DeepKE: A Deep Learning Based Knowledge Extraction Toolkit for Knowledge Base Population

Ningyu Zhang[1], **Xin Xu**[1], **Liankuan Tao**[1], **Haiyang Yu**[1], **Hongbin Ye**[1], **Shuofei Qiao**[1], **Xin Xie**[1], **Xiang Chen**[1], **Zhoubo Li**[1], **Lei Li**[1], **Xiaozhuan Liang**[1], **Yunzhi Yao**[1], **Shumin Deng**[1], **Peng Wang**[1], **Wen Zhang**[1], **Zhenru Zhang**[2], **Chuanqi Tan**[2], **Qiang Chen**[2], **Feiyu Xiong**[2], **Fei Huang**[2], **Guozhou Zheng**[1], **Huajun Chen**[1] *

[1] Zhejiang University & AZFT Joint Lab for Knowledge Engine

[2] Alibaba Group

http://deepke.zjukg.cn/

Abstract

We present an open-source and extensible knowledge extraction toolkit DeepKE, supporting complicated low-resource, document-level and multimodal scenarios in knowledge base population. DeepKE implements various information extraction tasks, including named entity recognition, relation extraction and attribute extraction. With a unified framework, DeepKE allows developers and researchers to customize datasets and models to extract information from unstructured data according to their requirements. Specifically, DeepKE not only provides various functional modules and model implementation for different tasks and scenarios but also organizes all components by consistent frameworks to maintain sufficient modularity and extensibility. We release the source code at GitHub[1] with Google Colab tutorials and comprehensive documents[2] for beginners. Besides, we present an online system[3] for real-time extraction of various tasks, and a demo video[4].

1 Introduction

As Information Extraction (IE) techniques develop fast, many large-scale Knowledge Bases (KBs) have been constructed. Those KBs can provide back-end support for knowledge-intensive tasks in real-world applications, such as language understanding (Che et al., 2021), commonsense reasoning (Lin et al., 2019) and recommendation systems (Wang et al., 2018). However, most KBs are far from complete due to the emerging entities and relations in real-world applications. Therefore, Knowledge Base Population (KBP) (Ji and Grishman, 2011) has been proposed, which aims to extract knowledge from the text corpus to complete the missing elements in KBs. For this target, IE is an effective technology that can extract entities and relations from raw texts and link them to KBs (Yan et al., 2021; Sui et al., 2021).

To date, a few remarkable open-source and long-term maintained IE toolkits have been developed, such as Spacy (Vasiliev, 2020) for named entity recognition (NER), OpenNRE (Han et al., 2019) for relation extraction (RE), Stanford OpenIE (Martínez-Rodríguez et al., 2018) for open information extraction, RESIN for event extraction (Wen et al., 2021) and so on (Jin et al., 2021). However, there are still several non-trivial issues that hinder the applicability of real-world applications.

Firstly, there are various important IE tasks, but most existing toolkits only support one task. Secondly, although IE models trained with those tools can achieve promising results, their performance may degrade dramatically when there are only a few training instances or in other complex real-world scenarios, such as encountering document-level and multimodal instances. Therefore, it is necessary to build a knowledge extraction toolkit facilitating the knowledge base population that supports multiple tasks and complicated scenarios: **low-resource**, **document-level** and **multimodal**.

In this paper, we share with the community a new open-source knowledge extraction toolkit called **DeepKE** (MIT License), which supports knowledge extraction tasks (named entity recognition, relation extraction and attribute extraction) in the standard supervised setting and three complicated scenarios: low-resource, document-level and multimodal settings. To facilitate usage, we design a unified framework for data processing, model training and evaluation. Developers and researchers can quickly customize their datasets and models for

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

[1]Github: https://github.com/zjunlp/DeepKE

[2]Docs: https://zjunlp.github.io/DeepKE/

[3]Project website: http://deepke.zjukg.cn/

[4]Video: http://deepke.zjukg.cn/demo.mp4

Proceedings of the The 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 98 - 108

