Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.041395

ARTICLE

The Entity Relationship Extraction Method Using Improved RoBERTa and Multi-Task Learning

Chaoyu Fan[*]

Department of Electrical Engineering, Columbia University, New York, 10027, USA

*Corresponding Author: Chaoyu Fan. Email: cf2859@columbia.edu; jakefan179@163.com

Received: 20 April 2023 Accepted: 13 September 2023 Published: 29 November 2023

ABSTRACT

There is a growing amount of data uploaded to the internet every day and it is important to understand the volume of those data to find a better scheme to process them. However, the volume of internet data is beyond the processing capabilities of the current internet infrastructure. Therefore, engineering works using technology to organize and analyze information and extract useful information are interesting in both industry and academia. The goal of this paper is to explore the entity relationship based on deep learning, introduce semantic knowledge by using the prepared language model, develop an advanced entity relationship information extraction method by combining Robustly Optimized BERT Approach (RoBERTa) and multi-task learning, and combine the intelligent characters in the field of linguistic, called Robustly Optimized BERT Approach + Multi-Task Learning (RoBERTa + MTL). To improve the effectiveness of model interaction, multi-task teaching is used to implement the observation information of auxiliary tasks. Experimental results show that our method has achieved an accuracy of 88.95 entity relationship extraction, and a further it has achieved 86.35% of accuracy after being combined with multi-task learning.

KEYWORDS

Entity relationship extraction; Multi-Task Learning; RoBERTa

1 Introduction

The goal of information extraction is to identify entities and events in text and their interconnections and extract specific information from unstructured or semi-structured natural language text and organise it into a structured form [1]. This allows applications such as Question Answering (QA), Information Etrieving (IE), Knowledge Reasoning (KR), etc., to better process textual information. Because the entities in the text are annotated and scattered in the text, the possible relationships among the entities cannot be determined and cannot be generalized as structured information. Therefore, also need to use entity-relationship extraction techniques to determine the interrelationships between entities in the text and extract them to make the data structured.

The result of entity relationship extraction is essential. Taking Google's self-built Knowledge Graph released in 2012 as an example, the Knowledge Graph can systematize users' search results with knowledge. When users search for a keyword, the search engine can present the complete knowledge system of search results. For example, when searching the keyword "China," the search engine can not only return the original sentence with the keyword but also provide information such as China's population, the capital city, neighboring countries, land area, and other relevant details. In the knowledge graph, knowledge is stored in the form of a triad of (entity1, relation, entity2), such as (China, neighbouring, and Russia), and the entity relationship extraction technique is used to automatically obtain the relationship of the triads. The relationship extraction model based on template matching first matches the extracted input text with the designed template and when the match is successful the entities of the input text are assigned to the relationship. The templates for matching need to be pre-designed based on the context of the input text including vocabulary, syntax, etc. The pre-designed template requires high cost but remains poor generality because it requires redesigning the matching templates when migrating to a new domain. Those approaches use the syntactic structure of the sentence, entity lexicality, etc., as features and train them with classifiers such as Support vector machine (SVM) to reach good results. However, these methods also require artificially designed features based on a specific corpus and lack generality. Therefore, entity-relationship extraction plays an important role and achieves considerable results in many aspects such as building knowledge graphs. However, there are still many challenges that deserve further research as discussed in Section 2.

2 Related Work

The concept of entity relationship extraction was first introduced in 1998 when the 7th Message Understanding Conference (MUC) first introduced the entity relationship extraction task as a subtask of information extraction. There were only three types of entity relations in the corpus used in this measurement task: location (LOCATION_OF), position (EMPLOYEE_OF), and product (PRODUCT_OF). The best solution when the concept was first introduced was the template matching-based model was the early approach for entity relationship extraction.

2.1 Traditional Methods

This approach requires experts and linguists in related fields to manually write corresponding extraction rules based on the domain-specific corpus. In entity relationship extraction, the input text is first pre-processed and then matched with the text to be extracted using a pre-defined manually designed extraction template [2,3]. However, all these systems have a common drawback in that they require experts with domain-specific knowledge as well as linguists to exhaust all possible expressions of the relations as templates for extraction. Therefore, this approach not only requires significant labour and time costs but also inevitably leads to omissions and errors. Meanwhile, the template needs to be redesigned when the model is transferred to a new domain corpus. As a result, the template is less general and scalable.

Traditional machine learning-based approaches generally view the relationship extraction problem as a classification problem and the solution is to apply specific machine learning algorithms to construct classifiers on a manually labelled corpus.

2.2 Deep Learning-Based Methods

Recurrent Neural Network (RNN) is a class of neural networks dedicated to sequence modelling, and its ability to incorporate the above information in sequence modelling is very suitable for modelling text, so many researchers build entity relationship extraction models [4]. Convolutional Neural Network (CNN) [5] was used in the field of computer vision at the beginning and was also used by researchers to solve the entity relationship extraction problem with good results because it can be parallelized and can capture some local features and local word order information in the text well. Reference [6] proposed the Convolutional Neural Network for Image Compact-Resolution (CR-CNN) model, which uses an interval-based ranking loss as a loss function based on Pulse Coupled Neural Network (PCNN). The Attention CNN model proposed by [7] based on Zeng's model, introduced the attention mechanism in the input stage and pooling layer respectively, and finally trained with a ranking loss function similar to that in CR-CNN to obtain the best results at that time.

Reference [8] presented an innovative SDP-LSTM neural network model designed for relation classification. This model learns features in an iterative manner, focusing on the shortest dependency path. By leveraging multiple types of information, the proposed approach achieves superior performance compared to existing methods on the SemEval-2010 relation classification task. Reference [9] used LSTM and LSTMP with Connectionist Temporal Classification to investigate continuous note recognition on a piano for robotics. In terms of recognition rate (99.8%) and processing time, the results showed that a single layer of LSTMP outperformed a single layer of LSTM. However, deep LSTM with multiple layers achieved a perfect recognition rate of 100% but required more training time. Reference [10] proposed an Improved Graph Convolutional Network (ImprovedGCN) for collaborative filtering in recommendation systems. By utilizing neighborhood aggregation, the ImprovedGCN outperforms the Neural Graph Collaborative Filtering (NGCF) approach, demonstrating significant improvements in performance. Reference [11] proposeed an efficient baseline for skeleton-based action recognition, using Graph Convolutional Network (GCN) with MIB, Residual GCN module, and Part-wise Attention block. This method outperforms current state-of-the-art (SOTA) models while requiring fewer parameters, especially surpassing DGNN, another leading SOTA approach for skeleton-based action recognition.

With the rise of Graph Neural Networks (GNN) [12], some researchers have attempted to construct relationship extraction models based on graph neural networks. The Graph Convolutional Network (AGGCN) model proposed by [13] uses a Graph Convolutional Network to encode dependency trees and introduces an attention mechanism for soft pruning, allowing the model to selectively focus on structures useful for relational representation. In 2018, Google released the massive pre-trained language model Bidirectional Encoder Representations from Transformers (BERT) [14] which produced amazing results. BERT sets the best results on 11 Natural Language Processing (NLP) tasks on General Language Understanding Evaluation (GLUE) [15] and Stanford Question Answering Dataset (SQUAD) [16], among others. The R-BERT model by [17] first used BERT for entity relationship extraction, which models sentence and entity pairs with BERT and then accesses a fully connected network to achieve good results. Deep Multi-Task Learning (MTL) models are investigated in this paper, with the objective of improving performance by sharing learned structures across related tasks. However, the dynamics of multi-task learning in deep neural networks remain unknown, as does the significance of different task pairs. On three MTL datasets, the study compares various MTL approaches that use a shared trunk with task-specific branches architecture. we presented results and analysis of MTL for both the heterogeneous case (NYU v2) and homogeneous case (IMDB-WIKI) using different state-of-the-art deep learning models. Surprisingly, the results show that for a user-defined combination of tasks, multi-task learning frequently does not outperform single-task learning. To achieve the desired performance gains in multi-task learning, careful task pair selection and weighting strategies are necessary. Deep learning-based methods are now the main research direction in the field of entity relationship extraction because they can outperform traditional methods while not taking a lot of time to construct a large number of features manually. Entity Recognition (ER) is critical in Natural Language Processing because it enables tasks such as Knowledge Extraction, Text Summarization, and Key Extraction. A partially layered network design with a common Sequence Layer and stacking component with several Tagging Layers avoids overfitting and outperforms earlier CR techniques. Event extraction and the recognition of argumentative components using this architecture show positive outcomes [18].

