International Journal of Computational Science and Engineering

ISSN online: 1742-7193 - ISSN print: 1742-7185
https://www.inderscience.com/ijcse

A new binary tagging-based model integrated with unsupervised tree hierarchy for relational triple extraction

Hua Yin, Zhiquan Chen

DOI: 10.1504/IJCSE.2024.10066802

Article History:

Received:	06 May 2024
Last revised:	27 July 2024
Accepted:	02 August 2024
Published online:	21 December 2024

A new binary tagging-based model integrated with unsupervised tree hierarchy for relational triple extraction

Hua Yin

School of Digital Economy,
Guangdong University of Finance and Economics,
Guangzhou, Guangdong, China
Email: yinhua@whu.edu.cn

Zhiquan Chen*

Quality Assurance Center,
NetEase Games,
Guangzhou, Guangdong, China
Email: chenzhiquan@corp.netease.com
*Corresponding author

Abstract: Relational triple extraction (RTE) extracts entities and relations from unstructured text, serving as a crucial task for various NLP applications. Traditional pipeline approaches often face error propagation issues. The cascade binary tagging (CBT) method was introduced to mitigate this by linking entity recognition and relation extraction through shared parameters. However, CBT struggles with long-distance dependencies between subject and object entities, weakening performance. To address this, the COnRel model is proposed, integrating shallow and deep hierarchical information into the CBT framework. An unsupervised hierarchy parser generates multi-level tree structures, and a weight-transform method assigns higher weights to words closer in hierarchy to subject entities. This improves semantic representation of the subjects. In experiments, COnRel with shallow hierarchy outperforms the baseline model CasRel on the WebNLG dataset, and the full model, including deep hierarchy, excels on both WebNLG and NYT datasets, particularly for sentences 20–50 words in length.

Keywords: relational triple extraction; RTE; tree hierarchy; ordered neurons LSTM; BIO-like tagging scheme; cascade binary tagging scheme.

Reference to this paper should be made as follows: Yin, H. and Chen, Z. (2025) 'A new binary tagging-based model integrated with unsupervised tree hierarchy for relational triple extraction', *Int. J. Computational Science and Engineering*, Vol. 28, No. 1, pp.21–31.

Biographical notes: Hua Yin is an Associate Professor of Computer Science. She is currently the Associate Dean of the School of Digital Economy at Guangdong University of Finance and Economics. She received her PhD and MS from Wuhan University. Her main research interests include big data analysis, knowledge graphs, natural language processing, and legal artificial intelligence.

Zhiquan Chen received his MS from Guangdong University of Finance and Economics. His research interests include natural language understanding with LLMs and information extraction. He is currently working as a Software Engineer in NetEase Games, Guangzhou.

1 Introduction

Extracting entities and their semantic relations from unstructured texts is a fundamental task in text mining and knowledge graph construction (Xu et al., 2021; Cheng et al., 2024). It could be transformed to relational triple extraction (RTE) task. Given a sentence, RTE is to detect all possible relational triples (*subject, relation, object*), which include head entities (subjects), tail entities (objects) and their relationships. E.g., two relational triples, *(William Anders, born in, British Hong Kong)* and *(William Anders, selected by, NASA)*, would be extracted from the sentence shown in Figure 1.

There are two typical approaches for the RTE task: one is the pipeline approach consisting of two models for the two sub-tasks, named entity recognition (NER) and relation extraction (RE), respectively. Another is the joint approach that models the two sub-tasks jointly

(Yan et al., 2022). Traditional pipeline approaches ignore the bidirectional influences between the sub-tasks, which results in error propagation. Empirical study shows that properly designed joint approaches outperform the pipeline approaches (Ye et al., 2021). According to the text representation, feature-based approaches (Miwa and Sasaki, 2014) depend on the lexical, syntactic, and semantic manual features. However, obtaining such features was always domain-related and expensive. The neural network-based (NN-based) approaches, automatically extracting features, show better performances than the state-of-the-art feature-based approaches, but they still suffered from the *overlapping problem* (Fei, 2020; Liu et al., 2020). The models would miss some relation triples when different relations shared the same subjects or objects.

A series of cascade binary tagging schemes (Hu et al., 2020; Wei et al., 2020) were proposed to solve the above problem. They learn relation-specific taggers by modelling relations as functions that map subjects to objects (Wei et al., 2020). These methods allow different relations to share the same subjects and objects, but identifying the triples precisely is hard when the subjects are far from the objects in some long sentences. It is difficult for neural network models to learn long-range dependencies, because the forward and backward signals must traverse the long-range dependency path (Huang et al., 2021).

In computational linguistics, linguistic structures could be used to understand the rules regarding language use that native speakers know (Islam and Hossain, 2022). The constituency parse trees, a kind of linguistic structures, present the hierarchies of the sentences as trees. As is shown in Figure 2, subject 'William Anders' and object 'NASA' obtain a shorter distance in the tree view, corresponding to the longer distance in the origin view. Experiments had proved that the addition of linguistic structures could improve the classifiers' ability of processing long sentences (Mintz et al., 2009). It inspires us to construct hierarchy representation in a tree view to shorter the distance between subjects and objects.

Most common parsers, producing hierarchy information based on human annotated treebanks, are expensive and domain-dependent. The ordered neurons LSTM (ON-LSTM) (Shen et al., 2019), an unsupervised parser, utilises the life cycle of information to model the sentence hierarchies as trees shown in Figure 2. We introduce the hierarchy information from ON-LSTM into the tagging scheme to enhance the semantic understanding of the model.

The trees produced by ON-LSTM provide the hierarchical level information, but it cannot be directly used to compute the distances between subjects and objects. A weight-transform method is needed to map the trees into proper weights (stand for the possibilities of becoming an object) to search the most potential object. Furthermore, the binary tagger is a kind of linear structures, while hierarchy information is a kind of tree structures. For breaking the barrier of diverse structures, a new feature fusion method is considered to combine hierarchy feature with linear semantic features together.

The contributions of this paper are as follows:

- To improve the performance of RTE task in long sentences, a novel semantic-enhanced model COnRel is prompted. It provides a universal machinism to introduce the constituency parse trees provided by ON-LSTM into the CBT scheme.
- To make the hierarchy information to be computable, a weight-transform method is proposed. We think that the words with similar hierarchies in a constituency parse tree have more possibility to contain relations between them, so the tree hierarchy information is mapped to weight.
- To solve the problem from different representation of hierarchy feature and linear semantic feature, a feature fusion method is designed. Experiments, on two public datasets NYT and WebNLG, show the performance improvement of model even if the sentences elongate.

