BTPK-based learning: An Interpretable Method for Named Entity Recognition

Yulin Chen¹, Zelai Yao¹, Haixiao Chi¹, Dov Gabbay², Bo Yuan¹, Bruno Bentzen¹ and Beishui Liao¹*

¹Zhejiang University
²King’s College London
{sylvia_cyl, byuan.186}@qq.com, zelai_yao@163.com, {haixiaochi, b.bentzen, baiseliao}@zju.edu.cn, dov.gabbay@kcl.ac.uk

Abstract

Named entity recognition (NER) is an essential task in natural language processing, but the internal mechanism of most NER models is a black box for users. In some high-stake decision-making areas, improving the interpretability of an NER method is crucial but challenging. In this paper, based on the existing Deterministic Talmudic Public announcement logic (TPK) model, we propose a novel binary tree model (called BTPK) and apply it to two widely used Bi-RNNs to obtain BTPK-based interpretable ones. Then, we design a counterfactual verification module to verify the BTPK-based learning method. Experimental results on three public datasets show that the BTPK-based learning outperform two classical Bi-RNNs with self-attention, especially on small, simple data and relatively large, complex data. Moreover, the counterfactual verification demonstrates that the explanations provided by the BTPK-based learning method are reasonable and accurate in NER tasks. Besides, the logical reasoning based on BTPK shows how Bi-RNNs handle NER tasks, with different distance of public announcements on long and complex sequences.

1 Introduction

Named Entity Recognition (NER) is an information extraction task aimed at classifying words in unstructured text. Most works focus on improving the model performance and the recognition accuracy [Fu et al., 2020; Lin et al., 2020]. Several methods based on Recurrent Neural Network (RNN) [Žukov-Gregorić et al., 2018; Katiyar and Cardie, 2018; Li et al., 2020; Eligüzel et al., 2021] have been proposed due to their ability to establish dependencies in neighboring words [Li et al., 2022]. However, their inherent black box nature make them unable to explain decision results, especially those involving ambiguous or polysemous words. In application areas where NER technology provides extensive underlying support such as health-care or autonomous driving, interpretable algorithms and a transparent internal decision system is critical for the system reliability and user trust.

It is worth noting that many explanation works have been carried out for RNN [Hou and Zhou, 2020; Wisdom et al., 2016; Krakovna and Doshi-Velez, 2016]. However, there are few research efforts on explainability for NER, although models with explainability are crucial [Agarwal et al., 2021]. In this paper, we use the Talmudic public announcement logic (TPK) model [Abraham et al., 2013] as a tool to explain the process of NER and bring transparency to the RNN-based models, since the reversible and modifiable recognition process in NER is very much in line the problem that TPK is trying to deal with. We propose a new binary TPK model (called BTPK) based on the original TPK model, which can deal with actions depending on future determination by public announcements [Abraham et al., 2013]. Through modifying the accessibility relation in a temporal tree structure, the public announcement at a future state will tell which path should be chosen. In order to map the TPK structure to the internal mechanism of RNN, we need to generate the public announcement in TPK. In semi-supervised learning algorithms, the Pseudo label [Lee and others, 2013] is used to mark the dataset without labels [Wang and Wu, 2020; Wang and Wu, 2020], and this fits the need of inducing public announcement. Moreover, in order to verify the accuracy of a public announcement, we use counterfactual verification [Byrne, 2019].

We summarize our main contributions as follows: (1) We propose a BTPK-based learning method based on the original TPK model and apply it to two widely used Bi-RNNs to obtain BTPK-based interpretable ones. The interpretable BTPK tree shows how Bi-LSTM and Bi-GRU handle NER tasks, and their differences on long and complex sequences. (2) We also design a counterfactual verification module to verify the BTPK-based learning method; the results indicate that explanations from BTPK trees are reasonable and accurate in NER tasks. (3) Experimental results on three public datasets also show that the BTPK-based Bi-RNNs performance is much better than self-attention based Bi-RNNs on small simple data and relatively large and complex data.

2 Problem Statement and Approach

We consider the black box problem for NER tasks, which is formulated by means of the following two aspects:
Semantic ambiguity. In real life, there are many ambiguous name entities, such as product and company names (e.g. “Apple”). Since the context dependency of sentences is essential to resolve ambiguities, thus, it’s necessary to explore how the context affect entity recognition process. To solve this, it will be helpful to understand the internal mechanism of NER models.