Effective entity relation extraction is essential in natural language processing, but the lack of sufficient data in specific domains such as agriculture and the metallurgical industry poses a challenge for developing accurate models. To address this, a collaborative model utilizing multiple neural networks (RBF) was developed using a small balanced dataset. This model combines Roberta as the coding layer and BiGRU as the decoding layer. The evaluation of the model's performance showed a significant improvement, with an F1 value that was 25.9% higher than the traditional Word2vec–BiGRU–FC model and 18.6% higher than the Bert–BiLSTM model, highlighting its effectiveness [19]. Entity relationship extraction in the Chinese language is vital in natural language processing and can be approached through two main methods: joint extraction and pipeline extraction. However, joint extraction, while capable of producing relation triples, lacks external knowledge integration and handling of nested entities. To address these limitations, this article introduces a novel approach that frames the problem as a machine reading comprehension task, leveraging the power of the Roberta pre-training model. By employing this method, the model surpasses traditional pointer networks and achieves superior accuracy, recall, and F1 scores compared to other existing methods [20]. A novel model for relation extraction in natural language processing (NLP) incorporates external knowledge and semantic roles of entities. The model, which utilizes RoBERTa, semantic role embeddings, entity attention, and multi-task learning, outperforms existing methods in terms of performance. The study emphasizes the importance of leveraging semantic role information and employing multi-task learning for enhancing relation extraction in NLP [21]. Using deep learning techniques, the model effectively extracts unit features from target entity pairs and merges them into fusion features, allowing it to capture abstract semantics and sentence structure. The model's ability to extract comprehensive knowledge is demonstrated by experimental results. However, there are some disadvantages, such as the model's reliance on external features, its high complexity, and the unreliability of its scope [22]. Reference [23] proposed SMHS, a joint entity relation extraction model designed to address issues such as entity overlap, and exposure bias found in current methods. Span-tagger, span-embedding, LSTM, multi-head self-attention, span feature extraction, span-level relation decoding, and a span classification task are all integrated into a multi-task learning approach in the model. The effectiveness of the SMHS model is demonstrated through experiments on the NYT and DuIE 2.0 datasets, illustrating significant improvements over existing techniques.

The novel research contribution of the paper is discussed as follows:

- The entity relationship based on deep learning is explored to introduce semantic knowledge by using the prepared language model.
- An advanced entity relationship information extraction method is developed by combining Robustly Optimized BERT Approach (RoBERTa) and multi-task learning.
- Also, the intelligent characters in the field of linguistics called Optimized BERT Approach + Multi-Task Learning (RoBERTa + MTL) is combined with the extraction method.
- The RoBERTa + MTL Model with input layer-based information interaction performs better than the Joint-Attention Model with attention mechanism-based information interaction.

3 Methods

The flow of this paper is using improved RoBERTa pre-training for sequence-based relationship extraction tasks as shown in Fig. 1.

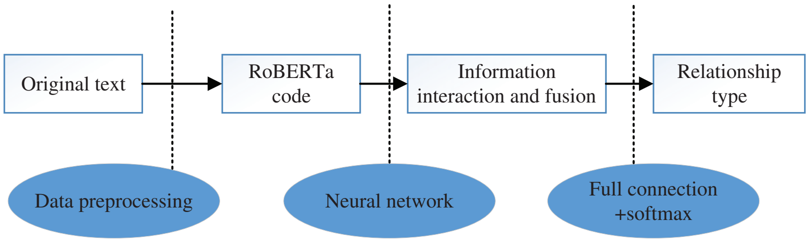


Figure 1: Flowchart of RoBERTa-based sequence-based relation extraction task

RoBERTa uses pre-training and adopts Transformer to build the model, and finally generates deep bi-directional language representations that can integrate contextual knowledge. The Transformer can be stacked to form a deeper neural network, and after multi-layer stacking of the Transformer structure, the RoBERTa structure is formed. RoBERTa, an enhanced version of BERT, utilizes a multi-layer stacking of transformers to improve its representation learning. By increasing the number of transformer layers to 12 or 24, RoBERTa can capture complex patterns in input data and generate more refined representations. This structure enhances performance while balancing computational resources. The basic structure of RoBERTa is shown in Fig. 2.

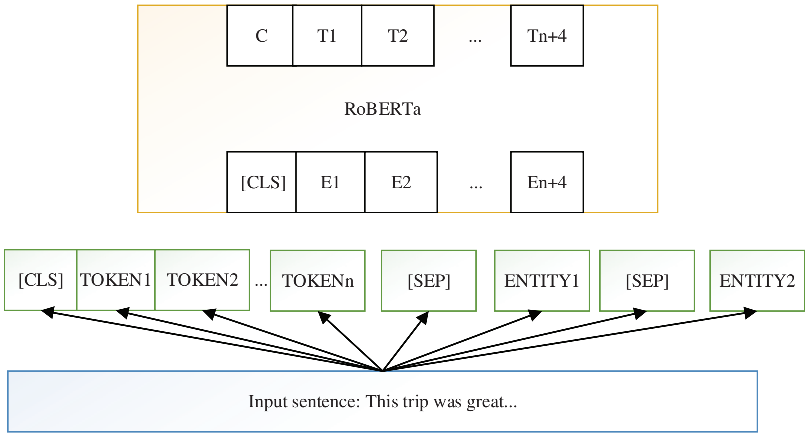


Figure 2: RoBERTa infrastructure

Convolutional layers use various filter sizes to extract multiple features and maximize pooling. These layers capture patterns at various scales, allowing the model to collect a wide range of data. Maximum pooling keeps crucial characteristics while reducing dimensions while activation functions introduce non-linearity. The resulting features are combined for further processing, which improves the model's ability to extract complex information from the input. The convolution operation is equivalent to extracting information from the convolution kernel of 3 sizes in the sentence, multiple convolutions are performed to extract multiple features, and maximum pooling will retain the most important information, and the model structure is shown in Fig. 3.