The rest of this paper is organised as follows. Section 2 reviews related research in RTE by making use of parse trees. Section 3 illustrated the methodology of the proposed model. Section 4 presents the experimental settings, main results and effectiveness analysis. Section 5 discusses the influences of embedding positions towards the COnRel model. Section 6 concludes the work.

2 Related work

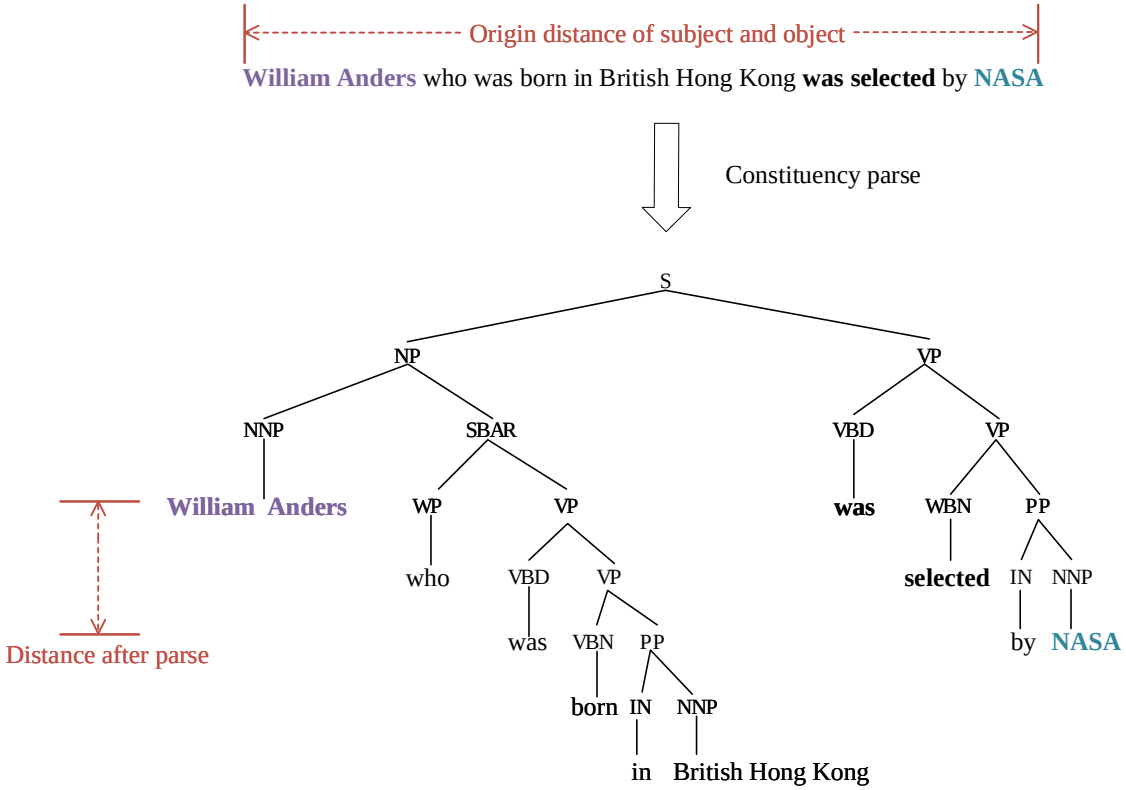
Traditional pipeline approaches suffer from overlapping problem because of the separation of NER and RE. Tagging scheme, as a way to connect NER (Zhang et al., 2022) and RE (Miwa and Bansal, 2016), was proposed and transformed the RTE task to the sequence labelling task. According to the types of tags used in tagging scheme, it could be classified into two kinds: BIO-like tagging scheme and cascade binary tagging scheme.

BIO-like tagging schemes take use of 'begin, inside, outside' and other tags ('end, single', etc.) to mark the positions of subjects and objects in the sentence. Relations are designed as subclass of BIO tags such as 'beginning of company founder', 'inside of company founder' and 'outside'. Zheng et al. (2017) first proposed a novel tagging scheme based on begin, inside, end, single (BIES) tags that unified tagging scheme of entities and relations. It was lack of the ability of processing sentences with overlapping relational triples because one tag could only appear one time in a sentence. Several researchers attempted to solve it. Dai et al. (2019) utilised position-attentive sequence labelling and BIES tagging to address the overlapping problem, which labelled multiple times for each word in sentences. Takanobu et al. (2019) proposed a hierarchical reinforcement learning framework with BIO tags to address the problem, which tagged entities in a low-level agent and relations in a high-level agent. Jia and Xiang (2020) used ON-LSTM to analyse the shallow hierarchy information for open-domain relation extraction task with BIOES tags.

Figure 1 The example of sentence and existing triples (see online version for colours)

Text	**William Anders** who was born in **British Hong Kong** was selected by **NASA**
Triples	(**William Anders**, born in, **British Hong Kong**)
	(**William Anders**, selected by, **NASA**)

Figure 2 The distance of subject and object in the origin view and the tree view (see online version for colours)

