

Article

Not peer-reviewed version

Biomedical Relation Extraction with Forest-Based Tagging Framework

Karam Paul , Fei Lee , [Woods Ali](#) *

Posted Date: 28 November 2023

doi: 10.20944/preprints202311.1817.v1

Keywords: Information extraction; Biomedical relation extraction; Dual-binary tree structure



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

Biomedical Relation Extraction with Forest-Based Tagging Framework

Karam Paul ¹, Fei Lee ² and Woods Ali ^{1,*}

¹ University of Charleston

² National University of Singapore

* Correspondence: aliwoods12345@gmail.com

Abstract: The groundbreaking Advanced Forest-Based Tagging Framework (AFTF) represents a paradigm shift in the domain of information extraction from medical texts. AFTF introduces a novel dual-binary tree structure that redefines how entity-relation triples are extracted. This innovative approach directly tackles the limitations of traditional linear and graph-based methods, effectively addressing challenges related to overlapping triples and computational efficiency. The AFTF model stands as a beacon of excellence, surpassing established baselines by significant margins in comprehensive evaluations on two pivotal medical datasets. Notably, AFTF achieves remarkable improvements in F1 scores, showcasing its prowess in accurate information extraction from complex medical narratives. Beyond its exceptional performance in the medical domain, AFTF exhibits remarkable versatility. This adaptability is vividly demonstrated through its robust performance across three diverse public datasets, further affirming its position as a versatile and reliable solution for information extraction tasks. This paper provides a comprehensive exposition of the AFTF architecture, shedding light on its innovative design principles and its efficient handling of intricate medical texts. AFTF represents a groundbreaking step forward in the realm of information extraction, promising enhanced accuracy, efficiency, and adaptability for a wide range of applications.

Keywords: information extraction; biomedical relation extraction; dual-binary tree structure

1. Introduction

The extraction of relational data from biomedical documents, encompassing electronic health records and medication package inserts, plays a pivotal role in the construction of detailed biomedical knowledge graphs [1–7]. Termed as biomedical relation extraction (RE), this intricate task involves pinpointing and deciphering the interrelations among entities embedded in these unstructured textual sources.

In the realm of RE, traditional methodologies have predominantly followed a sequential pipeline strategy. Initially, this involved the identification of entities within the textual corpus through a process known as named entity recognition (NER), followed by the elucidation of inter-entity relationships through relation classification (RC) modules [2,8]. Nevertheless, this sequential approach was fraught with a fundamental shortcoming: inaccuracies in the NER phase had the propensity to adversely affect the subsequent RC phase, thus compromising the overall relation classification accuracy. To overcome these shortcomings, a new wave of integrated approaches emerged, synthesizing both NER and RC into a singular, cohesive process aimed at reducing the cascading errors [9–11].

A notable innovation in this field was the adaptation of the RE task into a sequential tagging paradigm. The vanguard in this approach [10] attempted to linearly align relation triples, transforming them into a sequential tagging format. Despite its innovation, this method grappled with computational inefficiencies, marked by a complexity factor of $O(s|\mathcal{R}|)$ - where s signifies the sentence's length and $|\mathcal{R}|$ denotes the dimension of the pre-established relation set. A particularly challenging aspect was its inability to effectively manage overlapping relation triples, typified by scenarios such as *EntityPairOverlap* (EPO) and *SingleEntityOverlap* (SEO) [11]. Later advancements in tagging methodologies [12–15] sought to rectify these overlapping dilemmas by conceptualizing

relation triples in a graph-based format, albeit at the cost of escalating the computational complexity from $O(s|\mathcal{R}|)$ to $O(s^2|\mathcal{R}|)$.

Departing from these generalist approaches, our research zeroes in on the distinct relational architectures inherent in medical texts. A quintessential illustration of this is observed in the tree-like relational networks within EHRs and MPIs, where an individual pharmaceutical can be linked to a myriad of effects or components, thereby weaving a multifaceted nexus of relationships.

In response to these complexities, we put forth the Advanced Forest-Based Tagging Framework (AFTF). Diverging from previous models, AFTF adopts a forest-based structural representation over a graph-based one to articulate relation triples in medical sentences. This groundbreaking methodology transforms these forest structures into a tagging sequence with a length of 20, all the while preserving a computationally manageable complexity of $O(s|\mathcal{R}|)$. The AFTF operates through a tripartite structure: initially by aggregating relation triples within a sentence; subsequently, by architecting these aggregates into dual-binary trees based on their textual sequence; and finally, by forging a bi-directional linkage between these binary trees and the token-level sequence tags. This innovative configuration facilitates an efficient and expansive representation of relation triples.

The principal contributions of this paper are threefold:

- The unveiling of the Advanced Forest-Based Tagging Framework (AFTF), an avant-garde tagging paradigm employing a forest structure to succinctly encapsulate relation triples within medical texts.
- The formulation of a comprehensive medical RE model that harnesses the power of AFTF to autonomously predict relation triples.
- Empirical validation of AFTF's efficacy, underscored by its exceptional performance on two dedicated medical datasets and its versatility across three diverse public datasets, thereby evidencing its proficiency in managing overlapping relational structures with ease.

2. Related Work

Relation extraction (RE), a fundamental component in text mining and information extraction, has witnessed diverse advancements with the evolution of neural network architectures. Several pipeline-based RE models [8,16,17] have emerged, offering enhanced performance through their innovative structures. To address the cascading error issue inherent in pipeline models, integrated RE approaches were introduced. Notably, some studies [9,18] improved upon this by sharing encoder representations between the named entity recognition (NER) and relation classification (RC) modules. Although these approaches reduced error propagation, they continued to process entity and relation extraction sequentially.

