RelMatter's System for TAC-KBP 2020: Recognizing Ultra Fine-grained Entities

Jianling Li*, Pancheng Wang*, Kunyuan Pang*, Wuhang Lin*,
Shasha Li, Jintao Tang, Ting Wang
School of Computer, National University of Defense Technology, Changsha 410073, China
{jianlingl,wangpancheng13,pangkunyuan,wuhanglin}@nudt.edu.cn
{shashali,tangjintao,tingwang}@nudt.edu.cn

Abstract

This paper gives a detailed description of RelMatter's system for TAC-KBP 2020 RUFES (Recognizing Ultra Fine-grained EntitieS) task. The RUFES task requires systems to recognize name, nominal, and pronominal mentions of entities in news articles, from a newly developed ontology with over 200 types that cover a variety of topics in the news. Our system consists of a two-step pipeline architecture, which includes a mention detection module and an entity typing module. The former module aims to detect the candidate mentions for the given corpus, while the latter module links the mentions to the fine-grained ontology.

1 Introduction

Many real world applications in scenarios such as disaster relief and technical support require systems that recognize a wide variety of entity types (e.g., technical terms, lawsuits, disease, crisis, vehicles, food, biomedical entities) with limited training data for each type. The KBP 2020 RUFES task (Recognizing Ultra Fine-grained EntitieS) challenges systems to recognize name, nominal, and pronominal mentions of entities in news articles, from a newly developed ontology with over 200 types that cover a variety of topics in the news.

The task is defined as: given an input document, a system is required to automatically identify an entity as a cluster of name, nominal, and/or pronominal mentions, and classify the entity into one or more of the types defined in the ontology. The ontology is developed by NIST (National Institute of Standards and Technology) with approximately 200 fine-grained entity types, which follows a three-level x.y.z hierarchy.

*Authors contributed equally

In this paper, we propose a two-step pipeline, including a mention detection module and a entity typing module, to solve this task. We use DYGIE++ (Wadden et al., 2019), which is trained on ACE05 corpus, to detect candidate mentions. And we train a modified two-step mention-aware attention model FET (Lin and Ji, 2019) for entity typing. We also utilize the extra information from the typing ontology list to alleviate the problem for lack of data.

2 Mention Detection Module

Our mention detection module uses DYGIE++ (Wadden et al., 2019) to detect the mentions. DYGIE++ is a unified multitask framework for three information extraction tasks: named entity recognition, relation extraction, and event extraction. Shared span representations are constructed by refining contextualized word embeddings via span graph updates.

The model is trained on four different datasets, but only ACE05 corpus has the most similar annotation specification and data format with contrast to TAC-KBP. Therefore, we train DYGIE++ on ACE05 and use the model to detect the candidate mentions of the evaluation corpus.

3 Entity Typing Module

To simplify the task, we directly link the mention to the fine-grained ontology instead of using the entity. For the specified ontology types of KBP2020 RUFES task (Recognizing Ultra Fine-grained EntitieS) contains the types that differ from the other dataset such as Ontonote5.0 or English data in (Pan et al., 2017), we improved the model Fet (Lin and Ji, 2019) which performs excellently on the entity typing task.

3.1 Baseline

Our baseline model applies a two-step mention-aware attention mechanism to extract the most rel-

