

A Knowledge-Enriched and Span-Based Network for Joint Entity and Relation Extraction

Kun Ding¹, Shanshan Liu¹, Yuhao Zhang², Hui Zhang¹, Xiaoxiong Zhang^{1,*},
Tongtong Wu^{2,3} and Xiaolei Zhou¹

¹The Sixty-Third Research Institute, National University of Defense Technology, Nanjing, 210007, China

²School of Computer Science and Technology, Southeast University, Nanjing, 211189, China

³Faculty of Information Technology, Monash University, Melbourne, 3800, Australia

*Corresponding Author: Xiaoxiong Zhang. Email: zxxandxx@163.com

Received: 29 December 2020; Accepted: 29 January 2021

Abstract: The joint extraction of entities and their relations from certain texts plays a significant role in most natural language processes. For entity and relation extraction in a specific domain, we propose a hybrid neural framework consisting of two parts: a span-based model and a graph-based model. The span-based model can tackle overlapping problems compared with BILOU methods, whereas the graph-based model treats relation prediction as graph classification. Our main contribution is to incorporate external lexical and syntactic knowledge of a specific domain, such as domain dictionaries and dependency structures from texts, into end-to-end neural models. We conducted extensive experiments on a Chinese military entity and relation extraction corpus. The results show that the proposed framework outperforms the baselines with better performance in terms of entity and relation prediction. The proposed method provides insight into problems with the joint extraction of entities and their relations.

Keywords: Entity recognition; relation extraction; dependency parsing

1 Introduction

The extraction of entities and their interrelations is an essential issue in understanding text corpora. Determining the token spans in texts that compose entities and assigning types to these spans (i.e., named entity recognition, NER) [1] as well as assigning relations between each pair of entity mentions (i.e., relation classification, RC) [2,3] are critical steps in obtaining knowledge from texts for further possible applications, such as knowledge graph construction [4], knowledge-based question answering [5], and sentiment analysis [6].

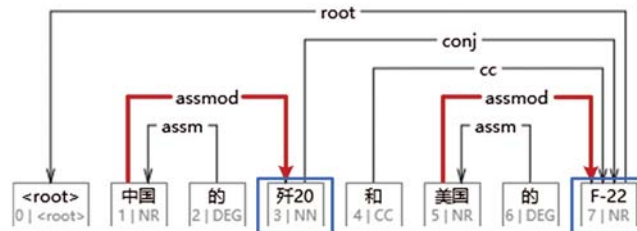
Currently, pre-trained language models, such as BERT [7], have achieved outstanding performance in various natural language processing (NLP) tasks, including entity and relation extraction. However, the performances of these BERT-based models on Chinese specific-domain corpora are not as effective as on English datasets. We argue that this is mainly due to two reasons. First, terminology is common in a specific domain, for example, weapons and equipment



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

nomenclature in the military domain, bringing about the possible problem that important entities with terminology that have never appeared in the open domain are understood as unregistered words and are thus difficult to identify. Second, when used with Chinese corpora, BERT generates embeddings at the character level. However, Chinese words contain more semantic information than characters. As a result, it is difficult for BERT-based models to extract dependency relations between Chinese words, which is important in relation extraction. As in the example shown in Fig. 1, dependency parsing can mine the two pairs of relationships, ‘中国’ (Chinese) and ‘歼’ (J-20), ‘美国’ (American) and F-22, which contributes to relation extraction.

To address the above-mentioned issues, we propose a novel architecture for joint entity and relation extraction. The key insight of our proposed model is to leverage external lexical and syntactic knowledge to overcome the limitations of BERT-based models encountered during Chinese specific-domain joint extraction. Specifically, lexical knowledge refers to domain dictionaries, such as weapons and equipment nomenclature, which helps to improve NER performance on entities with terminology. Moreover, syntactic knowledge refers to the dependency structure discussed in the current text. As the result of dependency parsing can be transformed into a tree structure, we further adopt the graph-based model to manage the dependency tree and capture the interaction between relations, which compensates for the lack of existing BERT-based models to extract dependencies and, in turn, contributes to relation extraction.