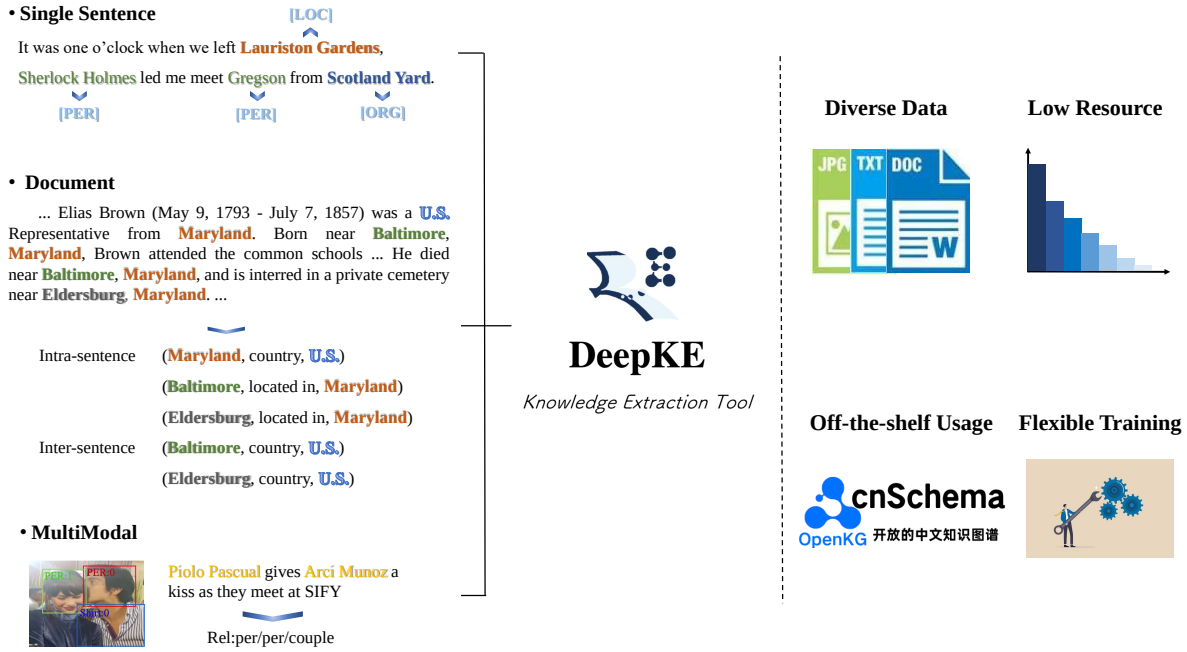

Figure 1: The examples of tasks with different scenarios in DeepKE.

various tasks without knowing too many technical details, writing tedious glue code, or conducting hyper-parameter tuning. We will provide maintenance to meet new requests, add new tasks, and fix bugs in the future. We highlight our major contributions as follows:

- We develop and release a knowledge base population toolkit that supports low-resource, document-level and multimodal information extraction.
- We offer flexible usage of the toolkit with sufficient modularity as well as automatic hyper-parameter tuning; thus, developers and researchers can implement customized models for information extraction.
- We provide detailed documentation, Google Colab tutorials, an online real-time extraction system and long-term technical support.

2 Core Functions

DeepKE is designed for different knowledge extraction tasks, including named entity recognition, relation extraction and attribute extraction. As shown in Figure 1, DeepKE supports diverse IE tasks in standard single-sentence supervised, low-resource few-shot, document-level and multimodal settings, which makes it flexible to adapt to practical and complicated application scenarios.

2.1 Named Entity Recognition

As an essential task of IE, named entity recognition (NER) picks out the entity mentions and classifies them into pre-defined semantic categories given plain texts. For instance, given the sentence "It was one o'clock when we left Lauriston Gardens, and Sherlock Holmes led me meet Gregson from Scotland Yard.", NER models will predict that "Lauriston Gardens" as a location, "Sherlock Holmes" and "Gregson" as persons, and "Scotland Yard" as an organization. To achieve supervised NER, DeepKE adopts the pre-trained language model (Devlin et al., 2019) to encode sentences and make predictions. DeepKE also implements NER in the few-shot setting (including in-domain and cross-domain) (Chen et al., 2022a) and the multimodal setting.

2.2 Relation Extraction

Relation Extraction (RE), a common task in IE for knowledge base population, predicts semantic relations between pairs of entities from unstructured texts (Wu et al., 2021). To allow users to customize their models, we adopt various models to accomplish standard supervised RE, including CNN (Zeng et al., 2015), RNN (Zhou et al., 2016), Capsule (Zhang et al., 2018a), GCN (Zhang et al., 2018c, 2019), Transformer (Vaswani et al., 2017) and BERT (Devlin et al., 2019). Meanwhile, DeepKE provides few-shot and document-level

support for RE. For low-resource RE, DeepKE re-implements[5] *KnowPrompt* (Chen et al., 2022b), a recent well-performed few-shot RE method based on prompt-tuning. Note that few-shot RE is significant for real-world applications, which enables users to extract relations with only a few labeled instances. For document-level RE, DeepKE re-implements *DocuNet* (Zhang et al., 2021) to extract inter-sentence relational triples within one document. Document-level RE is a challenging task that requires integrating information within and across multiple sentences of a document (Nan et al., 2020). RE is also implemented in the multimodal setting described in Section 4.4.

2.3 Attribute Extraction

Attribute extraction (AE) plays an indispensable role in the knowledge base population. Given a sentence, entities and queried attribute mentions, AE will infer the corresponding attribute type. For instance, given a sentence "诸葛亮，字孔明，三国时期杰出的军事家、文学家、发明家。" (Liang Zhuge, whose courtesy name was Kongming, was an extraordinary strategist, litterateur and inventor in the Three Kingdoms period.), an entity "诸葛亮" (Liang Zhuge), and an attribute mention "三国时期" (Three Kingdoms period), DeepKE can predict the corresponding attribute type "朝代" (Dynasty). DeepKE adopts various models for AE (Table 1).

3 Toolkit Design and Implementation

We introduce the design principle of DeepKE as follows: 1) **Unified Framework**: DeepKE utilizes the same framework for various task objectives with respect to *Data*, *Model* and *Core* components; 2) **Flexible Usage**: DeepKE offers convenient training and evaluation with auto-hyperparameter tuning and the docker for operational efficiency; 3) **Off-the-shelf Models**: DeepKE provides pre-trained models (Chinese models with pre-defined schemas) for information extraction. We will introduce details of components in DeepKE and the unified framework in the following sections.