A significant focus within the field has shifted towards sequential tagging-based methods [10,12,19], reconceptualizing extraction tasks as sequence labeling challenges. This shift aimed to bridge the gap between RC and NER, specifically targeting the issue of overlapping triples. While these methodologies marked substantial progress, they still fell short in certain complex scenarios or were time-intensive. Alternatively, seq2seq framework-based approaches [11,20,21] have been explored for directly generating relational triples. Although these methods circumvent the need for intricate tagging schemes, they face challenges in determining the optimal length for generated sequences and often require extensive decoding time.

In the specialized realm of medical RE, adaptations of generic RE models have been undertaken to meet the unique challenges of medical texts. For instance, [22] enhanced medical entity recognition by incorporating entity-related information like part-of-speech (POS) tags and medical ontologies into a joint RE model. Similarly, [23] introduced trainable character embeddings to address the out-of-vocabulary (OOV) problem prevalent in medical texts. Further, [24] leveraged domain-specific knowledge, integrating medical databases into an RE model for Chinese medical instructions, showcasing the importance of domain-specific adaptations in medical RE.

To build upon these developments, we introduce the Advanced Forest-Based Tagging Framework (AFTF). AFTF represents a paradigm shift in medical RE by utilizing a forest-based structure instead of traditional linear or graph-based models. This approach is specifically tailored to address the complexities and unique characteristics of medical texts, offering a more efficient and accurate method for extracting relational data. AFTF not only improves upon the limitations of existing models but also demonstrates versatility and robustness across various datasets, marking a significant advancement in the field of medical information extraction.

3. Proposed Framework

In this section, we delineate our approach for the nuanced segmentation of sentences with overlapping relational triples. We then elucidate the process of translating these segmented sentences into AFTF tag sequences and subsequently extracting triples from the AFTF tags. Lastly, we present our integrated RE model designed for the prediction of AFTF tags.

Prior studies like [11] classified sentences with overlapping relational triples into categories like *EPO* and *SEO*. A sentence falls under *EPO* if it contains triples with identical entity sets ($\{e_1, e_2\}$), and *SEO* if the entity sets, while different, share at least one overlapping entity. Notably, some sentences may exhibit characteristics of both *EPO* and *SEO*. In our refined approach, we further dissect *SEO* into two sub-categories: *ExcludeLoopSentences (ELS)* and *IncludeLoopSentences (ILS)*, contingent on the presence of relational loops. A sentence is classified as *ILS* if it meets the following criteria: (1) distinct entity sets; (2) at least one overlapping entity in each set; (3) a minimum of two overlapping entities in some sets. Conversely, a sentence is deemed an *ELS* if it aligns with *SEO* but does not meet the *ILS* criteria. Notably, an *ILS* sentence's relational graph features at least one loop, whereas an *ELS* sentence's relational graph does not, disregarding edge direction. According to our dataset analysis across various domains (see Table 1), a significant proportion of overlapping medical sentences are categorized as *ELS*.

3.1. Advanced Forest-Based Tagging Framework (AFTF)

We introduce the Advanced Forest-Based Tagging Framework (AFTF), a novel tagging paradigm employing dual-binary tree structures to encapsulate the three sentence types. Algorithm 1 illustrates our approach and algorithm for handling these varied sentence categories. In *EPO* sentences, even though some triples share identical entity sets, they differ in their relation categories. Therefore, we categorize such triples based on their relation category within a sentence and label them distinctly. For instance, the triple (*America, Capital, Washington*), categorized under *Capital*, is grouped independently, while triples categorized under *Contains* are grouped separately. It is essential to note that, while these triple groups are labeled individually, all triples are predicted concurrently by our joint RE model, ensuring the preservation of their inter-relationships.

Algorithm 1 Relation-to-Forest Transformation

Require: A sentence with s words, S ; An array of m relational triples sharing a common relation category in S, RT ; An array of n entities in S, EN ;

Ensure: A forest-based binary relation tree, B ;

```

1: /* Construct a relational forest. */
2: Initialize arrays  $L = []$  and  $F = []$ ;
3: for each entity index  $i \in [1, n]$  do
4:   Set  $l$  as an array of all location pairs of  $EN_i$  in  $S$ ;
5:   Append elements of  $l$  to  $L$ ;
6: end for
7: Sort  $L$  in ascending order based on the starting index;
8: while  $L$  is non-empty do
9:   Start a new tree  $T$  with root  $L_1$ , then remove  $L_1$  from  $L$ ;
10:  for each index  $i = 2$  to  $length(L)$  do
11:    if  $L_i$  is not in  $T$  and a valid relation exists between  $L_i$  and any node in  $T$  then
12:      Remove  $L_i$  from  $L$  and incorporate it into  $T$ ;
13:    end if
14:  end for
15:  Append  $T$  to forest  $F$ ;
16: end while
17: /* Convert the forest into a binary tree. */
18: Start with an empty stack  $S_t$ , push root nodes of  $F$  onto  $S_t$  in sequence;
19: Form a binary tree  $B$  with the root node as  $F_1$ 's root;
20: for each index  $i = 2$  to  $length(F)$  do
21:   Add  $F_i$ 's root as the right child of  $F_{i-1}$ 's root in  $B$ ;
22: end for
23: while  $S_t$  is not empty do
24:   Set  $node_{cur} = Pop(S_t)$ , and  $C$  as the children array of  $node_{cur}$  in  $F$ ;
25:   Attach  $C_1$  as the left child of  $node_{cur}$  in  $B$ ;
26:   for each index  $i = 2$  to  $length(C)$  do
27:     Add  $C_i$  as the right child of  $C_{i-1}$  in  $B$ ;
28:   end for
29:   Push children of  $node_{cur}$  onto  $S_t$  in sequence;
30: end while

