HOSMEL: A Hot-Swappable Modularized Entity Linking Toolkit for Chinese

Daniel Zhang-Li[1], Jing Zhang[2]*, Jifan Yu[1], Xiaokang Zhang[2], Peng Zhang[1,3], Jie Tang[1], Juanzi Li[1]

[1]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[2]School of Information, Renmin University of China Beijing, China, [3]ZHIPU.AI
{zlnn21,yujf21}@mails.tsinghua.edu.cn,
{zhang-jing,zhang2718}@ruc.edu.cn,
peng.zhang@aminer.cn, {jietang,lijuanzi}@tsinghua.edu.cn

Abstract

We investigate the usage of entity linking (EL) in downstream tasks and present the first modularized EL toolkit for easy task adaptation. Different from the existing EL methods that deal with all the features simultaneously, we modularize the whole model into separate parts with each feature. This decoupled design enables flexibly adding new features without retraining the whole model as well as flow visualization with better interpretability of the EL result. We release the corresponding toolkit, HOSMEL, for Chinese, with three flexible usage modes[1], a live demo[2], and a demonstration video[3]. Experiments on two benchmarks for the question answering task demonstrate that HOSMEL achieves much less time and space consumption as well as significantly better accuracy performance compared with existing SOTA EL methods. We hope the release of HOSMEL will call for more attention to study EL for downstream tasks in non-English languages.

1 Introduction

Entity linking (EL) is to extract the candidate mentions in the sentences and link them to their corresponding entities in the knowledge bases (KB) such as Freebase, Wikidata, and DBPedia. The linked entities encode rich knowledge from the KB, which can enhance many downstream tasks such as information retrieval (Raviv et al., 2016), recommendation (Guo et al., 2020), question answering (Feng et al., 2021; Zhang et al., 2021a), and language model pre-training (Zhang et al., 2019). As EL is usually deployed in the pre-processing stages of these tasks, an urgent demand for EL models is to guarantee a high accuracy to prevent potential error propagation.

Existing researches have fully explored the EL problem. From the matching-based methods (Chen et al., 2020, 2022; Logeswaran et al., 2019; Yamada et al., 2019; Wu et al., 2020; Zhang et al., 2021c) to the generation-based methods (Nicola De et al., 2020; De Cao et al., 2021), the SOTA models such as BLINK (Wu et al., 2020), GENRE (Nicola De et al., 2020), and EntQA (Zhang et al., 2021c), have considered different features such as mention, entity subtitle, and entity description, resulting in outstanding performances on various published EL benchmarks.

However, the existing advanced models usually aggregate all the features for training. Despite their excellent performance, they are difficult to be adapted to specific downstream tasks. Figure 2 illustrates an EL example for answering the question "*What religion does Luke's master believe in?*", by which multiple mentions are detected and linked to the entities in XLore[4] (Jin et al., 2019a) — one of the largest Chinese KBs. Depending only on the subtitle description of an entity, the mention, "*Luke*", can be highly probably linked to both the entity "*Luke Skywalker*" and "*Luke Cage*"[5]. Whereas in this scenario, we can accurately find that the former better matches because it has a relation "*master*" in XLore which is exactly the questioned aspect. This case presents a common phenomenon that downstream tasks usually require additional information invocation such as relation for better EL results.

Generally, developers need to annotate specific data and perform after treatments for EL model adaption. (For the above example, we need to annotate a new dataset such as the one including relation as the additional feature for question answering and retrain the EL model on the new dataset). However, in the era of advanced large EL models, such data

*Corresponding author.

[1] https://github.com/THUDM/HOSMEL

[2] https://www.aminer.cn/el/#/

[3] https://drive.google.com/drive/folders/1eh-dJnKWJulPuZGsORii4fPW-zCmWS5k?usp=sharing

[4] https://xlore.org/

[5] *Luke Skywalker* is a character in Star Wars and *Luke Cage* is a Marvel superhero.

annotation and model retraining is quite costly and inefficient, which raises a natural question: *Can we develop an effective EL tool that can be easily adapted to downstream tasks?*

Presented work. We propose a **HO**t-**S**wappable **M**odularized **E**ntity **L**inking toolkit (HOSMEL) to solve the above problem. Compared with existing EL methods or toolkits, HOSMEL is more suitable for the downstream tasks because of its following characteristics:

- **Low coupled modules.** We modularize mention filtering, mention detection, and entity disambiguation by each entity attribute, ensuring each module can be trained separately and combined freely.
- **Incremental development.** The decoupled design turns the module of each step into a hot-swappable module, which enables flexibly adding the new features that were not previously considered without retraining the whole model.
- **Flexible to use (three usage modes).** We develop a corresponding toolkit for Chinese EL as Chinese has gained less attention than English. For flexible usage, we release three usage methods. The first one is a ready-to-use release for directly invoking the API or accessing the web application. The second one is a partial release for users who prefer to include parts of the release as a pre-step to improve the recall of their model. The third one is an easy-to-change release that enables adding additional features or training with self-defined data.
- **Flow visualization.** The decoupled design also enables a more explainable way for visualizing the results of each module, which provides user engineers a more effortless experience in deciding the useful features for optimizing the best outcome.

We select question answering as the downstream task to evaluate the proposed HOSMEL. We conduct extensive experiments on two question answering benchmarks. The results reveal three major advantages: (1) the training time of the lightweight HOSMEL is reduced by 4-5 times compared with two SOTA EL models, GENRE (Nicola De et al., 2020) and EntQA (Zhang et al., 2021c), in advance, the storage occupancy rate is also reduced by 78% compared with EntQA. (2) HOSMEL can achieve much better performance (+8.49-17.06% accuracy) than the best baseline EntQA on less training data. (3) We additionally evaluate the hot-swappable ability of HOSMEL and find that when adding a new feature relation, HOSMEL can be quickly updated and further improves 3.71-5.02% of accuracy.

Contributions. (1) We investigate the usage of EL in downstream tasks and raise the problem of adaptation for EL in downstream tasks. (2) We design a hot-swappable modularized EL system and release the corresponding toolkit in Chinese[1] and a live demo[2].

2 Problem Definition

A **knowledge base** (KB) $\mathcal{E}$ contains n entities denoted by $\mathcal{E} = \{e_i\}_{i=1}^{n}$. Each entity e_i is associated with a set of attributes denoted by $A_i = \{A_i^t\}_{t=1}^{T}$ where A_i^t is the attributes of type t and T is the total number of attribute types. A_i^t is further denoted by $\{a_{ij}^t\}_{j=1}^{n_i^t}$ with a_{ij}^t as the j-th attribute of type t and n_i^t as the total number of attributes with type t. For example, an entity usually contains a title, a subtitle, a description, and multiple relations.

Problem 1. ***Entity Linking (EL):*** *Given an input text $d = \{w_1, \cdots, w_n\}$ and a KB $\mathcal{E}$, the output of EL is a list of mention-entity pairs $\{(m_i, e_i)\}_{i=1}^{K}$, where each mention m_i is a text span extracted from d, and each entity e_i is included in $\mathcal{E}$.*

We assume each mention has a valid gold entity in the KB and leave the out-of-KB prediction (i.e., nil prediction) to future works.

3 The Proposed HOSMEL

HOSMEL modularizes mention filtering, mention detection, and entity disambiguation by each entity attribute separately. Generally, given an input text, HOSMEL first selects all the possible mentions and then detects the useful ones, whose candidate entities are then measured by each attribute of them independently. Apart from mention filtering, each subsequent step aggregates the scores of all the previous steps and outputs a new top-K result to its next. Figure 1 illustrates the overall framework of the proposed HOSMEL, each step explained below.

3.1 Mention Filtering

Mention filtering is to filter out the possible mentions that can be linked to certain entities in KB. For example, in Figure 2, the input contains mentions "卢克(*Luke*)", "信仰(*religion*)", "师父(*master*)",