Original sentence: 中国的歼20和美国的F-22

Chinese J-20 and American F-22

Figure 1: Examples of dependency tree in the Chinese specific corpus. The dependency parsing result is generated by HanLP [8]. Arrows in red present dependencies that can contribute to relation extraction. Blue blocks show that domain-dictionaries can help recognize entities with terminology

Our proposed model incorporates the joint modeling of span-based and graph-based components by taking advantage of two different structures. More specifically, span-based components feature entity recognition and relation classification by making full use of a localized marker-free context representation [9]. As an extension of the previous work in [9], we incorporate a graph-based component in our model using graph neural networks (GNNs) in the relation classification component. We allow the simultaneous classification of entities and relations with higher accuracy. The main contributions of this study are summarized as follows:

- For the extraction of entities and relations in specific domains, a hybrid framework based on a knowledge-enriched and span-based network is proposed.
- The dependency structure is incorporated in our model, which can leverage external lexical and syntactic knowledge effectively in Chinese specific-domain joint extraction.

- Comparative experimental results demonstrated that our model outperforms the state-of-the-art model, and achieves an absolute 4.13% improvement in the F1 score in relation extraction.

The remainder of this paper is structured into five sections. Related works on entity and relation extraction are presented in Section 2. Section 3 defines the problem and provides details on the architecture of our proposed model for the joint extraction of entities and relations. Our experiments are presented in Section 4, and comparative experiments are implemented in Section 5. Finally, the conclusions are presented in Section 6.

2 Related Work

Most systems adopt a two-stage pipeline framework for the extraction of entities and relations. First, the entities in a given sentence are recognized using NER. Then, certain classification models are used to test each entity pair [10–12]. This method is easy to implement, and each component can be more flexible, but it lacks the interaction between different tasks, leading to error propagation [13,14].

Unlike pipelined methods, the joint extraction framework focuses on extracting entities together with relations using a single model. The advantages lie in joint entity and relation extraction to capture the inherent linguistic dependencies between relations and entity arguments to resolve error propagation. Most initial joint models are feature-based structured systems that require complicated feature engineering. For example, Roth et al. [15] investigated a joint inference framework based on integer linear programming to extract entities and relations. Li et al. [16] proposed a transition-based model for entity recognition and relation classification simultaneously. Ren et al. [17] investigated a new domain-independent framework that focused on a data-driven text segmentation algorithm for the extraction of entities and relations.

For the sake of reducing the manual work in the extraction process, models with different neural networks have been proposed with the characteristics of automatic feature extraction. These models adopt low-dimensional dense embedding to denote features. Gupta et al. [18] proposed a table-filling multi-task recurrent neural network (RNN) for the joint extraction of entities and relations. Adel et al. [19] introduced globally normalized convolutional neural networks and relation extraction. Katiyar et al. [20] presented a novel attention-based long short-term memory (LSTM) network for the joint extraction of entity mentions and relations.

Recently, Hong et al. [21] presented an end-to-end neural model based on graph convolutional networks for jointly extracting entities and relations. Kok et al. [22] provided a brief introduction to named entity (NE) extraction experiments performed on datasets of open-source intelligence and post-training mission analytics. Recently, Wang et al. [23] investigated a relation extraction method combining a bidirectional LSTM (Bi-LSTM) neural network, character embedding, and attention mechanism to solve a military named entity relation extraction. Takanobu et al. [24] later proposed a hierarchical reinforcement learning (HRL) framework for the sake of enhancing the interaction between entity and relation types. Trisedya et al. [25] adopted an N-gram attention mechanism with an encoder-decoder model for the completion of knowledge bases using distant supervised data.

Despite the great efforts in this field, they still leave an open question: how to efficiently capture the semantic information in texts, especially in a Chinese specific domain. In this study, a novel framework with a knowledge-enriched and span-based network is proposed for jointly

extracting entities and their relations simultaneously. Compared with other state-of-the-art models, this model improves the F1 score of entity and relation recognition in a specific domain.

3 Methodology

This section provides details on the implementation of our knowledge-enriched and span-based BERT (KSBERT) network, and its overall framework is presented in Fig. 2.

3.1 Problem Definition

This paper focuses on extracting entities and relations jointly; the input is a sentence S of N tokens, $S = \{t_1, t_2, \dots, t_N\}$ and the output contains two structures: entity type E and relations R . Specifically, we considered two subtasks. First, *entity recognition* extracts all possible entity spans $s_i = \{t_i, t_{i+1}, \dots, t_{i+k}\}$ from sentence S and predicts its best entity type e_i . Second, *relation classification* focuses on predicting the relation type r_{ij} between each set of two entities $\{s_i, s_j\}$.