3.1 Data Module

The data module is designed for preprocessing and loading input data. The tokenizer in DeepKE implements tokenization for both English and Chinese (in Appendix A.3). Global images and local visual objects are preprocessed as visual information in the multimodal setting. Developers can feed their own datasets into the tokenizer and preprocessor through the dataloader to obtain the tokens or image patches.

[5]The code is re-organized in a unified format for flexible usage in DeepKE.

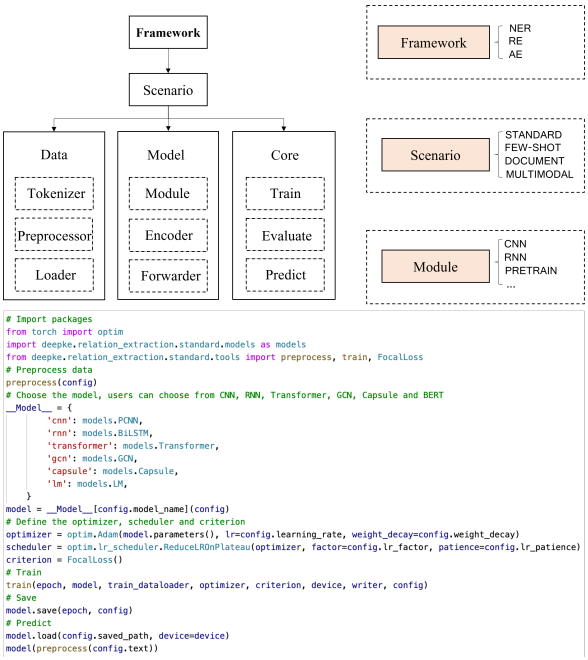

Figure 2: The architecture and example code.

3.2 Model Module

The model module contains main neural networks leveraged to achieve three core tasks. Various neural networks, including CNN, RNN, Transformer and the like, can be utilized for model implementation, which encodes texts into specific embedding for corresponding tasks. To adapt to different scenarios, DeepKE utilizes diverse architectures in distinct settings, such as *BERT* for standard RE and *BART* (Lewis et al., 2020) for few-shot NER. We implement the `BasicModel` class with a unified `model loader` and `saver` to integrate multifarious neural models.

3.3 Core Module

In the core code of DeepKE, `train`, `validate`, and `predict` methods are pivotal components. As for the `train` method, users can feed the expected parameters (e.g., the model, data, epoch, optimizer, loss function, .etc.) into it without writing tedious glue code. The `validate` method is for evaluation. Users can modify the sentences in the configuration for prediction and then utilize the `predict` method to obtain the result.

3.4 Framework Module

The framework module integrates three aforementioned components and different scenarios. It supports various functions, including data processing, model construction and model implementation. Meanwhile, developers and researchers can customize all hyper-parameters by modifying configuration files formatted as *"*.yaml"*, from which we apply *Hydra*[6] to obtain users' configuration. We also offer an off-the-shelf automatic hyperparameter tuning component. In DeepKE, we have implemented frameworks for all application functions mentioned in Section 2. For other future potential application functions, we have reserved interfaces for their implementation.

4 Toolkit Usage

4.1 Single-sentence Supervised Setting

All tasks, including NER, RE and AE, can be implemented in the standard single-sentence supervised setting by DeepKE. Every instance in datasets only contains one sentence. The datasets of these tasks are all annotated with specific information, such as entity mentions, entity categories, entity offsets, relation types and attributes.

4.2 Low-resource Setting

In real-world scenarios, labeled data may not be sufficient for deep learning models to make predictions for satisfying users' specific demands. Therefore, DeepKE provides low-resource few-shot support for NER and RE, which is exceedingly distinctive. DeepKE offers a generative framework with prompt-guided attention to achieve in-domain and cross-domain NER. Meanwhile, DeepKE implements knowledge-informed prompt-tuning with synergistic optimization for few-shot relation extraction.

4.3 Document-Level Setting

Relations between two entities not only emerge in one sentence but appear in different sentences within the whole document. Compared to other IE toolkits, DeepKE can extract inter-sentence relations from documents, which predicts an entity-level relation matrix to capture local and global information.

[6] https://hydra.cc/

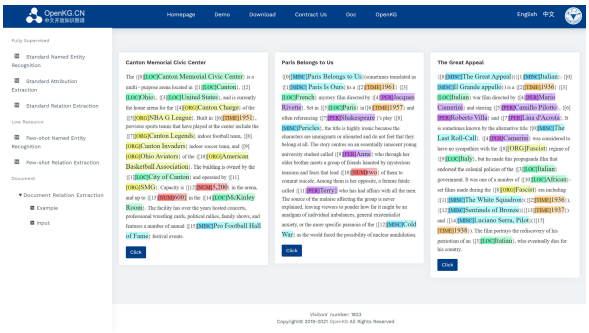

Figure 3: An example of the online system.

4.4 Multimodal Setting

Multimodal knowledge extraction is supported in DeepKE. Intuitively, rich image signals related to texts are able to enhance context knowledge and help extract knowledge from complicated scenarios. DeepKE provides a Transformer-based multimodal entity and relation extraction method named *IFAformer* with prefix-based attention for multimodal NER and RE. Specifically, *IFAformer* simultaneously concatenates the textual and visual features in keys and values of the multi-head attention at each transformer layer, which can implicitly align multimodal features between texts and objects in text-related images[7].

