

OpenUE: An Open Toolkit of Universal Extraction from Text

Ningyu Zhang^{1,2}, Shumin Deng^{1,2}, Zhen Bi^{1,2}, Haiyang Yu^{1,2}, Jiacheng Yang^{1,2},
Mosha Chen³, Fei Huang³, Wei Zhang^{2,3}, Huajun Chen^{1,2}*

¹ Zhejiang University

² AZFT Joint Lab for Knowledge Engine

³ Alibaba Group

{zhangningyu, 231sm, 21921233, yuhaiyang, 21951039, huajunsir}@zju.edu.cn
{chenmosha.cms, f.huang, lantu.zw}@alibaba-inc.com

Abstract

Natural language processing covers a wide variety of tasks with token-level or sentence-level understandings. In this paper, we provide a simple insight that most tasks can be represented in a single universal extraction format. We introduce a prototype model and provide an open-source and extensible toolkit called OpenUE for various extraction tasks. OpenUE allows developers to train custom models to extract information from the text and supports quick model validation for researchers. Besides, OpenUE provides various functional modules to maintain sufficient modularity and extensibility. Except for the toolkit, we also deploy an online demo¹ with restful APIs to support real-time extraction without training and deploying. Additionally, the online system can extract information in various tasks, including relational triple extraction, slot & intent detection, event extraction, and so on. We release the source code, datasets, and pre-trained models to promote future researches in <http://github.com/zjunlp/openue>.

1 Introduction

A large number of natural language processing (NLP) tasks exist to analyze various aspects of human language. Most of them focus on token-level classification (e.g., named entity recognition, slot filling, and argument role classification) or sentence-level understanding (e.g., relation classification, intent detection, and event classification). Previous researchers usually use specifically designed neural network architectures for those tasks. Note that most of those tasks share similar encoder and decoder modules (Jiang et al., 2019). It is beneficial to achieve a unified model for all diverse information extraction tasks without task-specific architectures.

Intuitively, we rethink most of the previous tasks and find that most tasks fall in two categories: **token-oriented tasks**, where the goal is to predict labeled spans (e.g., named entities, slots, aspects) and **sentence-oriented tasks**, where the goal is to predict labels regarding the semantic understanding of sentences (e.g., relations, intents, sentiments). The commonality of these tasks inspires us whether there is a universal framework. Moreover, in the domain of efficient human annotation interfaces, it is already standard to use unified representations for a wide variety of NLP tasks. Taking the BRAT (Stenetorp et al., 2012) annotation as an example, this framework has a single unified format which consists of spans (e.g., the span of an entity), and labeled relations between the spans (e.g., “born-in” and “live-in”).

Motivated by this, we formulate those tasks regarding both token and sentence as *universal extraction* and design a simple unified model. Our prototype model studies the possibility of bridging the gap between tasks with single architecture and providing future insight for unified natural language understanding. Furthermore, as there is a lack of practical and stable toolkit to support the implementation, deployment and evaluation of those tasks, we develop a toolkit which is a complement for existing toolsets such as Spacy² for named entity recognition (NER), TagMe (Ferragina and Scaiella, 2010) for entity linking (EL), OpenKE (Han et al., 2018) for knowledge embedding, Stanford OpenIE (Angeli et al., 2015) for open information extraction, and OpenNER (Han et al., 2019) for relation extraction.