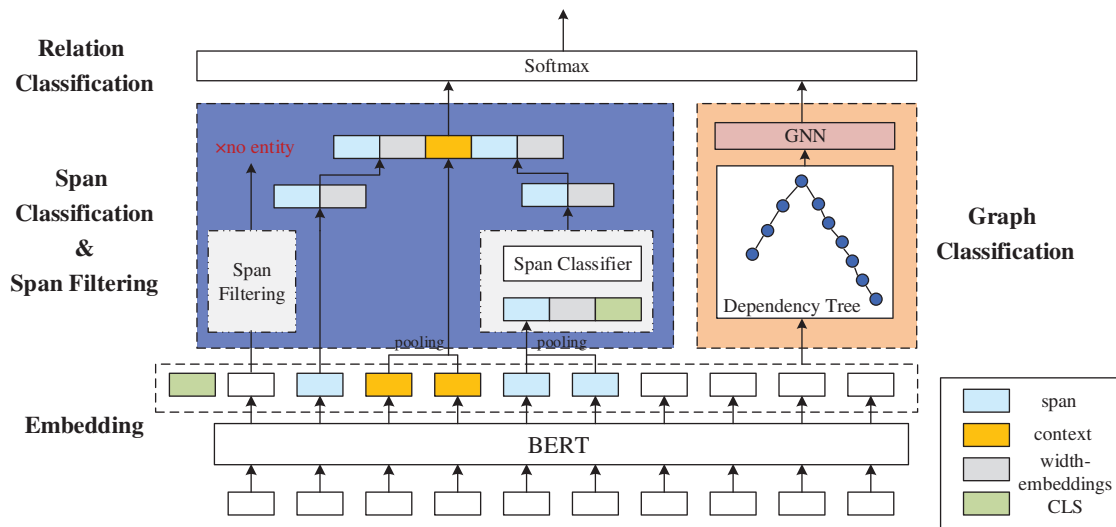


Figure 2: The framework of our KSBERT network

3.2 KSBERT

We developed a knowledge-enriched and span-based network, KSBERT, to recognize entities and classify relations jointly from the given sentences. Because of the outstanding performance of BERT [7] in natural language processing tasks, the pre-trained BERT model is applied to encode each character into embeddings with a special classifier token extracting information from the whole sentence. Then, the character embeddings of the sentence are fed into two models: the span-based model and the graph-based model. The span-based model takes the selected candidate span as input and judges whether or not the span is an entity; if it is, the model predicts its type; otherwise, it filters the span. The graph-based model includes knowledge of a specific domain, the core of which is GNNs. Fed with BERT embedding, the graph-based model first changes the dependency parsing tree of the sentence into an adjacent matrix, taking embeddings as node labels, dependency relations between words as edge labels, and relation types of the sentence as graph labels. The adjacent matrix is then fed into the GNNs to predict the

classes of the graph, which are the relation types of the given sentences. Finally, three losses, including the entity recognition loss and relation classification loss of the span-based model and the graph classification loss of the graph-based model, are trained jointly to predict the entity and relation types.

Next, we introduce the main components of our KSBERT in detail.

Embeddings By taking advantage of a multi-layer bidirectional transformer architecture, BERT can encode both left-to-right and right-to-left contexts for word representations, with semantic and syntactic information extracted. Given a sentence of N words, the BERT encoder generates an embedding sequence of length $N + 1$, $(w_{cls}, w_1, w_2, \dots, w_N)$, where w_{cls} denotes the special classification embedding capturing information of the whole sentence.

Span-based Model As presented in Fig. 2, the block in blue is the span-based model, containing two elements: span classification and span filtering. In contrast to the previous model [9], which takes a span of arbitrary length as input, our model proposes a novel negative sampling method to generate candidate spans for a specific domain. We first build a set C containing as many entities in the given dataset as possible. We then segment sentences in the dataset using the Jieba toolkit, filtering out words other than nouns. The similarities between nouns and entities in set C are computed, and the top similar Ne ones are selected as negative samples.

Using the above method, we obtain the candidate span $\{w_i, w_{i+1}, \dots, w_{i+k}\}$ and feed it to the span-based model. The BERT embeddings are then combined using max pooling, denoted as $f(w_i, w_{i+1}, \dots, w_{i+k})$. The output is concatenated with the width-embedding w_{width} learned by back-propagation and special classification embedding w_{cls} .

$$x_{s_i} = [f(w_i, w_{i+1}, \dots, w_{i+k}); w_{width}; w_{cls}] \quad (1)$$

Then, the concatenation is fed into a softmax function to obtain the logtis of entity type y^{e_i} .

$$y^{e_i} = \text{softmax}(W_i \cdot x_{s_i} + b_i) \quad (2)$$

Moreover, our span-based model includes a span-filtering function. In Eq. (2), if a candidate span is not an entity, our model classifies it as none type.