4.5 Online System & *cnSchema*-based Off-the-shelf Models

Besides this toolkit, we release an online system in http://deepke.zjukg.cn. As shown in Figure 3, we train our models in different scenarios with multilingual support (English and Chinese) and deploy them for online access. The system can be directly applied to recognize named entities, extract relations, classify attributes from plain texts, and visualizes extracted relational triples as knowledge graphs. The models are trained **with the pre-defined schema** (The system cannot extract knowledge out of the schema scope.) and offer flexible usage for users to obtain their customized models with their own schemas. Furthermore, DeepKE provides off-the-shelf extraction models with Chinese pre-trained language models (Cui et al., 2021b) based *cnSchema*[8] supporting 28 entity types and 50 relation categories.

[7] Implementation details in https://github.com/zjunlp/DeepKE/tree/main/example/ner/multimodal.

[8] http://cnschema.openkg.cn/

Scenario	Task	Dataset	Method	F1
Single-sentence	NER	CoNLL-2003	BERT	94.73
		People's Daily		95.62
	RE	DuIE	CNN	96.74
			RNN	94.43
			Capsule	96.23
			GCN	96.74
			Transformer	96.54
			BERT	95.79
	AE	Online	CNN	94.16
			RNN	93.06
			Capsule	94.57
			GCN	94.50
			Transformer	94.15
			BERT	99.03
Document	RE	DocRED	BERT_base*	53.20
			CorefBERT$^{*}_{base}$	56.96
			ATLOP-BERT$^{*}_{base}$	61.30
			DeepKE (BERT$_{base}$)	**61.86**
			RoBERTa_large*	59.62
			CorefRoBERTa$^{*}_{large}$	60.25
			ATLOP-RoBERTa$^{*}_{large}$	63.40
			DeepKE (RoBERTa$_{large}$)	**64.55**
Multimodal	NER	Twitter17	AdapCoAtt-BERT-CRF*	84.10
			ViLBERT$^{*}_{base}$	85.04
			UMT*	85.31
			DeepKE (IFAformer)	**87.39**
	RE	MNRE	BERT+SG*	62.80
			BERT+SG+Att*	63.64
			MEGA*	66.41
			DeepKE (IFAformer)	**81.67**

Table 1: F1 Score (%) of the single-sentence, document-level and multimodal scenarios. * means these baselines are from other papers.

5 Experiment and Evaluation

5.1 Single-sentence Supervised Setting

The performance of the standard single-sentence supervised setting is reported in Table 1.

Named Entity Recognition We conduct NER experiments on two datasets: CoNLL-2003 (Sang and Meulder, 2003) for English and People's Daily[9] for Chinese. The English part of CoNLL-2003 contains four types of entities: persons (PER), locations (LOC), organizations (ORG) and miscellaneous (MISC). People's Daily dataset is a Chinese dataset containing 45,518 entities classified into three categories PER, LOC and ORG. It is observed that DeepKE yields comparable performance with various encoders for these datasets. Meanwhile, DeepKE supports any English and Chinese NER datasets with BIO tags.

Relation Extraction We conduct RE experiments on the Chinese DuIE dataset[10] with 10 relation categorie Each sample contains one original sentence, one head entity, one tail entity in the sentence, their offsets, and the relation between them. We utilize six different neural networks in DeepKE for evaluation. Users can select models before training by changing only one hyper-parameter[11]. We report the performance of all models in Table 1.

Model	Entity Category				
	PER	ORG	LOC*	MISC*	Overall
LC-BERT	76.25	75.32	61.55	59.35	68.12
LC-BART	75.70	73.59	58.70	57.30	66.82
Template.	84.49	72.61	71.98	73.37	75.59
DeepKE (LightNER)	**90.96**	**76.88**	**81.57**	**82.08**	**78.97**

Table 2: F1 scores of in-domain low-resource NER on CoNLL-2003. * indicates low-resource entity types (100-shot).

Model	Dataset		
	MIT Movie	MIT Restaurant	ATIS
Neigh.Tag.	1.4	3.6	3.4
Example.	29.6	26.1	16.5
MP-NSP	36.8	48.2	74.8
LC-BERT	45.2	40.9	78.5
LC-BART	30.4	11.1	74.4
Template.	54.2	60.3	88.9
DeepKE (LightNER)	**75.6**	**67.4**	**89.4**

Table 3: F1 scores of cross-domain few-shot NER (20-shot).

Attribute Extraction The Chinese dataset for AE is from an online resource[12]. In each sample, one entity is annotated with its attribute type, value, and offset. Attributes in the dataset are classified into 6 categories. The training set contains 13,815 samples. The validation set contains 3,131 samples, and the test set includes 5,921 samples. Like RE, we leverage six neural models to extract attributes from the given sentence to evaluate DeepKE.

5.2 Low-resource Setting

We report the performance of the low-resource setting (NER and RE) in Table 2, 3, and 4.

Named Entity Recognition We conduct experiments in both in-domain and cross-domain few-shot settings with LightNER (Chen et al., 2022a). Following Cui et al. (2021a), for the in-domain few-shot scenario, we reduce the number of training samples for certain entity categories by down-sampling one dataset. Specifically, from CoNLL-

[9] https://github.com/OYE93/Chinese-NLP-Corpus/tree/master/NER/People's%20Daily

[10] http://ai.baidu.com/broad/download

[11] The hyper-parameter *-model* to select networks is in https://github.com/zjunlp/DeepKE/blob/main/example/re/standard/conf/config.yaml.