```

3.1.1. Advanced Handling of ELS Sentences

The defining characteristic of *ELS* sentences is the absence of loops within their relational graphs, allowing for a complete representation via a forest structure. We address *ELS* sentences with a two-step Tree Tagging approach within the Advanced Forest-Based Tagging Framework (AFTF).

From Relations to Trees Detailed in **Algorithm 1**, our initial step involves pinpointing all entities within a sentence and constructing a relational forest. The process begins by designating the first entity as the root of tree T_1 and methodically adding related entities to T_1 . This process continues until no further entities can be integrated into the current tree, at which point a new tree T_i is initialized. This iterative procedure continues until all entities are accommodated, culminating in a comprehensive forest F . Subsequently, F is transformed into a binary tree B . In this tree, for any entity node e , the first child in F becomes e 's left child in B , while the immediate right sibling becomes e 's right child. Edges in B are annotated with *Brother* if they originally connected root nodes in F , and all edges are directed from e_1 to e_2 in a given triple.

Tree to Tag Conversion Each word in the sentence is tagged in accordance with the binary relation tree B . Words not forming part of any entity receive the tag "O". For entity-inclusive words, their tag comprises four components:

- Part 1 (P_1) signifies the word's position within entity node e , using the "BIES" (Begin, Inside, End, Single) system.
- Part 2 (P_2) relates to the edge between e and its parent in B , indicating root or sibling relationships, or showcasing the child's position and role in the entity pair.
- Part 3 (P_3) and Part 4 (P_4) denote the relationship between e and its left and right children in B , respectively, or indicate the absence of such children.

3.1.2. Handling ILS Sentences in AFTF

Given the presence of at least one loop in the relation graph of an *ILS* sentence, a single forest structure is insufficient for a full representation. To address this, we construct an additional forest based on the reverse order of entity appearance, transforming it into a backward binary tree. This

enables the generation of backward tree tags for each sentence. By combining forward and backward tags, AFTF can accommodate a broader range of triples.

3.2. From Tags to Triples in AFTF

In processing forward tags, we first identify all root nodes using the criteria $P_2 = \text{Root}$ or $P_2 = \text{Brother}$. Starting from these roots, we systematically reconnect nodes based on matching P_3 or P_4 with other nodes' P_2 . In cases of multiple potential child nodes, the nearest following node is selected, adhering to the original entity order in forest construction. Any missed child or sibling nodes, and their subtrees, are omitted during tree reconstruction. This reconstructed forest then serves as the basis for recursively extracting relational triples.

The approach for backward tags mirrors that of the forward tags, with the distinction that the nearest preceding node is chosen when multiple child nodes are viable. This method ensures comprehensive extraction of relational information, even in complex sentence structures.

3.3. Joint Relation Extraction Model

This section outlines the architecture of our Joint Relation Extraction Model, the Advanced Forest-Based Tagging Framework (AFTF), which comprises four integral components: the text embedding module, the encoder module, the decoder module, and our specifically tailored loss function.

3.3.1. Text Embedding

For a given word w in an input sentence, its representation $e \in \mathbb{R}^d$ in the AFTF model is a fusion of four distinct embeddings:

$$e = \text{Linear}([e^w; e^l; e^c; e^p]) \quad (1)$$

Here, $e^w \in \mathbb{R}^{d_w}$ denotes the traditional word embedding; $e^l \in \mathbb{R}^{d_l}$ represents the contextualized word embedding obtained from advanced language models like BERT [25]; $e^c \in \mathbb{R}^{d_c}$ is the character-level embedding capturing finer syntactic details; and $e^p \in \mathbb{R}^{d_p}$ corresponds to the part-of-speech (POS) embedding, providing syntactic context.

3.3.2. Encoder

The encoder in AFTF aims to generate rich context-aware vector representations for each word. It utilizes a combination of Bi-directional Long Short-Term Memory (Bi-LSTM) and multi-head self-attention layers, inspired by the Transformer model [26]. The Bi-LSTM layer, comprising forward and backward LSTMs [27], processes the word embeddings $V = [e_1, \dots, e_s]$ of a sentence with s words. The output and hidden state of each word are computed as follows:

$$o_t, h_t = \text{lstm_block}(e_t, h_{t-1}) \quad (2)$$

The final Bi-LSTM state for each word is obtained by concatenating its corresponding forward and backward LSTM hidden states:

$$\dot{h}_t = [\overrightarrow{h}_t, \overleftarrow{h}_{s-t+1}] \quad (3)$$

The multi-head self-attention mechanism then refines these states through a series of attention-based transformations:

$$\text{MultiHead}(\mathbf{H}) = [\text{head}_1(\mathbf{H}); \dots; \text{head}_h(\mathbf{H})] \quad (4)$$

$$\text{head}_i(\mathbf{H}) = \text{Atten}(\mathbf{H}\mathbf{W}_i^Q, \mathbf{H}\mathbf{W}_i^K, \mathbf{H}\mathbf{V}_i^Q) \quad (5)$$

$$\text{Atten}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \quad (6)$$

Here, \mathbf{H} denotes the hidden states from the Bi-LSTM layer, and the matrices \mathbf{W}_i^Q , \mathbf{W}_i^K , and \mathbf{V}_i^Q are learnable parameters.