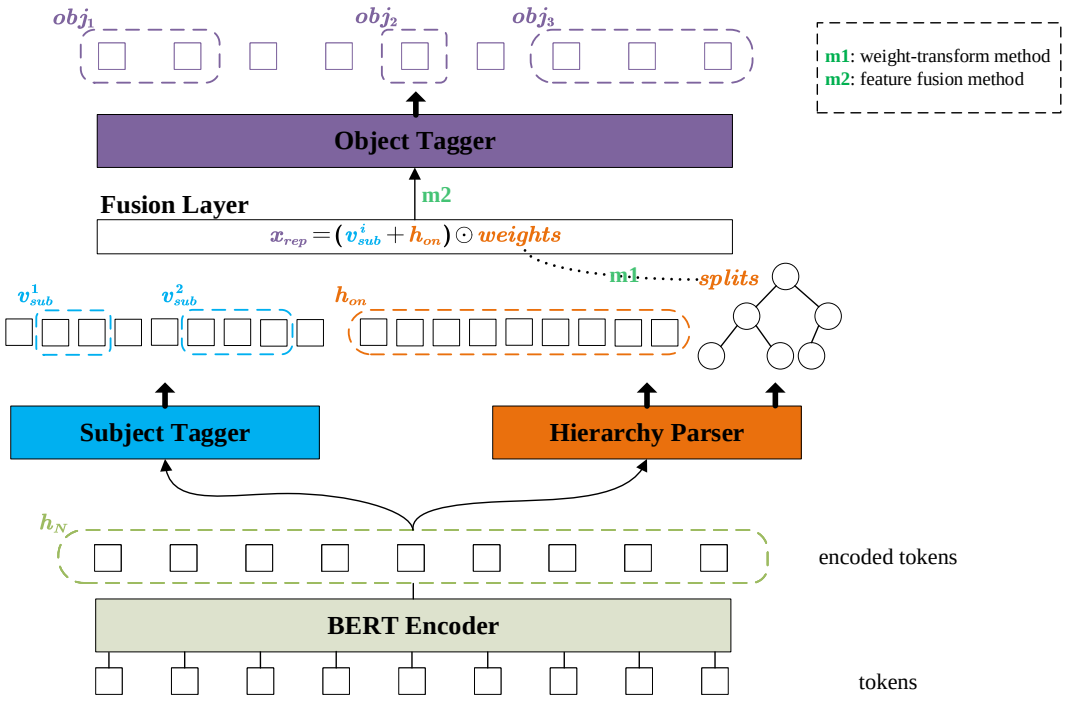


Note: In the constituency parse tree, the leaf nodes are words in sentences, and the non-leaf nodes which are in italics stand for the constituency of the words. 'S', 'NP' and 'VP' are the abbreviation of 'sentence', 'noun phrase' and 'verb phrase', respectively.

Cascade binary tagging schemes label words' spans by a pair of binary taggers, and usually first label subjects and then label relation and object together. These schemes do not need specific BIOES tags and perform better in speed. Wei et al. (2020) presented CasRel which shows great performance in dealing with overlapping relational triples. Ren et al. (2021) proposed confidence threshold-based cross entropy loss to alleviate the influence of imbalance data in the binary tagging scheme. Ren et al. (2022) proposed the unidirectional problem of such cascade architectures, which meant that the objects and relations extraction must be based on extraction of subjects firstly. They also proposed a bidirectional framework with subject-to-object binary tagger and object-to-subject binary tagger to address this problem.

The two types of tagging schemes both transform the RTE task to classification of tags. These tagging schemes face a same problem. When the distance of subject and object elongates, the performances of models would decline rapidly. Syntactic structures, such as dependency parse trees and constituency parse trees, are most used to guide the tagging process and relief this problem (Tuo et al., 2023; Zhu et al., 2024). Such approaches with dependency parse trees suffer from two drawbacks. One is dependency parse trees cannot reflect the word hierarchies, which comes from the property of dependency linguistic structures. They only display the dependency relationships between each two words rather than word hierarchies on the whole. The other is the lack of generalisation beyond the parse trees. For example, in the cross-domain RTE task, the domains of the training data and test data are different which often leads to a mismatch between the parse trees of the training data and the test data (Veyseh et al., 2020). The parse trees are generated by supervised learning which need provide labelled treebanks.

Figure 3 The overview of COnRel (see online version for colours)



Note: The BERT encoder encodes input sentence to tokens. Subject tagger tags out the spans of subjects as v_{sub}. Hierarchy parser works with encoded tokens and generates shallow hierarchy information h_{on} and deep hierarchy information $splits$, note that $splits$ could be visualised as a constituency parse tree. A weight-transform method converts $splits$ to multiple $weights$. A feature fusion method fuses all features and yields x_{rep} to tag objects.

It is feasible to produce word hierarchies by unsupervised constituency parse trees (Shen et al., 2019). Then the linguistic information from text will be utilised to enhance the semantic representation.

3 Methodology

We propose a semantic-enhanced model with binary tagging scheme named COnRel, the architecture of COnRel is shown in Figure 3. It consists of five components: a bidirectional encoder representations from transformers (BERT) encoder, a subject tagger, a hierarchy parser, a fusion layer and an object tagger.

3.1 Encoder

Firstly, the input sentences are segmented into tokens and transformed into word vectors (denoted as h_N) by a BERT-base model (Devlin et al., 2019). BERT is a pre-trained transformer-based model that utilises a bidirectional approach to understand the context of words in a sentence by considering both the preceding and following words.

The embedding process in BERT involves tokenisation, where sentences are divided into subword tokens using WordPiece tokenisation. Each token is then mapped to an embedding vector. Positional encodings are added to these token embeddings for retaining the order of tokens, and the combined embeddings are processed through multiple transformer encoder layers.

Each transformer encoder layer includes a self-attention mechanism and feed-forward neural networks. The self-attention mechanism computes attention scores for each token based on its relationships with all other tokens in the sequence, allowing the model to capture complex dependencies and contextual information.

The output of the BERT encoder is a sequence of hidden states h_N, where each hidden state corresponds to a token in the input sentence with enriched contextual information. These hidden states serve as the input for downstream tasks, leveraging the bidirectional nature of BERT to capture the nuances of natural language effectively.

The BERT embedding process ensures that words are represented by various semantic vectors based on their context, enhancing the model's ability to perform accurately on various natural language processing tasks.

3.2 Subject tagger

A pair of binary taggers are used to tag the start positions and end positions of subjects respectively. For obtaining the locations of subjects, the two kinds of positions by nearest start-end position pair are summed up. There are two groups of subject taggers. Each tagger is represented by 1 or 0 when it is or not the start (end) position. Denote subjects as s, $sstart$ indicates where s start and $send$ indicates where

s end. The probability of the i^{th} subject tagger is calculated as equations (1)–(2):

$$p^i_{\text{sstart}} = \sigma\left(W_{\text{sstart}} h^i_N + b_{\text{sstart}}\right), \tag{1}$$

$$p^i_{\text{send}} = \sigma\left(W_{\text{send}} h^i_N + b_{\text{send}}\right), \tag{2}$$

where h^i_N is the i^{th} token from encoded sentence h_N. $W_{(.)}$ stands for trainable weight, $b_{(.)}$ stands for bias and σ is sigmoid activation function. When p^i_{sstart}, p^i_{send} is greater than a threshold (here is 0.5), the tag in i^{th} would be '1'.

To identify the span of subjects LOC_{sub}, the likelihood function as equation (3) should be maximised. Eq. (4) shows the process of generating subject feature from encoded tokens.

$$\begin{aligned} &p_\theta(LOC_{sub} \mid h_N) \\ &= \prod_{t\in\{sstart,send\}} \prod_{i=1}^{L} \left(p^i_t\right)^{I\{y^i_t=1\}} \left(1-p^i_t\right)^{I\{y^i_t=0\}}, \end{aligned} \tag{3}$$

$$v^j_{sub} = h_N^{LOC^j_{sub}}, j \in [s, e], \tag{4}$$

where LOC_{sub} indicates the range of all subjects in the input sentence, v_{sub} is the subject feature, s and e are the start position and end position shown in LOC_{sub}, the number of subjects is $|LOC_{sub}|$.