[12] https://github.com/leefsir/triplet_extraction

Method	Split		
	K=8	K=16	K=32
Fine-Tuning	41.3	65.2	80.1
GDPNet	42.0	67.5	81.2
PTR	70.5	81.3	84.2
DeepKE (KnowPrompt)	**74.3**	**82.9**	**84.8**

Table 4: F1 scores of few-shot relation extraction

2003, we choose **100** "LOC" and **100** "MISC" as the low-resource entities and 2,496 "PER" and 3,763 "ORG" as the rich-resource entities. We leverage DeepKE to carry out the few-shot experiments and adopt BERT and BART with label-specific classifier layers as strong baselines denoted as *LC-BERT* and *LC-BART*. We also use template-based BART *(Template.)* (Cui et al., 2021a) as the competitive few-shot baseline. From Table 2, DeepKE outperforms other methods for both rich- and low-resource entity types, which illustrates that DeepKE has an outstanding performance on in-domain few-shot NER. In the cross-domain setting where the target entity categories and textual style are different from the source domain with limited labeled data available for training, we adopt the CoNLL-2003 dataset as an ordinary domain, and MIT Movie Review (Liu et al., 2013), MIT Restaurant Review (Liu et al., 2013) and Airline Travel Information Systems (ATIS) (Hakkani-Tür et al., 2016) datasets as target domains. The few-shot NER model in DeepKE is trained on CoNLL-2003 and fine-tuned on 20-shot target domain datasets (randomly sampled per entity category). We employ prototype-based *Neigh.Tag.* (Wiseman and Stratos, 2019), *Example.* (example-based NER) (Ziyadi et al., 2020), *MP-NSP* (Multi-prototype+NSP) (Huang et al., 2020), *LC-BERT*, *LC-BART* and *Template.* as competitive baselines. From Table 3, we notice that DeepKE achieves the most excellent few-shot performance.

Relation Extraction For few-shot relation extraction, we use SemEval 2010 Task-8 (Hendrickx et al., 2010), a conventional dataset of relation classification with nine bidirectional relations and one unidirectional relation OTHER. We utilize a SOTA few-shot RE method, *KnowPrompt* (Chen et al., 2022b) which incorporates knowledge into prompt-tuning with synergistic optimization, to conduct 8-, 16-, and 32-shot experiments compared with other baselines, such as *GDPNet* (Xue et al., 2021) and *PTR* (Han et al., 2021). Table 4 shows that DeepKE outperforms those baseline methods.

5.3 Document-level Setting

DeepKE can extract intra- and inter- sentence relations among multiple entities within one document. We leverage a large-scale document-level RE dataset, DocRED (Ye et al., 2020), containing 3,053/1,000/1,000 instances for training, validation and testing, respectively. We use cased BERT-base and RoBERTa-large (Liu et al., 2019) as encoders. Compared with BERT-based and RoBERTa-based models, including *Coref* (Ye et al., 2020), and *ATLOP* (Zhou et al., 2021), DeepKE appears the better or comparable performance than baselines as shown in Table 1.

5.4 Multimodal Setting

We report the performance of NER and RE in the multimodal scenario in Table 1.

Named Entity Recognition Multimodal NER experiments are conducted on Twitter-2017 (Lu et al., 2018) including texts and images from Twitter (2016-2017). The baselines for comparison are *AdapCoAtt-BERT-CRF* (Zhang et al., 2018b), ViLBERT (Lu et al., 2019) and UMT (Yu et al., 2020). We notice DeepKE can obtain a performance improvement compared with baselines.

Relation Extraction We use MNRE (Zheng et al., 2021b), a multimodal RE dataset containing sentences and images containing 23 relation categories. Previous SOTA models including *BERT+SG* (Zheng et al., 2021a), *BERT+SG+Att* (*BERT+SG* with attention calculating semantic similarity between textual and visual graphs) and *MEGA* (Zheng et al., 2021a), are leveraged for comparison. We further observe that DeepKE yields better performance than baselines.

6 Conclusion

In practical application, the knowledge base population struggles with low-resource, document-level and multimodal scenarios. To this end, we propose DeepKE, an open-source and extensible knowledge extraction toolkit. We conduct extensive experiments that demonstrate the models implemented by DeepKE can achieve comparable performance compared to some state-of-the-art methods. Besides, we provide an online system supporting real-time extraction (**with the pre-defined schemas**) without training. We will offer long-term maintenance to fix bugs, solve issues, add documents (tutorials) and meet new requests.

Broader Impact Statement

As noted in Manning (2022), linguistics and knowledge-based artificial intelligence were rapidly developing, and knowledge (explicit or implicit) as potential dark matter[13] for language understanding still faces obstacles to acquisition and representation. To this end, IE technologies that aim to extract knowledge from unstructured data can serve as valuable tools to not only govern domain resources (e.g., medical, business) but also benefit deep language understanding and reasoning ability. Note that the proposed toolkit, DeepKE, can offer flexible usage in widespread IE scenarios with pre-trained off-the-shelf models. We hope to deliver the benefits of the proposed DeepKE to the natural language processing community.