Uncertain Bi-RNN workflow. RNNs are very effective for data with sequence characteristics. The output of the current state in a bidirectional RNN model is not only related to its previous states, but also to its subsequent ones. However, it is unknown which state plays a decisive role in the output of the current state, because each Bi-RNN model is a black box model, and the output of the model is only the final labels. Therefore, exploring the influence of the future hidden state on the current state in Bi-RNN models is of great significance to understand how the model works. Therefore, we introduce TPK to show how it works on NER tasks.

2.1 Overview of our method
The overall framework is shown in Figure 1. We firstly train a pseudo learner on training data and get the pseudo labels of the test data. Any classifier can be used as a pseudo learner. Secondly, we calculate the public announcements of the sequences by an attention mechanism. The definition of public announcement in binary TPK tree model (BTPK) is given by Section 2.2. Then we learn BTPK-based Bi-RNNs by training. Finally we measure the explanations of BTPK-based Bi-RNNs by counterfactual verification. In the third step of training BTPK-based Bi-RNNs, we generate the path of BTPK by the hidden states of a Bi-RNN to explore the Bi-RNN property.

In the following, we elaborate the learning steps of the BTPK-based Bi-RNN model and counterfactual verification.

2.2 BTPK-based learning
Task definition. We regard NER as a sequence labeling problem, whose input includes a set of sequences and labels. For any sequence $W = (w_1, w_2, ..., w_n)$, the corresponding labels are $Y = (y_1, y_2, ..., y_n)$, where $w_i$ denotes an entity in the sequence, and $y_i$ comes from BIO tagging schema for labeling elements from the sentence.

The original TPK model. The original TPK model is a deterministic Talmudic K frame based on a time-action tree structure. The time-action model shown in Figure 2 (a) is a tree structure with a set of states $S = \{s_0, s_1, s_2, s_3, \ldots\}$ ($s_0$ is the root), and a set of actions $A = \{a_1, a_2, a_3, \ldots\}$. The elements of $A$ are actions moving the agent from any state to a new one. This corresponds to a successor function $R_k$ (denoted by $\rightarrow$), and can be written in the form $s_0 \rightarrow s_1$. A time-action sequence has the form $s_0 a_1 a_2 \ldots a_n$. In Figure 2, action $a_2$ is ambiguous and the next state is undetermined: either from $s_1$ to $s_2$, or from $s_1$ to $s_3$. So, here, we introduce the idea of public announcement, which would make a clarification to the previous undetermined path, and point out the correct successor of the branch point. It can be written in the form $s_0 \rho \delta s_3$ (denoted by $\rightarrow$). This means that the successor of $s_1$ should be clarified to be $s_2$.

Then, a deterministic TPK model [Abraham et al., 2013] can be defined as a 6-tuple $(S, R, \rho, s_0, \pi)$ where $(S, R, s_0)$ is a tree with root $s_0$ and successor relation $R_k$. $R$ is the transitive closure of $R_k$, $\rho$ is the public announcement function and $\pi$ is an assignment for each atom $q$, such that $s \rightarrow q$ iff $s \in \pi(q)$. $D$ is the distance from the root, and if $s_0 \rho \delta s_3$ then $D(s_3) = D(s_0) + 1$. The semantics for Deterministic TPK Model is as follows: As for the relation $R_k; t \rightarrow A$ iff $\forall s: tR_k s \rightarrow s \in A$. As for the relation $\rho; t \rightarrow A$ iff $\forall s: \rho s \rightarrow s \in A$. As for the relation $\pi; t \rightarrow A$ iff $\forall s; s \pi t \rightarrow s \in A$. $D_n$ is a time constant: $t \rightarrow D_n$ iff the distance of $t$ from $s_n$ is $n$.

BTPK tree. We view the final output of one RNN as the final option, so there are at most two options for each entity in the Bi-RNN. According to the original TPK model, we propose a binary TPK logic tree (BTPK) model, which is defined as a tree $T = \langle V, E \rangle$ with public announcements $A$ and height $|H|$, where $|V|$ is the order and $|V| = V_1 \cup V_2$, $|E|$ is the size and $E$ is represented by the successor relation $R_k$.