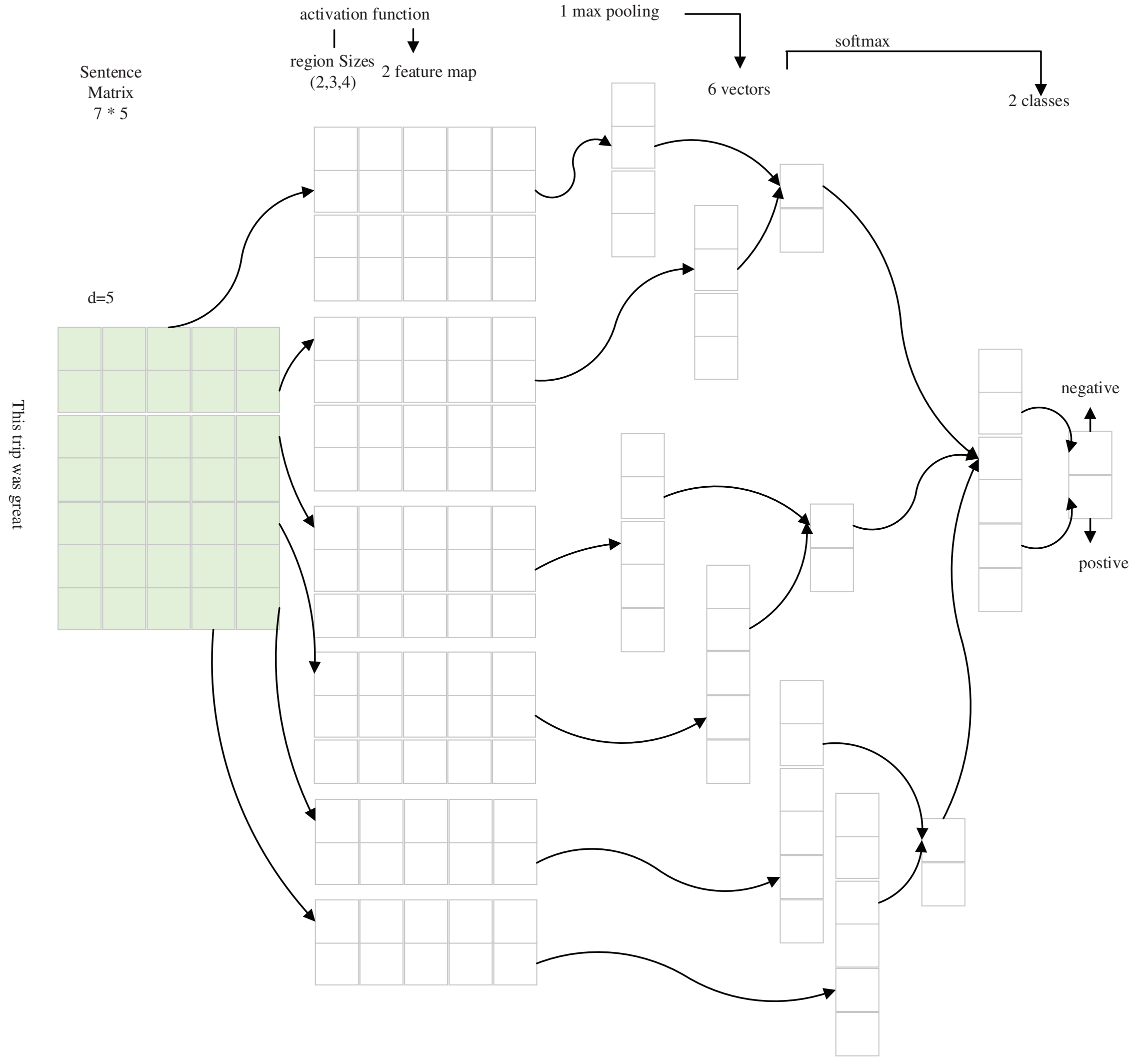


Figure 3: TextCNN infrastructure

The execution of TextCNN is similar to extracting information from N-Gram (algorithm based on statistical language model). TextCNN and N-Gram are both techniques used in text analysis, but they have different approaches. N-Gram extracts features by splitting text into contiguous sequences of words or characters, capturing local patterns. TextCNN, on the other hand, applies convolutional layers to extract local features from text using a sliding window approach. It uses filters of different window sizes to capture patterns at various scales. The extracted features are then pooled and fed into fully connected layers for classification. While N-Gram focuses on fixed-size sequences, TextCNN adapts to different window sizes and scales, making it effective for text classification tasks. Because TextCNN is only one layer, it is difficult to learn long-distance features. The regional embedding of Deep Pyramid Convolutional Neural Networks (DPCNN) is the TextCNN with the pooling layer removed, and then the convolutional layers are superimposed, each layer is 250 convolutional kernels of size 3, and finally, the labels are predicted using full connectivity and softmax activation function normalization.

TextCNN and DPCNN are convolutional neural network architectures used for text classification. TextCNN applies one-dimensional convolutions with max-pooling and multiple filter sizes, while DPCNN uses convolutional blocks with downsampling, shortcut connections, and global max-pooling. DPCNN is designed to handle longer texts and capture hierarchical representations efficiently. Both models involve embedding words, but DPCNN includes shortcut connections and global max-pooling to address the limitations of standard CNNs for longer texts. The structure of the DPCNN model in the experiment. There are advantages and disadvantages to using TextCNN and DPCNN as the main models to achieve entity relationship extraction using deep learning. The left context, word and right context of a word are combined as a word representation, and the contextual information of an utterance is extracted using a DPCNN. TextCNN provides simplicity, parallel processing, and the ability to capture local patterns in text. It is robust to varying input sizes and performs well on text classification tasks. On the other hand, DPCNN efficiently handles longer texts, captures global information, and benefits from residual connections. It excels in tasks that require a holistic understanding of the text and achieves strong performance in text classification.

$$x_i = [c_l(\omega_i); e(\omega_i); c_r(\omega_i)] \tag{1}$$

The left context vector $c_l(\omega_i)$ of a word ω_i, the word embedding representation vector $e(\omega_i)$, and a word are stitched together to obtain the expression x_i. And the maximum pooling operation learns the most important potential semantic information in the sentence.

$$y^{(3)} = \max_{i=1}^{n} y_i^{(2)} \tag{2}$$

Then the entity relationship extraction is performed using the activation function softmax, which is calculated as follows:

$$y^{(4)} = W^{(4)}y^3 + b^{(4)} \tag{3}$$

$$p_i = \frac{\exp\left(y_i^{(4)}\right)}{\sum_{k=1}^{n}\exp\left(y_k^{(4)}\right)} \tag{4}$$

The RoBERTa base framework has 12 encoder layers excluding the first input layer, and the first category token (CLS, classification) vector of each encoder layer can be treated as a sentence vector. The purpose is to get both the features about the words and the semantic features, the model is specifically done by sending the CLS vectors from layer 1 to layer 12 and using the RoBERTa as the embedding layer to the input of other TextCNN/DPCNN networks. The model framework is shown in Fig. 4.