To be specific, we develop an open and extensible toolkit named “OpenUE”. The toolkit prioritizes operational efficiency based on TensorFlow

* Corresponding author: C.Hua(huajunsir@zju.edu.cn)

¹<http://openue.top>

²<https://spacy.io/>

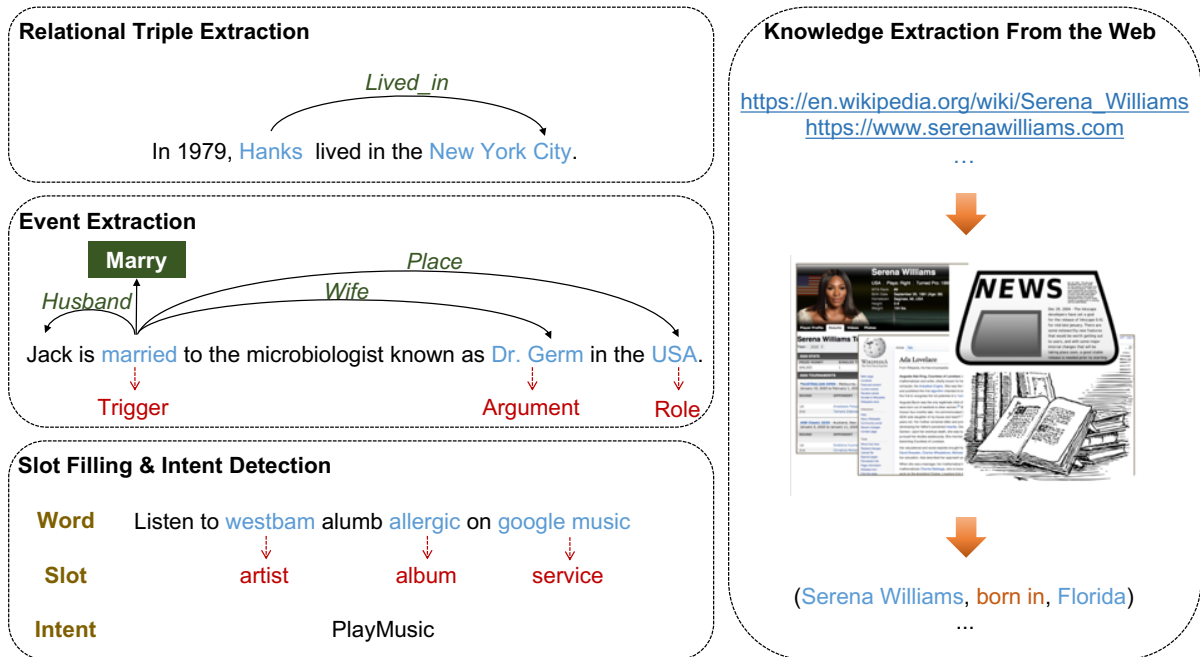


Figure 1: The examples of all application scenarios in OpenUE.

and Pytorch³, which supports quick model training and validation. Besides, OpenUE is able to meet some individual requirements of incorporating new models with system encapsulation and model extensibility. OpenUE provides interfaces for developers aiming at custom models; thus, it is convenient to start up an extraction system based on OpenUE without writing tedious glue code and knowing too many technical details. We provide an online system to extract structured relational facts, slots as well as intents or events from the text with friendly interactive interfaces and fast reaction speed. We will provide maintenance to meet new requests, add new tasks, and fix bugs in the future. This toolkit may benefit both researchers and industry developers. We highlight our contributions as follows:

- We provide a simple prototype implementation of one single model to perform various NLP tasks.
- We provide an open and extensible toolkit to train, evaluate, and serve with multilingual support for universal extraction.
- We open-source our code and release dataset, as well as pre-trained models with open restful APIs for future researchers.

³Pytorch version is under development.

2 Application Scenarios

OpenUE is designed for various tasks, including relational triple extraction, slot filling, intent detection, event extraction, and knowledge extraction from the Web, etc. As shown in Figure 1, we give some examples of these application scenarios.