3.3.3. Decoder

The decoder in the AFTF model interprets the contextualized word representations to predict AFTF tags for each word. It comprises several Linear layers, designed either as one-head or multi-head configurations for tag prediction. For instance, in predicting a forward tree tag, the one-head approach (following [10]) combines all tag components into a single label predicted by one Linear layer. Conversely, the multi-head approach predicts each tag component with separate Linear layers, significantly reducing model complexity and computational demand. Our implementation favors the multi-head structure for its efficiency.

3.3.4. Loss Function

The loss function for training AFTF employs a weighted bias objective, formulated as:

$$L = \lambda_1 L_1 + \lambda_2 L_2 + \lambda_3 L_3 + \lambda_4 L_4 \quad (7)$$

Here, $L_j (j \in \{1, 2, 3, 4\})$ are bias cross entropy functions (as per [10]) for the four parts of AFTF tags, with λ_j serving as their respective weights. We define separate loss functions L^f and L^b for forward and backward tags, respectively, and compute the total loss L^T as:

$$L^T = L^f + \gamma L^b \quad (8)$$

The parameter γ here is a weight hyperparameter, optimizing the balance between forward and backward tag predictions.

4. Experiments

4.1. Experimental Setup

Our AFTF scheme is rigorously evaluated across five diverse datasets, including two from the medical field (ADE [28] and CMeIE [29]) and three from other domains (NYT [30], WebNLG [31], and DuIE [32]). The ADE dataset, derived from English medical reports, encompasses 4.2k samples focusing on the *Adverse-Effect* relation. As per [33], we exclude samples with overlapping entities and implement a 10-fold cross-validation strategy. The CMeIE dataset, a comprehensive Chinese medical compilation, contains 28k sentences and 44 relation types, originating from medical texts and clinical practices. The NYT dataset, comprising 66.1k sentences with 24 relation types, is sourced from New York Times articles. The WebNLG dataset, initially designed for Natural Language Generation, includes 6.2k sentences and 246 relation types, adapted for RE. Lastly, the DuIE dataset, provided by Baidu Inc., features 214.7k sentences across 49 relation categories. Due to the unavailability of DuIE's test set, we repartition the original training set for our experiments.

Table 1 showcases statistics of overlapping triples in these datasets, highlighting the prevalence and distribution of such samples, particularly in the medical datasets where *ELS* samples are notably predominant. The effectiveness of AFTF is gauged using micro Precision (*Prec*), Recall (*Rec*), and F1 score (*F1*). We consider a triple as correctly predicted only if all its components (e_1, r, e_2) are accurately identified. Unlike some baselines that employ Partial Matching, our evaluation strictly requires the entire triple to be correct.

In applying AFTF to medical datasets, we use BiTT-Med (now referred to as AFTF-Med), as discussed in Section 3.3. The embeddings in the text embedding module are set to $d_w = d_c = d_p = 100$ and $d_l = d = 768$. We utilize GloVe vectors for word embeddings and pre-trained BERT-base embeddings for contextual information. Character embeddings, crucial for handling OOV issues,

are computed via an LSTM. POS tags, generated by SpaCy, also contribute to the model’s input. The encoder module includes two layers of Bi-LSTM and two layers of multi-head self-attention. The loss function weights are set to balance the contribution of different parts of AFTF tags. For non-medical datasets, we adapt AFTF-Med to create simplified versions, AFTF-LSTM and AFTF-BERT, to accommodate the peculiarities of these domains.

Table 1. Statistics of the overlapping samples in the five datasets, highlighting the proportion of *ELS* cases.

Dataset	<i>EPO</i>	<i>ELS</i>	<i>ILS</i>	Overlap Samples	<i>ELS Ratio</i>
ADE [28]	118	1,216	159	1,391	0.874
CMeIE [29]	381	8,805	457	9,213	0.956
NYT [30]	17,004	10,740	2,006	25,422	0.422
WebNLG [31]	622	2,894	1,294	3,957	0.731
DuIE [32]	15,672	94,891	11,780	109,675	0.865

Table 2 contrasts AFTF-Med with various baselines, highlighting its superiority in extracting medical relations. Notably, our model demonstrates exceptional precision and recall, outperforming existing methods.

For the other datasets, the AFTF variants (LSTM and BERT) were tested to assess the generalizability of the AFTF framework beyond medical contexts. These simplified models, adapted from AFTF-Med, show the flexibility and robustness of the AFTF scheme in handling diverse data domains. For the non-medical datasets, we restructured the data partitions to suit our experimental requirements.

We compared AFTF against 13 baseline models, which can be categorized into one-stage models (simultaneously outputting entities and relations) and two-stage models (first identifying entities, then classifying relations). Some of these models include Neural Joint [22], NovelTagging [10], Multi-head [34], CasRel [19], and others.

Table 2. Performance comparison of AFTF-Med and other models on ADE and CMeIE datasets, demonstrating AFTF’s proficiency in extracting medical relations.