The embeddings of entity spans s_i and s_j extracted by the span classifier are then combined with context c_{s_i, s_j} , which ranges from the end of entity s_i to the beginning of entity s_j , to obtain the relation representation.

$$x_{r_{i,j}} = [x_{s_i}; c_{s_i, s_j}; x_{s_j}] \quad (3)$$

$$x_{r_{j,i}} = [x_{s_j}; c_{s_i, s_j}; x_{s_i}] \quad (4)$$

Generally, as relations are asymmetric, two relation representations exist between two entities.

Relation representations are fed to a fully connected layer and activated by a sigmoid function to perform relation classification.

$$y^{r_{i,j}} = \sigma(W_{i,j} \cdot x_{r_{i,j}} + b_{i,j}) \quad (5)$$

$$y^{r_{j,i}} = \sigma(W_{j,i} \cdot x_{r_{j,i}} + b_{j,i}) \quad (6)$$

Graph-based Model As the previous models cannot address challenges in specific domains, we add the graph-based model to our KSBERT to introduce external knowledge, particularly dependencies, which play an important role in capturing relations among words. In the graph-based

model (the orange block in Fig. 2), we regard the relation classification task as graph classification of dependency trees; that is, given a set of graphs (dependency trees) $\{G_1, G_2, \dots, G_n\}$ and their labels (relation types) $\{y_1, y_2, \dots, y_n\}$, we aim to learn a representation vector \mathbf{h}_G to predict the label of the graph. Given a sentence S_i , we use HanLP [8] to obtain its dependency tree and then change the tree into an adjacent matrix to obtain the input of graph G_i . The graph is then fed into Graph Isomorphism Network (Gin) implemented by CogDL. Gin is a variant of GNN proposed by Xu et al. [26] and is good at representing different multisets. The representation vector of the entire graph \mathbf{h}_G is learned by an aggregating feature vector of node neighbors recursively.

$$h_{G_i} = GNN(G_i) \quad (7)$$

Similar to the span classifier, representation vector \mathbf{h}_{G_i} is fed into a softmax layer:

$$y_{G_i} = \text{softmax}(W_{G_i} \cdot h_{G_i} + b_{G_i}) \quad (8)$$

Final Prediction The final prediction is a joint training process and the loss function can be given as follows:

$$\gamma = \gamma_e + \gamma_r + f(\cdot) \gamma_g \quad (9)$$

where γ_e denotes the entity classification loss by the span-based part, γ_r refers to the relation classification loss by the span-based model, and γ_g represents the graph classification loss by the graph-based model. $f(\cdot)$ is a linear function.

4 Experiments

This section provides an introduction to our experiments in detail. First, the dataset is introduced in brief. Next, the metrics used in this study are presented. Then, the experimental settings are tabulated. Three baselines are compared with the proposed model to illustrate its superiority. Finally, we analyze the results in detail.

4.1 Dataset

As artificial construction features are insufficient in a specific domain, such as the military field, and Chinese word segmentation errors are inevitable, we focus on extracting entities and relations jointly in the Chinese military field. Owing to the lack of a dataset, we built one by ourselves. After crawling several representative military news sites, we obtained 840,000 articles. Based on the keywords relating to the military, we filtered out articles that were not closely related to the military or from which military relations could not be extracted. Ultimately, 85,000 articles were obtained in total. Because labeling a dataset in a specific domain is more time-consuming and also requires rich domain knowledge, we invited military experts to label the entity location, entity type, and relation type for approximately 300 articles. Fig. 3 presents an example of this human-labeled data. When labeling, each entity span in the given article is assigned a unique ID. Then, experts mark the beginning and end positions of the entity span in the article and judge its entity type. The relation between two entities is also labeled with relation types, head entity ID, and tail entity ID. There are seven entity types—equipment, person, organization, location, military activity, title, and engineering for preparedness against war—as well as three relation types: deploy, have, and locate.

