{KAWR} for the words "Marvin" and "Gaye" carry the entity information from KG.

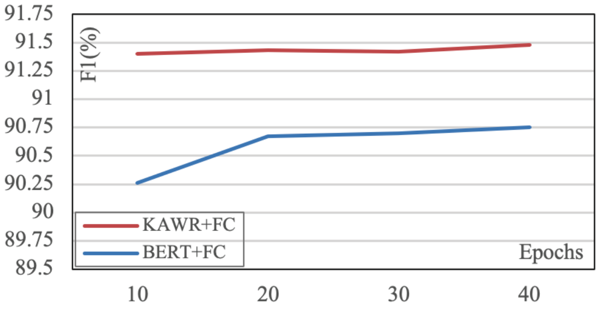

Figure 4 Learning curve on CoNLL2003.

In case 3 (from domain specific text Genia), the baseline model predicts the "MnSOD" as O because it has never seen the word in the training corpus, while KAWR makes the right prediction as B-protein thanks to the entity information from KG.

Task		BERT-FC	KAWR-FC	BERT+BiLSTM+CRF	KAWR+BiLSTM+CRF
CoNLL2003	F1	90.89	**91.78** (+0.89)	91.56	**91.80** (+0.24)
	Prec.	90.08	91.10	90.99	91.40
	Recall	91.71	92.47	92.13	92.20
Genia	F1	72.22	**72.68** (+0.46)	72.48	**72.69** (+0.21)
	Prec.	67.82	68.40	68.13	68.64
	Recall	77.22	77.52	77.42	77.25
NCBI	F1	78.40	**79.43** (+1.01)	78.78	**80.55** (+1.77)
	Prec.	75.56	77.13	77.21	78.21
	Recall	81.46	81.88	80.42	83.02
SEC	F1	79.03	**81.1** (+2.07)	80.22	**83.08** (+2.86)
	Prec.	76.73	77.27	76.95	80.95
	Recall	81.47	85.33	83.78	85.33
WNUT16	F1	56.94	**58.91** (+1.97)	59.65	**61.78** (+0.74)
	Prec.	55.56	58.70	57.43	65.57
	Recall	58.39	59.12	62.04	58.39

Table 3 Overall performance on different data sets.

No	Cases	Task	Model	Predictions
1	SOCCER - VIEIRA(**B-PER**) SAVES ARSENAL(B-ORG) WITH LAST-MINUTE EQUALISER	CoNLL2003	BERT	VIEIRA (**B-PER**) ARSENAL (**B-PER**)
			KAWR	VIEIRA (**B-PER**) ARSENAL (**B-ORG**)
2	#BlackHoliday Marvin(**B-person**) Gaye(**I-person**) Day	WNUT16	BERT	Marvin (**B-other**) Gaye (**I-other**)
			KAWR	Marvin (**B-person**) Gaye (**I-person**)
3	ethanol increased HepG2 cell MnSOD(**B-protein**) activity...	Genia	BERT	MnSOD (**O**)
			KAWR	MnSOD (**B-protein**)
4	…contributions made by the investor and Ashford (**ORG**)…	SEC	BERT	Ashford (**PER**)
			KAWR	Ashford (**ORG**)

Table 4 Typical Case Comparison on NER tasks.

In the last case (from SEC dataset), KAWR recognizes "Ashford" as "ORG" correctly because it is an organization-entity in KG while BERT gives a wrong prediction "PER".

Models	F1 score
BERT	80.22
BERT+EntityEmbedding (without GERU)	80.59
BERT+EntityEmbedding (with GERU)	82.58
BERT+RelationEmbedding	81.09
KAWR	83.08

Table 5 Ablation Study on SEC

Task		BERT-FC	KAWR-FC
CoNLL2003	F1	90.13	**90.78** (+0.65)
	Prec.	90.09	90.99
	Recall	90.16	90.56
Genia	F1	70.63	**71.12** (+0.49)
	Prec.	67.40	68.04
	Recall	74.19	74.48
NCBI	F1	70.95	**72.71** (+1.76)
	Prec.	69.01	71.53
	Recall	73.00	73.93
SEC	F1	49.49	**58.88** (+9.39)
	Prec.	47.57	56.86
	Recall	51.58	61.05
WNUT16	F1	52.97	**55.66** (+2.69)
	Prec.	55.24	60.20
	Recall	50.88	51.75

Table 6 Performances on new entities.

We did ablation study on SEC to evaluate the importance of the proposed entity and relation embeddings in KAWR. The results are shown in Table 5. The Entity-embedding (with GERU) contributes stronger than relation-embeddings do. And the effectiveness of GERU has also been demonstrated. The second model (BERT+EntityEmbedding without GERU) simply averages the related entity embeddings and gets less F1 scores compared with GERU version.

To further investigate the capabilities of generalization (which is important for domain-specific tasks where training data are often insufficient) for both representations, we evaluate their performances on a subset of testing dataset that contains only the entities which are unseen in the train data. The results are reported in Table 6.

Not surprisingly, KAWR shows even more significant improvement over BERT on the sub test sets than on the full test sets for most cases. The reason is that the representation from the entity side w_i^{entity} carries the information of a set of related entities, which can be used to give a hint on a new entity that shares the same word.

We also compare KAWR with BERT on different languages, such as German from CoNLL2003, Spanish and Dutch from CoNLL2002. The experiments show consistent results (Table 7) with that on English, where KAWR gives higher F1 scores.