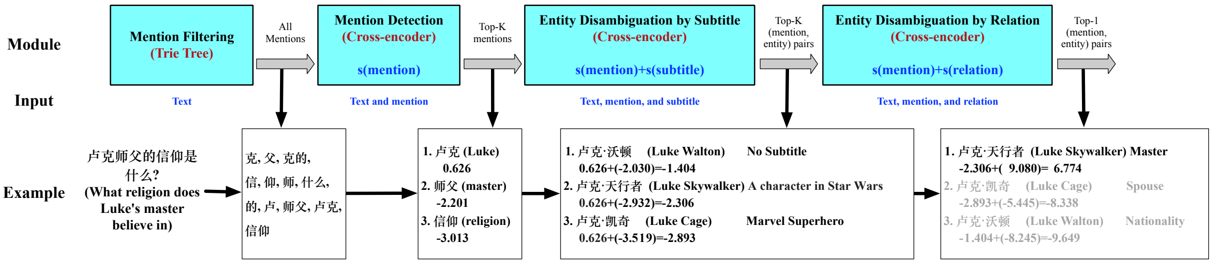

Module
Input
Example
Mention Filtering
(Trie Tree)
Text
All Mentions
Mention Detection
(Cross-encoder)
s(mention)
Text and mention
Top-K mentions
Entity Disambiguation by Subtitle
(Cross-encoder)
s(mention)+s(subtitle)
Text, mention, and subtitle
Top-K (mention, entity) pairs
Entity Disambiguation by Relation
(Cross-encoder)
s(mention)+s(relation)
Text, mention, and relation
Top-1 (mention, entity) pairs
卢克师父的信仰是什么?
(What religion does Luke's master believe in)
克, 父, 克的, 信, 仰, 师, 什么, 的, 卢, 师父, 卢克, 信仰
1. 卢克 (Luke) 0.626
2. 师父 (master) -2.201
3. 信仰 (religion) -3.013
1. 卢克·沃顿 (Luke Walton) No Subtitle
0.626+(-2.030)=-1.404
2. 卢克·天行者 (Luke Skywalker) A character in Star Wars
0.626+(-2.932)=-2.306
3. 卢克·凯奇 (Luke Cage) Marvel Superhero
0.626+(-3.519)=-2.893
1. 卢克·天行者 (Luke Skywalker) Master
-2.306+(9.080)= 6.774
2. 卢克·凯奇 (Luke Cage) Spouse
-2.893+(-5.445)=-8.338
3. 卢克·沃顿 (Luke Walton) Nationality
-1.404+(-8.245)=-9.649

Figure 1: Illustration of the overall framework, where mention filtering, mention detection, entity disambiguation by subtitle, and disambiguation by relation are modularized. Each module aggregates the scores of all the previous modules and outputs a new top-K result to the next step.

etc. For this purpose, we build a Trie tree (Wilkes, 1974) with all the possible mentions, collected using titles and available alias names (Cf. Section A.2 for collecting details.) of all the entities in the KB. Previous works usually omit the steps of mention filtering and mention detection under the assumption that the mentions in an input text are known (Logeswaran et al., 2019; Yamada et al., 2019; Wu et al., 2020). Without this assumption, we can simply resort to the above Trie tree to find the mentions, because in our two EL benchmarks for question answering, by this kind of mention filtering, we can obtain an exceptionally high recall of the ground truth entities (Cf. Section A.2 in Appendix for details). For the datasets with mentions not exactly the same with the titles or alias names of the entities, users can change this Trie tree to other more suitable methods such as bi-encoder (Zhang et al., 2021c).

3.2 Mention Detection

Mention detection determines the top-K important mentions from all the possible mentions returned by the previous step. For example, in Figure 2, "卢克(*Luke*)" is more crucial to answer the question than the other mentions. For this purpose, we concatenate the input text d and a mention m_i into "d; $[SEP]; m_i$" as the input of a cross encoder, which is instantiated as MacBERT(Cui et al., 2020) in this and the subsequent steps. Then we apply a MLP layer on the CLS embedding of MacBERT to obtain the probability of m_i given d. We take the logarithm of the probability as the mention's score $s(m_i) = \log(P(m_i|d))$ and output the top-K ranked mentions by $s(m_i)$ to the next step.

3.3 Entity Disambiguation

Entity disambiguation is to seek the correct entities from the KB for the detected mentions. Thanks to the Trie tree, we can quickly obtain the entity candidates stored as (title/alias, entity identifier) pairs with their title or aliases. For disambiguating an entity candidate, we match the input text and the mention with each type of attribute independently in the same way. Specifically, given an attribute type t, we concatenate the input text d, the mention m_i, and an attribute a^t_{ij} of entity e_i into "d; $[SEP]; m_i; a^t_{ij}$" as the input of MacBERT. We also apply a MLP layer on the CLS embedding to obtain the probability of attribute a^t_{ij} given d and m_i. We take the logarithm of the probability as a^t_{ij}'s score and get the maximal score from all the attributes A^t_i as the pooling score of A^t_i, *i.e.*, $s(A^t_i) = \max_j \log(P(a^t_{ij}|d, m_i))$. When a type only has one attribute, such as a single subtitle, the maximal score is the score of the single attribute.