2.1 Relational Triple Extraction

Relational Triple Extraction is an essential task in Information Extraction (IE) for Natural Language Processing (NLP) and Knowledge Graph (KG) (Zhang et al., 2018b; Yu et al., 2017; Zhang et al., 2019; Huang et al., 2020; Nan et al., 2020; Zhang et al., 2020a; Ye et al., 2020; Zhang et al., 2020b), which is aimed at detecting a pair of entities along with their relations from unstructured text. For instance, there is a sentence “*Paris is known as the romantic capital of France.*”, and in this example, an ideal relational triple extraction system should extract the relational triple $\langle Paris, Capital_of, France \rangle$, in which *Capital_of* is the relation between *Paris* and *France*. In this paper, we provide a simple implementation which firstly classifies relations with the sentence and then conduct sequence labeling to extract entities. The *relation first* approach is beneficial in the real-world setting as most of the sentences contain *NA* relations; therefore, openUE can filter out noisy candidates that do not have relations to improve computation efficacy.

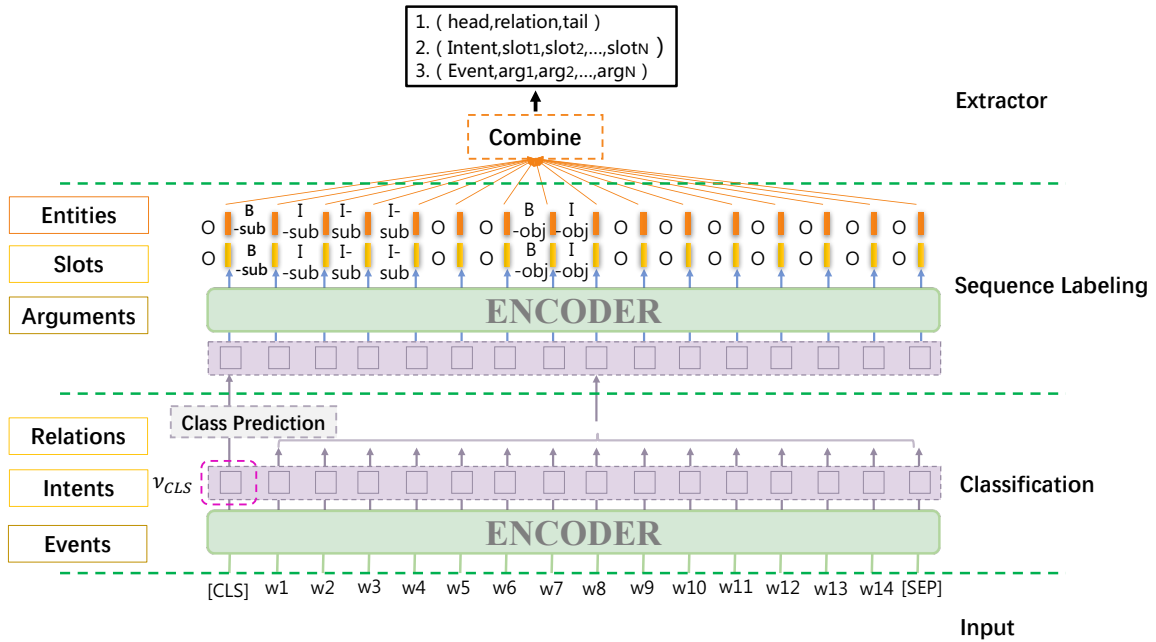


Figure 2: The architecture of OpenUE. Best view in color.

Moreover, we also provide a simple implementation of knowledge extraction from the Web. We implement crawler to obtain raw web pages and apply our approach to extract fact knowledge. Note that the recent knowledge graph is far from complete, while vast numbers of facts exist in web pages. OpenUE is suitable to serve as a schema-based never ended learner from the Web.

2.2 Event Extraction

Extracting events from natural language text is an essential yet challenging task for natural language understanding (Deng et al., 2020). When given a document, event extraction systems need to recognize event triggers with their specific types and their corresponding arguments with the roles. In real-world settings, classifying documents with specific event types and extracting arguments with role types is necessary. We integrate event extraction into OpenUE (without trigger identification).