For articles that were not manually labeled, we designed a regular template together with experts to label the dataset automatically, the steps of which are shown in Fig. 4. After analyzing 100 articles randomly selected from the corpus, we designed regular expressions for predefined

entities and relations. Regular expressions were then tested on a human-labeled dataset. If the accuracy of entities and relations exceeded the threshold, the regular template was considered credible and was applied to the automatic data labeling process. In the end, all of the labeled data were randomly split into a training set and a test set, with a ratio of 10:1.

have

Original				
空地-88 (KD-88) 中国 空地-88 (KD-88) 是我国新一代的防区外空地导弹, 由“鹰击”-83K 空舰导弹改进而来。				
KongDi-88(KD-88) China KongDi-88(KD-88) is a new generation of air-to-surface missiles outside the defense zone in our country, improved from the "YingJi"-83K air-to-ship missile.				
Label	ID	Entity span	BeginEnd	Type
	#aa10ebcc9ce548acacf1633d331e9c00	空地-88	0 4	equipment
	#aed1118468ea4dbecb875b922eaa8b9	KD-88	6 10	equipment
	#10a53c2480714a66b2fe46d2defe5a57	中国	12 13	organization
	#b5539bdd8e324c24aa0aff3d8372c540	空地-88	16 20	equipment
	#ca7bf0c6dd52496d89553f6c8fc33262	KD-88	22 26	equipment
	#9fb7575ebd164d188077800e0336a5e5	“鹰击”-83K 空舰导弹	45 56	equipment
#10a53c2480714a66b2fe46d2defe5a57 持有 (have) #aa10ebcc9ce548acacf1633d331e9c00				

Labeling entities and their locations
 Labeling relations

Figure 3: An example of dataset

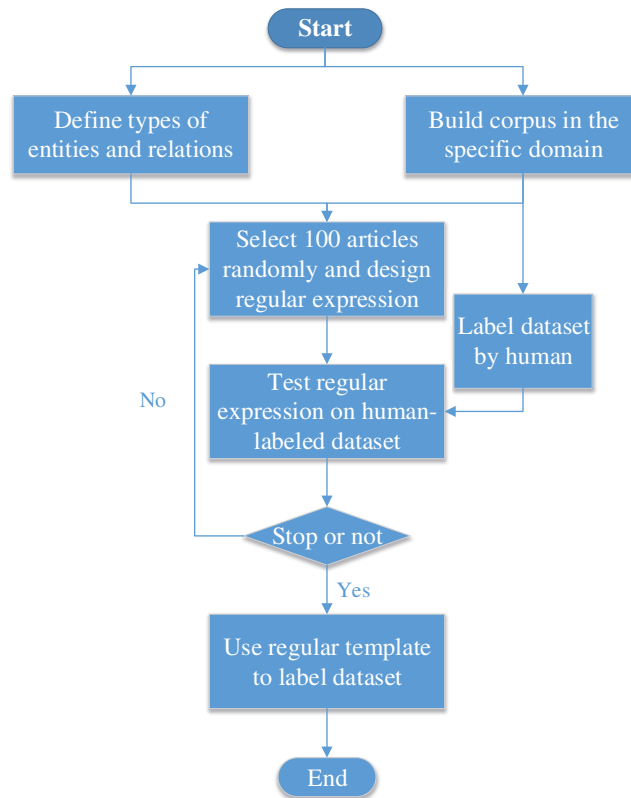


Figure 4: The steps of designing regular templates to label the military corpus

4.2 Metrics

In this study, three commonly used metrics, precision, recall, and F1 score, are adopted to evaluate model performance.

First, with true positive (TP), false positive (FP), true negative (TN), and false negative (FN), precision, recall, and F1 score can be computed as follows:

$$precision = \frac{TP}{TP + FP} \quad (10)$$

$$recall = \frac{TP}{TP + FN} \quad (11)$$

$$F1 = \frac{2 \times P \times R}{P + R} \quad (12)$$

where P stands for precision and R for recall. Note that F1 is the harmonic average of the other two metrics.

In our experiments, a relation is considered correct only if the relation type and the two related entities are correctly predicted.

4.3 Experiment Settings

The parameter settings of the training process are listed in [Tab. 1](#), along with the parameter names and their illustrations. The determination of these parameters was not a major concern in this study. In realistic applications, however, these values can be estimated based on historical data, expert elicitation, or experiments.

4.4 Baselines

To illustrate the superiority of our proposed method, we compared the model to three other competitive baselines, as follows:

SpERT: A span-based model for the joint extraction of entities and relations proposed by Eberts et al. [9]. In contrast to other BILOU-based models, SpERT can search over all spans in given sentences with span filtering and localized context representation and can identify overlapping entities efficiently. Note that BILOU is a common scheme for tag tokens in NLP.

