PLOS ONE

RESEARCH ARTICLE

A mutually-exclusive binary cross tagging framework for joint extraction of entities and relations

Xuan Liu[1,2]*, **Wanru Du**[1,2], **Xiaoyin Wang**[2], **Ruiqun Li**[2], **Pengcheng Sun**[2], **Xiaochuan Jing**[2]

1 China Aerospace Academy of Systems Science and Engineering, Beijing, China, **2** Aerospace Hongka Intelligent Technology (Beijing) CO., LTD., Beijing, China

* liux@cau.edu.cn

Check for updates

OPEN ACCESS

Citation: Liu X, Du W, Wang X, Li R, Sun P, Jing X (2022) A mutually-exclusive binary cross tagging framework for joint extraction of entities and relations. PLoS ONE 17(1): e0260426. https://doi.org/10.1371/journal.pone.0260426

Editor: Sergio Consoli, European Commission, ITALY

Received: April 26, 2021

Accepted: November 9, 2021

Published: January 21, 2022

Copyright: © 2022 Liu et al. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All data are available and links is: https://www.kaggle.com/hongka2021/binaryct.

Funding: This research was financially supported by Aerospace Hongka Intelligent Technology (Beijing) CO., LTD. The funder provided support in the form of salaries for authors X.L and W.D, but did not have any additional role in the study design, data collection and analysis, decision to publish, or preparation of the manuscript. The specific roles of these authors are articulated in the 'author contributions' section.

Abstract

Joint extraction from unstructured text aims to extract relational triples composed of entity pairs and their relations. However, most existing works fail to process the overlapping issues that occur when the same entities are utilized to generate different relational triples in a sentence. In this work, we propose a mutually exclusive Binary Cross Tagging (BCT) scheme and develop the end-to-end BCT framework to jointly extract overlapping entities and triples. Each token of entities is assigned a mutually exclusive binary tag, and then these tags are cross-matched in all tag sequences to form triples. Our method is compared with other state-of-the-art models in two English public datasets and a large-scale Chinese dataset. Experiments show that our proposed framework achieves encouraging performance in F1 scores for the three datasets investigated. Further detailed analysis demonstrates that our method achieves strong performance overall with three overlapping patterns, especially when the overlapping problem becomes complex.

Introduction

Relation extraction (RE) from natural language texts is widely studied in information extraction (IE). RE aims to detect specific types of entities from unstructured text and the semantic relations between entity pairs. It is the basis and data source for building knowledge bases (KBs) such as YAGO [1], Freebase [2], DBpedia [3] and NELL [4]. The relation is often formalized as a relational triple T, which consists of two entities ($E1$, $E2$) and the semantic relation Rs between them: $T = <E1, Rs, E2>$. For example: <Beijing, the capital of, China>.

Early works on RE adopted a pipeline approach [5, 6]. First, entity recognition is conducted using a named entity recognition (NER) module, then relations are further classified for each entity pair with a relation classification (RC) module [7]. Since errors in the early stage cannot be fixed in the following stage, such an approach suffers from propagation errors. Subsequent research introduced a joint learning method to ease error extraction. The joint extracting methods include feature-based models [8–11] and neural network models [12–15], and are able to extract and leverage the deep associations between entities and relations at the same

Competing interests: The authors declare that there are no conflicts of interest on the commercial affiliation, employment, consultancy, patents, products in development and marketed products regarding the publication of this paper. The commercial affiliation does not alter our adherence to all PLOS ONE policies on sharing data and materials.

time. However, in situations where sentences contain multiple overlapping triples, the effect of existing models fail to meet expectations.

The relational triples in the sentence have been divided into three types according to the overlap degree: *Normal*, *EntityPairOverlap* (EPO) and *SingleEntityOverlap* (SEO) [14] as shown in Fig 1. A sentence belongs to the EPO pattern if some of the triples have overlapping entity pairs. For instance, the two different relational triples <[Washington], capital, [United States]> and <[Washington], country, [United States]> share the same entity pairs. Alternatively, a sentence belongs to the SEO pattern if two triples contain at least one overlapping entity and do not share the same entity pairs. Consider the example sentence (as shown in Fig 1): "[Jackie R. Brown] was born in [Washington], and now lives in [New York].". In this case, two different relational triples share the single entity [Jackie R. Brown]. In an even more sophisticated case, "[Jackie R. Brown] was born in [Washington], the capital city of the [United States of America].", every single entity has an overlapping issue.

Subsequent works proposed some novel tagging schemes to handle the overlapping issues. Zeng et al. [14] introduced a sequence-to-sequence model with a copy mechanism to extract triples and to further investigate the impact of the extraction order [16]. Fu et al. [15] also built a Graph Convolutional Networks (GCNs) based model to study the overlapping triple problem. Nevertheless, these models predicted a single relation class for an entity pair and were not able to extract all the triples in a sentence. As a result, the overlapping triple is not addressed.

In our work, we aimed to design a framework that can extract entities and relations from Normal, EPO and SEO sentences while addressing the challenge mentioned above. Unlike previous tagging methods, we propose a novel mutually exclusive binary cross tagging scheme and developed the end-to-end BCT framework to improve the RE performance.

First, each token in a sentence is identified by the mutually exclusive binary tags, which represent the word position in the entity span. These binary tags are further distinguished by different predicates and subject/object dichotomy. Second, the binary tags of the two entities are cross-matched in all tag sequences to form the entity pairs that share the same relation type. If two or more triples have the same relation in an input sentence, the triples are formed based on the nearest principle. Finally, we establish a tagging scheme that can treat the task as token-

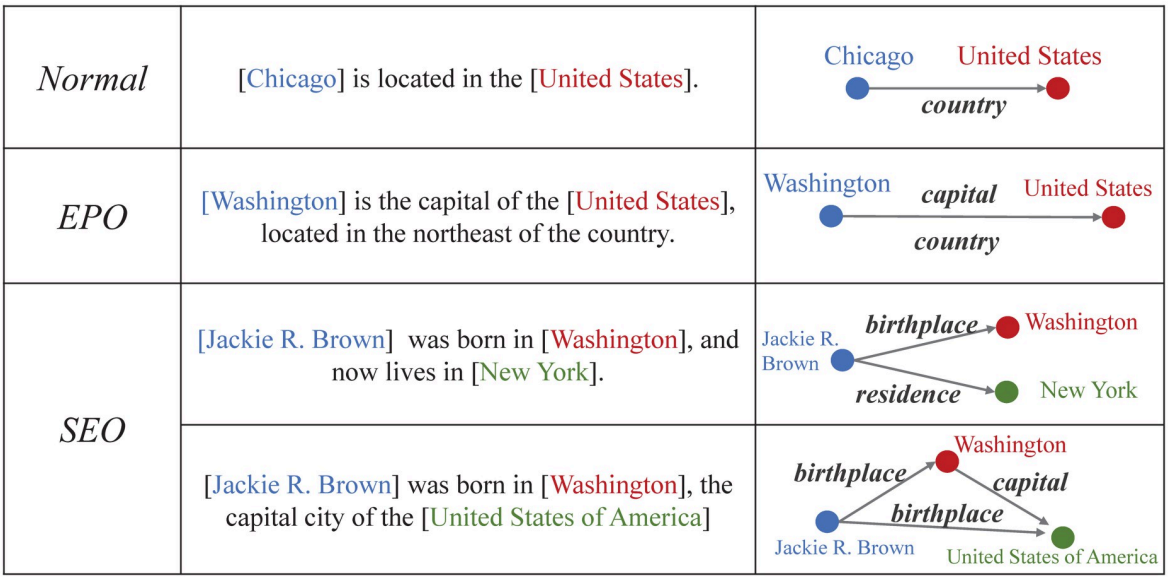

Fig 1. Examples of *Normal*, *EntityPairOverlap* (EPO) and *SingleEntityOverlap* (SEO) overlapping patterns.

https://doi.org/10.1371/journal.pone.0260426.g001

level multi-label classification. Thus, a novel sequential tagging scheme representing the overlapping triple is used for relation extraction.