Formally, BTPK model is a 2-option full binary TPK tree, generated by induction on height $|H| = 1$, as Figure 2 (b) shows. A tree of height $|H| = 1$ is a single node $s_0$. The root of BTPK (the same as $s_0$ in TPK) is an empty state representing the start of hidden state. We begin to add elements after that, and we write $x < y$ to mean that $x$ is the predecessor of $y$. The inductive steps are as follows: Assume there is a 2-option full binary tree of height $|H| = n$, with junction nodes of each height $k < n$, and top nodes $x_1, x_2, \ldots, x_m$. As Figure 2, we show two options to add new nodes above the top nodes of the tree: 1) to split: add two nodes above each previous top nodes, then $y_1, y_2, \ldots, y_m$ become the top nodes of the tree of height $|H| = n + 1$. 2) not to split: add one node above each top node, then $y_1, y_2, \ldots, y_m$ become the top nodes. If the junction node is a split junction, it is annotated by a word of possible several labels, then we can construct a tree which represents all options in Bi-RNN to recognize a sequence.

Learning a BTPK from Bi-RNNs. For any sequence $W = (w_1, w_2, \ldots, w_n)$ and the corresponding labels $Y = (y_1, y_2, \ldots, y_n)$, we can map the bi-directional hidden states to the path in a BTPK tree $T' = \langle V, E \rangle$, where the hidden states of Bi-RNN constitute the vertices of BTPK. We present the mapping from a Bi-RNN network to a BTPK tree of height $|H| = n + 1$ as follows:

\[
V_1 = \{h_1^f, h_2^f, \ldots, h_n^f\} \\
V_2 = \{h_1^b, h_2^b, \ldots, h_n^b\} \\
H = \{w_0, w_1, w_2, \ldots, w_n\} \\
n \leq |E| < |V| * (|V| - 1) \tag{1}
\]

where $h_i^f$ means the hidden state (feature vector) of the $i^{th}$ element $w_i$ in forward RNN, and $h_i^b$ means the feature vector of the $i^{th}$ element $w_i$ in backward RNN, $w_0$ denotes the start. $V$ is the vertices of graph, which is composed of $V_1$ and $V_2$, where $V_1$ and $V_2$ denote the vertices of forward RNN and backward RNN, respectively. $H$ is denoted by the elements in the sequence. As mentioned above, for all $x, y \in V$, $xy \in E$.
from a Bi-RNN are shown in Algorithm 1. The public announcement. The details of learning a BTPK tree is a pact. What's more, and the label of word \( s_i \) and the system announcement in BTPK model. Overall, the framework consists of four mains steps, namely, training a pseudo learner, training the public announcement, outputting the BTPK-based Bi-RNN, and finally verifying the BTPK-based method by counterfactual reasoning.

**Algorithm 1** BTPK-based learning.

**Input:**
- The Pseudo labels for the dataset \( \Lambda \);
- Current dataset \( N \);

**Output:**
- The labels of the dataset \( T_N \);
- Embedding the input dataset;
- Embedding the Pseudo labels \( \Lambda \);
- Extracting the hidden states \( H_T \) of a bidirectional RNN on step 1., including forward hidden states and backward hidden states;
- Generating the different paths in BTPK model by \( H_T \) according to Equation (1);
- Extracting the hidden states \( H_P \) of an RNN on step 2.;
- Calculating the similarity degree of \( H_T \) and \( H_P \) by attention module to get a public announcement, \( S \) according to Equation (4);
- \( H_S = H_T \cup S \);
- Calculating the labels of \( H_N \) in Softmax layer, denoted as \( T_N \);

9. **return** \( T_N \);

![Figure 3](image-url) Counterfactual verification. (a) shows the causal diagram for the label of word \( x_3 \). (b) shows the counterfactual reasoning process, we observe that the label of \( x_3 \) is “B-video” and ask what would have happened if word \( x_6 \) was deleted.

2.3 **Counterfactual verification**

In NER tasks, labels are decided by the information of former and future words. It’s crucial to know whether the explanation for NER models is accurate and reasonable, i.e., the inner mechanism of NER models or the semantic relations contained in sentences should be right. Counterfactual verification can provide an in-depth view of the essence of this problem. Through its introduction we verify the rationality and accuracy of the explanation of our method.