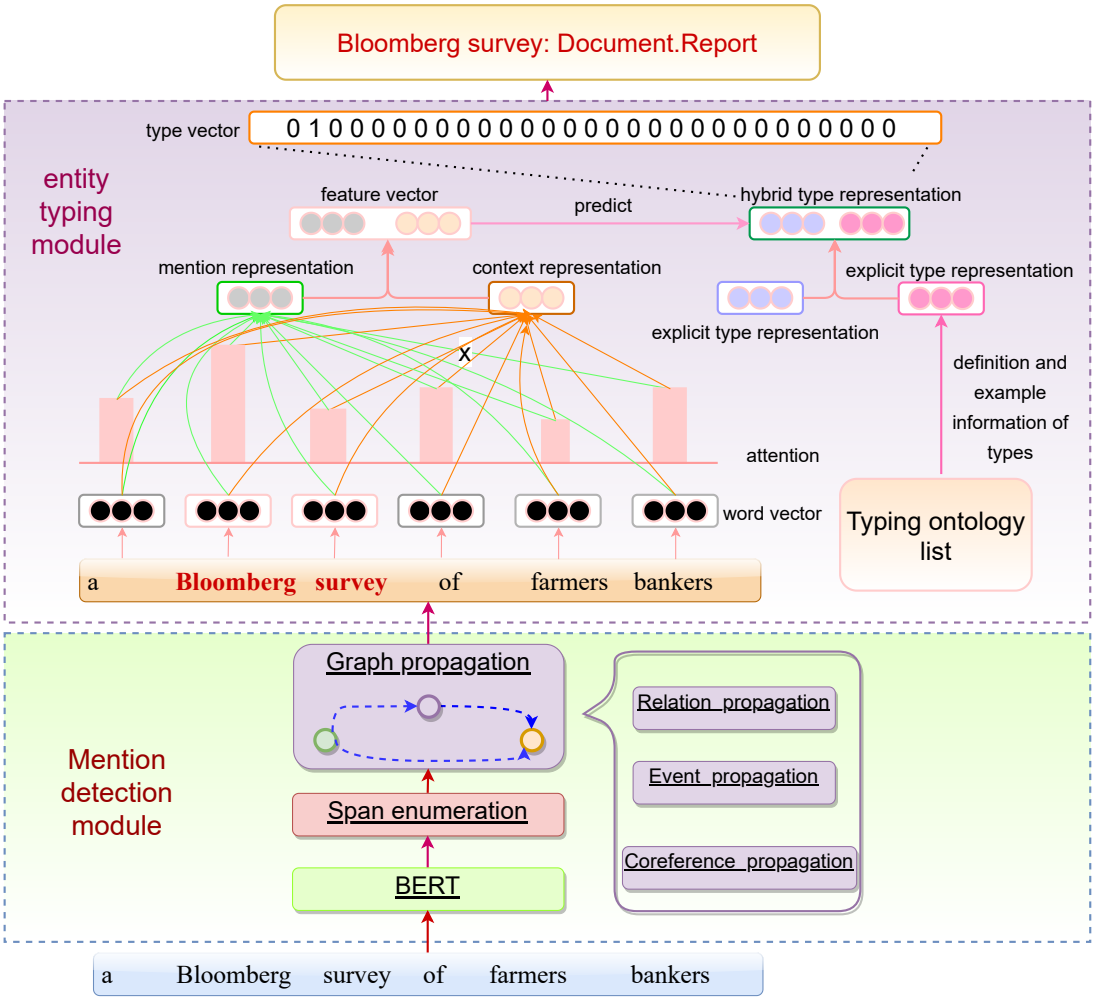

Figure 1: The overall framework of our two-step architecture for KBP 2020.

evant features from the sentences to form the feature vector. the Fet employs a hybrid classifier to predict the types of each mention, which uses contextual information and the mention representation. The sentence encoder utilizes ELMO (Peters et al., 2018), which takes characters as input rather than the words. To focus on more information for the mention representation, the Fet model uses an attention mechanism to sum the weighted contextualized word representation. The context representation involves context word vectors for the mention with a mention-aware mechanism.

To tackle the problem that each type of prediction does not consider their inter-dependency, the Fet model employs a latent type representation to consider the information for the same type.

3.2 Our entity typing model

Since the entity typing task of RUFES 2020 shares no labeled data, we consider the ontology information like the definition and examples for each type in RUFES ontology 2020 together with the latent typing representation in Fet. The detailed architecture is shown in figure 1

4 Experiments

Because this task shares only the test set with no data for training, we first prepare the data for our proposed model as well as the other baselines.

To mapping the other dataset, including Ontonotes, English typing of (Pan et al., 2017) as well as the test data from EDL 2019, to the target ontology of RUFES 2020. We utilize two methods to organize the target dataset for training. If the data has the same Yago ID with the target 266 ontology types, like the data in English typing

of (Pan et al., 2017) as well as the test data from EDL 2019. For others, we use the cosine similarity based on the ELMO representation to calculate the corresponding types in 266 ontology types in RUFES 2020. Finally, we obtain the data with the size of 765,947 to training our proposed entity typing model. However, our prepared data only overlap around 52% of the target ontology with 266 types in 2020 RUFES.

To compare the performance of the baseline, as well as our model, we employ the cosine similarity with some specified parameters, such as the weight for the similarity with the definition is more important than that of the examples listed in the ontology types. We test the three methods for classification on the test set published from RUFES 2020 using the Scorer. The experimental results show that The Fet surpasses the cosine method by 0.50, while our model surpasses the Fet by 0.08. Experiments show that our improvements based on Fet are effective.

5 Conclusion

We decompose RUFES task as mention detection and entity typing. DYGIE++ in the mention detection module guarantees enough coarse-grained candidate mentions to be typed. In the entity typing module, beyond the original FET model, we take advantage of the definition information and examples of each entity in the typing ontology list to enrich type representation. This strategy allviates the problem for lack of data and is proved effective.

References

Ying Lin and Heng Ji. 2019. An attentive fine-grained entity typing model with latent type representation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6198–6203.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada, July. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana, June. Association for Computational Linguistics.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. *arXiv preprint arXiv:1909.03546*.