Model	Encoder	<i>Prec</i>	<i>Rec</i>	<i>F1</i>
ADE				
Neural Joint [22]	L	64.0	62.9	63.4
Multi-head [34]	L	72.1	77.2	74.5
Multi-head + AT [35]	L	-	-	75.5
Rel-Metric [36]	L+C	<u>77.4</u>	<u>77.3</u>	77.3
Table-Sequence [37]	ALB	-	-	<u>80.1</u>
PFN [33]	Bb	-	-	80.0
AFTF-Med (Ours)	Bb	83.1	81.3	82.1
CMeIE				
NovelTagging [10] [♡]	Bb	51.4	17.1	25.6
GraphRel-1p [38] [♡]	Bb+G	31.2	26.0	28.4
GraphRel-2p [38] [♡]	Bb+G	28.5	23.1	25.5
CasRel [19] [♡]	Bb	<u>53.5</u>	<u>28.2</u>	37.0
ER+RE [39]	ALB	-	-	<u>47.6</u>
AFTF-Med (Ours)	Bb	55.6	45.5	50.1

4.2. Main Results

Comparison of Results on Medical Datasets Table 2 presents the performance comparison of AFTF on two medical datasets. Our AFTF model achieves robust *F1* scores of 82.1% and 50.1% on ADE

and CMeIE, respectively. Notably, our model outperforms Table-Sequence by 2.0% in *F1* on ADE and surpasses Rel-Metric by 5.7% in *Precision* and 4.0% in *Recall*. On CMeIE, AFTF outperforms ER+RE by 2.5% in *F1* and surpasses CasRel by 2.1% in *Precision* and 17.3% in *Recall*. These results underscore the effectiveness of our AFTF model in the joint extraction of medical entities and relations.

Comparison of Results on Common Datasets To further assess the performance of our AFTF scheme in joint relation extraction tasks, we deploy AFTF-LSTM and AFTF-BERT models on three general datasets. Table 3 presents the performance comparison of our models with baseline methods on NYT, WebNLG, and DuIE datasets.

Table 3. Performance comparison on NYT, WebNLG, and DuIE datasets, showcasing the effectiveness of AFTF in various domains.

Model	Encoder	<i>Prec</i>	<i>Rec</i>	<i>F1</i>
NYT				
NovelTagging [10] [♣]	L	62.4	31.7	42.0
CopyRE-Mul [11] [♣]	L	61.0	56.6	58.7
GraphRel-2p [38] [♣]	L+G	63.9	60.0	61.9
PA [12]	L	49.4	59.1	53.8
CopyMTL-Mul [21]	L	75.7	68.7	72.0
NovelTagging [10] [♡]	Bb	<u>89.0</u>	55.6	69.3
CopyRE-Mul [11] [♡]	Bb	39.1	36.5	37.8
GraphRel-2p [38] [♡]	Bb+G	82.5	57.9	68.1
CasRel [19] [♣]	Bb	89.7	89.5	89.6
AFTF-LSTM (Ours)	L	66.5	76.3	71.1
AFTF-BERT (Ours)	Bb	89.7	<u>88.0</u>	<u>88.9</u>
WebNLG				
NovelTagging [10] [♣]	L	52.5	19.3	28.3
CopyRE-Mul [11] [♣]	L	37.7	36.4	37.1
GraphRel-2p [38] [♣]	L+G	44.7	41.1	42.9
CopyMTL-Mul [21]	L	58.0	54.9	56.4
TPLinker [14]	Bb	<u>88.9</u>	84.5	86.7
AFTF-LSTM (Ours)	L	83.8	66.0	73.8
AFTF-BERT (Ours)	Bb	89.1	<u>83.0</u>	<u>86.2</u>
DuIE				
NovelTagging [10] [♡]	Bb	<u>75.0</u>	38.0	50.4
GraphRel-1p [38] [♡]	Bb+G	52.2	23.9	32.8
GraphRel-2p [38] [♡]	Bb+G	41.1	25.8	31.8
CaseRel [19] [♡]	Bb	75.7	<u>80.0</u>	<u>77.8</u>
AFTF-BERT (Ours)	Bb	75.7	80.6	78.0

For AFTF-LSTM, it achieves impressive *F1* scores of 71.1% and 73.8% on NYT and WebNLG, respectively. AFTF-LSTM outperforms CopyMTL-One by 7.6% and 11.1% in *Recall* on NYT and WebNLG. The notable improvement in *Recall* highlights the utility of the AFTF scheme when handling overlapping triples.

AFTF-BERT also performs well, achieving solid *F1* scores of 88.9%, 86.2%, and 78.0% on NYT, WebNLG, and DuIE, respectively. AFTF-BERT excels in *Precision* on all three datasets, with scores of 89.7%, 89.1%, and 75.7%. Additionally, sequential tagging-based models (NovelTagging, CasRel, TPLinker, and AFTF-BERT) demonstrate higher *Precision* compared to other models, emphasizing their superiority in conservative prediction. However, AFTF-BERT does not perform as well as the best method on NYT and WebNLG, possibly due to the lower proportion of "ELS" sentences in samples with overlapping triples in these datasets compared to DuIE. This highlights the advantages of our AFTF scheme in handling "ELS" sentences.