2.3 Slot Filling and Intent Detection

Natural language understanding (NLU) is critical to the performance of goal-oriented spoken dialogue systems. NLU typically includes the intent detection and slot filling tasks, aiming to form a semantic parse for user utterances. For example, given an utterance from the user, the slot filling annotates the utterance on a word-level, indicating the slot type mentioned by a specific word such as

the slot *artist* mentioned by the word *westbam* as shown in Figure 1. At the same time, the intent detection works on the utterance-level to give intent label(s) to the whole sentence. As slot filling and intent detection rely on both token-level and sentence-level understanding, we integrate this task into OpenUE.

Note that the OpenUE can also be applied to more related tasks such as aspect-based sentiment analysis (Pontiki et al., 2016), semantic role labeling (Carreras and Màrquez, 2005), and so on.

3 Toolkit Design and Implementation

To implement a single prototype model for all tasks, we introduce our OpenUE approach, as Figure 2 shows. We design the prototype implementation with *separated sentence classification and sequence labeling modules* based the following three empirical observations: 1) joint optimization of sequence labeling and sentence classification requires labor-intensive hyper-parameters fine-tuning (Chen et al., 2019); 2) sentence classification first can filter out vast amounts of instances which can reduce computation for sequence labeling; 3) sentence labels with additional information (like a machine reading comprehension style) can provide more signals for sequence labeling (Li et al., 2019).

To design the toolkit, we build a unified underlying platform. OpenUE encapsulates various data

processing and training strategies, which implies that developers can maximize the reuse of code to avoid redundant and unnecessary model implementations. We design OpenUE based on TensorFlow and PyTorch, enabling developers to train models on GPUs for operational efficiency. We introduce the model and design details in the following sections.

3.1 Tokenization

The tokenization module is designed for tokenizing input text into several tokens. In OpenUE, we implement both word-level tokenization and subword-level tokenization. These two kinds of tokenization can satisfy most tokenization demands; thus, developers can avoid spending too much time writing glue code for data processing. Developers can also build customer tokenizer by extending the `BasicTokenizer` class and implementing specific tokenization operations.

3.2 Classification

The classification module is designed for the sentence-level task. We adopt pre-trained language models as default instance encoders in OpenUE. For each sentence $x = \{w_1, w_2, \dots, w_n\}$ in the training set, where $w_i \in x$ is the word token in sentence x , we first construct input sentence in the form: $\{[CLS], w_1, w_2, \dots, w_n, [SEP]\}$. Then we leverage the output of `[CLS]` representation to encode the entire sentence information. We apply an MLP layer with a cross-entropy loss to perform sentence classification. In OpenUE, we have also implemented other common encoders such as XLNet (Yang et al., 2019).

3.3 Sequence Labeling

The sequence labeling module is designed for the token-level task. We utilize the same encoder from the previous section to represent instance. We concatenate the output of the classification module (e.g., relations, event types or intents) with the raw sentence as input for sequence labeling. Specifically, take the relational triple extraction as an example, the input is $\{[CLS], relation, [SEP], w_1, w_2, \dots, w_n\}$. To perform sequence labeling, we provide different kinds of implementations. Traditionally, when the hidden states of the words in the sentence are learned, it is convenient to apply the softmax function to obtain final logits. Moreover, we also provide sequence labeling implementations such as

CRF (Ye et al., 2009) to tag dependencies for each transition pattern between adjacent tags.

3.4 Extractor

To obtain final results, we implement an extractor module to combine the outputs of classification and sequence labeling. For entity and relation extraction, we utilize greedy methods to combine the final results. For other tasks such as slot filling and intent detection, we group those outputs as final predictions.

4 Experiment and Evaluation

In this section, we evaluate our toolkit OpenUE on several datasets in different tasks. The experimental results illustrate that our implementation with OpenUE can achieve comparable or even better performance compared to some state-of-the-art results.