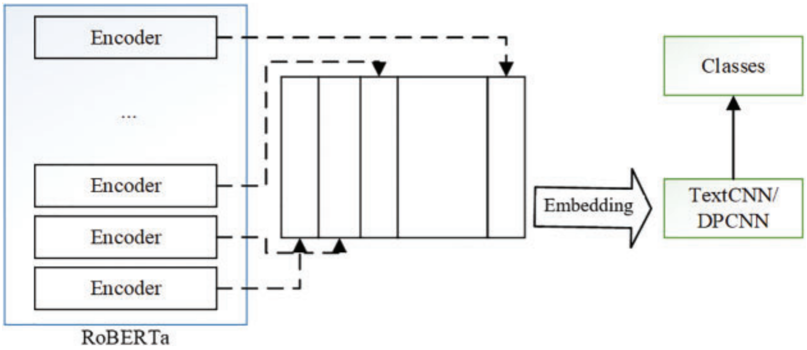
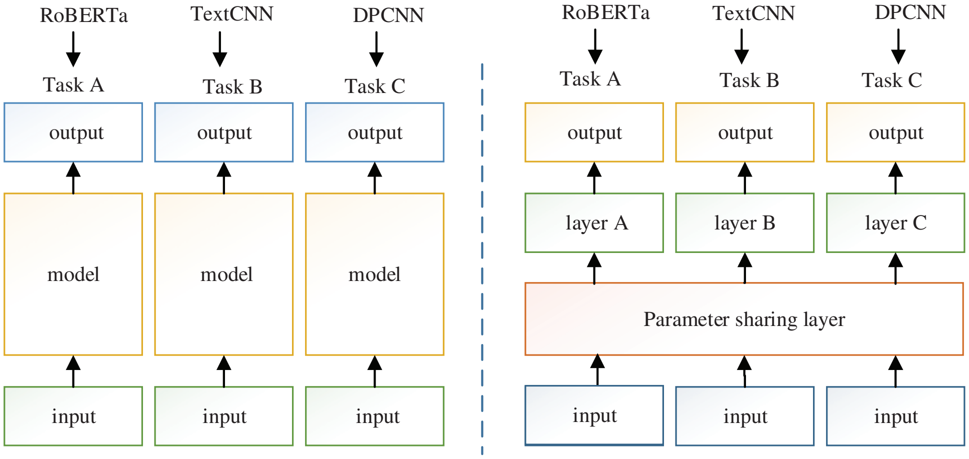


Figure 4: Improved structure of RoBERTa

Multi-Task Learning (MTL) is a learning method as opposed to Single-Task Learning (STL). Single-Task Learning (STL) focuses on solving one specific task at a time using task-specific models and datasets. It excels at the given task but may struggle with new tasks. Multi-Task Learning (MTL) simultaneously learns multiple related tasks by leveraging shared information between them. MTL models have a shared feature representation and separate task-specific branches. They benefit from knowledge transfer between tasks, improve learning efficiency, and potentially generalize better to new tasks. In STL, tasks are assumed to be independent, while MTL recognizes task interdependencies. MTL is more data-efficient, complex, and capable of transferring knowledge between tasks. On the other hand, STL is simpler and better suited for solving a single task in isolation. The human brain can learn several different tasks at the same time and uses the same architecture of the brain when processing several different tasks. Moreover, when the knowledge gained from previous learning of related tasks helps to learn a new task. For example, when learning a first foreign language, the knowledge in one's native language enhances the learning of the foreign language, and individuals who have learned multiple foreign languages learn a new foreign language much faster than those who have not learned it. Researchers believe that machine learning models should do the same, learning the goals of different tasks simultaneously during training, so that the knowledge learned in one task mutually enhances the model's effectiveness on other different tasks.

For three different tasks A, B and C, (a) in Fig. 5 shows the classical single-task learning approach. The single-task learning approach uses three mutually independent models to solve the tasks separately without any connection between these models, and the features extracted in one task do not have any effect on the other tasks. (b) in Fig. 5 shows the approach to multi-task learning. This approach uses a parameter-sharing layer to share the information learned from different tasks and then access the task-specific layer for output, which can facilitate the sharing of features extracted from multiple tasks.



(a) STL (b) MTL

Figure 5: Single-task learning (STL) *vs*. multi-task learning (MTL)

Multitask learning is specifically defined as follows: given m a collection of different learning tasks $\{T_i\}_{i=1}^{m}$. Multi-task learning can be a good solution when several different related tasks need to be obtained results at the same time.

In multi-task learning, a single model is trained to solve multiple related tasks simultaneously. This approach typically involves using shared layers to capture common patterns across tasks, task-specific layers to handle task-specific characteristics, and joint optimization to train the model on a combined objective. This method improves performance, enables knowledge transfer, and enhances generalization across tasks. In addition, even if focusing on only one specific task, multitask learning can still be used to help the model improve its performance on the main task by constructing auxiliary tasks for training. It focuses on only one specific task and is generally called Learning with Auxiliary Task. In this paper, the TextCNN model, DPCNN model and RoBERTa model are integrated to construct a multi-task model.

Deep learning models such as TextCNN, DPCNN, and RoBERTa are used in natural language processing tasks. While they have different architectures and purposes, they can be combined to form a multi-task model. The TextCNN model captures local patterns, the DPCNN model focuses on the global context, and the RoBERTa model recognizes word relationships using attention mechanisms. In a multi-task model, separate branches are created for each task, with the lower layers shared for shared representations. This allows the model to learn and optimize multiple tasks at the same time. The multi-task model can capture both local and global contexts by integrating these models, making it useful for a variety of NLP tasks. The model focuses on the extraction of entity relationships, and hope that the model can learn more supervised information and improve the effectiveness of entity relationship extraction by introducing an auxiliary task.

4 Experiments

4.1 Comparison of Entity Relationship Extraction Effects

Several publicly available datasets are used in this experiment. The datasets are the TAC Relation Extraction Dataset (TACRED) dataset is a large-scale dataset for relation extraction, consisting of 106,264 examples from newswire and web text sources. It covers 41 relation types used in TAC KBP challenges, with labels for "no_relation" when no specific relation is present. The dataset combines human annotations and crowdsourcing, making it a valuable resource for developing and evaluating relation extraction models [24], the SemEval 2010 Task 8 dataset is designed for multi-way classification, specifically focusing on identifying mutually exclusive semantic relations between pairs of nominals [25], the DuIE dataset stands out as the pioneering extensive and top-notch dataset designed for Chinese Information Extraction (IE). Comprising an impressive 450,000 instances, 49 distinct relation types, 340,000 unique Subject-Predicate-Object (SPO) triples, and 210,000 sentences, it boasts an expansive scope. The dataset encompasses a wide range of real-world applications, spanning areas such as news, entertainment, and user-generated content. Annotations in DuIE include both single-valued and multi-valued triples, thereby accurately mirroring real-world scenarios [26], and the Wiki80 dataset [27]. Because the TACRED dataset is large and provides high-quality pre-processed results, the TACRED dataset is used in the experimental part. DuIE suffers from noisy labels, limited domain coverage, and lacks fine-grained annotations. Wiki80 has imbalanced classes, lacks contextual information, and focuses primarily on English language relations.

4.2 Introduction to the Dataset

The TACRED dataset contains a total of 106,264 texts with 42 types of relationships, including a "no_relation" type, which is a large-scale relationship extraction dataset. Table 1 gives the basic statistics of the TACRED data sample.

Table 1: Statistics of the TACRED dataset

Split	Original corpus	Number of examples
Train	TAC KBP 2009–2012	68,124
Dev	TAC KBP 2013	22,631
Test	TAC KBP 2014	15,509
Total 106,264		

Each sample of TACRED contains information on the text sequence after disambiguation, the relationship type, the lexical annotation sequence, the Named entity recognition (NER) sequence, the position of the entity pair, the entity NER type, and the dependent syntactic tree. An example of a specific data sample is given in Table 2. When the relationship of an entity pair is marked as a "no_relation" relationship type, the corresponding sample is negative. Statistically, 79.5% of the samples are marked as "no_relation", which shows that the data type distribution of TACRED is unbalanced.