4.3. Efficiency of AFTF-based Models

Our AFTF-based models can predict AFTF tags with low computational complexity of $O(s|\mathcal{R}|)$, making them more efficient compared to graph-based and two-stage models. First, we compare our AFTF-BERT model with one-stage baselines. In our experiments, most baseline models with BERT-base encoder converge at around the 15th epoch, demonstrating fast convergence. Moreover, AFTF-BERT has a similar number of decoder parameters (48.9M) as NovelTagging (47.3M) but significantly fewer parameters than other one-stage baseline models such as GraphRel-2p (106.4M) and TPLinker (1293.3M). This indicates that our framework achieves faster convergence than Graph-Rel and TPLinker with the same encoder.

Second, we compare AFTF-BERT with the competitive two-stage baseline CasRel in terms of training efficiency. On a NVIDIA GeForce RTX 2080 Ti, AFTF-BERT takes 255.0 seconds per training epoch, while CasRel takes 1701.2 seconds. CasRel requires approximately 6.7 times more time to traverse the dataset than AFTF-BERT due to the two-stage model's strategy of copying sentences into multiple samples for training based on the number of head entities.

4.4. Ablation Study

Table 4 presents the results of ablation tests conducted on the AFTF-BERT model using the NYT dataset. We investigate the effects of handling methods for overlapping triples in the AFTF scheme and the decoder architecture.

In the ablation tests, "AFTF-BERT w/o Group" omits the "EPO Handling" and instead incorporates relation categories information into P_2 , P_3 , and P_4 of the AFTF tags. "AFTF-BERT w/o Bidirectional" involves building only the forward forest in a sentence and generating forward tags. "AFTF-BERT w/o Multi-head" replaces the multi-head structure with a one-head structure in the decoder module.

The results show that the grouping operation significantly improves the performance of our framework, not only for "EPO" (17.2%) but also for "ELS" (11.1%) and "ILS" (17.2%) sentences. It reduces the complexity of relation graphs in a sentence and reduces the number of AFTF tag categories. The backward forest effectively complements triples that cannot be represented by a single forward forest, leading to a boost in $F1$ scores for "ILS" sentences by 10.8%. The multi-head structure decreases decoder parameters and improves $F1$ scores by 1.2% compared to the one-head structure, indicating that multi-head enhances training for labels with fewer occurrences.

Table 4. Results of ablation study on the AFTF-BERT model using the NYT dataset. $F1$ -EPO, $F1$ -ELS, $F1$ -ILS represent $F1$ scores for EPO, ELS, and ILS sentences, respectively. $F1$ -All is the overall $F1$ score. Decoder parameters are in millions (M), and training time is in seconds per epoch (s/epoch).

Metrics	w/o Group	w/o Bidirectional	w/o Multi-head	AFTF-BERT
$F1$ -EPO	74.0	90.2	90.2	91.2
$F1$ -ELS	76.2	84.5	85.5	87.3
$F1$ -ILS	68.3	74.7	81.8	85.5
$F1$ -All	81.9	88.2	87.7	88.9
Decoder Params	71.3	48.0	68.5	48.9
Training Time	-	-	2125.0	1739.0

5. Conclusion

In this study, we have drawn inspiration from the tree-like structures commonly found in medical texts and introduced the Bidirectional Tree Tagging (BiTT) scheme for the precise labeling of overlapping entities and relations within medical sentences. Our approach achieves remarkable accuracy and efficiency in this challenging task. We have developed a unified Relation Extraction (RE) model named BiTT-Med, along with two simplified variants, BiTT-LSTM and BiTT-BERT, for experimental evaluation. The results obtained on both publicly available medical datasets and general-purpose datasets underscore the superiority of our proposed methodology, particularly in

handling complex cases such as sentences with overlapping entities and relations (ELS). Our future research endeavors will focus on further enhancing the BiTT scheme and refining our RE models in the following ways: (1) Investigating the possibility of reconstructing binary forests instead of binary trees when extracting information from BiTT tags. This approach aims to minimize error propagation in cases where nodes are inadvertently dropped. (2) Proposing additional rule constraints for BiTT to enhance its robustness. (3) Exploring the integration of more potent pre-trained encoders into our extraction framework to achieve even better performance. The AFTF framework has shown great promise in advancing the field of information extraction, and we are committed to its continual improvement and refinement in pursuit of even greater accuracy and efficiency in real-world applications.