4.1 Relational Triple Extraction

We carry out experiments on four datasets of relational triple extraction: NYT (Riedel et al., 2010), WebNLG (Gardent et al., 2017), SKE and ChMedIE. NYT dataset was originally produced by the distant supervision method. It consists of 1.18M sentences with 24 predefined relation types. WebNLG dataset was originally created for Natural Language Generation (NLG) tasks and adapted by (Zeng et al., 2018) for relational triple extraction task. It contains 246 predefined relation types. Different from the two previous English datasets, SKE is a Chinese dataset for information extraction, which is released in the 2019 Language and Intelligence Challenge⁴. SKE contains 50 relation types, and training texts exceed 200,000. We build our training set, development set, and test set by randomly selecting 50,000, 5,000, and 5,000 texts. ChMedIE is also a Chinese dataset for information extraction in the medical domain. We crawl corpus from the Chinese health website⁵ and build this dataset via distant supervision. It contains 4 relation types.

We compare our OpenUE with four baseliens. **Tagging** (Zheng et al., 2017) is an end-to-end method with a novel tagging scheme. **CopyR** (Zeng et al., 2018) is a Seq2Seq learning framework with a copy mechanism. **HRL** (Takanobu

⁴<http://lic2019.ccf.org.cn/kg>

⁵<http://www.39.net/>

Model	SNIPS-NLU			ATIS		
	Slot (F1)	Intent (Acc)	Overall (Acc)	Slot (F1)	Intent (Acc)	Overall (Acc)
CNN TriCRF	-	-	-	0.944	-	-
Joint Seq	0.873	0.969	0.732	0.942	0.926	0.807
Attention BiRNN	0.878	0.967	0.741	0.942	0.911	0.789
Slot-Gated Full Atten	0.888	0.970	0.755	0.948	0.936	0.822
Capsule-NLU	0.918	0.973	0.809	0.952	0.950	0.834
Joint-BERT	0.986	0.970	0.928	0.975	0.961	0.882
OpenUE	0.985	0.988	0.930	0.953	0.960	0.874

Table 1: Evaluation results of slot filling and intent detection on SNIPS-NLU and ATIS datasets.

Model	NYT	WebNLG	SKE	ChMedIE
Tagging	42.0	28.3	-	-
CopyR	58.7	37.1	-	-
HRL	68.3	66.0	-	-
CasRel	89.6	91.8	78.4	81.0
OpenUE	89.9	89.9	79.3	81.2

Table 2: Evaluation results of entity and relation extraction on NYT, WebNLG, SKE and ChMedIE datasets.

Model	ACE		DuEE	
	Type	Arg	Type	Arg
DMCNN	69.1	53.5	-	-
dbRNN	71.9	58.7	-	-
JMEE	73.7	60.3	80.2	79.5
OpenUE	75.5	60.5	86.2	85.3

Table 3: Evaluation results of event extraction extraction on ACE and DuEE datasets.

et al., 2019) addresses relation extraction by regarding related entities as the arguments of relation via hierarchical reinforcement learning. **CasRel** (Wei et al., 2019) provides a novel cascade binary tagging framework which models relations as functions that map subjects to objects in a sentence. From Table 2 we observe that OpenUE can archive comparable results with CasRel,

4.2 Event Extraction

We carry out experiments on two datasets of event extraction: ACE05⁶, DuEE. The ACE 2005 dataset annotates 33 event subtypes and 36 role classes, along with the NONE class and BIO annotation schema, we classify each sentence into 67 categories in event classification and 37 categories in argument extraction. DuEE is a Chinese dataset for event extraction, which is released in the 2020 Lan-

⁶<https://catalog.ldc.upenn.edu/LDC2006T06>

guage and Intelligence Challenge⁷. DuEE contains 65 event types and 121 argument roles.

