

PAPER • OPEN ACCESS

THT: Transfer Head and Tail for Joint Entity Relation Extraction

To cite this article: Huike Zou *et al* 2023 *J. Phys.: Conf. Ser.* **2504** 012015

View the [article online](#) for updates and enhancements.

You may also like

- [A token system for incentivising equal contributions in group work](#)
Siming Guo, Scott Carr and David Kumi
- [Electronic and spin-orbit properties of the kagome MOF family \$M_3\(1,2,5,6,9,10\text{-triphenylenehexathiol}\)_2\$ \(\$M = \text{Ni, Pt, Cu and Au}\$ \)](#)
Orlando J Silveira and Helio Chacham
- [OmniJet: the first cross-task foundation model for particle physics](#)
Joschka Birk, Anna Hallin and Gregor Kasieczka

THT: Transfer Head and Tail for Joint Entity Relation Extraction

Huikou Zou¹[0000–0002–4107–3535], Jiawei Jiang² and Yan Fu¹

¹ Big Data Research Center, University of Electronic Science and Technology of China, Chengdu 611731, P.R. China

² Chengdu Union Big Data Tech. Inc., Chengdu 610041, P.R. China

E-mail: fuyan@uetsc.edu.cn

Abstract. Extracting entities and relations from unstructured text is the basic task of building knowledge graphs. Various methods based on sequence labeling have been proposed recently, but most of them use uniform token representations to identify entities. It is clear that the first and last words of the head and tail entities should have different representations. For this reason, we propose a method based on word interaction that provides different representations for the first and last words of entities. Specifically, initial token representation is obtained by BERT. Each token interacts with the next k tokens for the entity's first token recognition, interacts with the previous k tokens for the entity's last token recognition, and merges them to obtain the head entity. Then recognize the tail entities based on the head entities and form entity pairs. Finally, get the triples based on a simple multi-classification for relation determination. Extensive experimental results on four datasets show that THT performs better than state-of-the-art baselines.

1. Introduction

Extracting triples composed of entities and relations from unstructured text is a fundamental task of building knowledge graphs. Previous researches [1, 2] take pipeline strategies to decompose triple extraction into two tasks: entity identification and relation classification. In order to reduce the impact of error propagation, end-to-end strategies are often adopted to construct joint models, especially those based on sequence labeling.

The core of the joint extraction model based on sequence labeling method is entity recognition and relation extraction. MultiHead [3] uses character-level encoding to obtain token representations for head entities and takes a multi-head selection method to obtain tail entities under a particular relation. NovelTagging [4] directly views the joint extraction as a whole sequence labeling task, with triple reorganization via Tag. ETL-Span [5] decomposes the task into multiple sequence labeling tasks, each including two multi-label binary classifications, then combine results to obtain triples. CasRel [6] considers the relation as a function of the head entity to the tail entity and takes Span method to introduce the entity information to the identification of the tail entity under a specific relation. TPLinker [7] transforms triple identifying into a token linking problem. PRGC [8] uses potential relation extraction to reduce relation redundancy, and subsequently performs entity identification as well as entity alignment with the help of table filling to form triples. In addition, joint extraction model includes both table filling and text generation. GraphRel [9] introduces Bi-GCN for entity information propagation and extraction



of relations at once. CopyRE [10] and CopyMTL [11] progressively generate head entities, relations and tail entities to compose triples.

The sequence labeling model treats joint extraction as a multi-task for entity recognition and entity alignment. Intuitively, taking different representations for the first and last tokens of an entity can better identify it. The first token of entity should be related to many tokens after it, and the last token of entity should be related to tokens before it. Inspired by this idea, a sequence labeling based model named Transfer Head and Tail (THT for short) is proposed in this paper. Specifically, given a sentence, initial representation is obtained by BERT firstly. Secondly, encode transformation is used to obtain a representation for identifying the first and last tokens of entities and a representation for determining relations. Thirdly, entity recognition module identifies the head entity and corresponding tail entity. Finally, relation determination module is used to determine the relationship of entity pairs and to form a triple.

The main contributions of this work can be summarized as follows:

- (i) Using token interaction to improve the accuracy of entity recognition.
- (ii) Information both before and after the token is used in the model.
- (iii) Extensive experiments have been conducted on four common datasets, and the results are comparable to state-of-the-art models.