Then we rank the entities by the score of each (m_i, e_i) pair, which is computed by the logarithm of the joint probability of m_i and all the processed attributes of e_i given the input text d, *i.e.*,

$$
\begin{aligned}
s(m_i, e_i) &= \log P(A^1_i, A^2_i, \cdots, A^t_i, m_i|d), \\
&= s(m_i) + \sum_{\tau=1}^{t} s(A^\tau_i), \qquad (1)
\end{aligned}
$$

where $s(m_i)$ and $s(A^\tau_i)$ are the scores of the mention m_i and attributes A^τ_i respectively. The second equation is obtained according to the assumption of the independence of the mention and different attributes. The derivation details can be referred to Eq.(2) in Appendix. We return top-K ranked (mention,entity) pairs by $s(m_i, e_i)$ to the next step.

3.4 Training Strategy

The parameters of MacBERT are learned via optimizing the cross-entropy between the predicted scores and the ground truth mentions or the entity attributes. We train a separate MacBERT for computing each score, including the score of the mention and the score of each attribute type respectively. The training data is organized following the setting of multiple-choice question answering. For example, a data instance for training the mention detection model needs to include the input text and four candidate mentions, with one labeled as the ground truth. While for training the entity disambiguation model by an attribute such as the subtitle, it needs to include the input text, the mention to be linked, and four candidate subtitles with the ground truth label. Thanks to this separate training, we can adjust each module without influencing other modules. A new feature can be easily added as we only need to annotate a small amount of the training data about the new feature rather than re-annotate a new one with both the old and the new features.

4 The Usage of HOSMEL

We consider three different usage scenarios of the proposed HOSMEL and release the corresponding toolkit usage scripts with a live demo.

4.1 Ready-to-Use Release

The ready-to-use release is for users who need to link the input text to the general Chinese open domain KB. For this purpose, we train HOSMEL on XLore with the mention and entities' title, subtitle, and relations as features and release the model checkpoints. Users can download all the checkpoints and use them by the following scripts:

```
text = "卢克的师父信仰什么"
url = "http://localhost:9899/
    readyToUse/"
data = rq.urlopen(url+urllib.parse.
    quote(text)).read()
data = json.loads(data.decode("UTF-8"
    ))
# {"data": ["卢克", "bdi9202050",
# 6.774, "卢克的师父"]}
```

Live Demo. For this ready-to-use release, we also provide a live demonstration to observe each step's outputs in our pipeline, including mention filtering, mention detection, entity disambiguation by subtitle, and disambiguation by relation. In addition, it also comes with clickable links to XLore for closer observation of the entity. This could be useful for users who prefer a visualized front-end webpage for interpretability.

4.2 Partial Release

The partial release is for users interested in completing the EL process inside their downstream models or using parts of our release for entity candidate retrieval from XLore instead of the whole release. In this scenario, we expose each pipeline step for users to determine where to stop according to their needs. For example, if users only want to use mention filtering, mention detection, and disambiguation by subtitle, they can use the following scripts:

```
text = "卢克的师父信仰什么"
filtered_m = filter_mention(text)
# ["卢", "卢克", "什么", etc.]
detected_m = detect_mention(text,
    filtered_m,K=3)
# ["卢克", "师父", "信仰"]
entities = disambiguate_by_subtitle(
    text,detected_m,K=3)
# [["卢克", "bdi9203099", -1.404],
# ["卢克", "bdi9202050", -2.306],
# ["卢克", "bdi9201727", -2.893]]
```

Since loading the Trie tree into memory is time-consuming, which would bring a poor experience when debugging, we encapsulate the Trie tree into a web service using flask.