Table 2: Example of samples in TACRED data

Relation	a_t Org: top _ members/employees D'
Token	['sharpton', 'is', 'persident', 'of', 'the', 'National', 'action', 'Network', '-']
Subj (start, end)	(5,7)
Obj (start, end)	(0,0)
Subj_type	ORGANIZATION
Obj_type	PERSON
Stanford_pos	['NNP', 'VBZ', 'NN', 'IN', 'DT', 'NNP', 'NNP', 'NNP', 'O']
Stanford_ner	['PERSON', 'O', 'O', 'O', 'O', 'ORAGNIATION', 'ORAGNIATION', 'ORAGNIATION', 'O']
Stanford_head	[3,3,0,8,8,8,8,3,3]
Stanford_deperl	['nsubj', 'cop', 'ROOT', 'case', 'det', 'compound', 'compound', 'nmod', 'punct']

The SemEval dataset is smaller than TACRED due to manual annotation effort, task complexity, and limited data availability. TACRED's narrower focus on relation extraction allows for a potentially larger dataset. There are 9 types of relationships for entity pairs and one special relationship "Other". The SemEval dataset only provides the raw text, so it needs to be pre-processed by NLP-related tools, including lexical annotation and syntactic analysis. Lexical annotation and syntactic analysis are essential NLP tools for processing human language by computers. Lexical annotation involves categorizing words with their grammatical information, such as part-of-speech tags and named entity recognition. The Syntactic analysis focuses on analyzing the structure and relationships between words in a sentence, including parsing and dependency parsing. These tools enable various NLP applications like machine translation, sentiment analysis, and information extraction. For example, Stanfordnlp can be used for English text and Hanlp, Pyltp, etc., for Chinese text. Since the annotation results of TACRED are also derived from Stanfordnlp, Stanfordnlp is also used to preprocess the data on the

SemEval dataset. The meanings, True Positive (TP), False Positive (FP), False Negative (FN), True Negative (TN) are shown in Table 3.

Table 3: Confusion matrix of classification results

		Forecast results	
		Positive	Negative
Physical truth	True	TP	FN
	False	FP	TN

The official evaluation metric for the TACRED dataset is Micro-F1, which does not include the statistics of the "no_relation" relationship type. The Micro-F1 is used for multi-category problems with unbalanced categories and is calculated by first counting the confusion matrix for each category, including TP_i, FP_i, FN_i, TN_i, "summing" the confusion matrices, and calculating the precision and recall rates corresponding to P_{micro}, R_{micro}, and finally obtaining the Micro-F1, calculated as follows:

$$P_{\text{micro}} = \frac{\sum_i TP_i}{\sum_i TP_i + FP_i} \tag{5}$$

$$R_{\text{micro}} = \frac{\sum_i TP_i}{\sum_i TP_i + FN_i} \tag{6}$$

$$\text{Micro-F1} = 2 \times \frac{P_{\text{micro}} \times R_{\text{micro}}}{P_{\text{micro}} + R_{\text{micro}}} \tag{7}$$

On the SemEval data, the performance of the model is officially tested using the Macro-F1 score, again, the other category is not taken into account. Macro-F1 is also suitable for multi-category problems, which are not affected by data imbalance, but are vulnerable to categories with high recall or precision rates. It is calculated by counting the TP_i, FP_i, FN_i, TN_i of each category, calculating their respective Precision$_i$, Recall$_i$, getting the corresponding $F1_i$ values, and finally averaging them to obtain Macro-F1.

An experimental result comparison between the TACRED and SemEval with different evaluation parameters such as precision, recall, Micro-F1 and Macro-F1 is shown in Table 4, where the Shortest Dependency Path Long Short-Term Memory (SDPLSTM) and PALSTM are LSTM-based relationship extraction models, and CGCN and the proposed models are both graph volume network-based relationship extraction models. Shortest Dependency Path Long Short-Term Memory (SDPLSTM) is a model designed for natural language processing tasks, combining LSTM neural networks with shortest dependency paths. By focusing on the most direct syntactic relationships between words, SDPLSTM enhances the LSTM's capacity to grasp long-range dependencies and contextual information, significantly improving performance in tasks like sentiment analysis and named entity recognition with 66.2% precision and 52.8% of recall rate in TACRED dataset and 58.6% and 83.3% micro and Macro-F1 rate in SemEval 2010 dataset. While, Projection Augmented Long Short-Term Memory (PALSTM), on the other hand, extends the LSTM's capabilities by introducing location awareness. This means incorporating position information as additional features in the model. By considering the relative positions of elements in the input sequence, PALSTM gains a better understanding of the order and positional dependencies, making it particularly well-suited for tasks such as machine translation and question answering. Also, it attains an outcome of 65.8% precision and 64.4% recall rate in TACRED dataset; 65.2% and 82.8% of F1 rate in SemEval 2010 dataset. Contextual Graph Convolutional Network (CGCN) represents an enhanced version of the Graph Convolutional Network (GCN) architecture. What sets CGCN apart is its ability to integrate contextual information into the learning process. This could include word embeddings, syntactic data, or other relevant features derived from the data. By leveraging this contextual understanding, CGCN becomes more effective in capturing the semantics and relationships between entities in the graph with 69.8% precision and 63.4% recall rate, making it a valuable tool for tasks like node classification, link prediction, and recommendation systems. LSTMs are widely utilized for understanding, analyzing, and categorizing sequential data due to their ability to capture and comprehend long-term relationships among different time steps. They find extensive applications in various fields such as sentiment analysis, language modelling, speech recognition, and video analysis.

Table 4: Comparison of experimental results

Model	TACRED			Semeval 2010
	Precision	Recall	Micro-F1	Macro-F1
SDP-LSTM† (2015)	66.2	52.8	58.6	83.8
PA-LSTM† (2017)	65.8	64.4	65.2	82.8
C-GCN‡ (2022)	69.8	63.4	66.3	84.5
GCN (baseline) (2020)	67.6	50.1	63.1	82.4
ROBERTA + MTL (ours)	70.9	64.9	67.5	85.2
ROBERTA + MTL (ensemble)	71.5	66.2	68.7	85.3

The specific values of precision and recall are not given in the table because the SemEval dataset uses the official evaluation code in the evaluation phase, which directly outputs the results of F1. In the table, "†" indicates the results directly quoted from the paper, and "‡" indicates the results reproduced from the paper.

By analyzing the experimental results of the TACRED data set, it is concluded that the F1 grade is improved by 1.2% compared with the same type of CGCN model. Especially, compared with the baseline model, the model has some improvements in F1 precision evaluation, recall and evaluation. Finally, through the integration of several improved RoBERTa models and multi-task learning, the results of the softmax function are averaged, and then the prediction results are improved by 1%. Further comparative experimental results of joint entity relationship extraction are shown in Table 5, and the Micro-F1 for each model is presented using a bar chart, as shown in Fig. 6.

Table 5: Experimental results of joint and pipeline models

Algorithmic model	Micro-precision (%)	Micro-recall (%)	Micro-F1 (%)
Novel tagging	79.89	56.33	66.04
Multi-head	79.35	72.48	75.75
CasRel	79.47	76.64	78.02
ETL-span	80.31	76.70	78.46

(Continued)

Table 5 (continued)

Algorithmic model	Micro-precision (%)	Micro-recall (%)	Micro-F1 (%)
Pipeline	80.16	78.74	79.46
Joint-attention model	82.95	83.11	83.04
RoBERTa + MTL model	83.22	83.77	83.48