2. Method

The proposed model THT consists 3 modules, encode transformation module, entity recognition module and relation determination module. The architecture of model is shown in Figure 1.

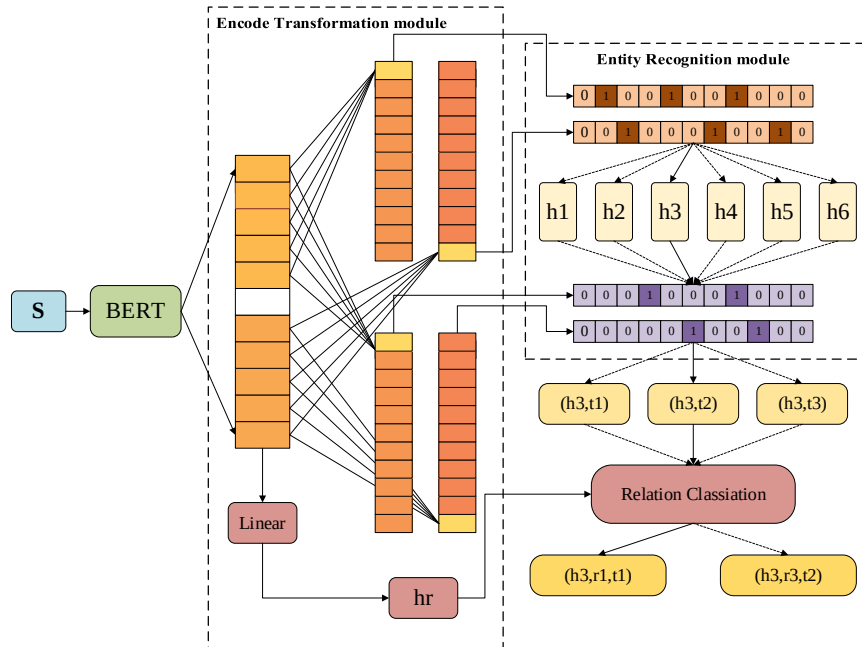


Figure 1. The architecture of THT.

2.1. Encode Transformation Module

An initial representation for each word in the sentence is firstly generated with the help of pre-training model BERT, and then the representation of head entity, tail entity, and relation

is generated respectively. For entity boundary recognition, a word representation obtained by absorbing information from surrounding words would be beneficial. For example, for the entity “apple tree”, “apple” can be better judged as the first token with the help of “tree”, and similarly, “tree” can be further judged as the last token with the help of “apple”. The first token of entity is recognized by the token representation obtained by interacting with the next k tokens using local-attention. The last token of entity is recognized by the token representation obtained by interacting with the previous k tokens using local-attention. For relation judgments, the representation mapping generated directly by BERT is used as a new representation. The encode process can be formulated as:

$$\mathbf{h} = \{e_1, e_2, \dots, e_n\} = \text{BERT}(x_1, x_2, \dots, x_n) \quad (1)$$

$$\left. \begin{aligned} \mathbf{Q}_{hs} &= \mathbf{W}_1 \mathbf{h} + \mathbf{b}_1 = \{q_{hs_1}, q_{hs_2}, \dots, q_{hs_n}\} \\ \mathbf{K}_{hs} &= \mathbf{W}_2 \mathbf{h} + \mathbf{b}_2 = \{k_{hs_1}, k_{hs_2}, \dots, k_{hs_n}\} \\ \mathbf{V}_{hs} &= \mathbf{W}_3 \mathbf{h} + \mathbf{b}_3 = \{v_{hs_1}, v_{hs_2}, \dots, v_{hs_n}\} \\ \alpha_{ij} &= \frac{\exp(q_{hs_i}^T \cdot k_{hs_j})}{\sum_{l=i}^{i+k} \exp(q_{hs_i}^T \cdot k_{hs_l})} \\ e_{hs_i} &= \sum_{j=i-k}^i \alpha_{ij} v_{hs_j} \\ \mathbf{h}_{hs} &= \{e_{hs_1}, e_{hs_2}, \dots, e_{hs_n}\} \end{aligned} \right\} \quad (2)$$