We compare our OpenUE with three baseliens. **DMCNN** (Chen et al., 2015) uses dynamic multi-pooling to keep multiple events’ information. **dbRNN** (Sha et al., 2018) adds dependency which bridges over Bi-LSTM for event extraction. **JMEE** (Liu et al., 2018) proposes an approach which jointly extract multiple event triggers and arguments by introducing syntactic shortcut arcs to enhance information flow and attention-based graph convolution networks to model graph information. From Table 3 we observe that OpenUE can archive comparable results with JMEE,

4.3 Slot Filling and Intent Detection

We conduct experiments on two benchmarks NLU datasets: SNIPS Natural Language Understanding benchmark⁸ (SNIPS-NLU) and the Airline Travel Information Systems (ATIS) dataset (Tur et al., 2010). SNIPS-NLU dataset is collected from the Snips personal voice assistant. There are 72 slot labels and 7 intent types in the SNIPS dataset. ATIS dataset is a widely used dataset in NLU research, which includes audio recordings of people making flight reservations. There are 120 slot labels and 21 intent types in the ATIS dataset.

We compare OpenUE with six baselines as follows: **CNN TriCRF** (Xu and Sarikaya, 2013) introduces a Convolution Neural Network (CNN) based sequential labeling model for slot filling. **Joint Seq** (Hakkani-Tür et al., 2016) proposes a Recurrent Neural Network (RNN) for slot filling and utilizes the last hidden state of the RNN to pre-

⁷<https://aistudio.baidu.com/aistudio/competition/detail/32>

⁸<https://github.com/snipsco/nlu-benchmark/>

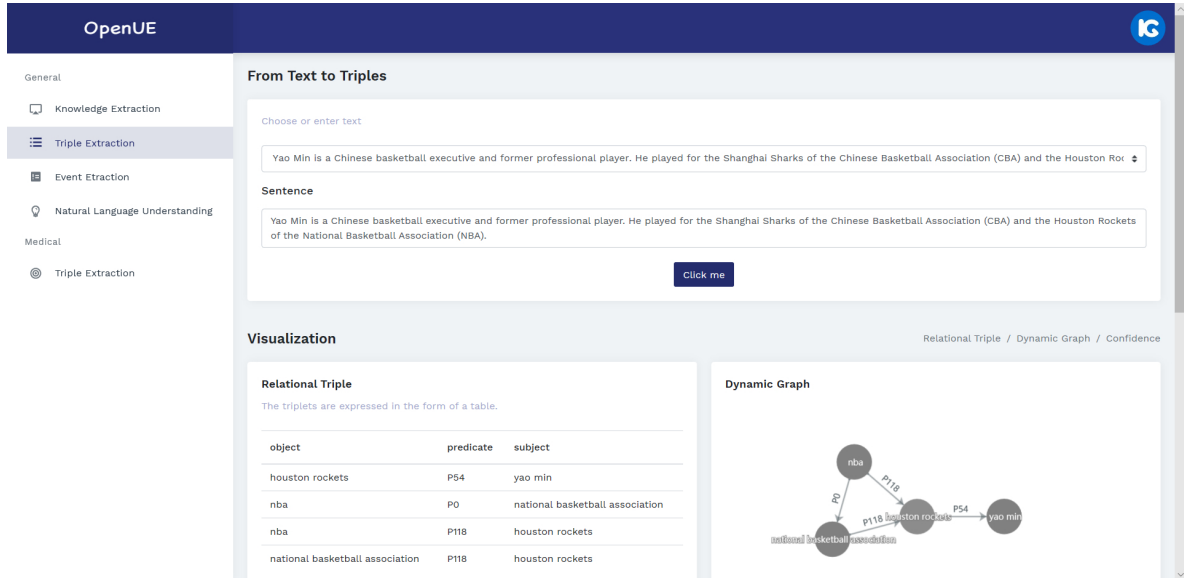


Figure 3: An example of the online system.

dict the utterance intent. **Attention BiRNN** (Liu and Lane, 2016) proposes a RNN based encoder-decoder model for joint intent detection and slot filling. **Slot-gated Full Atten** (Goo et al., 2018) proposes a slot gate that focuses on learning the relationship between intent and slot attention vectors in order to obtain better semantic frame results by the global optimization. **Capsule-NLU** (Zhang et al., 2018a) proposes a capsule-based neural network model that accomplishes slot filling and intent detection via a dynamic routing-by-agreement schema. **Joint-BERT** (Chen et al., 2019) adapts the standard BERT classification, and token classification pipeline to jointly model the slot and intent. From Table 1, we observe that OpenUE can archive comparable performance with Capsule-NLU.