To enable our scheme, we propose an end-to-end framework for training that can perform joint extraction of relational triples from sentences following EPO and SEO patterns.

The key contributions are summarized as follows:

- We propose a mutually exclusive binary cross tagging scheme to label the overlapping triples in sentences of EPO and SEO types;
- We built an end-to-end framework based on our proposed tagging scheme to perform the joint extraction of overlapping triples in a sentence;
- Experiments on the three public datasets show that the end-to-end BCT framework achieves encouraging performance and consistent improvements in F1 score, obtained by effectively handling the overlapping issue through the mutually exclusive binary cross tagging scheme.

Related work

Relation Extraction is a core task in Information Extraction and its goal is to extract relational triples from unstructured natural language text. It is critical for ontology learning works, particularly for building large-scale relational knowledge graphs [17]. In addition, it has been introduced to a variety of NLP applications such as text summarization and sentiment analysis.

Current IE research is mainly divided into fixed Information Extraction and Open Information Extraction (OIE). OIE was introduced as an open variant of the conventional IE task [18]. It focuses on extracting the text description of relations from plain texts without predefining a set of relationships. The first OIE system called Textrunner [19] utilizes a self-supervised learning framework to extract relational facts. Subsequent rule-based system ReVerb [20] employs handcrafted rules implemented by regular expressions to establish relations. However, both Textrunner and ReVerb only extract relational triples from the sentences connected to the verb. Later systems OLLIE [21] and ClausIE [22] were able to extract non-verbs as relation words using dependency-analysis algorithms. Even more recently, a study in never-ending learning called NELL [4] acquires knowledge through continuous reading and can reason its knowledge base.

Although the OIE task does not limit the relation classes, the extracted relationships tend to lack semantic information. Unlike the research in the OIE field, the fixed IE in this work utilizes pre-labeled datasets and predefined relationships for extracting semantic relationships. Despite the coverage rate being lower than OIE, the method is simple and easy to maintain in specific scenarios. Our work focuses on training an entity and relation extractor for predefined relations, intending to extract high-precision semantic relations for overlapping problems.

With the development of neural networks, researchers increasingly pay attention to fixed IE based on deep learning. More and more efforts are made to deal with relation extraction in complex scenarios. Early RE works [23, 24] mainly followed the pipelined method. The pipelined method extracts relational triples in two separate steps. First, the entities in a sequence are identified. Next, the relations between entities are distinguished by running relation classification (RC). Several pipeline methods based on neural network models have been proposed to improve the effectiveness. For example, Vu et al. [25] introduces Bidirectional RNN for the relation classification. Xu et al. [26] used LSTM for the RC module to obtain the information along the shortest dependency path. However, the accuracy of the RC module will be affected by preliminary errors using these pipeline methods. In addition, the pipeline method usually

neglects the interaction between the two steps and creates unnecessary redundancy as a result. By introducing a tagging framework, we convert the relation extraction task into an end-to-end prediction problem where we simultaneously extract entities and relations, thereby alleviating the propagation error of the pipeline method.

Compared with pipeline methods, joint models can limit the error propagation by integrating entities and relations. Several joint models [8–11] have been proposed to detect entities and relations simultaneously, but these models rely on substantial manual work. Subsequently, Zheng et al. [13] proposed a novel tagging scheme for extracting multiple relations which converts the joint extracting models into a sequence tagging problem. However, this model can only assign one label to each word and cannot deal with overlapping triples, as the number of tags is too large to learn. Zeng et al. [14] classified the sentences with overlapping triples into EPO and SEO, and they used a Seq2Seq learning framework with a copy mechanism to ease the overlapping issues. Nevertheless, their NER module relies heavily on high-precision word segmentation tools. More recently, Fu et al. [15] proposed a novel joint method based on graph convolutional networks (GCNs). Luo et al. [27] employed two binary tree structures to solve the overlapping triple problem, but they would need to design a different handling approach to handle all three types of sentence modes. Despite their initial successes, none of these methods can fully extract the overlapping triples, and the aforementioned models fail to achieve satisfactory results when the problem of overlapping triples becomes relatively more complex. Our extraction framework is constructed based on the sequence tagging approach. We perform mutually exclusive binary labeling for each entity, then jointly extract entities and relations by a cross-match constraint. Subsequently, we adopt joint learning and utilize the end-to-end BCT framework to process the relational triples.

In this work, we propose a sequential tagging scheme and show how we efficiently employed an end-to-end neural model to extract relations without NER and RC. Our end-to-end BCT framework solves overlapping patterns with a sequential tagging scheme and reduces the number of tags to learn. Compared with the previous tagging schemes, our model achieves better results for sentences that include any of the types of overlapping triples.

Methodology

Our proposed end-to-end BCT framework enables the extraction of multiple relational triples at once. In this section, we will introduce each part of the end-to-end BCT framework in detail. As Fig 2 shows, the proposed framework is composed of a BERT-based encoder, a decoder module (BCT tagger) and a loss function module. Our BCT scheme is mainly employed in the decoder part of the extraction framework, and we will illustrate in detail how to simplify the extraction problem to a concise tagging issue based on the BCT scheme later in this section.

End-to-end extraction framework

Our proposed end-to-end BCT framework is illustrated in Fig 2. It contains the encoder module, decoder module, and loss function. The input of our model is a sentence $w = [e_1, e_2, \ldots, e_n]$, where e_t represents the t-th word in the sentence of length n. The output semantic representation through the BERT-based encoder module is Z_n. Next, we apply the BCT scheme to decode Z_n and lastly the entity and triples are jointly extracted.

BERT-based encoder. The encoder module aims to extract feature information for each word from the sentence w. In this paper, a pre-trained BERT model [28] is adopted as the encoder for input sentences. We will briefly introduce the BERT, a multi-layer bidirectional Transformer-based language representation model.