**Figure 3** (a) shows a diagram representing how the label of element \( x_3 \) is decided in the sentence \( s \), where

\[
s = \{x_0, x_1, x_2, x_3, x_4, x_5, x_6\}
\]

\[
x_i = True / False (0 \leq i \leq 6, i \neq 3)
\]

Each \( x_3 \) is a variable ranging over values \( True \) or \( False \) of \( i^{th} \) word encoded by the data dictionary. If the value of the variable is reset to 0, it’s deleted and False. For example, \( x_6 = True \) means that \( x_6 \) exists in the current sentence and its position is the same as that of the original sentence \( s \). On the other hand, \( x_6 = False \) means that the value of \( x_6 \) reset to 0, because it’s deleted.

**Figure 3** (b) shows counterfactual verification about the label of \( x_3 \) in \( s \). Suppose that \( x_6 \) does not occur in \( s \), would
the label of $x_3$ not be “B-video”? This question requires us to compare the real world with a fictitious and conditional world (called counterfactual world) where $x_6$ was deleted. In the counterfactual world, the value of $x_6$ is reset to 0. If $x_6$ is a public announcement of $x_3$ in $S$, the explanation of the proposed model is considered to be rational and accurate when the label of $x_3$ isn’t “B-video” in the counterfactual world. The counterfactual verification module of the BTPK-based learning model is presented in Algorithm 2.

Algorithm 2 The counterfactual verification of the BTPK-based learning

Input:
- Dataset $N$;
- The output $O_1$ of Bi-RNN models;
- The public announcement matrix $S$;
- A trained Bi-RNN model $M$;
- The value of $k$;

Output:
- The Boolean value of the counterfactual verification, $T$ or $F$;

1: Extract the $k$ elements in the matrix $S$ that have the greatest impact on the entity labels;
2: Reset the encoding of the $k$ elements to 0 to get a counterfactual dataset $N'$;
3: Calculate the label $T_{N'}$ of $N'$;
4: Compare the $T_{N'}$ with $O_1$;
5: If the label of entities in $T_{N'}$ is different with that in $O_1$, then the output $O'$ is $T$; Otherwise, $O'$ is $F$;
6: return $O'$;

3 Experiments

In this section, we mainly evaluate our method across three NER datasets, including two MIT datasets in English, and one Chinese NER dataset.

3.1 Datasets

CBVM CBVM is a Chinese public NER dataset on Github, including 7 label categories. We extract 7814 available sequences from it, including 8791 train samples and 977 test samples.

MIT-V The MIT-V is a well-defined and fine-grained dataset for NER and we use its trivia10k13 corpus. We extract 9769 available sequences from it, including 7816 train samples and 1953 test samples.

MIT-R The MIT-R is another MIT dataset related to restaurants including 17 categories. We extract 9081 available sequences from it, including 7660 train samples and 1921 test samples.

To explore how Bi-RNNs work on ambiguous entities, we divide entities into simple entities and variable entities. Simple entities refer to all entities marked with “O” (means Others), and variable entities refer to those labeled with various labels (excluding “O”), such as “B-book” or “B-video” or “B-music”.

3.2 Setup

In our experiments, we evaluate our method in two settings: self-attention based models and BTPK-based models. We also set up five groups of experiments for two English datasets. In each group, we only select $N$ (100, 200, 300, 400, 500) sentences from the train set to train models for evaluating performance under limited observational instances. In addition, we select sentences (10%, 15%, 18%, 20%, 22%, 30% of training data) from a more complex Chinese dataset to show how BTPK-based learning method performance. We always keep test datasets unchanged in all experiments.

3.3 Evaluation of the experiment

We mainly consider the performance at the entity level, especially the variable entities with ambiguities. Generally, when the categories of labels are extremely unbalanced, macro average can pay more attention to the kind of labels with a small number than micro average. However, it cannot reflect the accuracy of entities’ boundaries. Weighted average considers the proportion of the number of labels in each category in total labels. To explore the recognition of ambiguous entities, macro average and weighted average are both used to measure the performance of models.

4 Results and Discussion

In this section, we will firstly analyze our experimental results on three public datasets. Secondly, we will give the explanation of real instances in the form of BTPK trees, and show how to extend the BTPK-based learning method to an Bi-RNN-CRF. Thirdly, we try to explore the workflow of two gate Bi-RNNs, and reason on ambiguous entities through TPK logic. Finally we verify our method against the real and counterfactual world.