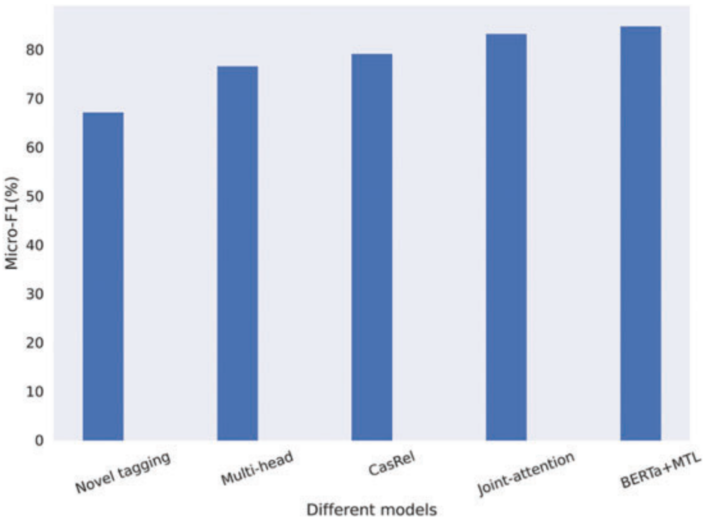


Figure 6: Comparison of all models on Micro-F1

Compared with the CasRel method, the Micro-F1 of the Joint-Attention Model and the RoBERTa + MTL Model proposed in this paper are 5.46 and 5.0 percentage points ahead, respectively. The CasRel method utilizes the Joint-Attention Model and the RoBERTa + MTL Model for relation extraction. Micro-F1, an evaluation metric, measures the performance of these models. The Joint-Attention Model captures dependencies between entities and context words using attention mechanisms, while the RoBERTa + MTL Model combines RoBERTa's language understanding with Multi-Task Learning. The models' Micro-F1 scores indicate their ability to accurately extract relations. The CasRel method adopts a cascade decoding scheme similar to this paper in modelling the joint entity-relationship extraction task, but it adopts a simple feature fusion approach in the information interaction between the entity model and the relationship model, and completely ignores the important feature information of entity type, thus resulting in the final performance of the model is worse than that of this paper. The cascade decoding scheme is a key component of the CasRel method for relation extraction. It involves a hierarchical approach where relation labels are iteratively refined. Contextual encoding is used to capture the surrounding context, and a relation classification model predicts the relation labels. The labels are updated based on the predictions, and the process continues until all entity pairs have been processed. This scheme improves relation extraction accuracy by leveraging global information and refining predictions step-by-step. This result reflects the importance of the way of information interaction between the entity model and relational model, especially the entity class information is a very important feature constraint. The different number of triples in DuIE's dev set data distribution is shown in Table 6.

Table 6: Data distribution of the different number of triples in DuIE's dev set

Number of triples	$N = 1$	$N = 2$	$N = 3$	$N = 4$	$N \geq 5$
Data volume	9225	7159	2593	1288	1347

The trends of Micro-F1 for each model are shown in Fig. 7.

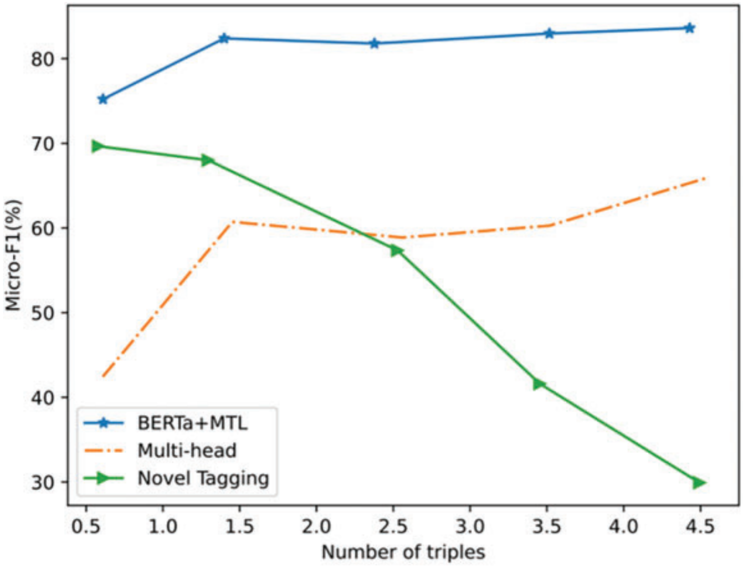


Figure 7: Variation trend of Micro-F1 of each model under different numbers of triples samples

4.3 Influence of Different Mechanisms

In the experiments, the Pipeline model with no information interaction between the entity model and the relational model is used as the baseline model. The overall Micro-F1 gain of the model brought by various mechanisms compared to the baseline model is analyzed by histogram as shown in Fig. 8.

The experimental results show that the multi-task learning approach of sharing parameters between the entity model and the relational model and joint optimization has a positive effect on the joint entity-relational extraction task with 3.21% increase in multi-task learning, 5.4% increase in joint-attention network and 5.8% increase in BERTa + MTL, resulting in an aMicro-F1 gain of 3.20% points, indicating that the joint optimization, compared with independent training, does not update the parameters for the entity or relational task alone.

The changes of Micro-F1 in the first 14 rounds of training of the RoBERTa + MTL Model and Joint-Attention Model are shown in Fig. 9. The RoBERTa + MTL Model and Joint-Attention Model achieve the optimal performance in the 4th and 6th rounds of training, respectively, due to the use of pre-trained models for fine-tuning and a certain basis for parameter updating. From the experiment result, it can be found that the RoBERTa + MTL Model with input layer-based information interaction performs better than the Joint-Attention Model with attention mechanism-based information interaction.

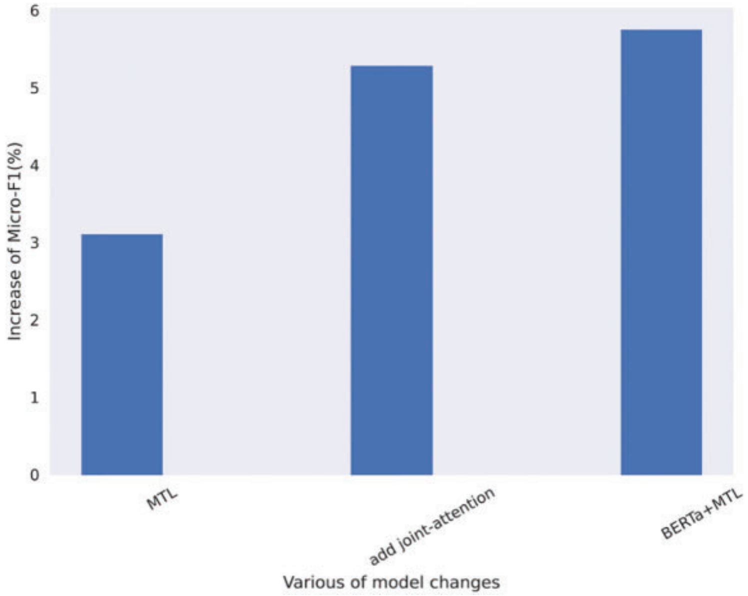


Figure 8: Micro-F1 gains of various model changes

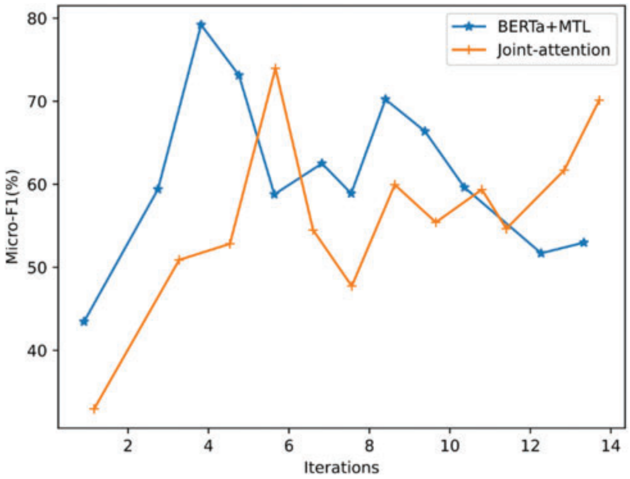


Figure 9: Comparison of Micro-F1 in the process of model training with different information interaction methods

4.4 Analysis of the Role of Multi-Task Learning Module

RoBERTa + MTL is improved as the base entity-relationship extraction single-task model and different tasks are used as auxiliary tasks to improve the model.