3.3 Hierarchy parser

For finding the potential objects by the constituency structures of the sentences, ON-LSTM is applied to parse the sentences and get the hierarchy information. The hierarchy information includes two kinds, one is the vectors (denoted as h_{on}) which contain the long-range information according to a special updating mechanism of ordered neurons (Shen et al., 2019), and the other is the hierarchical structures (denoted as $splits$) which could be visualised as binary constituency parse trees. Both of them are generated by the hierarchy parser ON-LSTM.

h_{on} are the hidden states of ON-LSTM, which are semantic representations of embedded word vector h_N. Operations of producing h_{on} are as follows:

$$h^0_{on} = c_0 = 0, \tag{5}$$

$$h^t_{on} = ONCell(h^t_N, h^{t-1}_{on}, c_{t-1}), t \in [1, N]. \tag{6}$$

ON-LSTM has isometric cells with transformer blocks. Denote transformer block as $Trans(x)$ and ON-LSTM cell as $ONCell(x_t, h_{t-1}, c_{t-1})$. Equations (5)–(6) present the update rule of ON-LSTM cell:

1 Firstly fill two matrices with 0 as initial states.

2 Then feed each cell with a part of h_N (the result of word embedding of BERT) and ON-LSTM hidden state and ON-LSTM cell state from previous time step.

3 At last, gather all cells. As a result, h_{on} is produced. Note that the recurrent structures are helpful to represent sentence features in a syntactic avenue.

Here we briefly introduce $splits$. $splits$ are comprised of d^f_t which are float numbers standing for hierarchies of words, i.e., a greater $\hat{d}^f_t$ means the word is in higher level. And $splits$ are vectors gathered all $\hat{d}^f_t$ by cell order as equation (7):

$$splits = [\hat{d}^f_t \in ONCell_i], i \in [1, N], \tag{7}$$

d^f_t are split points of sentence. Smooth estimate $\hat{d}^f_t$ are computed to take place of d^f_t. d^f_t is produced as equation (8):

$$\begin{aligned} p_f &= softmax\left(W_f h^t_N + U_f h_{t-1} + b_f\right), \\ \hat{d}_f &= argmax\left(p_f\right), \end{aligned} \tag{8}$$

where p_f is the probability distribution of split points indicated by ON-LSTM master forget gate $\tilde{f}$. p_f indicates the hierarchies of history information and ON-LSTM use it to distinguish how much input information to forget in a timestep. W_f and U_f are the trainable weights and b_f is the trainable bias in master forget gate. h^t_N are the t^{th} vectors of encoded tokens and h_{t-1} are the hidden states in $t-1$ timestep. Through a top-down greedy parsing algorithm proposed by Shen et al. (2018), $splits$ could be visualised as a binary parse tree. For a more comprehensive description of ON-LSTM and $\hat{d}^f_t$, we refer readers to Shen et al. (2019).

3.4 Fusion layer

To obtain the ability of perceive sentence hierarchies, the introduction of constituency parse tree $splits$ is in need. Nevertheless, $splits$ that stand for hierarchy structures of sentences could be applied into neither h_N nor v_{sub} directly. The reason is a word with a higher level than other words does not mean it is more possible to be an object. As the hypothesis we proposed before, the words whose levels are closer contain higher likelihood that relations exist between them. According to this hypothesis, a weight-transform method is presented for granting higher weights to words adjacent to the subjects in Section 3.4.1. Furthermore, subject features v_{sub}, semantic features h_{on} and level features $weights$ would be fused by a feature fusion method. The fused features are denoted as x_{rep}, and the method is described in Section 3.4.2.

3.4.1 Weight-transform method

A weight-transform method transforms constituency parse tree $splits$ to $weights$ by paying more attention to words whose hierarchies are nearer to subjects. To assign higher weight to words that are more adjacent to subjects, we calculate the absolute distance $dist$ of every word towards the first subject in a sentence. A kind of normalisation approach is applied to keep $dist \in [1, 2]$. Further of all, a

scaling factor θ is introduced to adapt the scaling degree of $weights$, which makes $weights$ appropriate. Above processes are shown as equation (9):

$$\begin{aligned} dist_i &= \frac{|\hat{d}_i - \hat{d}_{sub}|}{(\hat{d}_{max} - \hat{d}_{min})/2} \\ w_i &= -dist_i + 2 \\ \theta &= \sigma(W_a w + b_a) \\ weights &= [\theta w_i], i \in [1, N], \end{aligned} \tag{9}$$

where $\hat{d}_i$ presents the level of i^{th} word, $\hat{d}_{sub}$ is the level of the first subject whose indices are same with v_{sub} in $splits$, $\hat{d}_{max}, \hat{d}_{min}$ are the maximum and minimum number in $splits$. w stand for temporary vectors which contain the weight without scaling. And $weights$ are the final weight vectors which indicate how near each word is far from the first subject in the sentence. After the transform operation, the hierarchy structures $splits$ would be converted to $weights$ that could be multiplied to sentence vectors directly. Formally, the method is as follows:

Algorithm 1 Weight-transform method

Input: $splits = [d_0, d_1, ..., d_{n-1}]$: d_i stands for the hierarchies of the i^{th} token in a sentence; in_sub: index of the first subject in a sentence
Output: $weights = [w_0, w_1, ..., w_{n-1}]$: the weights stand for the degree of how much the token is near from the subject token

$dist = []$
$w = []$
$d_{max} = \text{Max}(d_0, d_1, ..., d_{n-1})$
$d_{min} = \text{Min}(d_0, d_1, ..., d_{n-1})$
$d_{in\_sub} = splits[in\_sub]$
for $i = 0$ to $n - 1$ **do**
    $dist[i] = 2 * |splits[i] - d_{in\_sub}|/(d_{max} - d_{min})$
    $w[i] = -dist[i] + 2$
end for
$\theta = \text{FullConnectLayer}(w)$
$weights = []$
for $i = 0$ to $n - 1$ **do**
    $weights[i] = \theta w[i]$
end for
return $weights$

3.4.2 Feature fusion method

The object tagging process is based on three kinds of features, includes: subject feature, shallow hierarchy feature h_{on} and transformed deep hierarchy feature $weights$. The hierarchy feature can be used to target the potential objects especially in long and difficult sentences. A feature fusion method is proposed to fuse the above features to get the fused feature representation x_{rep}. The process of generating x_{rep} could be described as following: subject taggers tag out the spans of subjects, and obtain the subjects feature vectors v_{sub}. In the meanwhile, hierarchy parser yields shallow hierarchy feature h_{on} and deep hierarchy feature $splits$ to improve the model's understanding of sentence hierarchies. To introduce $splits$ into word vectors, $splits$ would be transformed to $weights$ according to weight-transform method. x_{rep} is calculated as equation (10):

$$x_{rep} = \left(h_{on} + v^k_{\text{sub}}\right) \odot weights, \tag{10}$$

where v^k_{sub} is the k^{th} subject feature extracted by subject tagger. Element-wise multiplication is applied to merged features and $weights$. The feature fusion algorithm is as follows:

Algorithm 2 Feature fusion method

Input: h_N: BERT encoded word vector; v_{sub}: the segments of vectors corresponded to subjects; s: mask vectors of subjects, s[i] would be '1' if i^{th} word is a subject
Output: x_{rep}: semantic-enhanced feature vector

$h_{on} = []$
$c = []$
$h_{on}[0] = c[0] = 0$
for $t = 1$ to n **do**
    $h_{on}[t] = \text{ONCELL}(h_N[t], h_{on}[t-1], c[t-1])$
end for
$isub = 0$
for $i = 0$ to $n - 1$ **do**
    if $s[i] == 1$ **then**
        $isub = i$
        break
    end if
end for
$weights = \text{WeightTransformMethod}(splits, isub)$
$x_{rep} = []$
for $i = 0$ to $n - 1$ **do**
    $x_{rep}[i] = (h_{on}[i] + v_{sub}[i]) * weights[i]$
end for
return x_{rep}

3.5 Object tagger

Object tagger also contains a pair of taggers. The probability of the i^{th} tagger is calculated as equations (11)–(12).

$$p^i_{\text{ostart}} = \sigma\left(W_{\text{ostart}} x^i_{rep} + b_{\text{ostart}}\right), \tag{11}$$

$$p^i_{\text{oend}} = \sigma\left(W_{\text{oend}} x^i_{rep} + b_{\text{oend}}\right), \tag{12}$$

where x^i_{rep} is the i^{th} token of fused feature x_{rep}. $W_{(.)}$ stands for trainable weight, $b_{(.)}$ stands for bias and σ is sigmoid activation function. p^i_{ostart}, p^i_{oend} is the probability value of i^{th} token, which indicates object's start and end position.

Likewise, objects are tagged through maximising likelihood function equation (13), and each pair of object taggers are only for a specific relation.

$$\begin{aligned} &p_{\phi_r}(LOC_{obj} \mid x_{rep}) \\ &= \prod_{t \in \{ostart, oend\}} \prod_{i=1}^{L} \left(p^i_t\right)^{I\{y^i_t = 1\}} \left(1 - p^i_t\right)^{I\{y^i_t = 0\}}, \end{aligned} \tag{13}$$

where LOC_{obj} indicates the range of all subjects in the input sentence, and $|LOC_{obj}|$ is the number of objects.

4 Experiments

4.1 Experiment setting

4.1.1 Datasets

Two datasets, NYT (Riedel et al., 2010) and WebNLG (Gardent et al., 2017), are chosen to test COnRel. The overview of datasets is as Table 1. NYT was produced by the distant supervision method. There are 56,195 sentences of 2005–2006 for training, 5,000 sentences of 2007 for testing, and 24 predefined relation types in total. WebNLG was proposed for an natural language generation (NLG) task of DBpedia. There are 5,000 sentences for training and 703 sentences for testing in 246 relation types. To explore the performance of RTE in sentences with various lengths, the original datasets are split into several sub-datasets by word length.

Table 1 Sentence number and relation number of datasets

Dataset	*Train*	*Valid*	*Test*	*Relation*
NYT	56,195	5,000	5,000	24
WebNLG	5,019	500	703	171*

Note: *The relation number of WebNLG is not correct because the work of Zeng et al. (2018) and Wei et al. (2020) is based on subset rather origin WebNLG. The relation number has been fixed.

4.1.2 Baselines

We compare our model with five models: NovelTagging (Zheng et al., 2017), an end-to-end method using novel tagging scheme; CopyR (Zeng et al., 2018), a Seq2Seq model that utilises copy mechanism to address overlapping problem; GraphRel (Fu et al., 2019), jointly extract entities and relations with GCN; CopyR$_{RL}$ (Zeng et al., 2019), using reinforcement learning to learn triples order; CasRel (Wei et al., 2020), using binary taggers to tag entities and relations.

Our COnRel model is implemented by TensorFlow 1.13 and Adam. It utilises BERT-base English version (available at https://huggingface.co/bert-base-cased) with 110M parameters and tokenises with cased. The max length is set to 100 and the learning rate is set to 1e-5. The batch size is set as 6/32 in NYT/WebNLG. It is trained by Tesla V100 on each dataset at most 100 epochs with an early stopping strategy.

4.1.3 Evaluation metrics

For fair competition, the evaluation metrics are same as those used in NovelTagging, CopyRE and CasRel. Standard precision (Prec.), recall (Rec.) and F1-score are in used to evaluate the results.

Table 2 Main results of different method on NYT and WebNLG

Method	*NYT*			*WebNLG*		
	Prec.	*Rec.*	*F1*	*Prec.*	*Rec.*	*F1*
NovelTagging	62.4	31.7	42.0	52.5	19.3	28.3
CopyR$_{OneDecoder}$	59.4	53.1	56.0	32.2	28.9	30.5
CopyR$_{MultiDecoder}$	61.0	56.6	58.7	37.7	36.4	37.1
GraphRel$_{1p}$	62.9	57.3	60.0	42.3	39.2	40.7
GraphRel$_{2p}$	63.9	60.0	61.9	44.7	41.1	42.9
CopyR$_{RL}$	77.9	67.2	72.1	63.3	59.9	61.6
CasRel$_{LSTM}$	84.2	83.0	83.6	86.9	80.6	83.7
CasRel	89.7	89.5	89.6	**93.4**	90.1	91.8
COnRel$_{shallow}$	86.6	91.1	88.8	93.1	**92.8**	**93.0**
COnRel$_{shallow+deep}$	**90.0**	**91.6**	**90.8**	92.1	92.1	92.1

Note: COnRel offers shallow version and shallow+deep version. COnRel$_{shallow}$ means only h_{on} is been employed and the COnRel$_{shallow+deep}$ take use of h_{on} and $splits$ together.