NN_GS: A joint model extracting entities and relations based on a novel tagging scheme proposed by Zheng et al. [27]. This model can convert the joint extraction task into a tagging problem. Thus, neural networks can be easily used to model the joint extraction task without complicated feature engineering.

DYGIE: A joint framework proposed by David et al. [28]. This model extracts entities and relations by enumerating, refining, and scoring text spans designed to capture local (within sentence) and global (cross-sentence) contexts.

4.5 Results

[Tab. 2](#) presents the evaluation results of the models for our dataset. The first column is the index, and the second column is the model name. The third column presents the accuracy scores of graph-based models when performing graph classification. The fourth and fifth columns show the results of entity prediction and relation classification, respectively.

Table 1: Parameters settings utilized in KSBERT model

Model Name	Parameter name	Illustration	Value
Span-based model	Train_batch_size	The batch size of the training process	2
	Eval_batch_size	The batch size of the evaluation process	1
	Neg_entity_count	The number of negative entities	100
	Neg_relation_count	The number of negative relations	100
	Learning_rate	Learning rate	5e−5
	Lr_warmup	Proportion of total training iterations to warm up in linear increase or decrease schedule	0.1
	Weight_decay	Weight decay to apply in span-based model	0.01
	Max_grad_norm	Maximum gradient norm	1.0
	Rel_filter_threshold	Filter threshold for relations	0.4
	Size_embedding	Dimensionality of size embedding	25
	Prop_drop	Probability of dropout	0.1
	Max_span_size	Maximum size of spans	10
	Sampling_limit	Maximum number of samples batches in queue	100
Graph-based model	Epoch	Number of epochs	100
	Weight_decay	Weight decay to apply in graph-based model	5e−4
	Drop_out	Rate of dropout	0.5
	Pooling	Polling method	Sum
	Learning_rate	Learning rate	0.001
	Gamma	Value of gamma	0.5

Our model was compared with three other models. More specifically, the model in Row 1 is a span-based model without a graph-based component, whereas Rows 2 and 3 refer to two novel hybrid models. As shown in [Tab. 2](#), KSBET performs the best in relation extraction in terms of the F1 score, followed by DYGIE, SpERT, and NN_GS. Although the NN_GS model performs well in entity extraction compared with other models, it cannot improve the relation prediction F1 scores. Comparing the results of Rows 1 and 4, we can see that our KSBERT model, which applies Gin [26] using a graph-based model, performs well in both entity recognition

and relation classification, which indicates that incorporating external knowledge with the graph-based component can contribute to entity and relation extraction.

Table 2: The evaluation results of different models for our dataset

ID	Model	Graph classification	Entity			Relation		
			Precision	Recall	F1	Precision	Recall	F1
1	SpERT	/	92.33	64.22	75.75	74.29	45.5	56.43
2	DYGIE	/	72.49	73.44	72.96	68.34	50.75	58.25
3	NN_GS	/	83.70	87.62	87.66	63.91	49.63	55.87
4	KSBERT (our model)	95.82	84.42	69.22	76.07	68.12	54.50	60.56

5 Discussion

5.1 Ablation Study

To evaluate the performance of the different components of our proposed model, we further conducted ablation studies.

As shown in [Tab. 3](#), the results of our proposed model are presented in the first row, and their ablations are listed below. It is clear that both the graph classification and the domain dictionary components contribute to the model’s performance. More specifically, the graph classification results in a performance drop 0.32% and 4.13% in F1 score regarding entity and relation extraction, respectively. The domain dictionary, however, results in a relatively lower performance drop of 0.28% and 2.56% in F1 score regarding entity and relation extraction, respectively. In other words, graph classification makes a greater contribution to the improvement of performance than does the domain dictionary, but both of these components are indispensable.

Table 3: Ablation analysis results

ID	Model	Graph classification	Entity (F1)	Relation (F1)
1	KSBERT (our model)	95.82	76.07	60.56
2	—graph classification	/	−0.32	−4.13
3	—domain dictionary	−2.57	−0.28	−2.56

5.2 Comparison of Joint Training Methods

To determine the most efficient method to jointly train the span-based and graph-based models, in this section, we conduct a comparative analysis. There are three candidate approaches: adding loss of relation classification in the span-based model γ_r and loss of graph classification γ_g , multiplying γ_r and γ_g , or using a linear function to combine γ_r and γ_r together. The results are shown in [Tab. 4](#); the add method and linear function methods can perform joint training correctly,

while the multiply method cannot. Moreover, with the linear function, the model can obtain higher F1 scores of 76.07 and 60.56 in entity recognition and relation classification, respectively. In other words, the linear function method outperforms the other two methods in terms of the F1 score for both entity and relation extraction. Therefore, the linear function method was adopted in our KSBERT model.