4.1 Main results

Pseudo test

The essence of BTPK-based learning method is to understand how Bi-RNNs work on ambiguous entities, so we mainly focus on variable entities. The key part of BTPK-based learning is to learn public announcements, which are trained from sequences and pseudo labels. As Figure 4 shows, pseudo labels have a great impact on the performance of the model, which is proportional on the whole. In other words, the more accurate the pseudo labels are, the more likely the BTPK-based learning method is to achieve better prediction results. In addition, pseudo labels have a greater influence on BTPK-based Bi-GRU than BTPK-based Bi-LSTM.

| Dataset | Weighted average on MIT-V | Bi-LSTM | Bi-GRU |
|---------|---------------------------|---------|--------|
|         | P  | R  | F  | P  | R  | F  | P  | R  | F  |
| MIT-V   | 86.11 | 87.63 | 84.13 | 83.46 | 85.96 | 83.32 | 81.83 | 85.08 | 80.63 | 85.52 | 89.22 | 86.89 | 86.76 |
|         | 83.00 | 87.36 | 85.06 | 84.68 | 86.70 | 88.07 | 81.44 | 86.21 | 83.05 | 87.64 | 90.30 | 88.72 | 85.67 |
|         | 89.65 | 91.33 | 90.37 | 91.02 | 91.23 | 90.92 | 89.04 | 90.72 | 90.38 | 91.02 | 91.23 | 90.92 | 89.54 |
|         | 30.01 | 19.66 | 20.72 | 23.01 | 19.66 | 20.72 | 23.01 | 19.66 | 20.72 | 23.01 | 19.66 | 20.72 | 23.01 |
|         | 35.03 | 26.33 | 28.16 | 26.01 | 17.79 | 20.03 | 17.79 | 20.03 | 20.34 | 20.34 | 20.34 | 20.34 | 20.34 |

Table 1: Results of self-attention based Bi-RNNs and BTPK-based Bi-RNNs on MIT-V.
Comparison with self-attention based Bi-RNNs

In order to investigate how BTPK-based learning method influences the performance of Bi-RNNs, we carry out two experiments with BTPK-based Bi-LSTM and BTPK-based Bi-GRU. To ensure the fairness of the experiment, we control the consistency of pseudo labels using the pseudo learner by self-attention based Bi-RNNs. That means the accuracy of the pseudo labels is the same as self-attention based Bi-RNNs. Table 1 and Table 2 show the comparison between self-attention based Bi-RNNs and BTPK-based based Bi-RNNs on MIT-V and MIT-R, respectively. The results show that our method achieves an improvement of weighted average and macro average in all settings. Even though it’s hard for the pseudo learner to learn high-quality pseudo labels from small datasets, the BTPK-based learning method can still achieve better performance on variable entities and simple entities. This indicates that BTPK-based learning is a potential effective method to recognize the variable entities from limited data. As Table 1 shows, for MIT-V, the best results of our method yield a boost of 5.76% on weighted average and 10.04% on macro average. As Table 2 shows, for MIT-R, the best results of our method yields a boost of 5.82% on weighted average and 9.26% on macro average. What’s more, we observe that BTRK-based Bi-GRU outperforms BTPK-based Bi-LSTM in general on small and simple datasets.

We also conduct an experiment on CBVM to explore how BTPK-based learning performance on a more complex and larger dataset. As Figure 5(b) shows, 22% of the data with BTPK-based Bi-LSTM can achieve better results than 30% of the data with self-attention based Bi-LSTM. It’s worth noting that even small changes in weighted average may lead to large changes in macro average, because there are far more simple entities than variable entities in all datasets.

4.2 Semantic ambiguity explanation

Based on the experimental results, we can obtain the explanation of ambiguous entities by BTPK-based learning methods. The explanation consists of public announcements and a natural language template.

Figure 6: BTPK trees. (a) shows a BTPK tree of Bi-LSTM, (b) shows a BTPK tree of Bi-LSTM-CRF; where ρ denotes a public announcement and ρ' denotes a CRF announcement

Example 1 Consider the sentence task 103 = “Telling the death is a very good story, downloading free novels can also make money, killing two birds with one stone”.

Question: Why is “Telling the death” recognized as a book name rather than others (the label of simple entities)?

Explanation: Because the “novels” (public announcement) appears in the following words, it is more reasonable to be recognized as “book”.