Acknowledgments

We want to express gratitude to the anonymous reviewers for their kind comments. This work was supported by National Natural Science Foundation of China (No.62206246, 91846204 and U19B2027), Zhejiang Provincial Natural Science Foundation of China (No. LGG22F030011), Ningbo Natural Science Foundation (2021J190), and Yongjiang Talent Introduction Programme (2021A-156-G).

References

Wanxiang Che, Yunlong Feng, Libo Qin, and Ting Liu. 2021. N-LTP: An open-source neural language technology platform for Chinese. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 42–49, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiang Chen, Lei Li, Shumin Deng, Chuanqi Tan, Changliang Xu, Fei Huang, Luo Si, Huajun Chen, and Ningyu Zhang. 2022a. LightNER: A lightweight tuning paradigm for low-resource NER via pluggable prompting. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2374–2387, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022b. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pages 2778–2788. ACM.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021a. Template-based named entity recognition using BART. In *Proceedings of ACL/IJCNLP*, volume ACL/IJCNLP 2021 of *Findings of ACL*.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. 2021b. Pre-training with whole word masking for chinese BERT. *IEEE ACM Trans. Audio Speech Lang. Process.*, 29:3504–3514.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186. Association for Computational Linguistics.

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 *Proceedings of INTERSPEECH*, pages 715–719. ISCA.

Xu Han, Tianyu Gao, Yuan Yao, Deming Ye, Zhiyuan Liu, and Maosong Sun. 2019. Opennre: An open and extensible toolkit for neural relation extraction. In *Proceedings of EMNLP-IJCNLP*.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. PTR: prompt tuning with rules for text classification. *arXiv:2105.11259*.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of SemEval@ACL Workshop*.

Jiaxin Huang, Chunyuan Li, Krishan Subudhi, Damien Jose, Shobana Balakrishnan, Weizhu Chen, Baolin Peng, Jianfeng Gao, and Jiawei Han. 2020. Few-shot named entity recognition: A comprehensive study. *arXiv:2012.14978*.

Heng Ji and Ralph Grishman. 2011. Knowledge base population: Successful approaches and challenges. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 1148–1158. The Association for Computer Linguistics.

Zhuoran Jin, Yubo Chen, Dianbo Sui, Chenhao Wang, Zhipeng Xue, and Jun Zhao. 2021. CogIE: An information extraction toolkit for bridging texts and CogNet. In *Proceedings of ACL-IJCNLP*, pages 92–98.

[13] 2082: An ACL Odyssey: The Dark Matter of Language and Intelligence

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of ACL*, pages 7871–7880. Association for Computational Linguistics.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. Kagnet: Knowledge-aware graph networks for commonsense reasoning. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 2829–2839. Association for Computational Linguistics.

Jingjing Liu, Panupong Pasupat, Scott Cyphers, and James R. Glass. 2013. Asgard: A portable architecture for multilingual dialogue systems. In *Proceedings of ICASSP*, pages 8386–8390. IEEE.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *arXiv:1907.11692*.

Di Lu, Leonardo Neves, Vitor Carvalho, Ning Zhang, and Heng Ji. 2018. Visual attention model for name tagging in multimodal social media. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1990–1999, Melbourne, Australia. Association for Computational Linguistics.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Christopher D Manning. 2022. Human language understanding & reasoning. *Daedalus*, 151(2):127–138.

José-Lázaro Martínez-Rodríguez, Ivan López-Arévalo, and Ana B. Ríos-Alvarado. 2018. Openie-based approach for knowledge graph construction from text. *Expert Syst. Appl.*, 113:339–355.

Guoshun Nan, Zhijiang Guo, Ivan Sekulic, and Wei Lu. 2020. Reasoning with latent structure refinement for document-level relation extraction. In *Proceedings of ACL*, pages 1546–1557. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of HLT-NAACL*, pages 142–147. ACL.

Dianbo Sui, Chenhao Wang, Yubo Chen, Kang Liu, Jun Zhao, and Wei Bi. 2021. Set generation networks for end-to-end knowledge base population. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 9650–9660. Association for Computational Linguistics.

Yuli Vasiliev. 2020. *Natural Language Processing with Python and SpaCy: A Practical Introduction*. No Starch Press.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: deep knowledge-aware network for news recommendation. In *Proceedings of WWW*, pages 1835–1844. ACM.

Haoyang Wen, Ying Lin, Tuan Lai, Xiaoman Pan, Sha Li, Xudong Lin, Ben Zhou, Manling Li, Haoyu Wang, Hongming Zhang, Xiaodong Yu, Alexander Dong, Zhenhailong Wang, Yi Fung, Piyush Mishra, Qing Lyu, Dídac Surís, Brian Chen, Susan Windisch Brown, Martha Palmer, Chris Callison-Burch, Carl Vondrick, Jiawei Han, Dan Roth, Shih-Fu Chang, and Heng Ji. 2021. RESIN: A dockerized schema-guided cross-document cross-lingual cross-media information extraction and event tracking system. In *In Proceedings of NAACL-HLT*.

Sam Wiseman and Karl Stratos. 2019. Label-agnostic sequence labeling by copying nearest neighbors. In *Proceedings of ACL*, pages 5363–5369. Association for Computational Linguistics.

Tongtong Wu, Xuekai Li, Yuan-Fang Li, Gholamreza Haffari, Guilin Qi, Yujin Zhu, and Guoqiang Xu. 2021. Curriculum-meta learning for order-robust continual relation extraction. In *Proceedings of AAAI*.