References

1. Gehui Shen, Zhi-Hong Deng, Ting Huang, and Xi Chen. Learning to compose over tree structures via pos tags for sentence representation. *Expert Systems with Applications*, 141:112917, March 2020. ISSN 0957-4174. doi: 10.1016/j.eswa.2019.112917. URL <http://dx.doi.org/10.1016/j.eswa.2019.112917>.
2. Richard Socher, Andrej Karpathy, Quoc V. Le, Christopher D. Manning, and Andrew Y. Ng. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2:207–218, December 2014. ISSN 2307-387X. doi: 10.1162/tacl_a_00177. URL http://dx.doi.org/10.1162/tacl_a_00177.
3. Hao Fei, Meishan Zhang, and Donghong Ji. Cross-lingual semantic role labeling with high-quality translated training corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7014–7026, 2020.
4. Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. Next-gpt: Any-to-any multimodal llm, 2023.
5. Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, 2015. doi: 10.3115/v1/p15-1150. URL <http://dx.doi.org/10.3115/v1/p15-1150>.
6. Hao Fei, Yafeng Ren, and Donghong Ji. Retrofitting structure-aware transformer language model for end tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 2151–2161, 2020.
7. Baohua Wang, Junlian Huang, Haihong Zheng, and Hui Wu. Semi-supervised recursive autoencoders for social review spam detection. In *2016 12th International Conference on Computational Intelligence and Security (CIS)*. IEEE, December 2016. doi: 10.1109/cis.2016.0035. URL <http://dx.doi.org/10.1109/cis.2016.0035>.
8. D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao. Relation classification via convolutional deep neural network. in *Proceedings of COLING*, pages 2335–2344, 2014.
9. M. Miwa and M. Bansal. End-to-end relation extraction using lstms on sequences and tree structures. in *Proceedings of ACL*, pages 1105–1116, 2016.
10. S. Zheng, F. Wang, H. Bao, Y. Hao, P. Zhou, and B. Xu. Joint extraction of entities and relations based on a novel tagging scheme. in *Proceedings of ACL*, pages 1227–1236, 2017.
11. X. Zeng, D. Zeng, S. He, K. Liu, and J. Zhao. Extracting relational facts by an end-to-end neural model with copy mechanism. in *Proceedings of ACL*, pages 506–514, 2018.
12. D. Dai, X. Xiao, Y. Lyu, S. Dou, Q. She, and H. Wang. Joint extraction of entities and overlapping relations using position-attentive sequence labeling. in *Proceedings of AAAI*, pages 6300–6308, 2019.
13. Hao Fei, Shengqiong Wu, Jingye Li, Bobo Li, Fei Li, Libo Qin, Meishan Zhang, Min Zhang, and Tat-Seng Chua. Lasuie: Unifying information extraction with latent adaptive structure-aware generative language model. In *Proceedings of the Advances in Neural Information Processing Systems, NeurIPS 2022*, pages 15460–15475, 2022.
14. Y. Wang, B. Yu, Y. Zhang, T. Liu, H. Zhu, and L. Sun. Tplinker: single-stage joint extraction of entities and relations through token pair linking. in *Proceedings of COLING*, pages 1572–1582, 2020.
15. Hao Fei, Fei Li, Bobo Li, and Donghong Ji. Encoder-decoder based unified semantic role labeling with label-aware syntax. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12794–12802, 2021.

16. Y. Xu, L. Mou, G. Li, Y. Chen, H. Peng, and Z. Jin. Classifying relations via long short term memory networks along shortest dependency paths. *in Proceedings of EMNLP*, pages 1785–1794, 2015.
17. N. T. Vu, H. Adel, P. Gupta, and H. Schutze. Combining recurrent and convolutional neural networks for relation classification. *in Proceedings of ACL*, pages 534–539, 2016.
18. A. Katiyar and C. Cardie. Investigating lstms for joint extraction of opinion entities and relations. *in Proceedings of ACL*, pages 919–929, 2016.
19. Z. Wei, J. Su, Y. Wang, Y. Tian, and Y. Chang. A novel cascade binary tagging framework for relational triple extraction. *in Proceedings of ACL*, pages 1476–1488, 2020.
20. T. Nayak and H. Tou. Effective modeling of encoder-decoder architecture for joint entity and relation extraction. *in Proceedings of AAAI*, pages 8528–8535, 2020.
21. D. Zeng, H. Zhang, and Q. Liu. Copymtl: copy mechanism for joint extraction of entities and relations with multi-task learning. *in Proceedings of AAAI*, pages 9507–9514, 2020.
22. F. Li, Y. Zhang, and D. Ji. Joint models for extracting adverse drug events from biomedical text. *in Proceedings of IJCAI*, pages 2838–2844, 2016.
23. F. Li, M. Zhang, G. Fu, and D. Ji. A neural joint model for entity and relation extraction from biomedical text, " *bmc bioinform.*, vol. 18, no. 1., 198:1-198:11, 2017.
24. T. Qi, S. Qiu, X. Shen, H. Chen, S. Yang, and H. Wen. et al., "kemre: knowledge-enhanced medical relation extraction for chinese medicine instructions," *j. In Biomed. Informatics*, vol. 120., 103834, 2021.
25. J. Devlin, M. W. Chang, K. Lee, and K. Toutanova. Bert: pre-training of deep bidirectional transformers for language understanding. *in Proceedings of ACL*, pages 4171–4186, 2019.
26. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *In Advances in neural information processing systems*, pages 5998–6008, 2017.
27. S. Hochreiter and J. Schmidhuber. Long short-term memory. 1735-1780,, 1997.
28. H. Gurulingappa, A. Mateen, J. Fluck Angus, M. Hofmann-Apitius, and L. Toldo. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *In J*, pages 885–892. *Biomed. Informatics*, vol. 45, no. 5, 2012.
29. T. Guan, H. Zan, X. Zhou, H. Xu, and K. Zhang. *CMelE: construction and evaluation of Chinese medical information extraction dataset*. *In proceedings of NLPCC*, 2020.
30. S. Riedel, L. Yao, and A. McCallum. Modeling relations and their mentions without labeled text. *in Proceedings of ECML-PKDD*, pages 148–163, 2010.
31. C. Gardent, A. Shimorina, S. Narayan, and L. Perez-Beltrachini. Creating training corpora for nlg micro-planners. *in Proceedings of ACL*, pages 179–188, 2017.
32. S. Li, W. He, Y. Shi, W. Jiang, H. Liang, and Y. Jiang. Duie: a large-scale chinese dataset for information extraction. *in Proceedings of NLPCC*, pages 791–800, 2019.
33. Z. Yan, C. Zhang, J. Fu, Q. Zhang, and Z. Wei. A partition filter network for joint entity and relation extraction. *in Proceedings of EMNLP*, pages 185–197, 2021.
34. G. Bekoulis, J. Deleu, T. Demeester, and C. Develder. Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Systems with Applications*, pages 34–45, 2018.
35. G. Bekoulis, J. Deleu, T. Demeester, and C. Develder. Adversarial training for multi-context joint entity and relation extraction. *in Proceedings of EMNLP*, pages 2830–2836, 2018.
36. T. Tran and R. Kavuluru. Neural metric learning for fast end-to-end relation extraction. *CoRR*, 1905, 2019.
37. J. Wang and W. Lu. Two are better than one: Joint entity and relation extraction with table-sequence encoders. *in Proceedings of EMNLP*, pages 1706–1721, 2020.
38. T. J. Fu, P. H. Li, and W. Y. Ma. Graphrel: modeling text as relational graphs for joint entity and relation extraction. *in Proceedings of ACL*, pages 1409–1418, 2019.
39. N. Zhang, M. Chen, Z. Bi, X. Liang, L. Li, and X. Shang. et al., "cblue: a chinese biomedical language understanding evaluation. *in Proceedings of ACL*, pages 7888–7915, 2022.
40. Shengqiong Wu, Hao Fei, Yafeng Ren, Donghong Ji, and Jingye Li. Learn from syntax: Improving pair-wise aspect and opinion terms extraction with rich syntactic knowledge. *In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pages 3957–3963, 2021.
41. Ling Zhuang, Hao Fei, and Po Hu. Knowledge-enhanced event relation extraction via event ontology prompt. *Inf. Fusion*, 100:101919, 2023.