4.2 Experimental results

4.2.1 Main result

For testifying the influence of introducing shallow and deep hierarchy information, we compare COnRel$_{shallow}$ and COnRel$_{shallow+deep}$ with the baselines. The experimental results are presented as Table 2. Among all the baselines, CasRel performs best on both datasets. Therefore we only compare our results with CasRel. In NYT dataset, COnRel$_{shallow}$ is 1.6% higher than CasRel on recall, but lower on precision and F-1 score. However, it performs better on recall and F-1 score, and has close precision in WebNLG dataset. Observed from the performances of COnRel$_{shallow+deep}$ on both datasets, they apparently exceed CasRel on most of the evaluation metrics. It indicates COnRel outperforms most of the baselines. The addition of hierarchy information exactly improves the performance by increasing more semantic features to the model.

We split the original datasets to several sub-datasets according to different lengths of the sentences. The length gap is set to 10. Because the max length of sentences in NYT and WebNLG dataset is respectively 100 and 80 words, there are 10 sub-datasets in NYT and 8 sub-datasets in WebNLG. Figures 4 and 5 illustrate that COnRel shows better performance in sentences of varying length segments. With the elongating of sentences, the F1-score shows a slower decline, which indicates that COnRel has a stronger ability of processing long sentences. And it is noticed that we employ COnRel$_{shallow}$ splits version, because it works better in NYT dataset. One reason why COnRel$_{shallow}$ is greater than COnRel$_{shallow+deep}$, presumably, is that WebNLG is too narrow to adapt deep model. The number of data in WebNLG is much fewer than NYT (more than 11 times). Deeper architectures need more data. Another reason is that the sentences in NYT are obviously difficult and longer than in WebNLG. In such a situation, COnRel$_{shallow}$ model may work better in WebNLG.

Figure 4 The experimental result of CasRel and COnRel (shallow + deep) on NYT dataset in different length sentence (see online version for colours)

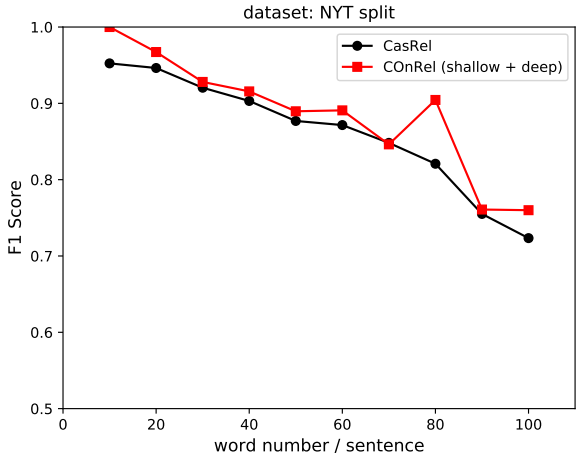
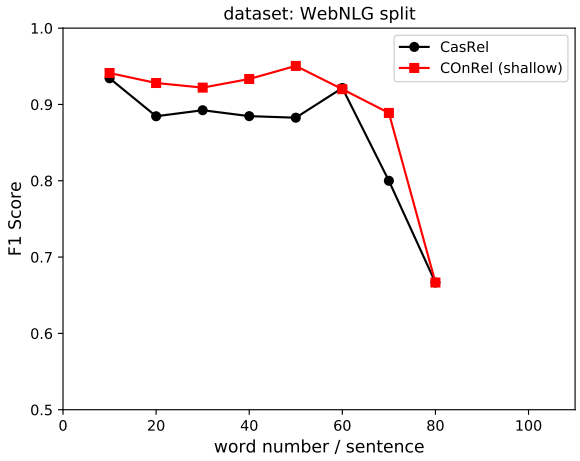


Figure 5 The experimental result of CasRel and COnRel (shallow) on WebNLG dataset in different length sentence (see online version for colours)



4.2.2 *Ablation study*

There are two main components in the proposed model:

1 h_{on}, the shallow hierarchy information

2 $weights$, the weights that are transformed from the deep hierarchy information yielded from ON-LSTM.

To analysis the contributions and effects of the two components, we perform ablation study on the NYT and WebNLG datasets.

Table 3 reports the performance of the different components. It shows that in NYT dataset, model with h_{on} and weights (COnRel) gets the best recall and F1-score, and the precision is almost the best. Apart from that, when h_{on} or weights are removed from the model, the precision, recall and F1-score significantly degrade. In WebNLG, model with h_{on} and weights indicates a more balanced precision and recall than the other groups. The F1-score is also close to the best one. In brief, the ablation study proves the effects of the two components.

Table 3 Ablation study on NYT and WebNLG datasets

System	*NYT*			*WebNLG*		
	Prec.	*Rec.*	*F1*	*Prec.*	*Rec.*	*F1*
COnRel	90.0	**91.6**	**90.8**	92.1	92.1	92.1
h_{on}	**90.1**	89.4	89.8	92.0	**92.4**	92.2
weights	89.6	90.3	90.0	**92.8**	92.0	**92.4**
h_{on} – weights	89.3	89.6	89.4	91.1	92.4	91.7

Note: The components listed on each row are removed from origin model.

4.2.3 *The effectiveness analysis of the generated weights*

To explore if the $weights$ take positive effects to the tagging process, the following experiment is designed. Given a sentence, we recognise the first subject's index, and then predict the index of the object $index_{pred\_obj}$. We get the bias of distance as $distance_{bias}$ from the subtraction between $index_{pred\_obj}$ and $index_{real\_obj}$. That is, for a sentence of length n,

$$distance_{bias} = |index_{pred\_obj} - index_{real\_obj}|$$

The $distance_{bias}$ could be regarded as a distance from a 'potential object' to a real object, thus a smaller bias means that the 'potential object' is closer to the real object, therefore a higher weights should be assign. As is said before, $splits$ have been transformed to $weights$. The word with a higher weight has a closer level with the subject. Theoretically, if the word with a higher $weights$ obtains a lower $distance_{bias}$, the $weights$ may take a positive effect.

There are 131,137 sample words in NYT, 4,606 sample words in WebNLG in this experiment. Figure 6 shows the result of experiments. The horizontal axis of the figure indicates the transformed $distance_{bias}$, and the vertical axis indicates the $weights$. Two datasets show analogical distribution of $distance_{bias}$ and $weights$: dots are assembled in the left top corner, there are few dots in the left bottom corner and right top corner, hardly dots in the right bottom, which illustrated $weights$ does not have a positive correlation with distance. Figure 7 illustrates WebNLG has a similar distribution of NYT in this experiment. The result of this experiment could be concluded as: if a word has a higher likelihood to be an object, it is more possible to take a higher weight. The reason why data distribute unevenly in most sentences in the dataset is not balanced, so the covering to the right side is sparser. And a word closer to each side of the sentence has a lower probability to be an object.