$$\left. \begin{aligned} \mathbf{Q}_{he} &= \mathbf{W}_4 \mathbf{h} + \mathbf{b}_4 = \{q_{he_1}, q_{he_2}, \dots, q_{he_n}\} \\ \mathbf{K}_{he} &= \mathbf{W}_5 \mathbf{h} + \mathbf{b}_5 = \{k_{he_1}, k_{he_2}, \dots, k_{he_n}\} \\ \mathbf{V}_{he} &= \mathbf{W}_6 \mathbf{h} + \mathbf{b}_6 = \{v_{he_1}, v_{he_2}, \dots, v_{he_n}\} \\ \beta_{ij} &= \frac{\exp(q_{he_i}^T \cdot k_{he_j})}{\sum_{l=i-k}^i \exp(q_{he_i}^T \cdot k_{he_l})} \\ e_{he_i} &= \sum_{j=i-k}^i \beta_{ij} v_{he_j} \\ \mathbf{h}_{he} &= \{e_{he_1}, e_{he_2}, \dots, e_{he_n}\} \end{aligned} \right\} \quad (3)$$

$$\mathbf{h}_r = \mathbf{W}_r \mathbf{h} + \mathbf{b}_r \quad (4)$$

where $\mathbf{W}_i (i = 1, 2, \dots, 6)$ is the trainable weight, $\mathbf{h}$ is the initial representation of the token vector, and $\mathbf{h}_{hs}$, $\mathbf{h}_{he}$ are used to identify the first and last tokens of head entity. Similarly, $\mathbf{h}_{ts}$, $\mathbf{h}_{te}$ can be obtained to identify the first and last tokens of tail entity.

2.2. Entity Recognition Module

The entity recognition module consists of four binary classification modules where two of them are used to recognize the first and last tokens of head entity and other two are used to recognize that of tail entity related to the head entity. Adopting two binary classification modules for entity, first and last token recognition can solve the problem of nested entities. Specifically, each token identified by the first token classification is combined with each token identified by the second token classification in the specified range to obtain entities so as to identify entities as many as possible. The probabilities of the first and last tokens of head entity are:

$$\left. \begin{aligned} p_{hs}^i &= \text{Sigmoid}(\mathbf{W}_{hs} \mathbf{h}_{hs}^i + \mathbf{b}_{hs}) \\ p_{he}^i &= \text{Sigmoid}(\mathbf{W}_{he} \mathbf{h}_{he}^i + \mathbf{b}_{he}) \end{aligned} \right\} \quad (5)$$

where $\mathbf{h}_{hs}^i$ and $\mathbf{h}_{he}^i$ denote the i -th token representations used to identify the first and last tokens of the head entity respectively, and p_{hs}^i and p_{he}^i are the probabilities of the first and last tokens of head entity respectively, $\text{Sigmoid}()$ is an activation function.

Intuitively, tail entities can be well identified by introducing relevant head entity information, so the head entity representation is connected to the tail entity representation for entity identification. The process of tail entity identification based on head entity can be formulated

as::

$$\left. \begin{aligned} \mathbf{v}_{hs} &= \text{avgpool}(\mathbf{h}_{hs}^s, \mathbf{h}_{hs}^{s+1}, \mathbf{h}_{hs}^{s+2}, \dots, \mathbf{h}_{hs}^{s+k=e}) \\ \mathbf{v}_{he} &= \text{avgpool}(\mathbf{h}_{he}^s, \mathbf{h}_{he}^{s+1}, \mathbf{h}_{he}^{s+2}, \dots, \mathbf{h}_{he}^{s+k=e}) \\ \mathbf{v}_h^h &= \text{mean}(\mathbf{v}_{hs}, \mathbf{v}_{he}) \end{aligned} \right\} \quad (6)$$

$$\left. \begin{aligned} p_{ts}^i &= \text{Sigmoid}(\mathbf{W}_{ts}(\mathbf{h}_{ts}^i; \mathbf{v}_h^h) + \mathbf{b}_{ts}) \\ p_{te}^i &= \text{Sigmoid}(\mathbf{W}_{te}(\mathbf{h}_{te}^i; \mathbf{v}_h^h) + \mathbf{b}_{te}) \end{aligned} \right\} \quad (7)$$

where $\text{avgpool}()$ is used to get the representation of the entity under a particular token representation, $\mathbf{v}_{hs}, \mathbf{v}_{he}$ are the representations corresponding to first and last tokens, $\mathbf{v}_h^h$ is the head entity representation, p_{ts}^i is the probability corresponding to the first or the last token of the entity, $\mathbf{W}_{ts}$ and $\mathbf{W}_{te}$ are the trainable weight.