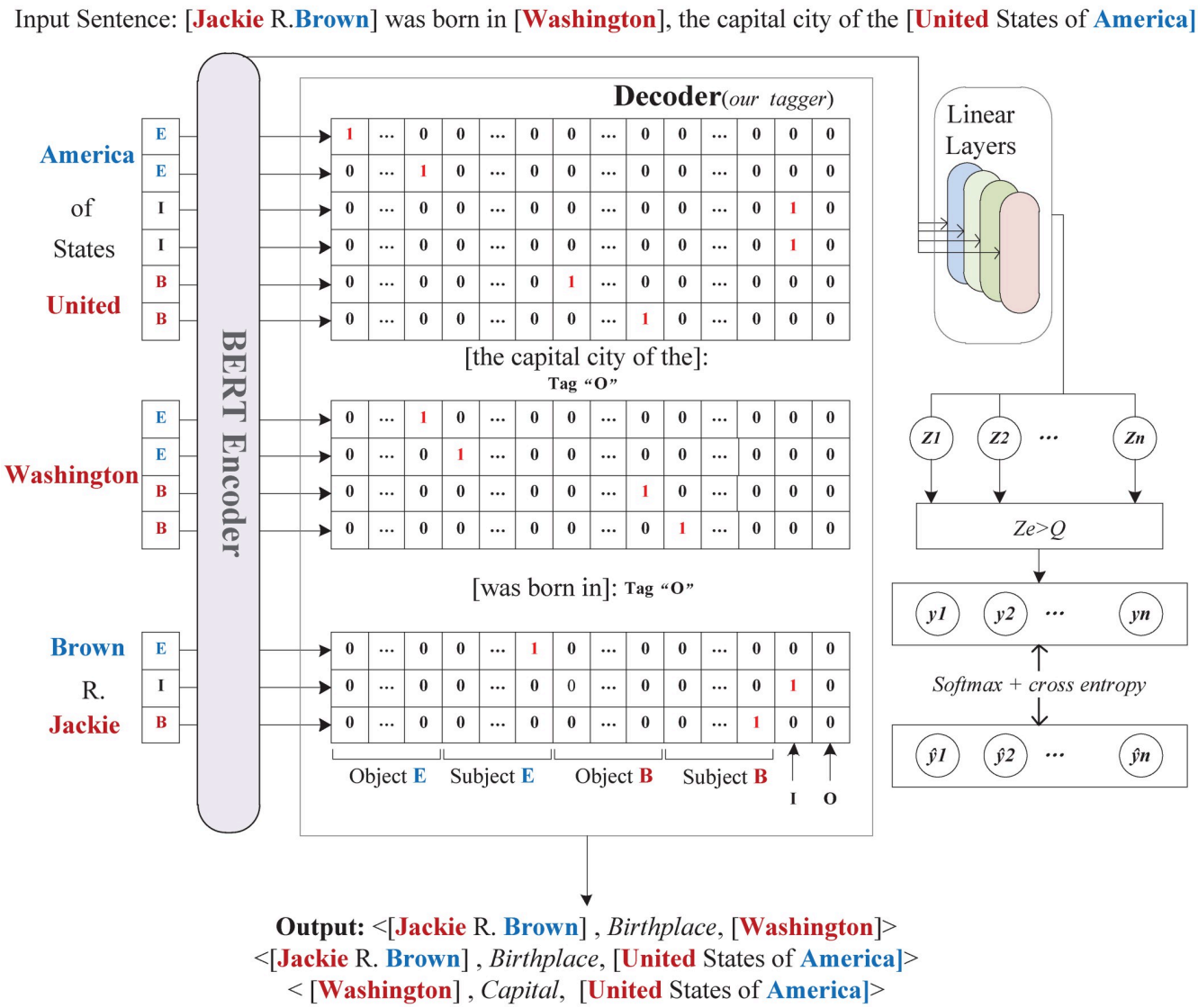

Fig 2. An illustration of our proposed end-to-end BCT extraction framework. In this example, red and blue respectively represent the "B" tags and "E" tags for the entities. We suppose there are $|N|$ predicates, and a total number of $4|N| + 2$ BCT tags, which are the $4|N|$ mutually exclusive "B" and "E" tags, plus two "I" and "O" tags.

https://doi.org/10.1371/journal.pone.0260426.g002

The pre-trained BERT model contains the Embedding module and N identical Transformers Block modules. The Transformer module uses multi-head attention to represent a word with a vector containing context information. We define the transformer module as $Trans(x)$ where x represents the input vector. The transformer module extracts each word feature containing context information based on the multi-head attention. We can write the formula as follows:

$$h = S * W_s + W_p \tag{1}$$

$$h_a = \mathrm{T}rans(\mathrm{x}), a \in [1, N] \tag{2}$$

We first convert the input sentence into the matrix of one-hot vectors of sub-words to get the matrix $S = [x_1, x_2, \ldots, x_n]$. S is a matrix comprised of one-hot vectors of token indices in the input sentence. The sub-words embedding matrix is represented as W_s and the position embedding matrix is denoted as W_p where p represents the position index in the input sequences. The hidden state vector is represented as h, and h_a is the hidden state of the input sentence at the a-th layer. Since our input is a single sentence instead of a text pair, we did not consider segmentation embedding.

Decoder(BCT tagger). Our BCT scheme is mainly employed in the decoder module. We first illustrate how to simplify the extraction problem to a concise tagging issue based on the BCT scheme and then introduce our decoding process.

Mutually exclusive binary cross tagging scheme. Our mutually exclusive binary cross tagging scheme is designed to extract multiple overlapped EPOs and SPOs in the RE task. We use an example to detail our tagging strategy as shown in Fig 3, and the pseudo-code is presented in **Algorithm 1**.

Based on the conventional BIO tagging scheme [29], each token in the input sentence is assigned a label that contributes to indicate its position in an entity span. The conventional BIO tagging scheme [29] works based on the positions within a labeled entity. BIO refers to the beginning, inside, and outside of an entity. Likewise, we employ BIEO signs to label entities in our tagging scheme. The tags consist of three parts: the word position in the entity, the relation type, and the relation role.

- The word position in the entity: "BIEO", where the signs B (begin), I (inside), E (end), and O (outside) represent the position information of a word in the entity.

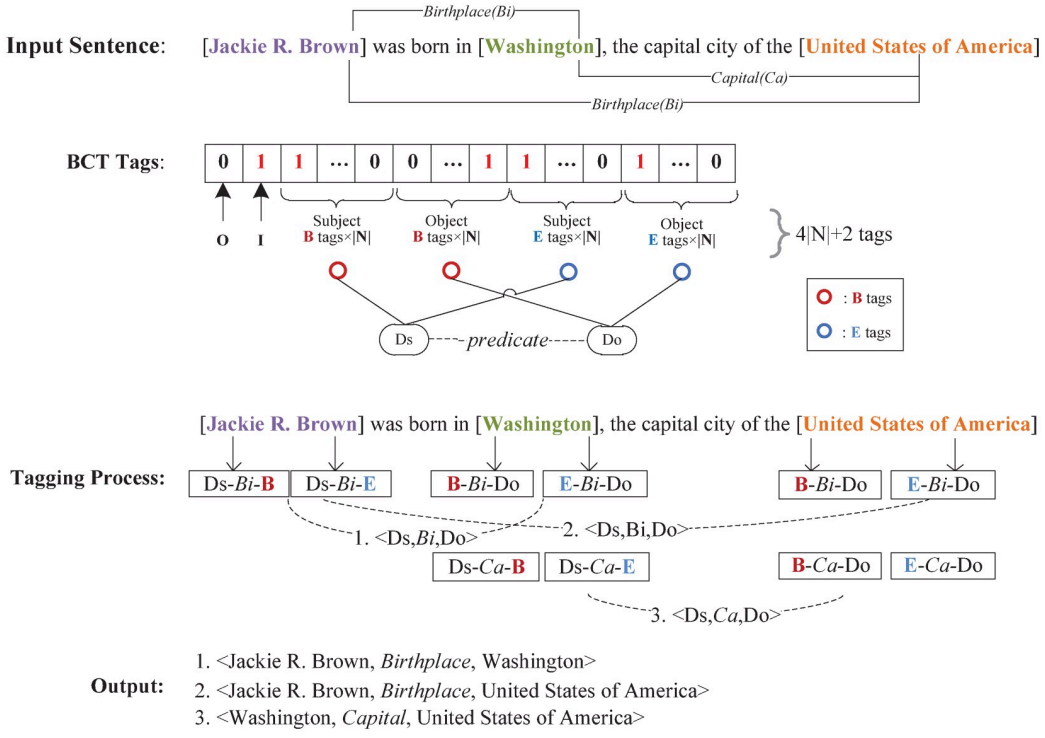

Fig 3. A visual illustration of our BCT scheme. In this example, red and blue circles represent respectively the "B" tags and "E" tags for the entities. We take an SEO sentence with 3 triples and 2 relation types as an example.

https://doi.org/10.1371/journal.pone.0260426.g003

- The relation type: The relation type information is obtained from a predefined set of predicates $|N|$.
- The relation role: The relation role information is represented by the subject *Ds* and the object *Do*. An extracted result is represented as a triple: <*Ds*, *predicate*, *Do*>. *Ds* means the subject entity in the triple, while *Do* belongs to the object entity.