Fuzhao Xue, Aixin Sun, Hao Zhang, and Eng Siong Chng. 2021. Gdpnet: Refining latent multi-view graph for relation extraction. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14194–14202. AAAI Press.

Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021. A unified generative framework for various NER subtasks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5808–5822. Association for Computational Linguistics.

Deming Ye, Yankai Lin, Jiaju Du, Zhenghao Liu, Peng Li, Maosong Sun, and Zhiyuan Liu. 2020. Coreferential reasoning learning for language representation. In *Proceedings of EMNLP*, pages 7170–7186. Association for Computational Linguistics.

Jianfei Yu, Jing Jiang, Li Yang, and Rui Xia. 2020. Improving multimodal named entity recognition via entity span detection with unified multimodal transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3342–3352, Online. Association for Computational Linguistics.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of EMNLP*, pages 1753–1762. The Association for Computational Linguistics.

Ningyu Zhang, Xiang Chen, Xin Xie, Shumin Deng, Chuanqi Tan, Mosha Chen, Fei Huang, Luo Si, and Huajun Chen. 2021. Document-level relation extraction as semantic segmentation. In *Proceedings of IJCAI*, pages 3999–4006. ijcai.org.

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-HLT*, pages 3016–3025. Association for Computational Linguistics.

Ningyu Zhang, Shumin Deng, Zhanling Sun, Xi Chen, Wei Zhang, and Huajun Chen. 2018a. Attention-based capsule network with dynamic routing for relation extraction. In *In Proceedings of EMNLP*.

Qi Zhang, Jinlan Fu, Xiaoyu Liu, and Xuanjing Huang. 2018b. Adaptive co-attention network for named entity recognition in tweets. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018c. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of EMNLP*, pages 2205–2215. Association for Computational Linguistics.

Changmeng Zheng, Junhao Feng, Ze Fu, Yi Cai, Qing Li, and Tao Wang. 2021a. *Multimodal Relation Extraction with Efficient Graph Alignment*, page 5298–5306. Association for Computing Machinery, New York, NY, USA.

Changmeng Zheng, Zhiwei Wu, Junhao Feng, Ze Fu, and Yi Cai. 2021b. Mnre: A challenge multimodal dataset for neural relation extraction with visual evidence in social media posts. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of ACL*, pages 207–212, Berlin, Germany. Association for Computational Linguistics.

Wenxuan Zhou, Kevin Huang, Tengyu Ma, and Jing Huang. 2021. Document-level relation extraction with adaptive thresholding and localized context pooling. In *In proceedings of AAAI*, pages 14612–14620. AAAI Press.

Morteza Ziyadi, Yuting Sun, Abhishek Goswami, Jade Huang, and Weizhu Chen. 2020. Example-based named entity recognition. *arXiv:2008.10570*.

Word	Named Entity Tag
U.N.	B-ORG
official	O
Ekeus	B-PER
heads	O
for	O
Baghdad	B-LOC
.	O
Israel	B-LOC
approves	O
Arafat	B-PER
s	O
flight	O
to	O
West	B-LOC
Bank	I-LOC
.	O

Table 5: Examples of the input format for NER.

A Toolkit Usage Details

In this section, we introduce how to use DeepKE exhaustively.

A.1 Build a Model From Scratch

Prepare the Runtime Environment Users can clone the source code from the DeepKE GitHub repository and create a runtime environment. There are two convenient methods to create the environment. Users can choose to either leverage *Anaconda* or run the docker file provided in the repository. Besides, all dependencies can be installed by running `pip install deepke` directly. If developers would like to modify the source code of DeepKE, the following commands should be executed: running `python setup.py install`, modifying code and then running `python setup.py develop`. Users can also use corresponding datasets (e.g., default or customized datasets) to obtain specific information extraction models. All datasets need to be downloaded or uploaded in the folder named *data*.

Named Entity Recognition As shown in Table 5, the input data files with BIO tags for standard and few-shot NER contain two columns separated by a single space. Each word has been put on a separate line, and there is an empty line after each sentence. The two columns represent two items: the word and the named entity tag. Before training, all datasets with the formats mentioned above should be fed into NER models through the data loader. Developers can implement training and evaluation by running example code *run.py* to obtain a fine-tuned NER model, which will be used in the prediction period. For inference, users can run *predict.py* with a single sentence and obtain the output recognized entity mentions and types.

Relation Extraction The training input with the CSV format of standard RE is shown in Table 6. There are five components in the format, including a sentence, a relation, the head and tail entity of the relation, the head entity offset and the tail entity offset. For few-shot RE, one input sample, as shown in Figure 4, contains sentence tokens including words and punctuation, the head entity and tail entities with their mention names and position spans, and the relation between them. For example, an input of few-shot relation extraction instance is the format of *{"token": ["the", "dolphin", "uses", "its", "flukes", "for", "swimming", "and", "its", "flippers", "for", "steering", "."], "h": {"name": "dolphin", "pos": [1, 2]}, "t": {"name": "flukes", "pos": [4, 5]}, "relation": "Component-Whole(e2,e1)"}* (h: head entity, t: tail entity, pos: position). The document-level RE training format is shown in Figure 5. One sample consists of a sample title, sentences separated into words and punctuation in one document, an entity set (including entity mentions, sentence IDs the entities are located in, entity position spans and entity types in the document) and a relation label set (including the head and tail entity IDs, relations and evidence sentence IDs). After training and validation, users can run the `predict` function given an input sentence with head and tail entity to obtain corresponding relations.