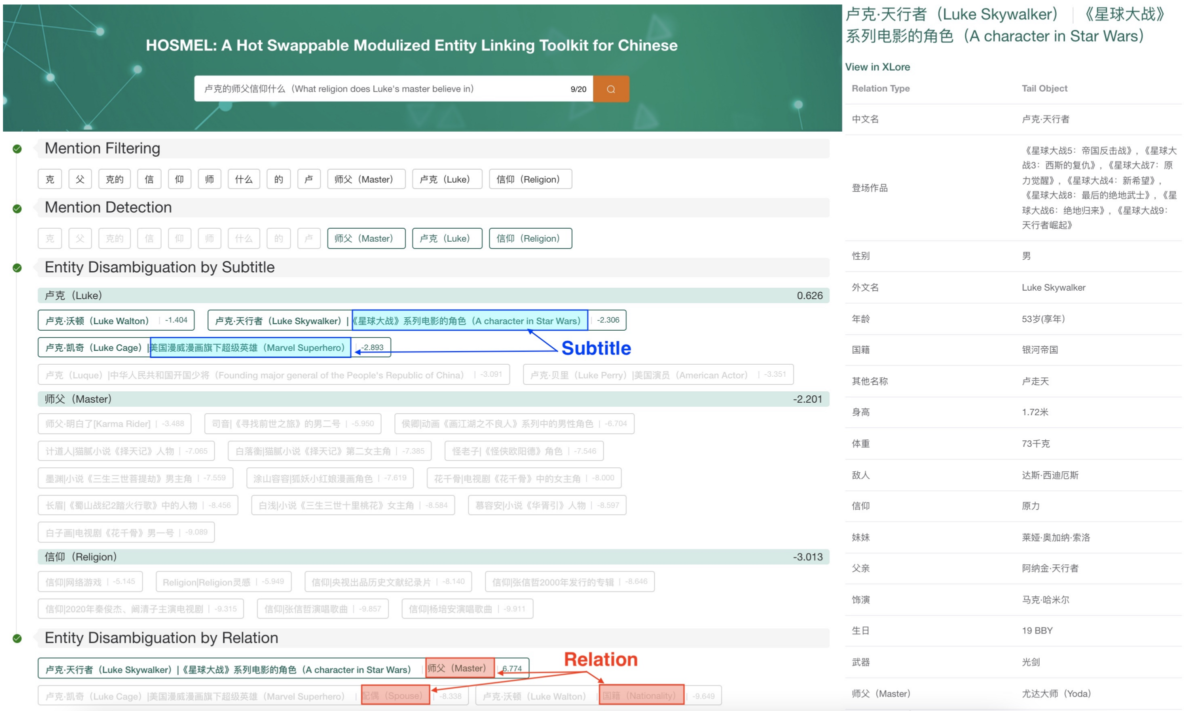
4.3 Easy-to-Change Release

As we illustrated in Figure 2, using specific features such as the relations of entities can potentially benefit the EL for downstream tasks. We provide a training script and a sample model usage implementation for users who have such a demand. In order to add a new feature, HOSMEL requires the users to: (1) format their training data into our format and (2) make a copy of the sample relation usage, and re-write the $generatePair$ method in it to retrieve the required feature. If the users prefer to change XLore into other KBs, they only need to rebuild the Trie tree. More usage details can be found in the released code and readme documents[6].

5 Experiment

In this section, we use two question answering benchmarks to evaluate the EL capacity of the proposed HOSMEL and also show its ability to easily add task-relevant features that can benefit the EL performance.

[6] https://github.com/THUDM/HOSMEL



HOSMEL: A Hot Swappable Modulized Entity Linking Toolkit for Chinese

卢克的师父信仰什么（What religion does Luke's master believe in） 9/20

Mention Filtering

克 | 父 | 克的 | 信 | 仰 | 师 | 什么 | 的 | 卢 | 师父（Master） | 卢克（Luke） | 信仰（Religion）

Mention Detection

克 | 父 | 克的 | 信 | 仰 | 师 | 什么 | 的 | 卢 | 师父（Master） | 卢克（Luke） | 信仰（Religion）

Entity Disambiguation by Subtitle

卢克（Luke） 0.626

卢克·沃顿（Luke Walton） | -1.404

卢克·天行者（Luke Skywalker）|《星球大战》系列电影的角色（A character in Star Wars） | -2.306

卢克·凯奇（Luke Cage）|美国漫威漫画旗下超级英雄（Marvel Superhero） | -2.893

卢克（Luque）|中华人民共和国开国少将（Founding major general of the People's Republic of China） | -3.091