Fig 3 illustrates an example of our tagging scheme. The input sentence "[Jackie R. Brown] was born in [Washington], the capital city of the [United States of America]" is a typical SEO overlapping pattern. It contains three overlapped triples: <Jackie R. Brown, Birthplace, Washington>, <Jackie R. Brown, Birthplace, United States of America> and <Washington, Capital, the United States of America>. Where "Birthplace" and "Capital" are the predefined relation types. The words "Jackie," "R.," "Brown," "Washington," "United," "States," "of," and "America" are all related to the final extracted entities. Thus, we tag these words based on our BCT strategy.

Next, we generate BCT tags for each word in the input sentence. As Fig 2 shows, each token in a sentence is identified by the mutually exclusive binary "B" and "E" tags. These tags are further distinguished by different relations and subject/object dichotomy. A "B" or "E" tag should have the form of (B (E)-predicate-*Do*) or (*Ds*-predicate-B (E)). If we suppose there are $|N|$ predicates, the results are $4|N|$ mutually exclusive "B" and "E" tags as shown in Fig 3.

In the example of our tagging process shown in Fig 3, the word "United" is the first word of the object entity "United States of America" and is related to the predicate "Capital". The resulting triple based on the BCT tag then becomes (B-predicate-*Ds*). Similarly, the BCT tag for the last word of the object entity "America" is (E-predicate-*Ds*). Any words between "United" and "America" are now tagged as "I (Inside)" while the subject entity "Washington" (which is corresponding to "United States of America") is tagged as (B-predicate-*Do*). Additionally, words unrelated to the final result are labeled as "O".

The binary "B" and "E" tags of each entity are cross-matched with an interval of $2|N|$ in all tag sequences to form the subject or object. Matching tags that differ by $|N|$ total tags will share the same predicate type. After tagging, we deal with the overlapping by using token-level multi-label classification, with a total of $4|N|$ labels plus the "I" and "O" tags. The decoder parses the word representations and predicts the BCT tags for each word. As shown in Fig 2, the h_a obtained from the BERT encoding layer is the input for the decoding layer. It is fed into linear layers to compute the probabilities of BCT tags based on the tag predicted vector.

The specific formula is as follows:

$$Z_k = W_k * x_i + b_k, \quad k \in [B, I, E, O] \tag{3}$$

$$\begin{aligned} p_i^{\mathrm{k(B)}} &= \mathbf{W}_{\mathrm{B}}\mathbf{x}_i + \mathbf{b}_{\mathrm{B}} \\ p_i^{\mathrm{k(E)}} &= \mathbf{W}_{\mathrm{E}}\mathbf{x}_i + \mathbf{b}_{\mathrm{E}} \end{aligned} \tag{4}$$

Where the x_i is the i-th vector of h_a, W represents the trainable weight and b is the bias. The encoding vector Z_k is the result of the output of the BERT and linear layers. The probabilities of recognizing the i-th token with "B" and "E" tag of an entity are indicated by respectively $p_i^{\mathrm{k(B)}}$ and $p_i^{\mathrm{k(E)}}$.

The essential decoding process of our BCT scheme is shown in **Algorithm 1**. The encoding vector Z_k generated by the BERT and linear layers is utilized as input of the decoder. Before the actual decoding process of joint extraction can start, we first have to perform some preparation and processing work to store the output of the encoder in a list *P*. No activation

functions were used to implement our subsequent loss function. Instead, we require the probabilities of the target class to be higher than the other classes. It can be described as follows:

$$d_i = \begin{cases} Q & Z_k \in [-\infty, 0] \\ 0 & Z_k \in (0, +\infty] \end{cases} \tag{5}$$

$$p(d_i, Z_{k'}) = p(Z_{k'} \mid d_i) p(d_i) \tag{6}$$

Algorithm 1 Mutually-exclusive Binary Cross tagging scheme

Input: The predict results $Z_{k'}$ for entities filtered by Q output by the BERT and linear layers.
Output: *Entity* list E and *SPO* list: T = <$E1$, Rs, $E2$>
1: Initialize list:P = [], entities list: E = [],*SPO* list: T = []
2: Sort elements of array $Z_{k'}$ in P by position
3:**for** i in *length*(P) **do**
4: Determine the initial position i of the entity;
5: **if** $id \in P$ **then**
6: id is defined as the triple matching identifier; Determine the end position j of the entity, j = 0;
7: **while** $i + j + 1 <$ *length*(P) **do**
8: **if** $id + 2|N|$ in $P[i + j + 1]$ **then**
9: Add 1 to j;
10: Form the entity with the start and the end index elements with an interval of $2|N|$ in P;
11: **end if**
12: **if** the $P[i + j + 1]$ is the value of Inside sign (I = 1). **then**
13: Add 1 to j
14: **else if** $i + j + 2 <$ *length*(P) **and** $P[i + j + 2]$ is the value of Inside sign (I = 1). **then**
15: Add 1 to j
16: **else**
17: Form the entity with the start and the end index elements with an interval of $2|N|$ in P;
18: **end if**
19: **end while**
20: **end if**
21: the "B" and "E" tags of two entities with an interval of $|N|$ are cross-matched to form the triple;Add all entities to entities list E; Add the <$E1$, Rs, $E2$> to the *SPO* list: T = [];
22: **end for**

Here Q is a certain threshold. Eq (5) now helps us to filter for a certain target and form binary labels when $Q = 1$. A new set $Z_{k'}$ is now generated under the constraint d_i, and this new set $Z_{k'}$ for entities filtered by Q is utilized as input for **Algorithm 1**. In this case, we sort elements of array $Z_{k'}$ by position, and construct the list of the predicted results for entities $p(d_i, Z_{k'})$ (List P in **Algorithm 1**).

In the decoding process based on the BCT scheme, we declare three assistant extraction identifiers: *id*, *i*, and *j*. Here *i* and *j* are the location identifiers of an entity, and the id is the triple matching identifier. First, the start and end index elements that differ by $2|N|$ in the list are matched to form an entity (subject or object entity), then the binary tags of two entities with an interval of $|N|$ are cross-matched to form a triple of the same predicate.

BCT tags to triples. In accordance to the tagged sequence in Fig 3, our BCT scheme can extract multiple triples at a time. As described above, we begin by extracting mutually exclusive dichotomous "B" and "E" tags for all subject and object entities. Then we start from the

position of these "B" tagged entities and recursively match "E" tagged entities to form the subjects and objects. Finally, the "B" and "E" tags of two entities with an interval of $|N|$ are cross-matched to form a triple of the same predicate, and we build the relations between the entities' pair.