2.3. Relation Determination Module

After combining head and tail entity to obtain entity pair, a simple relation judgment is performed to form triples. The relation determination module is simply defined as a multi-classifier. Specifically, for each entity pair, its head and tail entity representations are pooled on token representation corresponding to relation, and then connect them to judge relation. The process can be formulated as:

$$\left. \begin{aligned} \mathbf{v}_r^h &= \text{avgpool}(\mathbf{h}_r^{s_1}, \mathbf{h}_r^{s_1+1}, \mathbf{h}_r^{s_1+2}, \dots, \mathbf{h}_r^{s_1+k=e_1}) \\ \mathbf{v}_r^t &= \text{avgpool}(\mathbf{h}_r^{s_2}, \mathbf{h}_r^{s_2+1}, \mathbf{h}_r^{s_2+2}, \dots, \mathbf{h}_r^{s_2+k=e_2}) \\ p &= \text{Sigmoid}(\mathbf{W}_p(\mathbf{v}_r^h; \mathbf{v}_r^t) + \mathbf{b}_p) \end{aligned} \right\} \quad (8)$$

where $\mathbf{h}_r^{s_i+j}$ ($i = 1, 2; j = 0, 1, 2, \dots, k$) are token representations used for relation judgment, $\mathbf{v}_r^h$ and $\mathbf{v}_r^t$ are head and tail representations, p is the probability of the head and tail entities under all relations, $\mathbf{W}_p$ is the trainable weight.

We train THT using joint training. Loss functions for entity recognition module and relation determination are as follows:

$$ce(p, t) = -[t \log p + (1 - t) \log(1 - p)] \quad (9)$$

$$L_{er} = \frac{1}{2 * l} \sum_{m \in \{h, t\}} \sum_{n \in \{s, e\}} \sum_{i=1}^l ce(p_{mn}^i, t_{mn}^i) \quad (10)$$

$$L_{ed} = \frac{1}{|R|} \sum_{i=1}^{|R|} ce(p_r^i, t_r^i) \quad (11)$$

where $ce(p, t)$ is a cross entropy loss, p is probability and $t \in \{0, 1\}$ is true flag and l is the number of tokens.

The total loss is summed over two component assignments:

$$L = L_{er} + L_{ed} \quad (12)$$

3. Experiments

3.1. Datasets and Evaluation

Following up on previous work, experiments are conducted on two widely used datasets NYT [12] and WebNLG [13]. NYT is generated by remote supervision with 56k training samples and 5k test samples. WebNLG is originally constructed for NLG with 5k training samples and 703 test samples. The whole entity span is labeled in NYT and WebNLG. Modified versions of them,

i.e., NYT* and WebNLG* with only last word of each entity being labeled are also considered in this paper.

Similar to the previous work, the prediction is correct only if the triple is exactly the same as the ground truth, which means matching the last word of the entity in NYT* and WebNLG* or the whole entity span in NYT and WebNLG. Meanwhile, three metrics of standard micro Precision, Recall, and F1-score are taken for the evaluation of methods.

3.2. Implementation Details

All our experiments are conducted on RTX 3080. English based BERT released by Huggingface is taken as the pre-trained model. The threshold of first and last tokens of head and tail entities, and that of relation is set to 0.5. The batch size is set to 8 and 16 on NYT and WebNLG respectively. All parameters are optimized using AdamW. Maximal sequence length is set to 100. Experiments are attempted with the best hyperparameter according to the validation. The model with the highest F1 in the validation data is evaluated on the test data.

3.3. Results

In the experiments, THT is compared with eight Joint Extraction models, including NovelTagging [4], CopyRE [10], MultiHead [3], GraphRel [9], ETL-Span [5], CasRel [6], TPLinker [7] and PRGC [8].

Table 1. Precision(%), Recall(%) and F1-score(%) of THT and baselines.