In this example, the explanation is obtained by a logic reasoning process of a BTPK tree in Figure 6 (a). Let axiom p denotes that the entity is recognized as “I-book”, axiom q denotes that the entity is recognized as “Others”, and axiom p' denotes that the entity is recognized as “B-book”. From semantics of TPK logic, “Telling the death” is correctly recognized iff (s1 ⊨ p') ∧ (s2 ⊨ p') ∧ (s3 ⊨ p') ∧ (s4 ⊨ p'), and all the entities is recognized correctly iff (s1' ⊨ p') ∧ (s2' ⊨ p') ∧ (s3' ⊨ p') ∧ (s4' ⊨ p'), where s1 and s1' denote the states in height |H| = i. When the system gets words from s1 and goes forward to s1', the path can be represented as s1 ⊨ ρ. But there is a public announcement s12,s12,s12 in the system, so the system will go back to s1' and then go forward to the end state s23, generating a new path in red (see Figure 6 (a)), since there’s no other public announcement. The new path can be denoted by (s2 ⊨ ˘p) ∧ (s4 ⊨ ˘q), the relation between s12 and s12' can be represented as s12 → s12'. Thus, the recognition process of the ambiguous entity can be presented in a logical way by BTPK tree, and the public announcement function illustrates
However, in task$_{1001}$, Bi-LSTM still doesn’t recognize the boundaries of an entity correctly, since the label of first entity should be “B-book”. In this case, we extend the BTPK-based learning to the framework of Bi-RNN-CRF. Notice that there may be more announcements and paths in the BTPK-based Bi-RNN-CRF, so we define it as “CRF announcement”. Different from public announcements in Bi-RNNs, CRF announcements are generated by the CRF layer. For task$_{1003}$, Figure 6 (b) shows the explanation of Bi-LSTM-CRF based on BTPK-based learning method. Similarly, when there is a CRF announcement and a new path in green, we have that $\{u_i' \models p\} \land (u_2 \models p) \land (u_3' \models p) \land (u_4' \models q)$, where $u_i$ denotes the state in $|H| = i$. Therefore, a “CRF announcement” mainly appears due to the rule of the labels, and the public announcement mainly appears due to semantics of the words.

### 4.3 Bi-RNNs workflow explanation

Consider a relatively long and difficult Chinese sentence task$_{1000}$ = “The Lord of the Rings is very good, can you recommend similar films?”. To simplify the task, we use the axiom $p$ to denote the option of “video” (including “B-video” and “I-video”) and axiom $q$ to denote the option of “Others”.

We can visualize the BTPK tree of task$_{1000}$ in Figure 7 (c). Similarly, “The Lord of the Rings” is recognized correctly iff $(s_i' \models p) \land (s_i' \models p)$, and all the entities are recognized correctly iff $(s_i' \models \neg p) \land (s_i' \models \neg q)$. In the Bi-LSTM model in Figure 7 (a), the old path in blue can be represented as $(s_1' \models \neg q) \land (s_2' \models \neg q)$, but there is a public announcement $s_2 \rightarrow s_2'$, so it will return to $s_1$ and choose the new path in red, which is the right one. For the Bi-GRU model in Figure 7 (b), there is no public announcement, and only one path of the states is chosen, namely, $(s_1' \models \neg q) \land (s_2' \models \neg q)$, which is not the right one. Then, we can derive the distance of the public announcement $D_a$ based on this knowledge. $D_a$ in BTPK shows how long the future state influences the former state, which can also indicate how long the information can be transmitted in Bi-RNNs. In task$_{1001}$, $D_a$ in Bi-LSTM is 20, but $D_a$ in Bi-GRU is 0. Thus, even Bi-GRU learns more information of variable entities than Bi-LSTM from a small and simple data, but for a large and complex data, the public announcement distance of Bi-LSTM is longer than Bi-GRU, which indicates that the generalization ability of Bi-LSTM is better than that of Bi-GRU.

### 4.4 Counterfactual verification

It’s vital to know whether the explanation of the NER model is correct. For users it may be easy to identify whether it gets the right logic of sentences semantically, but it’s hard to tell whether the explanation can truly reflect the workflow of the model. This work verifies explanations from a causal len. Suppose that the explanation for tasks is accurate and reasonable if labels of entities in a counterfactual world is different from that in a real world. Therefore, when the explanation of a BTPK tree for any sequence $s$ is right, the label of variable entities cannot be recognized accurately in the counterfactual world. In this way, the $P$, $R$, and $F1$ of the counterfactual data would be smaller than that of the real data.