42. Shengqiong Wu, Hao Fei, Fei Li, Meishan Zhang, Yijiang Liu, Chong Teng, and Donghong Ji. Mastering the explicit opinion-role interaction: Syntax-aided neural transition system for unified opinion role labeling. In *Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence*, pages 11513–11521, 2022.
43. Shengqiong Wu, Hao Fei, Wei Ji, and Tat-Seng Chua. Cross2StrA: Unpaired cross-lingual image captioning with cross-lingual cross-modal structure-pivoted alignment. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2593–2608, 2023.
44. Hao Fei, Yue Zhang, Yafeng Ren, and Donghong Ji. Latent emotion memory for multi-label emotion classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7692–7699, 2020.
45. Hao Fei, Yafeng Ren, Yue Zhang, Donghong Ji, and Xiaohui Liang. Enriching contextualized language model from knowledge graph for biomedical information extraction. *Briefings in Bioinformatics*, 22(3), 2021.
46. Junshan Wang, Zhicong Lu, Guojia Song, Yue Fan, Lun Du, and Wei Lin. Tag2vec: Learning tag representations in tag networks. In *The World Wide Web Conference, WWW '19*. ACM, May 2019. doi: 10.1145/3308558.3313622. URL <http://dx.doi.org/10.1145/3308558.3313622>.
47. Shengqiong Wu, Hao Fei, Yixin Cao, Lidong Bing, and Tat-Seng Chua. Information screening whilst exploiting! multimodal relation extraction with feature denoising and multimodal topic modeling. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14734–14751, 2023.
48. Hao Fei, Yafeng Ren, and Donghong Ji. Boundaries and edges rethinking: An end-to-end neural model for overlapping entity relation extraction. *Information Processing & Management*, 57(6):102311, 2020.
49. Jingye Li, Hao Fei, Jiang Liu, Shengqiong Wu, Meishan Zhang, Chong Teng, Donghong Ji, and Fei Li. Unified named entity recognition as word-word relation classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10965–10973, 2022.
50. Jingye Li, Kang Xu, Fei Li, Hao Fei, Yafeng Ren, and Donghong Ji. MRN: A locally and globally mention-based reasoning network for document-level relation extraction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1359–1370, 2021.
51. Fengqi Wang, Fei Li, Hao Fei, Jingye Li, Shengqiong Wu, Fangfang Su, Wenxuan Shi, Donghong Ji, and Bo Cai. Entity-centered cross-document relation extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9871–9881, 2022.
52. Hu Cao, Jingye Li, Fangfang Su, Fei Li, Hao Fei, Shengqiong Wu, Bobo Li, Liang Zhao, and Donghong Ji. OneEE: A one-stage framework for fast overlapping and nested event extraction. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1953–1964, 2022.
53. Wenxuan Shi, Fei Li, Jingye Li, Hao Fei, and Donghong Ji. Effective token graph modeling using a novel labeling strategy for structured sentiment analysis. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4232–4241, 2022.
54. Hao Fei, Shengqiong Wu, Yafeng Ren, Fei Li, and Donghong Ji. Better combine them together! integrating syntactic constituency and dependency representations for semantic role labeling. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021*, pages 549–559, 2021.
55. Wenhao Zhu, Tengjun Yao, Wu Zhang, and Baogang Wei. Part-of-speech-based long short-term memory network for learning sentence representations. *IEEE Access*, 7:51810–51816, 2019. ISSN 2169-3536. doi: 10.1109/access.2019.2911983. URL <http://dx.doi.org/10.1109/access.2019.2911983>.
56. Hao Fei, Shengqiong Wu, Yafeng Ren, and Meishan Zhang. Matching structure for dual learning. In *Proceedings of the International Conference on Machine Learning, ICML*, pages 6373–6391, 2022.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.