Model	NYT*			WebNLG*			NYT			WebNLG		
	Prec.	Rec.	F ₁	Prec.	Rec.	F ₁	Prec.	Rec.	F ₁	Prec.	Rec.	F ₁
NovelTagging [4]	-	-	-	-	-	-	32.8	30.6	31.7	52.5	19.3	28.3
CopyRE [10]	61.0	56.6	58.7	37.7	36.4	37.1	-	-	-	-	-	-
MultiHead [3]	-	-	-	-	-	-	60.7	58.6	59.6	57.5	54.1	55.7
GraphRel [9]	63.9	60	61.9	44.7	41.1	42.9	-	-	-	-	-	-
ETL-span [5]	84.9	72.3	78.1	84	91.5	87.6	85.5	71.7	78	84.3	82	83.1
CasRel [6]	89.7	89.5	89.6	93.4	90.1	91.8	-	-	-	-	-	-
TPLinker [7]	91.3	92.5	91.9	91.8	92	91.9	91.4	92.6	92	88.9	84.5	86.7
PRGC [8]	93.3	91.9	92.6	94.0	92.1	93.0	93.5	91.9	92.7	89.9	87.2	88.5
THT*	-	-	-	69.2	85.5	76.6	-	-	-	89.5	87.1	83.1
THT	92.7	92.8	92.8	93.7	93.4	93.5	92.6	92.7	92.7	91.2	88.7	89.9

Table 1 shows the results of our model and 8 baselines on four datasets. It can be observed that THT achieves the highest F1 on four datasets, especially on the WebNLG, where the F1 gains 1.4 improvement and all three evaluation metrics get the best performance. THT* is a cut-down version of THT with a unified token representation for entity recognition and relation classification. It can be seen that all metrics of THT are better than THT* on WebNLG* and WebNLG, indicating that different token representations are effective. The advantages of the model are mainly focused on the encode module. Firstly, the head and tail entities in a sentence can be better identified by using different token representations. Secondly, for the first and last tokens of entities, combining different representations and then combining them can improve the accuracy of entity recognition and can find real entities with higher probability comparing with unified representation.

4. Conclusion and Future work

In this paper, a new joint extraction of entities and relations named THT is proposed. Current models with uniform token representation lead to incomplete recognition of head and tail entities. So, a word interaction method is proposed to solve this problem. Large number of experiments are conducted on four datasets to evaluate the proposed model from different aspects. The

experimental results show that the proposed method is effective . The complexity of the proposed method is relatively high since more tokens should be analyzed while identifying the entities. More efficient method will be studied in the future.

Acknowledgments

This work was supported by Key Research Project on Philosophy and Social Sciences of the Ministry of Education under Grant No 21JZD055, by the National Natural Science Foundation of China with Grant No 71872117 and by the Fundamental Research for the Central Universities with Grant No ZYGX2019J074.

References

- [1] Bunescu R, Mooney R 2008 *human language technology conference and conference on empirical methods in natural language processing* 724–731.
- [2] Zelenko D, Aone C, Richardella A 2003 *Journal of machine learning research* 1083-1106.
- [3] Bekoulis G, Deleu J, Demeester T 2018 *Expert Systems with Applications* 34-45.
- [4] Zheng S, Wang F, Bao H 2017 *55th Annual Meeting of the Association for Computational Linguistics* 1227-1236.
- [5] Yu B, Zhang Z, Shu X 2019 *24th European Conference on Artificial Intelligence*.
- [6] Wei Z, Su J, Wang Y 2020 *58th Annual Meeting of the Association for Computational Linguistics* 1476-1488.
- [7] Wang Y, Yu B, Zhang Y 2020 *28th International Conference on Computational Linguistics* 1572-1582.
- [8] Zheng H, Wen R, Chen X 2021 *59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing* 6225-6235.
- [9] Fu T, Li P, Ma W 2019 *57th Annual Meeting of the Association for Computational Linguistics* 1409-1418.
- [10] Zeng X, Zeng D, He S 2018 *56th Annual Meeting of the Association for Computational Linguistics* 506-514.
- [11] Zeng D, Zhang H, Liu Q 2020 *AAAI conference on artificial intelligence* 9507-9514.
- [12] Riedel S, Yao L, McCallum A 2010 *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. 148-163.
- [13] Gardent C, Shimorina A, Narayan S 2020 *55th annual meeting of the Association for Computational Linguistics (ACL)*.