In summary, we conclude that **there exist general architectures for diverse tasks and OpenUE can achieve comparable performance compared with baselines.**

5 Online System

Besides the toolkit, we also release an online system in <http://openue.top>. As shown in Figure 3, we train models in different scenarios with multilingual support (English and Chinese) and deploy the model for online access. The online system can be directly applied to extract structured facts, events, and slots & intents from plain text. We also visualize the graph of relational triples and the probabilities of sentence logits (e.g., relation probabilities) to help to analyze model performance.

Additionally, we deploy a schema-based never ended learner that can extract factual knowledge from the Web. Our system has already obtained millions of facts. More details can be shown in the <https://openue-docs.readthedocs.io/en/latest/>.

Moreover, we provide open restful APIs⁹ for diverse tasks by OpenUE. More tasks, such as aspect-based sentiment analysis, semantic role labeling, and more domains, will be supported in the future.

6 Conclusion

We provide a simple insight that lots of NLP tasks can be represented in a single format. To this end, we provide a prototype model implementation of universal extraction and introduce an open and extensible toolkit, namely, OpenUE. We conduct extensive experiments which demonstrate that the models implemented by OpenUE are efficient, effective, and can achieve comparable performance compared to the state-of-the-art results. Furthermore, we also provide an online system with restful APIs for meeting real-time extraction without training and deploying. In the future, we plan to utilize the multitask learning or meta-learning algorithms to enhance extraction performance. We will provide long-term maintenance to fix bugs and meet new requests.

⁹<https://github.com/zjunlp/openue/blob/master/API.md>

Acknowledgments

We want to express gratitude to the anonymous reviewers for their hard work and kind comments, which will further improve our work in the future. This work is funded by NSFC91846204/SQ2018YFC000004/2018YFB1402800, Alibaba CangJingGe (Knowledge Engine) Research Plan.

References

- Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. 2015. Leveraging linguistic structure for open domain information extraction. 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)*, pages 344–354.
- Xavier Carreras and Lluís Màrquez. 2005. Introduction to the conll-2005 shared task: Semantic role labeling. In *Proceedings of the ninth conference on computational natural language learning (CoNLL-2005)*, pages 152–164.
- Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*.
- Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural 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)*, pages 167–176.
- Shumin Deng, Ningyu Zhang, Jiaojian Kang, Yichi Zhang, Wei Zhang, and Huajun Chen. 2020. Meta-learning with dynamic-memory-based prototypical network for few-shot event detection. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 151–159.
- Paolo Ferragina and Ugo Scaiella. 2010. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1625–1628.
- Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for nlg micro-planning.
- Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757.
- Dilek Hakkani-Tür, Gökhan Tür, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *InterSpeech*, pages 715–719.
- Xu Han, Shulin Cao, Xin Lv, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. 2018. Openke: An open toolkit for knowledge embedding. In *Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations*, pages 139–144.
- Xu Han, Tianyu Gao, Yuan Yao, Demin Ye, Zhiyuan Liu, and Maosong Sun. 2019. Openre: An open and extensible toolkit for neural relation extraction. *arXiv preprint arXiv:1909.13078*.
- Luyang Huang, Lingfei Wu, and Lu Wang. 2020. Knowledge graph-augmented abstractive summarization with semantic-driven cloze reward. *ArXiv*, abs/2005.01159.
- Zhengbao Jiang, Wei Xu, Jun Araki, and Graham Neubig. 2019. Generalizing natural language analysis through span-relation representations. *arXiv preprint arXiv:1911.03822*.
- Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2019. A unified mrc framework for named entity recognition. *arXiv preprint arXiv:1910.11476*.
- Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. *arXiv preprint arXiv:1609.01454*.
- Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018. Jointly multiple events extraction via attention-based graph information aggregation. *arXiv preprint arXiv:1809.09078*.
- Guoshun Nan, Zhijiang Guo, Ivan Sekulić, and Wei Lu. 2020. Reasoning with latent structure refinement for document-level relation extraction. In *In ACL*.
- Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *10th International Workshop on Semantic Evaluation (SemEval 2016)*.
- Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *In KDD*, pages 148–163.
- Lei Sha, Feng Qian, Baobao Chang, and Zhifang Sui. 2018. Jointly extracting event triggers and arguments by dependency-bridge rnn and tensor-based argument interaction. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

- Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. Brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107.
- Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. 2019. A hierarchical framework for relation extraction with reinforcement learning. In *In AAAI*, volume 33, pages 7072–7079.
- Gokhan Tur, Dilek Hakkani-Tür, and Larry Heck. 2010. What is left to be understood in atis? In *2010 IEEE Spoken Language Technology Workshop*, pages 19–24. IEEE.
- Zhepei Wei, Yantao Jia, Yuan Tian, Mohammad Javad Hosseini, Mark Steedman, and Yi Chang. 2019. A novel cascade binary tagging framework for relational triple extraction. *arXiv preprint arXiv:1908.08672*.
- Puyang Xu and Ruhi Sarikaya. 2013. Convolutional neural network based triangular crf for joint intent detection and slot filling. In *2013 IEEE workshop on automatic speech recognition and understanding*, pages 78–83. IEEE.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.
- Hongbin Ye, Ningyu Zhang, Shumin Deng, Mosha Chen, Chuanqi Tan, Fei Huang, and Huajun Chen. 2020. Contrastive triple extraction with generative transformer. *arXiv preprint arXiv:2009.06207*.
- Nan Ye, Wee S Lee, Hai L Chieu, and Dan Wu. 2009. Conditional random fields with high-order features for sequence labeling. In *Advances in neural information processing systems*, pages 2196–2204.
- Mo Yu, Wenpeng Yin, Kazi Saidul Hasan, Cícero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. 2017. Improved neural relation detection for knowledge base question answering. *ArXiv*, abs/1704.06194.
- Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, Jun Zhao, et al. 2018. Extracting relational facts by an end-to-end neural model with copy mechanism.
- Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and Philip S Yu. 2018a. Joint slot filling and intent detection via capsule neural networks. *arXiv preprint arXiv:1812.09471*.
- Ningyu Zhang, Shumin Deng, Zhanlin Sun, Jiaoyan Chen, Wei Zhang, and Huajun Chen. 2020a. Relation adversarial network for low resource knowledge graph completion. In *Proceedings of The Web Conference 2020*, pages 1–12.
- Ningyu Zhang, Shumin Deng, Zhanlin Sun, Guanying Wang, Xi Chen, Wei Zhang, and Huajun Chen. 2019. Long-tail relation extraction via knowledge graph embeddings and graph convolution networks. In *Proceedings of NAACL*.
- Ningyu Zhang, Shumin Deng, Zhanling Sun, Xi Chen, Wei Zhang, and Huajun Chen. 2018b. Attention-based capsule networks with dynamic routing for relation extraction. In *Proceedings of EMNLP*.
- Ningyu Zhang, Luoqiu Li, Shumin Deng, Haiyang Yu, Xu Cheng, Wei Zhang, and Huajun Chen. 2020b. Can fine-tuning pre-trained models lead to perfect nlp? a study of the generalizability of relation extraction. *arXiv preprint arXiv:2009.06206*.
- Suncong Zheng, Yuexing Hao, Dongyuan Lu, Hongyun Bao, Jiaming Xu, Hongwei Hao, and Bo Xu. 2017. Joint entity and relation extraction based on a hybrid neural network. *Neurocomputing*, 257:59–66.