Table 4: The evaluation results of different joint training methods

Methods	Entity			Relation		
	Precision	Recall	F1	Precision	Recall	F1
Add	84.24	67.11	74.71	61.90	52.00	56.52
Multiply	84.94	60.78	70.85	3.24	17.50	5.47
Linear Function	84.42	69.22	76.07	68.12	54.50	60.56

5.3 Comparison of Aggregation of BERT Character Embeddings

A pre-trained BERT encoder can generate only character embeddings in Chinese, but Chinese words may contain more information than characters. To obtain word embeddings from BERT-encoded character embeddings, we compared two methods: sum and average. For each word, the sum method adds all embeddings of characters in the word as word embedding, whereas word embedding in the average method is obtained by dividing the sum value by the number of characters in the word.

Tab. 5 presents the evaluation results of these two different methods. As can be seen, the sum method has a larger precision metric value. However, the average method performs better than the sum method for all other metrics, especially F1 score, indicating its efficiency in fully extracting information of each character and representing the word.

Table 5: The evaluation results of different BERT character embedding aggregation approaches

Methods	Entity			Relation		
	Precision	Recall	F1	Precision	Recall	F1
Sum	89.85	65.89	75.93	67.48	48.25	56.27
Average	84.42	69.22	76.07	68.12	54.50	60.56

6 Conclusion

In this paper, we propose a hybrid framework based on a knowledge-enriched and span-based network for the joint extraction of entities and their relations in a specific domain. With our KSBERT network, dependency relation and domain dictionary, as external lexical and syntactic knowledge, can be incorporated into relation prediction, which is essential for improving performance. Extensive experiments have been conducted on a military entity and relation extraction corpus. The results show that our proposed model outperforms other state-of-the-art approaches

regarding F1 score and may be a more promising approach for future research. It should be further noted that our proposed model can be applied to other domains with slight modifications.

In the future, our model can be easily extended by allowing for richer assumptions. As for future research directions, integrating other knowledge such as pos-tags into our framework can be taken into account. Addressing these and other challenges will contribute to the expansion of this method for dealing with other problems.

Funding Statement: The research was supported by the Jiangsu Province “333” project BRA2020418, the NSFC under Grant Number 71901215, the National University of Defense Technology Research Project ZK20-46, the Outstanding Young Talents Program of National University of Defense Technology, and the National University of Defense Technology Youth Innovation Project.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] J. Li, A. Sun, J. Han and C. Li, “A survey on deep learning for named entity recognition,” *IEEE Transactions on Knowledge and Data Engineering*, arXiv preprint arXiv:1812.09449, 2020.
- [2] Y. Liu, F. Wei, S. Li, H. Ji, M. Zhou *et al.*, “A dependency-based neural network for relation classification,” in *ACL-IJCNLP*, Beijing, China, pp. 285–290, 2015.
- [3] K. Xu, Y. Feng, S. Huang and D. Zhao, “Semantic relation classification via convolutional neural networks with simple negative sampling,” in *EMNLP*, Lisbon, Portugal, pp. 536–540, 2015.
- [4] T. Li, H. Li, S. Zhong, Y. Kang, Y. Zhang *et al.*, “Knowledge graph representation reasoning for recommendation system,” *Journal of New Media*, vol. 2, no. 1, pp. 21–30, 2020.
- [5] B. Zhang, H. W. Wang, L. Q. Jiang, S. H. Yuan and M. Z. Li, “A novel bidirectional LSTM and attention mechanism based neural network for answer selection in community question answering,” *Computers, Materials & Continua*, vol. 62, no. 3, pp. 1273–1288, 2020.
- [6] D. J. Zeng, Y. Dai, F. Li, J. Wang and A. K. Sangaiah, “Aspect based sentiment analysis by a linguistically regularized CNN with gated mechanism,” *Journal of Intelligent & Fuzzy Systems*, vol. 36, no. 5, pp. 3971–3980, 2019.
- [7] J. Devlin, M. W. Chang, K. Lee and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *NAACL HLT*, Minneapolis, MN, United states, pp. 4171–4186, 2019.
- [8] H. He, “HanLP: Han Language Processing,” 2020. [Online]. Available: <https://github.com/hankcs/HanLP>.
- [9] M. Eberts and A. Ulges, “Span-based joint entity and relation extraction with transformer pre-training,” *ECAI*, vol. 325, pp. 2006–2013, 2020.
- [10] D. Zeng, Y. Xiao, J. Wang, Y. Dai and A. K. Sangaiah, “Distant supervised relation extraction with cost-sensitive loss,” *Computers, Materials & Continua*, vol. 60, no. 3, pp. 1251–1261, 2019.
- [11] Z. X. Ye and Z. H. Ling, “Distant supervision relation extraction with intra-bag and inter-bag attentions,” in *NAACL HLT*, Minneapolis, MN, United states, pp. 2810–2819, 2019.
- [12] L. Yin, X. Meng, J. Li and J. Sun, “Relation extraction for massive news texts,” *Computers, Materials & Continua*, vol. 60, no. 1, pp. 275–286, 2019.
- [13] M. Cui, L. Li, Z. Wang and M. You, “A survey on relation extraction,” in *CCKS*, Chengdu, China, pp. 26–29, 2017.
- [14] X. R. Zeng, D. J. Zeng, S. Z. He, K. Liu and J. Zhao, “Extracting relational facts by an end-to-end neural model with copy mechanism,” in *ACL*, Melbourne, VIC, Australia, pp. 506–510, 2018.