Attribution Extraction The input CSV files formatted as Table 7 should be given to train the attribution extraction (AE) model. One sample contains six components: a raw sentence, a queried attribute type, an entity and its offset, the entity's corresponding attribute value and the attribute mention offset. After training, users will obtain a fine-tuned AE model, which can be leveraged to infer attributes. Given a sentence with an entity and a candidate attribute mention, the AE model will predict the attribute type with confidence. Note that all operations mentioned above are guided in the example code file *run.py* and *predict.py*.

Sentence	Relation	Head	HO	Tail	TO
When it comes to beautiful sceneries in Hangzhou, West Lake first emerges in mind.	city: located in	West Lake	50	Hangzhou	40
Harry Potter, a wizard, graduated from Hogwarts School of Witchcraft and Wizardry.	school: graduated from	Harry Potter	0	Hogwarts School of Witchcraft and Wizardry	39

Table 6: Examples of the input format for standard RE. HO: Head Offset, TO: Tail Offset.

```
Data Format:
{
    'token': [tokens in a sentence],
    "h": {
        "name": mention_name,
        "pos" : [postion of mention in a sentence]
    },
    "t": {
        "name": mention_name,
        "pos" : [postion of mention in a sentence]
    },
    "relation": relation
}
```

Figure 4: The input format of few-shot RE.

A.2 Auto-Hyperparameter Tuning

To achieve automatic hyper-parameters fine-tuning, DeepKE adopts *Weight & Biases*, a machine learning toolkit for developers to reduce label-intensive hyper-parameter tuning. With DeepKE, users can visualize results and tune hyper-parameters automatically. Note that all metrics and hyper-parameter configurations can be customized to meet diverse settings for different tasks. For more details on automatic hyper-parameter tuning.

Sentence	Attribute	Entity	EO	AV	AVO
1903年，亨利·福特创建福特汽车公司	创始人	福特	9	亨利·福特	6
吴会期，字行可，号子官，明朝工部郎中	朝代	吴会期	0	明朝	12

Table 7: Examples of the input format AE. EO: Entity Offset, AV: Attribute Value, AVO: Attribute Value Offset.

```
Data Format:
{
  'title',
  'sents':     [
                  [word in sent 0],
                  [word in sent 1]
                ]
  'vertexSet': [
                  [
                    { 'name': mention_name,
                      'sent_id': mention in which sentence,
                      'pos': postion of mention in a sentence,
                      'type': NER_type}
                    {anthor mention}
                  ],
                  [anthoer entity]
                ]
  'labels':   [
                  {
                    'h': idx of head entity in vertexSet,
                    't': idx of tail entity in vertexSet,
                    'r': relation,
                    'evidence': evidence sentences' id
                  }
                ]
}
```

Figure 5: The input format of document-level RE.

Task	Scenario	Language
NER	Supervised	Chinese
	Few-shot	English, Chinese
	Multimodal	English
RE	Supervised	Chinese
	Few-shot	English
	Multimodal	English
	Document	English
AE	Supervised	Chinese

Table 8: Language supported in DeepKE.

please refer to the official document[14].

A.3 Language Support

The current version of DeepKE supports English and Chinese implementation for three IE tasks, as shown in Table 8.

A.4 Notebook Tutorials

We provide Google Colab tutorials[15] and jupyter notebooks in the GitHub repository as an exemplary implementation of every task in different scenarios. These tutorials can be run directly, thus, leading developers and researchers to have a whole picture of DeepKE's powerful functions.

[14] https://docs.wandb.ai

[15] https://colab.research.google.com/drive/1vS8YJhJltzw3hpJczPt24O0Azcs3ZpRi?usp=sharing

B Contributions

Ningyu Zhang from Zhejiang University, AZFT Joint Lab for Knowledge Engine, conducted the whole development of DeepKE and wrote the paper.

Xin Xu from Zhejiang University, AZFT Joint Lab for Knowledge Engine developed the standard NER and wrote the paper.

Liankuan Tao, Shuofei Qiao, Peng Wang, Haiyang Yu from Zhejiang University, AZFT Joint Lab for Knowledge Engine develop the standard RE and AE, the deepke python package, and documents and provides consistent maintenance.

Hongbin Ye from Zhejiang University, AZFT Joint Lab for Knowledge Engine developed the online system and constructed the online demo.

Xin Xie, Xiang Chen from Zhejiang University, AZFT Joint Lab for Knowledge Engine developed the few-shot relation extraction model KnowPrompt and the document-level relation extraction model DocuNet.

Zhoubo Li, Lei Li, Xiaozhuan Liang, Yunzhi Yao, Shumin Deng, Wen Zhang from Zhejiang University, AZFT Joint Lab for Knowledge Engine developed the Google Colab and proofread the paper.

Zhenru Zhang, Chuanqi Tan, Qiang Chen, Feiyu Xiong, Fei Huang from Alibaba Group, proofread the paper and advised the project.

Guozhou Zheng, Huajun Chen from Zhejiang University, AZFT Joint Lab for Knowledge Engine advised the project, suggested tasks, and led the research.