Task		BERT-FC	KAWR-FC
German (CoNLL2003)	F1	87.33	**87.45**(+0.12)
	Prec.	87.09	87.63
	Recall	87.57	87.28
Spanish (CoNLL2002)	F1	87.56	**87.81**(+0.25)
	Prec.	87.25	87.50
	Recall	87.87	88.12
Dutch (CoNLL2002)	F1	90.11	**91.23**(+1.12)
	Prec.	90.32	91.31
	Recall	89.90	91.14

Table 7 Performances on different languages.

4. Conclusion and Future Work

In this paper we proposed a new word representation KAWR for named entity recognition. KAWR is an augmented version of existing language representations by encoding entity information (from an external knowledge base) through a new gated recurrent unit GERU and by modeling the relation context between entities through a new attention function $attention^{rel}$. Our experiments show that KAWR outperforms BERT on 5 different data sets from different domains, especially in the cases where the training sets are small.

Since the knowledge information carried by KAWR may also facilitate other NLP tasks like text classifying, machine translation and question answering, we think that KAWR could be a general word representation. Therefore, we will explore the potentials of KAWR on other NLP tasks as future work.

References

Adam Lerer, Ledell Wu, Jiajun Shen, Timothee Lacroix, Luca Wehrstedt, Abhijit Bose, Alex Peysakhovich. 2019. PyTorch-BigGraph: A Large-scale Graph Embedding System. In *Proceedings of The Conference on Systems and Machine Learning 2019 (SysML 2019).*

Alexis Conneau, Holger Schwenk, Loïc Barrault and Yann Lecun. 2017. Very Deep Convolutional Networks for Text Classification. In *Proceedings of European Chapter of the Association for Computational Linguistics 2017 (EACL 2017)*, Valencia, Spain

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 231-235

Annervaz K M, Somnath Basu Roy Chowdhury and Ambedkar Dukkipati. 2018. Learning beyond Datasets: Knowledge Graph Augmented Neural Networks for Natural Language Processing. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2018)*, Volume 1, pages 313-322

Antoine Bordes, Nicolas Usunier, Alberto Garcia- Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi- relational data. In *Proceedings of Advances in neural information processing systems 2013 (NIPS 2013)*. pages 2787–2795.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser and Illia Polosukhin. 2017. Attention is All you Need. In *Proceedings of Advances in Neural Information Processing Systems 30 (NIPS 2017)*

Baoxu Shi and Tim Weninger. 2017. Proje: Embed- ding projection for knowledge graph completion. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence 2017 (AAAI 2017)*. pages 1236–1242.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss and Andrew McCallum. 2018. Linguistically-Informed Self-Attention for Semantic Role Labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, pages 5027-5038

Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge graph embedding via dy- namic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*. pages 687–696.

Han Xiao, Minlie Huang, Yu Hao, and Xiaoyan Zhu. 2015. Transg: A generative mixture model for knowledge graph embedding. *arXiv preprint*. arXiv:1509.05488.

Huaping Zhong, Jianwen Zhang, Zhen Wang, Hai Wan, and Zheng Chen. 2015. Aligning knowledge and text embeddings by entity descriptions. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*. pages 267–272.

Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova. 2018. Bert: Pretraining of deep bidirectional transformers for language understanding. *arXiv preprint*, arXiv:1810.04805.

Jason P.C. Chiu and Eric Nichols. 2016. Named Entity Recognition with Bidirectional LSTM-CNNs. In *Transactions of the Association for Computational Linguistics*, Volume 4, 2016. pages 357-370

Ji Xin, Yankai Lin, Zhiyuan Liu and Maosong Sun. 2018. Improving Neural Fine-Grained Entity Typing with Knowledge Attention. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence 2018 (AAAI 2018)*

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint*, arXiv: 1406.1078.

Lebret Rémi, Collobert and Ronan 2013. Word Emdeddings through Hellinger PCA. In *Proceedings of Conference of the European Chapter of the Association for Computational Linguistics 2013 (EACL 2013)*

Levy Omer and Goldberg Yoav. 2014. Neural Word Embedding as Implicit Matrix Factorization. In *Proceedings of Advances in Neural Information Processing System 27 (NIPS 2014)*.

Martin Sundermeyer, Ralf Schlüter and Hermann Ney. 2012. LSTM neural networks for language modeling. In *Proceedings of 13th Annual Conference of the International Speech Communication Association (INTERSPEECH-2012)*, pages 194-197.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word rep- resentations. In *NAACL* 2018.

Pengfei Liu, Xipeng Qiu and Xuanjing Huang. 2016. Recurrent Neural Network for Text Classification with Multi-Task Learning. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI 2016)*, pages 2874-2879

Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. 2016. Representation learning of knowledge graphs with entity descriptions. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence 2016 (AAAI 2016)*. pages 2659–2665.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of Advances in Neural Information Processing Systems 26 (NIPS 2013)*, pages 3111–3119.

Zhiheng Huang, Wei Xu and Kai Yu. 2015. Bidirectional LSTM-CRF Models for Sequence Tagging. *arXiv preprint*, arXiv:1508.01991.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou and Yoshua Bengio. 2017. A Structured Self-attentive Sentence Embedding. In *Proceedings of 5th International Conference on Learning Representations (ICLR 2017)*