Whenever an input sentence has at least two triples with the same predicate, the triples are formed based on the nearest principle. Consider for example the following sentence: "The new station will have its headquarters in [Doha], [Qatar], and operate broadcast newsrooms in London, Washington and [Kuala Lumpur], [Malaysia]". We know that "Doha", "Qatar", "Kuala Lumpur" and "Malaysia" share the same predicate "administrative-divisions". Therefore, if a BCT tagged subject ("Doha" or "Kuala Lumpur") can match more than one entity as its object while having more than one triple that shares the same predicate, we match the two entities that are closest in position in the sequence. In summary, we construct triples based on cross-matching with an interval of $|N|$ and the nearest principle.

In this study, we do not add activation functions like the Sigmoid or Softmax function, but we employ the multi-label categorical cross-entropy to ease the class imbalance problems instead. The details are described below.

Loss function. The loss function for training our BCT tags is a multi-label categorical cross-entropy function. For the multi-label classification task, we need to select m target categories from n candidate categories. The common practice is to use the sigmoid activation function and then convert it to an $|N|$ binary classification problem, with determination of the final loss through the Sigmoid cross-entropy. However, this approach will suffer from a severe category imbalance problem when $n \gg m$.

As we know, the single-label classification task is more straightforward than multi-label classification; it can apply the efficient Softmax cross-entropy to avoid class imbalance. Multi-label classification on the other hand tends to create category imbalance problems. Therefore, we would like to extend the Softmax cross-entropy function to our task.

We employ a pair similarity optimization viewpoint [30], aiming to maximize the within-class similarity and minimize the between-class similarity. Given a single sample x in the feature space, we assume that K within-class similarity scores and L between-class similarity scores are associated with x. Since each target-class score s_j needs to be higher than non-target class score s_i, we employ *logSumExp* as follows:

$$\log \mathrm{SumExp}(x_1 \dots x_n) = \log\left(1 + \sum_{i \in K, j \in L} e^{s_i - s_j}\right) \tag{7}$$

The K, L sets are also the category sets of respectively negative and positive samples. In this study, m target categories are selected from n candidate categories, and we define a threshold $Q = 0$ as presented in **Algorithm 1** to control the output when m is not fixed. The detailed operations are as follows:

$$\begin{aligned} L_m &= \log\left(1 + \sum_{i \in K, j \in L} e^{s_i - s_j} + \sum_{i \in K} e^{s_i - Q} + \sum_{j \in L} e^{Q - s_j}\right) \\ &= \log\left(e^{Q} + \sum_{i \in K} e^{s_i}\right) + \log\left(e^{-Q} + \sum_{j \in L} e^{-s_j}\right) \end{aligned} \tag{8}$$

Where L_m is the multi-label categorical cross-entropy function including both the positive and negative samples. We derive L_{pos} for the positive labels and L_{neg} for the negative labels by Eq

(8). The total loss L_m of our framework is then determined as follows:

$$\begin{aligned} L_m &= \log\left(1+\sum_{i\in K} e^{s_i}\right) + \log\left(1+\sum_{j\in L} e^{-s_j}\right) \\ &= L_{neg} + L_{pos} \end{aligned} \tag{9}$$

Experiments and results

Experimental setting

Datasets and evaluation metrics. The framework in this study has been evaluated on three widely used public datasets: NYT [31], WebNLG [32], and DuIE [33]. The NYT dataset was initially generated by the distant supervision method. It contains 1.18 million sentences with 24 predefined relations from New York Times articles. After removing sentences without valid triples, 61,195 sentences remain. We used the NYT datasets released by Zeng et al. [14], which includes a training set of 56,195 sentences, a validation set of 5000 sentences and a testing set of 5000 sentences. The WebNLG dataset was created for Natural Language Generation (NLG) tasks. We use the WebNLG datasets processed by Zeng et al. [14], which contains 246 predefined relation types. DuIE is a large-scale dataset built by Baidu Inc for Chinese information extraction, consisting of 210,000 sentences and 450,000 instances covering 49 predefined relation categories. We used the training set and development set from the DuIE dataset, which contains 173,108 sentences for training and 21,639 sentences for testing.

Table 1 describes the statistics of the training and testing sets from the three datasets. For each of the datasets, we divided the sentences into three different overlapping patterns of relational triples: Normal, *EntityPairOverlap* (EPO), and *SingleEntityOverlap* (SEO). Although several sentences might belong to both the EPO and SEO classes, the overlapping problem is common in these datasets.

We follow the evaluation metrics from Fu et al. [15], that is to say that a predicated relational triple $T = <E1, Rs, E2>$ is considered as a correct one only if the entities ($E1$, $E2$) and relation Rs are all correct. Specifically, we adopt the standard Precision (*Prec.*), Recall (*Rec.*) and F1 score ($F1$) to evaluate our framework.

Baseline methods. We selected several classical triple extraction models as our baselines. A list of the models can be found in Table 2. NovelTagging [13] uses the neural network to jointly extract relational triples by a novel sequential tagging scheme. CopyRE [14] is the first framework to consider the relational triple overlap problem. It proposed an end-to-end model based on sequence-to-sequence learning with a copy mechanism for relation extraction. Copy-RE_{RL} [16] is a reinforcement learning method based on a sequence-to-sequence model to handle the multiple relation extraction tasks. $Graph_{Rel}$ [15] is an end-to-end joint extraction model based on GCNs. BiTT [27] is an end-to-end extraction framework that labels the

Table 1. Statistics of NYT, WebNLG and DuIE datasets in our experiment.

Category	NYT		WebNLG		DuIE	
	Train	Test	Train	Test	Train	Test
Normal	37013	3266	1596	246	75600	9404
EPO	9782	978	227	26	7518	922
SEO	14735	1297	3406	457	95042	11957
ALL	56195	5000	5019	703	173108	21639

https://doi.org/10.1371/journal.pone.0260426.t001

Table 2. Results of different methods on NYT, WebNLG and DuIE datasets.

Method	NYT			WebNLG			DuIE		
	Prec.	*Rec.*	*F1*	*Prec.*	*Rec.*	*F1*	*Prec.*	*Rec.*	*F1*
NovelTagging [13]	62.4	31.7	42.0	52.5	19.3	28.3	-	-	-
NovelTagging $_{BERT}$ [13]	89.0	55.6	69.3	-	-	-	75.0	38.0	50.4
CopyRE $_{OneDecoder}$ [14]	59.4	53.1	56.0	32.2	28.9	30.5	-	-	-
CopyRE $_{MultiDecoder}$ [14]	61.0	56.6	58.7	37.7	36.4	37.1	-	-	-
CopyRE $_{RL}$ [16]	77.9	67.2	72.1	63.3	59.9	61.6	-	-	-
GraphRel $_{1p}$ [15]	62.9	57.3	60.0	42.3	39.2	40.7	52.2	23.9	32.8
GraphRel $_{2p}$ [15]	63.9	60.0	61.9	44.7	41.1	42.9	41.1	25.8	31.8
CasRel [34]	89.7	89.5	89.6	93.4	90.1	91.8	-	-	-
BiTT$_{BERT}$ [27]	89.7	88.0	88.9	-	-	-	75.7	80.6	78.0
BCT$_{BERT}$	**89.8**	**88.5**	**89.1**	**90.6**	**92.0**	**91.3**	**78.4**	**82.6**	**80.4**

https://doi.org/10.1371/journal.pone.0260426.t002

overlapping triples in a sentence based on two binary tree structures. CasRel [34] is a novel cascade binary tagging framework that is considered the state-of-the-art when it comes to results for the NYT and WebNLG datasets. CasRel regards the relations as mapping functions from subjects to objects. Note that in Table 2, CopyRE $_{OneDecoder}$ and CopyRE $_{MultiDecoder}$ are the CopyRE frameworks with one decoder and multiple decoders respectively. GraphRel$_{1p}$ is the 1st phase for GraphRel architecture and GraphRel$_{2p}$ is the complete version. We use these baseline models for further comparison, which have been upgraded through the pre-trained BERT [27].