- [15] D. Roth and W. Yih, "Global inference for entity and relation identification via a linear programming formulation," in *Introduction to Statistical Relational Learning*. Cambridge, Massachusetts, United States: MIT Press, pp. 553–580, 2007.
- [16] Q. Li and H. Ji, "Incremental joint extraction of entity mentions and relations," in *ACL*, Baltimore, Maryland, USA, pp. 402–412, 2014.
- [17] X. Ren, Z. Wu, W. He, M. Qu, C. R. Voss *et al.*, "Cotype: Joint extraction of typed entities and relations with knowledge bases," in *WWW*, Perth, WA, Australia, pp. 1015–1024, 2017.
- [18] P. Gupta, H. Schutze and B. Andrassy, "Table filling multi-task recurrent neural network for joint entity and relation extraction," in *COLING*, Osaka, Japan, pp. 2537–2547, 2016.
- [19] H. Adel and H. Schutze, "Global normalization of convolutional neural networks for joint entity and relation classification," in *EMNLP*, Copenhagen, Denmark, pp. 1723–1729, 2017.
- [20] A. Katiyar and C. Cardie, "Going out on a limb: Joint extraction of entity mentions and relations without dependency trees," in *ACL*, Vancouver, Canada, pp. 917–928, 2017.
- [21] Y. Hong, Y. Liu, S. Yang, K. Zhang, A. Wen *et al.*, "Improving graph convolutional networks based on relation-aware attention for end-to-end relation extraction," *IEEE Access*, vol. 8, pp. 51315–51323, 2020.
- [22] A. Kok, I. I. Mestric and M. Street, "Named entity extraction in a military context," in *ICMCIS*, Budva, Montenegro, pp. 1–6, 2019.
- [23] X. Wang, R. Yang, Y. Feng, D. Li and J. Hou, "A military named entity relation extraction approach based on deep learning," in *ACAI*, Sanya, China, pp. 1–6, 2018.
- [24] R. Takanobu, T. Y. Zhang, J. X. Liu and M. L. Huang, "A hierarchical framework for relation extraction with reinforcement learning," in *AAAI*, Honolulu, HI, United states, pp. 7072–7079, 2019.
- [25] B. D. Trisedya, W. K. Gerhard, J. Z. Qi and R. Zhang, "Neural relation extraction for knowledge base enrichment," in *ACL*, Florence, Italy, pp. 229–240, 2020.
- [26] K. Xu, W. Hu, J. Leskovec and S. Jegelka, "How powerful are graph neural networks," in *ICLR*, New Orleans, LA, United States, pp. 1–15, 2019.
- [27] S. C. Zheng, F. Wang, H. Y. Bao, Y. X. Hao, P. Zhou *et al.*, "Joint extraction of entities and relations based on a novel tagging scheme," in *ACL*, Vancouver, Canada, pp. 1227–1236, 2017.
- [28] W. David, W. Ulme, L. Yi and H. Hannaneh, "Entity, relation, and event extraction with contextualized span representations," in *EMNLP-IJCNLP*, Hong Kong, China, pp. 5784–5789, 2019.