From Table 7, the Multi-head method has some ability to handle nested triples, but the overall performance is still low. This is due to the difficulties associated with complex nested structures, attention diffusion, limited training data, and the models' capacity and complexity. These factors make it difficult to understand and represent nested relationships accurately. Compared with the Novel Tagging and Multi-head methods, the RoBERTa + MTL Model has a significant performance advantage in the case of natural statements containing more entity-relational triples. Roberta + MTL method provides several advantages such as improved performance by training on multiple tasks simultaneously, better data efficiency by leveraging shared knowledge, knowledge transfer between tasks, regularization to prevent overfitting, and scalability to handle various NLP tasks. Also, it enhances performance, efficiency, knowledge sharing, regularization, and scalability for tackling diverse NLP challenges. The main reason is that the multi-layer binary tagging method in this paper can better handle the nested entity-relationship triples by tagging the corresponding tail entities under each type of relationship separately for each head entity. Moreover, thanks to the information interaction based on the input layer, the joint model only needs to predict the tail entities and entity relationships associated with the current head entity in each decoding during the prediction process.

Table 7: Macro-F1 values (%) of the model when different auxiliary tasks are introduced

Auxiliary task	Main task	
	SemEval-2010 task 8	Wiki80
RoBERTa + MTL	89.52	85.19
+ SemEval-2010 task 8	–	85.61
+ Wiki80	89.64	–
+ Single sentence classification task	89.27	85.22
+ Sentence pair classification task	89.52	85.23
+ All auxiliary tasks	89.75	85.76

Table 5 was produced from the relevant data and the results for the highest Macro-F1 values are shown in bold. It can be determined by analyzing the data in Table 7.

1. The Macro-F1 values of the model with all auxiliary tasks introduced are the highest when the two datasets are used as the main task separately. The multi-task learning module does not need to directly change the network structure of the model. When the model is trained, it only needs to introduce other task models sharing parameters for training. This makes the model highly scalable.
2. Our model has a greater performance improvement than introducing only a single-sentence entity-relationship extraction task or sentence-to-entity relationship extraction task. This paper suggests that this is because the two tasks are both entity-relationship extraction and have a strong correlation. Furthermore, the model learns the semantic expressions related to the relationship more easily by performing multi-task learning on both tasks.
3. When using words about inter-entity relationships to extract tasks or using phrases to help multi-task learning, the quality of the model will decline. The quality decline of the model indicates that the relevance of these tasks to inter-entity relationships is not as good as introducing rules to let the model learn more general text expressions. This also shows that the introduction of subsidiary tasks cannot be infinite and the analysis of their relevance is also worth studying.

4.5 Effect of Training Sample Size on Performance

In this subsection, experiments are designed to investigate the difference between the performance of this paper's model and the traditional word vector-based model at different training data sizes. To reduce the impact of extracted data on performance, three sets of data were randomly selected for training using unused random seeds for each data volume setting. Calculating the relative decline rate of the model scores at different data amounts can obtain the curve of the relative decline rate of the scores as the amount of training data decreases. The higher the curve, the greater the demand for the model for the amount of data and the insensitivity to smaller training data.

The models involved in the comparison in this subsection are as follows:

-Our Model (RoBERTa + MTL): the complete model proposed in this paper.

-Improved RoBERTa + Semantic Role Information: this model removes the multi-task learning module from the full model presented in this paper, i.e., the model combining improved RoBERTa and semantic role information introduced previously.

-PCNN: the CNN-based model introduced, with the lexical features removed. The experimental results are derived from the model experiments reproduced in this paper, shown in Fig. 10.

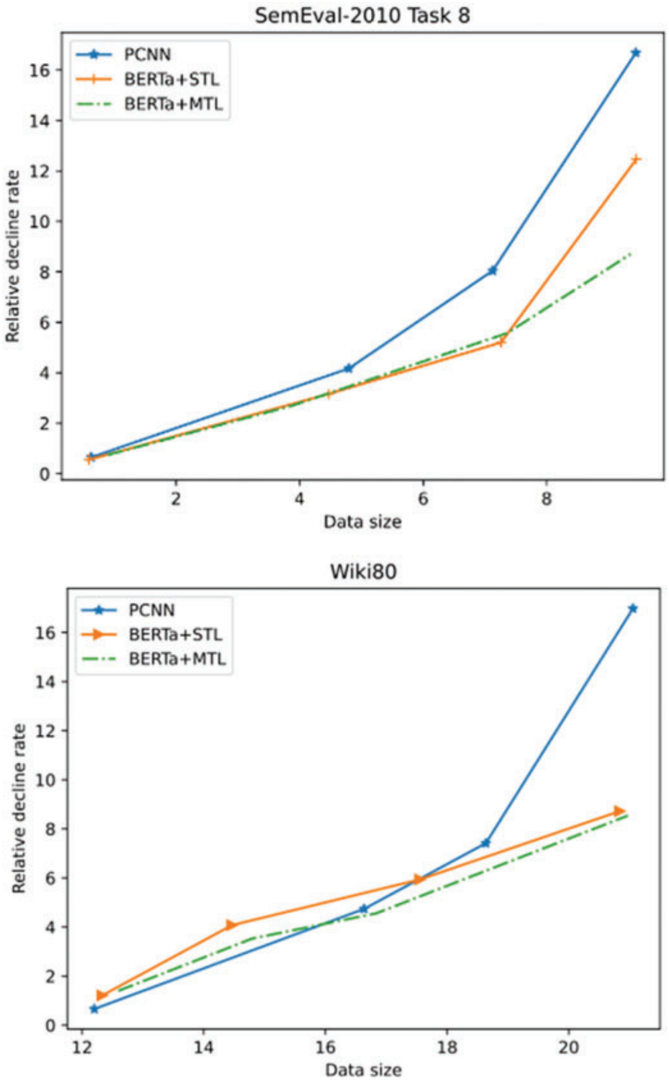


Figure 10: Relative decline rate of model scores under different data amounts

From Table 8 and Fig. 10,

(1) Compared with the traditional word vector-based PCNN, our model performs better based on pre-trained language models on various data volumes. The performance of the proposed model surpasses that of PCNN when trained with only 10% of the data. It shows that the pre-trained language model can obtain good modelling ability for contextual semantics by pre-training on the large-scale corpus, which provides good generalization performance for the downstream entity relationship extraction task, allowing the model to learn the semantic information implied by the samples and improving the relationship extraction ability of the model. Simultaneous use of a model with multi-task learning.

(2) Compared with the improved RoBERTa + SRL, the model RoBERTa + MTL with multi-task learning can still maintain better performance with less data volume. This indicates that the multi-task learning module can increase the generalization performance of the model by learning the semantic information embedded in different domain task annotation data, which can alleviate the problem of insufficient domain annotation corpus for entity relationship extraction.