We combined our proposed BCT scheme and the pre-trained BERT model to optimize performance. BCT$_{BERT}$ is the full-fledged framework using pre-trained BERT weights.

Implementation details. In this work, we employ two types of pre-trained BERT models for fine-tuning: The BERT-Base uncased model and the BERT-Base Chinese model [28].

The BERT-Base uncased model (12-layer Transformer, 768-hidden, 12-heads, 110M parameters) is trained on a large text corpus (Wikipedia and BookCorpus) [28], and it is applied for fine-tuning two English datasets NYT and WebNLG. Furthermore, it employs Word Piece tokenization to split words.

However, this approach is not appropriate for Chinese datasets, as Chinese is a continuous language, where contrary to English, whitespace characters do not exist. Secondly, Chinese characters are the smallest unit and cannot be further split. For these reasons, we employ the BERT-Base Chinese model (12-layer Transformer, 768-hidden, 12-heads, 110M parameters) [28] to the DuIE Chinese dataset. This model utilizes character-based tokenization and is trained on the relevant corpus of Chinese Wikipedia.

In our experiments, we adopted a mini-batch mechanism with a batch size of 4 to train our model. The learning rate was set to $1e^{-5}$. Additionally, we implemented the early-stopping mechanism to prevent our model from over-fitting. Specifically, the training process is terminated when the F1 scores on the validation set do not increase for at least ten sequential epochs. These hyperparameters were determined by the validation set.

Experimental results

Compared results. In this section, we compare our proposed method with the previously mentioned state-of-the-art models. We conducted experiments on all types of sentences (Normal, EPO and SEO) and compared the performance with results from previous works. Table 2 shows the results (*Prec.*, *Rec.*, and *F*1) of different baseline models, together with our framework for three datasets.

Our BCT_{BERT} framework achieves encouraging F1 scores in NYT, WebNLG, and DuIE datasets of 89.1%, 91.3%, and 80.4% respectively. Compared with the best current baseline method $BiTT_{BERT}$ [27] for the NYT dataset, our BCT based on pre-trained BERT achieves strong performance, specifically in terms of F1. Notably, the results of the WebNLG dataset even outperform the NYT results since the proportion of overlapping triples is higher in the former. Our framework is comparable to the current state-of-the-art method CasRel [34] on the WebNLG dataset in F1 score. For the DuIE dataset, our framework outperforms the top baseline method $BiTT_{BERT}$ [27] by 2.7% in precision, 2.0% in recall, and 2.4% in F1 score. From these results, we note that the BCT scheme helps to predict relations in terms of precision, recall, and F1 score in the general case.

NovelTagging [13], CopyRE $_{MultiDecoder}$ [14], and our framework all use a sequential framework. NovelTagging considers all entities belonging to a single relation type, resulting in high precision and low recall. CopyRE $_{MultiDecoder}$ applies multiple separated decoders to form relation triples, but the accuracy of extraction is low as a result of the restrictions on the copy mechanism. Our method, on the other hand, distinguishes the entities by different relations and subject/object dichotomy to generate more relation triples. This enables us to achieve high precision and high recall, yielding higher $F1$ scores.

A significant difference can be noted between the performance scores of NYT, WebNLG, and DuIE datasets in Table 2. The explanation for this lies in the differences in composition of specific datasets. The proportion of overlapping pattern sentences is relatively high in the WebNLG and DuIE datasets, while the NYT dataset mainly consists of Normal pattern sentences. Furthermore, the DuIE is a large-scale dataset with a larger proportion of EPO and SEO sentences. For most baseline methods these differences between the datasets generally cause the results for the NYT and WebNLG datasets to be better than the results for the DuIE dataset. However, our BCT model achieves competitive results on all three datasets, verifying the utility of the BCT scheme in solving overlapping triples.

Detailed analysis on different types of sentences. To further evaluate the capacity of the proposed end-to-end BCT framework in extracting overlapping relational triples, we experimented with different overlapping patterns and compared the results with prior works.

Fig 4 shows the detailed results for both NYT and WebNLG datasets for three different overlapping patterns. As can be seen, our framework achieves much higher performance on all three overlapping patterns. Since the overlapping patterns make it harder to extract relational triples from sentences, the extracting performance of EPO and SEO patterns is significantly lower than the Normal pattern extraction performance in baseline methods. Contrarily, our framework shows consistently strong overall performance with all three overlapping patterns. This is because the BCT scheme was specifically designed to be more suitable for overlapping relational triples.

Additionally, we validated the ability to extract relational triples from sentences with different numbers of triples. Table 3 presents our method in detail. The sentences from the NYT and WebNLG test set have been split into five subclasses based on the number of triples. Our framework achieves superior overall performance in all five subclasses.

The results compared to the sequence-to-sequence learning methods are shown in Fig 5. It can be seen that the F1 scores of the two sequential methods display a downward trend with the increasing number of relational triples. Our framework gives a consistently stable performance on both of the investigated datasets. Especially when it becomes difficult to extract triples (the number of triples in a sentence ≥ 5), our framework gains the most significant improvement of F1 score over the other methods. Since the BCT scheme considers the various word features of entities based on mutually exclusive cross dichotomy, our method is superior in dealing with overlapping triples, resulting in a higher F1 score.

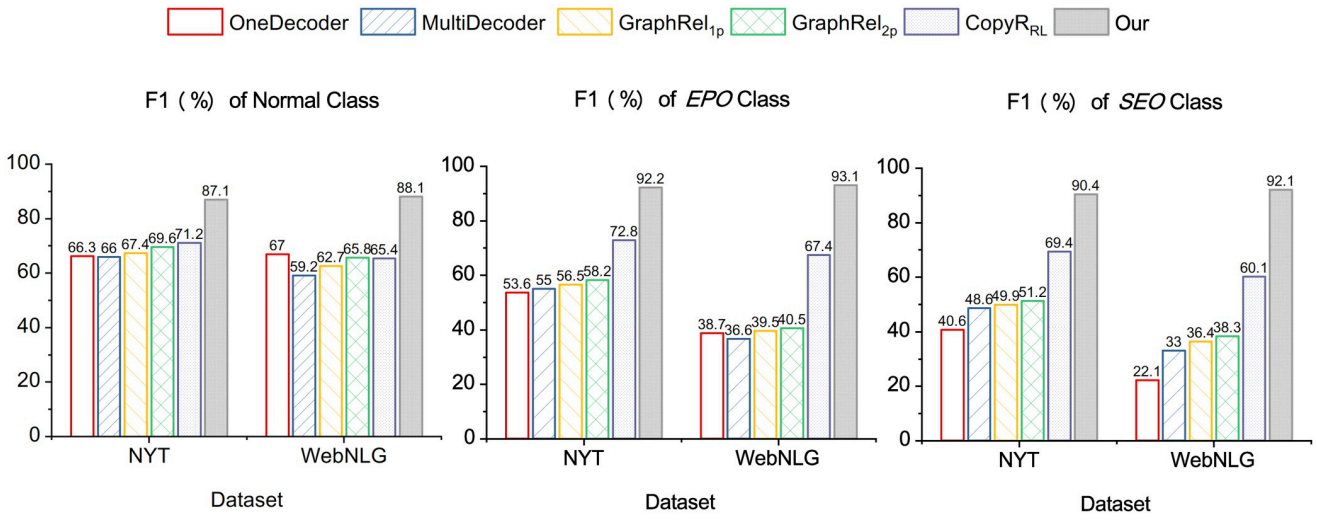

Fig 4. Results(F1-scores) of extracting relational triples from sentences with *Normal*,*EPO*, and *SEO* overlapping patterns.