5 Discussion

There are two ways to embed the hierarchy parser into the model: before-embedding and behind-embedding. COnRel before-embedding encodes the parser before

subject tagger, i.e., subject tagger is going to take use of h_{on} rather h_N, while COnRel behind-embedding encodes the parser after subject tagger. The foresaid experiments are with behind-embedding way. We also explored the before-embedding way. COnRel before-embedding has the same hyper parameters with COnRel behind-embedding except embedding position. An interesting phenomenon appears: the embedding position of hierarchy parser has a crucial effect on RTE tasks. In the beginning, we believe that before-embedding would bring more syntactic features to both subject and object tagger, thus it would perform better. But as a result shown in Figures 8 and 9, COnRel before-embedding has almost no positive influence on tagging compared with COnRel behind-embedding.

Figure 6 Correlation of hidden state distance from real object and weight in NYT (see online version for colours)

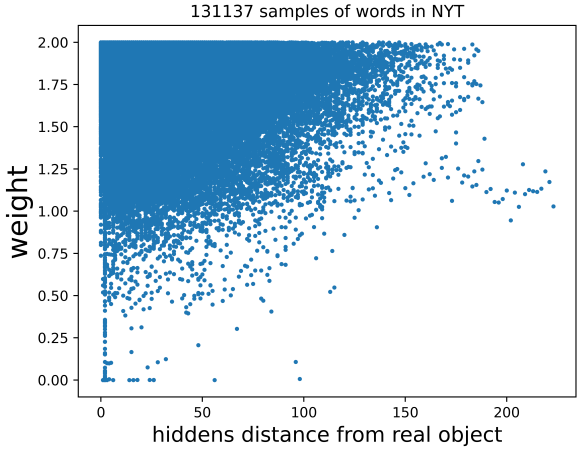
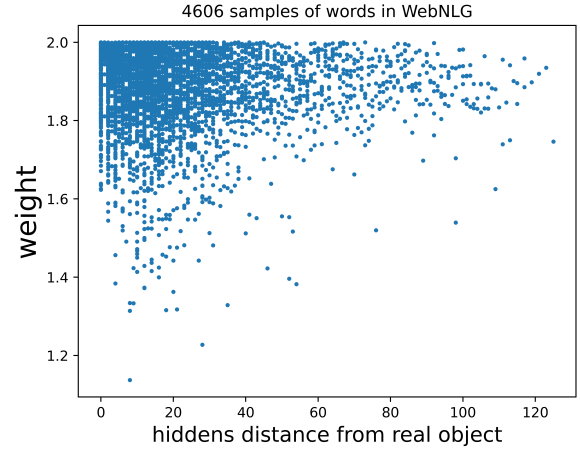


Figure 7 Correlation of hidden state distance from real object and weight in WebNLG (see online version for colours)



The subject tagging task is more simple than object tagging, which focus on both relations and objects. Therefore it is enough to tag subjects only with BERT and binary taggers, while it is not enough for tagging objects at the same time. That is why embedding hierarchy parser behind the subject tagger works.

6 Conclusions

In this paper, to improve the performance of RTE task especially in long sentences, we proposed a tagging-based RTE model with the tree hierarchy information. A weight-transform method and a feature fusion method are presented to overcome the hardship of introducing the information. The hierarchy information is extracted by ON-LSTM by an unsupervised way, which is more domain-independent than traditional methods. Experimental results on two open datasets indicate the model outperforms most baseline models.

Our future work aims to improve the way of utilising the tree hierarchies. In the proposed model, we apply all of the hierarchies to find out the potential objects, whose levels may be overmuch and misleading. We will try to pick several rough hierarchies rather all of them to improve the performance of the model.

Figure 8 Before-embedding and behind-embedding in NYT dataset with different length sentence (see online version for colours)

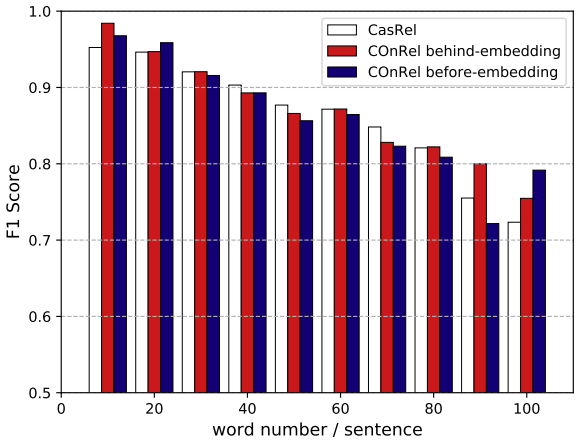
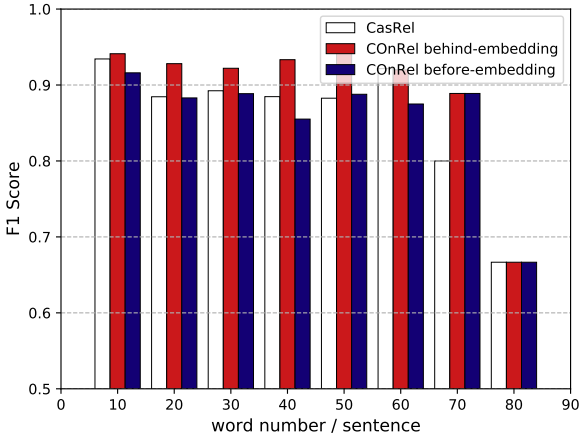


Figure 9 Before-embedding and behind-embedding in WebNLG dataset with different length sentence (see online version for colours)



Acknowledgements

This work was supported by the Humanity and Social Science Youth Foundation of Ministry of Education of China (21YJCZH202), the Innovation Team Project of Higher Education of Guangdong Province (2022WCXTD008) and Commission Project of Guangdong Province Law Society (GDLS(2024)C12).

References

Cheng, J., Zhang, T., Zhang, S., Ren, H., Yu, G., Zhang, X., Gao, S. and Ma, L. (2024) 'A cascade dual-decoder model for joint entity and relation extraction', *IEEE Transactions on Emerging Topics in Computational Intelligence*, pp.1–13.

Dai, D., Xiao, X., Lyu, Y., Dou, S., She, Q. and Wang, H. (2019) 'Joint extraction of entities and overlapping relations using position-attentive sequence labeling', *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, pp.6300–6308.

Devlin, J., Chang, M-W., Lee, K. and Toutanova, K. (2019) *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, arXiv:1810.04805 [cs].

Fei, H. (2020) 'Boundaries and edges rethinking: an end-to-end neural model for overlapping entity relation extraction', *Information Processing and Management*, Vol. 57, No. 6, p.102311.