Table 8: Is produced based on the relevant experimental data

Data set	SemEval-2010 task 8					Wiki80				
N	100	70	50	30	10	100	70	50	30	10
Our model	88.95	88.58	87.28	85.55	82.13	86.35	83.78	83.07	81.03	79.48
RoBERTa + SRL	87.91	88.15	87.11	85.41	78.15	85.96	83.32	82.02	80.54	78.92
PCNN	81.59	79.97	78.56	75.18	67.35	77.34	75.94	74.42	71.93	64.15

5 Conclusion

With the continuous development of deep learning and NLP, ER extraction research has achieved good results. However, further research work is needed for solving problems and shortcomings in the existing research methods. The classical deep learning-based entity relationship extraction models all use pre-trained vectors to map the text, and then design complex upper-layer networks based on CNN, RNN, etc., to extract features. However, such word vectors cannot express the contextual semantic information well and the current model does not consider the introduction of linguistics-related knowledge. As a result, the model has limitations in extracting insufficient semantic information from entities and cannot generate more relevant contextual information with entities. The ER extraction method focused on teaching materials and ignored external knowledge and limiting their preparation ability. Multi-task learning enables the model to study the observation information hidden in other learning materials. In contrast, the scarcity of research on using multi-objective learning to improve the performance of a sample model in inter-entity relationships. The purpose of this research in future is to overcome this gap by identifying shortcomings in the current approach and proposing specific methods that encompass multi-objective learning to effectively address these limitations.

Acknowledgement: The author would like to express their heartfelt gratitude to the techniques that have made valuable contributions to this research.

Funding Statement: The author received no specific funding for this study.

Author Contributions: Chaoyu Fan contributed to the design and methodology of this study, the assessment of the outcomes and the writing of the manuscript.

Availability of Data and Materials: All data generated or analysed during this study are included in the manuscript.

Conflicts of Interest: The author declares that he has no conflicts of interest to report regarding the present study.

References

[1] A. Rago, C. Marcos and J. A. Diaz-Pace, "Using semantic roles to improve text classification in the requirements domain," *Language Resources and Evaluation*, vol. 52, no. 3, pp. 801–837, 2018.

[2] S. N. Alsubari, S. N. Deshmukh, A. A. Alqarni, N. Alsharif, T. H. Aldhyani *et al.,* "Data analytics for the identification of fake reviews using supervised learning," *Computers, Materials & Continua*, vol. 70, no. 2, pp. 3189–3204, 2022.

[3] Q. Liu, C. Liu and Y. Wang, "Integrating external dictionary knowledge in conference scenarios the field of personalized machine-translation method," *Journal of Chinese Informatics*, vol. 33, no. 10, pp. 31–37, 2019.

[4] R. Ali, M. H. Siddiqi and S. Lee, "Rough set-based approaches for discretization: A compact review," *Artificial Intelligence Review*, vol. 44, no. 2, pp. 235–263, 2015.

[5] A. Warstadt, A. Singh and S. R. Bowman, "Neural network acceptability judgments," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 625–641, 2019.

[6] K. H. Thung and C. Y. Wee, "A brief review on multi-task learning," *Multimedia Tools and Applications*, vol. 77, no. 22, pp. 29705–29725, 2018.

[7] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman *et al.,* "Gradient surgery for multi-task learning," *Advances in Neural Information Processing Systems*, vol. 33, no. 489, pp. 5824–5836, 2020.

[8] Y. Xu, L. Mou, G. Li, Y. Chen, H. Peng *et al.,* "Classifying relations via long short-term memory networks along shortest dependency paths," in *Proc. of the 2015 Conf. on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, pp. 1785–1794, 2015.

[9] Y. Jia, Z. Wu, Y. Xu, D. Ke and K. Su, "Long short-term memory projection recurrent neural network architectures for piano's continuous note recognition," *Journal of Robotics*, vol. 2017, no. 2061827, pp. 1–7, 2017.

[10] S. Dhawan, K. Singh, A. Rabaea and A. Batra, "ImprovedGCN: An efficient and accurate recommendation system employing lightweight graph convolutional networks in social media," *Electronic Commerce Research and Applications*, vol. 55, pp. 101191, 2022.

[11] Y. F. Song, Z. Zhang, C. Shan and L. Wang, "Stronger, faster and more explainable: A graph convolutional baseline for skeleton-based action recognition," in *Proc. of the 28th ACM Int. Conf. on Multimedia*, New York, USA, pp. 1625–1633, 2020.

[12] C. Fifty, E. Amid, Z. Zhao, T. Yu, R. Anil *et al.,* "Efficiently identifying task groupings for multi-task learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 27503–27516, 2021.

[13] Y. Zhang and Q. Yang, "An overview of multi-task learning," *National Science Review*, vol. 5, no. 1, pp. 30–43, 2018.

[14] X. Sun, R. Panda, R. Feris and K. Saenko, "Adashare: Learning what to share for efficient deep multi-task learning," *Advances in Neural Information Processing Systems*, vol. 33, no. 732, pp. 8728–8740, 2020.

[15] O. Marfoq, G. Neglia, A. Bellet, L. Kameni and R. Vidal, "Federated multi-task learning under a mixture of distributions," *Advances in Neural Information Processing Systems*, vol. 34, pp. 15434–15447, 2021.

[16] N. Jin, J. Wu, X. Ma, K. Yan and Y. Mo, "Multi-task learning model based on multi-scale CNN and LSTM for sentiment classification," *IEEE Access*, vol. 8, pp. 77060–77072, 2020.

[17] T. Gong, T. Lee, C. Stephenson, V. Renduchintala, S. Padhy *et al.,* "A comparison of loss weighting strategies for multi-task learning in deep neural networks," *IEEE Access*, vol. 7, pp. 141627–141632, 2019.

[18] A. Waldis and L. Mazzola, "Nested and balanced entity recognition using multi-task learning," arXiv preprint arXiv:2106.06216, 2021.

[19] Y. Liu, Q. Zuo, X. Wang and T. Zong, "Entity relationship extraction based on a multi-neural network cooperation model," *Applied Sciences*, vol. 13, no. 11, pp. 6812, 2023.

[20] T. Shang, B. Deng and T. Jiang, "A Chinese entity-relation extraction method via improved machine reading comprehension," in *2022 4th Int. Conf. on Intelligent Information Processing (IIP)*, Guangzhou, China, pp. 175–180, 2022.

[21] Z. Zhu, J. Su and X. Hong, "Improving relation extraction using semantic role and multi-task learning," in *Knowledge Graph and Semantic Computing: Knowledge Graph and Cognitive Intelligence: 5th China Conf., CCKS 2020*, Nanchang, China, pp. 93–105, 2021.

[22] Y. Zhao, X. Yuan, Y. Yuan, S. Deng and J. Quan, "Relation extraction: Advancements through deep learning and entity-related features," *Social Network Analysis and Mining*, vol. 13, no. 1, pp. 92, 2023.

[23] D. Han, Z. Zheng, H. Zhao, S. Feng and H. Pang, "Span-based single-stage joint entity-relation extraction model," *PLoS One*, vol. 18, no. 2, pp. e0281055, 2023.

[24] Y. H. Zhang, V. Zhong, D. Q. Chen, G. Angeli and C. D. Manning, "Position-aware attention and supervised data improve slot filling," in *Conf. on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, pp. 35–45, 2017. https://nlp.stanford.edu/pubs/zhang2017tacred.pdf

[25] I. Hendrickx, S. N. Kim, Z. Kozareva, P. Nakov, D. Ó. Séaghdha *et al.,* "Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals," arXiv preprint, arXiv:1911.10422.

[26] X. Luo, W. Liu, M. Ma and P. Wang, "A bidirectional tree tagging scheme for joint medical relation extraction," in *Int. Joint Conf. on Neural Networks (IJCNN)*, Gold Coast, Australia, pp. 1–8, 2023.

[27] https://figshare.com/articles/dataset/Wiki80/19323371 (accessed on 01/10/2022).