https://doi.org/10.1371/journal.pone.0260426.g004

The experimental results on three public datasets show the effectiveness of our proposal, especially while dealing with overlapping entities and triples. Even so, still some shortcomings in the extraction of overlapping relations remain. Our BCT model focuses on extracting as many relational triples as possible from an input sentence, while the tagging process of entities considers all relation types simultaneously. Due to this characteristic, our framework is prone to mistakenly detect redundant triples when the number of triples in a sentence increases. Therefore, the relation identification between entity pairs needs to be further refined in subsequent work.

Case study. We conducted the case study by comparing our framework with the current best state-of-art method. Table 4 shows the results of different overlapping sentences containing Normal, EPO, and SEO patterns. It can be observed that both our framework and the CasRel [34] method present a solid ability to extract overlapping relational triples on the Normal and EPO sentences. However, CasRel could not accurately extract all the gold triples when dealing with a more complex SEO sentence. It failed to catch the triple < Brooklyn, contains, Island >, possibly due to this method's specific general bias caused by relations being modeled as functions. Our method accurately captured all the overlapping relational triples, showing

Table 3. F1-score of extracting relational triples from sentences that contain different numbers of triples.

Method	NYT					WebNLG				
	$N = 1$	$N = 2$	$N = 3$	$N = 4$	$N \geq 5$	$N = 1$	$N = 2$	$N = 3$	$N = 4$	$N \geq 5$
CopyRE $_{OneDecoder}$ [14]	66.6	52.6	49.7	48.7	20.3	65.2	33.0	22.2	14.2	13.2
CopyRE $_{MultiDecoder}$ [14]	67.1	58.6	52.0	53.6	30.0	59.2	42.5	31.7	24.2	30.0
GraphRel $_{1p}$ [15]	69.1	59.5	54.4	53.9	37.5	63.8	46.3	34.7	30.8	29.4
GraphRel $_{2p}$ [15]	71.0	61.5	57.4	55.1	41.1	66.0	48.3	37.0	32.1	32.1
CopyRE $_{RL}$ [16]	71.7	72.6	72.5	77.9	45.9	63.4	62.2	64.4	57.2	55.7
CasRel [34]	88.2	90.3	91.9	94.2	83.7	89.3	90.8	94.2	92.4	90.9
BCT$_{BERT}$	**87.3**	**90.5**	**88.7**	**94.4**	**87.3**	**87.8**	**90.4**	**94.6**	**93.0**	**91.2**

We divide the sentences of the NYT and WebNLG test set into 5 subclasses respectively. Each class contains sentences that have 1,2,3,4 or $\geq$ 5 triples.

https://doi.org/10.1371/journal.pone.0260426.t003

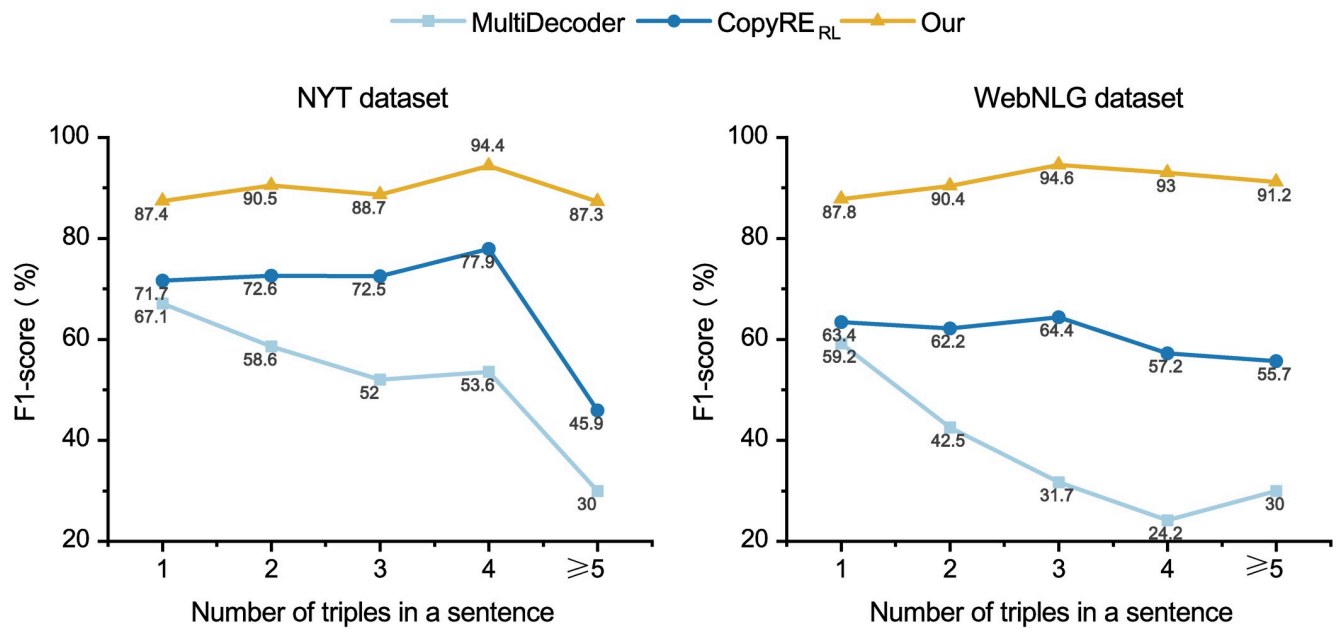

Fig 5. Performance(F1-scores) of extracting relational triples from sentences that contain different numbers of triples.

https://doi.org/10.1371/journal.pone.0260426.g005

Table 4. Case study of our BCT model and best baseline method.

Sentence	Gold triples	CasRel [34]	Our
Kalle Palander of Finland won the race, and Akira *Sasaki* of *Japan* was the runner-up.	<*Sasaki*,nationality, *Japan*>	<*Sasaki*,nationality, *Japan*>	<*Sasaki*,nationality, *Japan*>
The issue reflects the relentless battle over *Jerusalem*, which both *Israel* and the Palestinians claim as their capital.	<*Israel*,capital,*Jerusalem*> <*Israel*,contains, *Jerusalem*>	<*Israel*,capital,*Jerusalem*> <*Israel*,contains, *Jerusalem*>	<*Israel*,capital, *Jerusalem*> <*Israel*,contains, *Jerusalem*>
He trained for about six months, he said, running from his house on East 23rd Street in *Midwood*, *Brooklyn*, to the Coney *Island* boardwalk and back, he said.	<*Brooklyn*,contains, *Midwood*> <*Island*,neighborhood of, *Brooklyn*> <*Brooklyn*,contains, *Island*> <*Midwood*,neighborhood of,*Brooklyn*>	<*Brooklyn*,contains, *Midwood*> <*Island*,neighborhood of, *Brooklyn*> <*Midwood*,neighborhood of,*Brooklyn*>	<*Brooklyn*,contains, *Midwood*> <*Brooklyn*,contains, *Island*> <*Midwood*,neighborhood of,*Brooklyn*) <*Island*,neighborhood of, *Brooklyn*>

Gold triples: All triples we expect to be extracted from the example sentences.

https://doi.org/10.1371/journal.pone.0260426.t004

the effectiveness of the BCT scheme for dealing with overlapping cases. Although our framework achieved encouraging results in extracting overlapping triples, the process still requires refinement. We will continue to conduct research, with the goal of further improving our ability to reliably extract triples.