Fu, T-J., Li, P-H. and Ma, W-Y. (2019) 'GraphRel: modeling text as relational graphs for joint entity and relation extraction', *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Florence, Italy, pp.1409–1418.

Gardent, C., Shimorina, A., Narayan, S. and Perez-Beltrachini, L. (2017) 'Creating training corpora for NLG micro-planners', *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Vancouver, Canada, pp.179–188.

Hu, Z., Yin, H., Xu, G., Zhai, Y., Pan, D. and Liang, Y. (2020) 'An empirical study on joint entities-relations extraction of Chinese text based on BERT', *Proceedings of the 2020 12th International Conference on Machine Learning and Computing*, ACM, Shenzhen, China, pp.473–478.

Huang, W., Mao, Y., Yang, L., Yang, Z. and Long, J. (2021) 'Local-to-global GCN with knowledge-aware representation for distantly supervised relation extraction', *Knowledge-Based Systems*, Vol. 234, No. C, p.107565.

Islam, M.A. and Hossain, M.S. (2022) 'A comprehensive understanding of popular machine translation evaluation metrics', *International Journal of Computational Science and Engineering*, Vol. 25, No. 5, pp.467–478.

Jia, S. and Xiang, Y. (2020) *Hybrid Neural Tagging Model for Open Relation Extraction*, arXiv:1908.01761 [cs].

Liu, T., Lin, X., Jia, W., Zhou, M. and Zhao, W. (2020) 'Regularized attentive capsule network for overlapped relation extraction', in Scott, D., Bel, N. and Zong, C. (Eds.): *Proceedings of the 28th International Conference on Computational Linguistics*, International Committee on Computational Linguistics, Barcelona, Spain (Online), pp.6388–6398.

Mintz, M., Bills, S., Snow, R. and Jurafsky, D. (2009) 'Distant supervision for relation extraction without labeled data', *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, Association for Computational Linguistics, Suntec, Singapore, pp.1003–1011.

Miwa, M. and Bansal, M. (2016) 'End-to-end relation extraction using LSTMs on sequences and tree structures', *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Berlin, Germany, pp.1105–1116.

Miwa, M. and Sasaki, Y. (2014) 'Modeling joint entity and relation extraction with table representation', *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, Doha, Qatar, pp.1858–1869.

Veyseh, A.P.B., Dernoncourt, F., Dou, D. and Nguyen, T.H. (2020) 'Exploiting the syntax-model consistency for neural relation extraction', *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Online, pp.8021–8032.

Ren, F., Zhang, L., Yin, S., Zhao, X., Liu, S. and Li, B. (2021) 'A Conditional Cascade Model for Relational Triple Extraction', *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, ACM, Virtual Event, Queensland, Australia, pp.3393–3397.

Ren, F., Zhang, L., Zhao, X., Yin, S., Liu, S. and Li, B. (2022) 'A Simple but Effective Bidirectional Framework for Relational Triple Extraction', *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, ACM, Virtual Event, AZ USA, pp.824–832.

Riedel, S., Yao, L. and McCallum, A. (2010) 'Modeling relations and their mentions without labeled text', in Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J.M., Mattern, F., Mitchell, J.C., Naor, M., Nierstrasz, O., Rangan, C.P., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M.Y., Weikum, G., Balcázar, J.L., Bonchi, F., Gionis, A. and Sebag, M. (Eds.): *Machine Learning and Knowledge Discovery in Databases*, Vol. 6323, pp.148–163, Springer Berlin Heidelberg, Berlin, Heidelberg.

Shen, Y., Lin, Z., Huang, C-W. and Courville, A. (2018) *Neural Language Modeling by Jointly Learning Syntax and Lexicon*, arXiv:1711.02013 [cs].

Shen, Y., Tan, S., Sordoni, A. and Courville, A. (2019) *Ordered Neurons: Integrating Tree Structures into Recurrent Neural Networks*, arXiv:1810.09536 [cs].

Takanobu, R., Zhang, T., Liu, J. and Huang, M. (2019) 'A hierarchical framework for relation extraction with reinforcement learning', *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, pp.7072–7079.

Tuo, M., Yang, W., Wei, F. and Dai, Q. (2023) 'A novel Chinese overlapping entity relation extraction model using word-label based on cascade binary tagging', *Electronics*, Vol. 12, No. 4, p.1013.

Wei, Z., Su, J., Wang, Y., Tian, Y. and Chang, Y. (2020) 'A novel cascade binary tagging framework for relational triple extraction', in Jurafsky, D., Chai, J., Schluter, N. and Tetreault, J. (Eds.): *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Online, pp.1476–1488.

Xu, K., Wang, P., Chen, X., Luo, X. and Gao, J. (2021) 'Causal event extraction using causal event element-oriented neural network', *International Journal of Computational Science and Engineering*, Vol. 24, No. 6, p.621.

Yan, Z., Jia, Z. and Tu, K. (2022) 'An empirical study of pipeline vs. joint approaches to entity and relation extraction', in He, Y., Ji, H., Li, S., Liu, Y. and Chang, C-H. (Eds.): *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, Association for Computational Linguistics, Online only, pp.437–443.

Ye, H., Zhang, N., Deng, S., Chen, M., Tan, C., Huang, F. and Chen, H. (2021) 'Contrastive triple extraction with generative transformer', *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35, No. 16, pp.14257–14265.

Zeng, X., He, S., Zeng, D., Liu, K., Liu, S. and Zhao, J. (2019) 'Learning the extraction order of multiple relational facts in a sentence with reinforcement learning', *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Association for Computational Linguistics, Hong Kong, China, pp.367–377.

Zeng, X., Zeng, D., He, S., Liu, K. and Zhao, J. (2018) 'Extracting relational facts by an end-to-end neural model with copy mechanism', *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Melbourne, Australia, pp.506–514.

Zhang, S., Zhu, H., Xu, H., Zhu, G. and Li, K-C. (2022) 'A named entity recognition method towards product reviews based on BiLSTM-attention-CRF', *International Journal of Computational Science and Engineering*, Vol. 25, No. 5, pp.479–489.

Zheng, S., Wang, F., Bao, H., Hao, Y., Zhou, P. and Xu, B. (2017) *Joint Extraction of Entities and Relations Based on a Novel Tagging Scheme*, arXiv:1706.05075 [cs].

Zhu, X., Gao, W., Yu, Y., Zhang, L. and Deng, H. (2024) 'Syntax-based argument correlation-enhanced end-to-end model for scientific relation extraction', *Neurocomputing*, Vol. 586, No. C, p.127639, ISSN: 0925-2312.