We conduct two counterfactual tests on MIT-R to investigate how public announcements effect the model functioning according to Algorithm 2 with $k = 2$. The original and counterfactual datasets indicate a real and a counterfactual world, respectively. The results of the counterfactual test are shown in Figure 5(a), and indicate that the accuracy of a counterfactual world is much lower than that of a real world for all settings. This shows that the public announcements for sequences are accurate, and, moreover, that the BTPK-based learning can reflect the internal mechanism of the Bi-RNN models.

### 5 Conclusions and future work

We proposed a new BTPK-based learning method for NER tasks, which can effectively and logically capture the semantics in the context and give explanations in form of trees to show the internal mechanism of Bi-RNNs. We implement the BTPK-based learning method to Bi-RNNs and conduct experimental comparisons on three public datasets. The experimental results show that BTPK-based Bi-RNNs outperforms the self-attention based Bi-RNNs. In addition, the proposed counterfactual verification proves that the explanations of our method are accurate and reasonable. For future work, we plan to combine BTPK-based learning (as in this work) with transfer learning for cross-lingual NER tasks.

### References

[Abraham et al., 2013] Michael Abraham, Israel Belfer, Dov M Gabbay, and U Schild. Future determination of
entities in talmudic public announcement logic. Journal of Applied logic, 11(1):63–90, 2013.

[Agarwal et al., 2021] Oshin Agarwal, Yinfei Yang, Byron C Wallace, and Ani Nenkova. Interpretability analysis for named entity recognition to understand system predictions and how they can improve. Computational Linguistics, 47(1):117–140, 2021.

[Byrne, 2019] Ruth MJ Byrne. Counterfactuals in explainable artificial intelligence (xai): Evidence from human reasoning. In IJCAI, pages 6276–6282, 2019.

[Byrne et al., 2021] Nazmiye Eligüzel, Cihan Çetinkaya, and Türcay Dereli. Application of named entity recognition on tweets during earthquake disaster: a deep learning-based approach. Soft Computing, pages 1–27, 2021.

[Fu et al., 2020] Jinlan Fu, Pengfei Liu, and Graham Neubig. Interpretable multi-dataset evaluation for named entity recognition. arXiv preprint arXiv:2011.06854, 2020.

[Hou and Zhou, 2020] Bo-Jian Hou and Zhi-Hua Zhou. Learning with interpretable structure from gated rnn. IEEE transactions on neural networks and learning systems, 31(7):2267–2279, 2020.

[Katiyar and Cardie, 2018] Arzoo Katiyar and Claire Cardie. Nested named entity recognition revisited. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 861–871, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[Krakovna and Doshi-Velez, 2016] Viktoriya Krakovna and Finale Doshi-Velez. Increasing the interpretability of recurrent neural networks using hidden markov models. arXiv preprint arXiv:1606.05320, 2016.

[Lee and others, 2013] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In Workshop on challenges in representation learning, ICML, volume 3, page 896, 2013.

[Li et al., 2020] Jianqiang Li, Shenhe Zhao, Jijiang Yang, Zhisheng Huang, Bo Liu, Shi Chen, Hui Pan, and Qing Wang. Wcp-rnn: a novel rnn-based approach for bio-ner in chinese emrs. The journal of supercomputing, 76(3):1450–1467, 2020.

[Li et al., 2022] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. IEEE Transactions on Knowledge and Data Engineering, 34(1):50–70, 2022.

[Lin et al., 2020] Bill Yuchen Lin, Dong-Ho Lee, Ming Shen, Ryan Moreno, Xiao Huang, Prashant Shiralkar, and Xiang Ren. Triggerner: Learning with entity triggers as explanations for named entity recognition. arXiv preprint arXiv:2004.07493, 2020.

[Vaswani et al., 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in neural information processing systems, pages 5998–6008, 2017.

[Wang and Wu, 2020] Guo-Hua Wang and Jianxin Wu. Repetitive reprediction deep decipher for semi-supervised learning. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 6170–6177, 2020.

[Wisdom et al., 2016] Scott Wisdom, Thomas Powers, James Pitton, and Les Atlas. Interpretable recurrent neural networks using sequential sparse recovery. arXiv preprint arXiv:1611.07252, 2016.

[Žukov-Gregorič et al., 2018] Andrej Žukov-Gregorič, Yoram Bachrach, and Sam Coope. Named entity recognition with parallel recurrent neural networks. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 69–74, Melbourne, Australia, July 2018. Association for Computational Linguistics.