Conclusions

In this work, we introduced an end-to-end BCT framework to jointly extract the overlapping entities and relations. Unlike previous sequential frameworks, we utilize an efficient binary cross-matching method for constructing entities that participate in multiple triples. The experimental results on three datasets show the effectiveness of our proposal, especially while

handling overlapping issues. However, some shortcomings in the extraction of overlapping relations still remain. Our BCT model focuses on extracting as many relational triples as possible from an input sentence, while the tagging process of entities considers all relation types simultaneously. Due to this characteristic, our framework is inclined to detect redundant triples by mistake when the number of triples in a sentence increases. Therefore, the identification of relations between entity pairs needs refinement in further research. Our later work will be aimed at both enhancing our capacity to extract overlapping relational triples and applying our improved method to other tasks such as Medical Named Entity Recognition and Chinese event extraction.

Author Contributions

Conceptualization: Xuan Liu, Wanru Du.

Data curation: Xuan Liu.

Formal analysis: Xuan Liu, Xiaoyin Wang.

Funding acquisition: Xuan Liu, Wanru Du.

Methodology: Xuan Liu, Wanru Du.

Software: Xuan Liu, Ruiqun Li.

Validation: Xuan Liu.

Writing – original draft: Xuan Liu, Wanru Du, Xiaoyin Wang.

Writing – review & editing: Xuan Liu, Wanru Du, Xiaoyin Wang, Ruiqun Li, Pengcheng Sun, Xiaochuan Jing.

References

1. Suchanek F. M., Kasneci G., and Weikum G., "Yago: a core of semantic knowledge," in *Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 697–706.
2. Bollacker K., Evans C., Paritosh P., Sturge T., and Taylor J., "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 2008, pp. 1247–1250.
3. Lehmann J., Isele R., Jakob M., Jentzsch A., Kontokostas D., Mendes P. N., et al., "Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia," *Semantic web*, vol. 6, no. 2, pp. 167–195, 2015. https://doi.org/10.3233/SW-140134
4. Mitchell T., Cohen W., Hruschka E., Talukdar P., Yang B., Betteridge J., et al., "Never-ending learning," *Communications of the ACM*, vol. 61, no. 5, pp. 103–115, 2018. https://doi.org/10.1145/3191513
5. Zelenko D., Aone C., and Richardella A., "Kernel methods for relation extraction," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1083–1106, 2003.
6. Chan Y. S. and Roth D., "Exploiting syntactico-semantic structures for relation extraction," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011, pp. 551–560.
7. Liu C., Sun W., Chao W., and Che W., "Convolution neural network for relation extraction," in *International Conference on Advanced Data Mining and Applications*. Springer, 2013, pp. 231–242.
8. Yu X. and Lam W., "Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach," in *Coling 2010: Posters*, 2010, pp. 1399–1407.
9. Li Q. and Ji H., "Incremental joint extraction of entity mentions and relations," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2014, pp. 402–412.
10. M. Miwa and Sasaki Y., "Modeling joint entity and relation extraction with table representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1858–1869.

11. Ren X., Wu Z., He W., Qu M., Voss C. R., Ji H., et al., "Cotype: Joint extraction of typed entities and relations with knowledge bases," in *Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 1015–1024.

12. Gupta P., H. Schütze, and Andrassy B., "Table filling multi-task recurrent neural network for joint entity and relation extraction," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016, pp. 2537–2547.

13. Zheng S., Wang F., Bao H., Hao Y., Zhou P., and Xu B., "Joint extraction of entities and relations based on a novel tagging scheme," *arXiv preprint arXiv:1706.05075*, 2017.

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

15. Fu T.-J., Li P.-H., and Ma W.-Y., "Graphrel: Modeling text as relational graphs for joint entity and relation extraction," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 1409–1418.

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

17. Pan J. Z., Vetere G., Gomez-Perez J. M., and Wu H., *Exploiting linked data and knowledge graphs in large organisations*. Springer, 2017.

18. Etzioni O., Banko M., Soderland S., and Weld D. S., "Open information extraction from the web," *Communications of the ACM*, vol. 51, no. 12, pp. 68–74, 2008. https://doi.org/10.1145/1409360.1409378

19. Yates A., Banko M., Broadhead M., Cafarella M. J., Etzioni O., and Soderland S., "Textrunner: open information extraction on the web," in *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, 2007, pp. 25–26.

20. Fader A., Soderland S., and Etzioni O., "Identifying relations for open information extraction," in *Proceedings of the 2011 conference on empirical methods in natural language processing*, 2011, pp. 1535–1545.

21. Schmitz M., Soderland S., Bart R., O. Etzioni et al., "Open language learning for information extraction," in *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, 2012, pp. 523–534.

22. Del Corro L. and Gemulla R., "Clausie: clause-based open information extraction," in *Proceedings of the 22nd international conference on World Wide Web*, 2013, pp. 355–366.

23. Mintz M., Bills S., Snow R., and Jurafsky D., "Distant supervision for relation extraction without labeled data," in *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*, 2009, pp. 1003–1011.

24. Gormley M. R., Yu M., and Dredze M., "Improved relation extraction with feature-rich compositional embedding models," *arXiv preprint arXiv:1505.02419*, 2015.

25. Vu N. T., Adel H., Gupta P., and H. Schütze, "Combining recurrent and convolutional neural networks for relation classification," *arXiv preprint arXiv:1605.07333*, 2016.

26. Xu Y., Mou L., Li G., Chen Y., Peng H., and Jin Z., "Classifying relations via long short term memory networks along shortest dependency paths," in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 1785–1794.

27. Luo X., Liu W., Ma M., and Wang P., "A bidirectional tree tagging scheme for jointly extracting overlapping entities and relations," *arXiv preprint arXiv:2008.13339*, 2020.

28. Devlin J., M.-W. Chang, Lee K., and Toutanova K., "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

29. Sang E. F. and Veenstra J., "Representing text chunks," *arXiv preprint cs/9907006*, 1999.

30. Sun Y., Cheng C., Zhang Y., Zhang C., Zheng L., Wang Z., et al., "Circle loss: A unified perspective of pair similarity optimization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6398–6407.

31. Riedel S., Yao L., and McCallum A., "Modeling relations and their mentions without labeled text," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2010, pp. 148–163.

32. Gardent C., Shimorina A., Narayan S., and L. Perez-Beltrachini, "Creating training corpora for nlg micro-planning," in *55th annual meeting of the Association for Computational Linguistics (ACL)*, 2017.

33. Li S., He W., Shi Y., Jiang W., Liang H., Jiang Y., et al., "Duie: A large-scale chinese dataset for information extraction," in *CCF International Conference on Natural Language Processing and Chinese Computing*. Springer, 2019, pp. 791–800.

34. Wei Z., Su J., Wang Y., Tian Y., and Chang Y., "A novel cascade binary tagging framework for relational triple extraction," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 1476–1488.