卢克·贝里（Luke Perry）|美国演员（American Actor） | -3.351

师父（Master） -2.201

师父·明白了[Karma Rider] | -3.488

司音|《寻找前世之旅》的男二号 | -5.950

侯卿|动画《画江湖之不良人》系列中的男性角色 | -6.704

计道人|猫腻小说《择天记》人物 | -7.065

白落衡|猫腻小说《择天记》第二女主角 | -7.385

怪老子|《怪侠欧阳德》角色 | -7.546

墨渊|小说《三生三世菩提劫》男主角 | -7.559

涂山容容|狐妖小红娘漫画角色 | -7.619

花千骨|电视剧《花千骨》中的女主角 | -8.000

长眉|《蜀山战纪2踏火行歌》中的人物 | -8.456

白浅|小说《三生三世十里桃花》女主角 | -8.584

慕容安|小说《华胥引》人物 | -8.597

白子画|电视剧《花千骨》男一号 | -9.089

信仰（Religion） -3.013

信仰|网络游戏 | -5.145

Religion|Religion灵感 | -5.949

信仰|央视出品历史文献纪录片 | -8.140

信仰|张信哲2000年发行的专辑 | -8.646

信仰|2020年秦俊杰、阚清子主演电视剧 | -9.315

信仰|张信哲演唱歌曲 | -9.857

信仰|杨培安演唱歌曲 | -9.911

Entity Disambiguation by Relation

卢克·天行者（Luke Skywalker）|《星球大战》系列电影的角色（A character in Star Wars） | 师父（Master） | 6.774

卢克·凯奇（Luke Cage）|美国漫威漫画旗下超级英雄（Marvel Superhero） | 配偶（Spouse） | -8.338

卢克·沃顿（Luke Walton） | 国籍（Nationality） | -9.649

卢克·天行者（Luke Skywalker） | 《星球大战》系列电影的角色（A character in Star Wars）

View in XLore

Relation Type	Tail Object
中文名	卢克·天行者
登场作品	《星球大战5：帝国反击战》，《星球大战3：西斯的复仇》，《星球大战7：原力觉醒》，《星球大战4：新希望》，《星球大战8：最后的绝地武士》，《星球大战6：绝地归来》，《星球大战9：天行者崛起》
性别	男
外文名	Luke Skywalker
年龄	53岁(享年)
国籍	银河帝国
其他名称	卢走天
身高	1.72米
体重	73千克
敌人	达斯·西迪厄斯
信仰	原力
妹妹	莱娅·奥加纳·索洛
父亲	阿纳金·天行者
饰演	马克·哈米尔
生日	19 BBY
武器	光剑
师父（Master）	尤达大师（Yoda）

Figure 2: Illustration of the live demonstration. The input example “卢克的师父信仰什么？(*What religion does Luke’s master believe in?*)” contains the mention “卢克(*Luke*)” that can be linked to the entity “卢克·天行者(*Luke Skywalker*)” with a relation “师父(*master*)”.

5.1 Experimental Settings

Datasets.

KB. Since the 26,146,618 entities in XLore contain many uncommon entities, we create a subset of 16,095,248 entities from it for better usage and evaluation. To ensure precision, we consider various ways, such as ranking by edit times or removing certain entity types, during filtering.

Test set. We choose KgCLUE(Xu et al., 2020) and a hand-crafted dataset labeled by us as the test sets. Both of them are for one-hop question answering from the general Chinese open domain KB, each containing $1,673$ and $1,597$ questions labeled with the topic entities. We transfer KgCLUE’s questions to our KB to increase ambiguity in order to challenge the EL model as our KB is much larger than KgCLUE’s original KB (16M vs. 3M). Our dataset provides various formats of questions for the same answer, which also increases the EL difficulty.

Basic Training data. Unlike using hyperlinks and anchor texts in previous works (Logeswaran et al., 2019; Wu et al., 2020), we use the entity descriptions to construct a weak-supervised EL training dataset because hyperlinks are not always available in some KBs. Specifically, the description for each entity is parsed for its title or alias name to compose the training data, because the title or alias name occurred in its own description is highly possible to mention the entity, yielding a basic training data of 16 million (text, mention, entity subtitle) tuples.

Additional Training data for Question Answering. Since relation name is commonly used in question answering, we choose it as the additional feature. KgCLUE’s training data, including 18,000 (question, mention, entity relation) tuples, allows us to train a model based on relation.

Baselines. We choose GENRE (Nicola De et al., 2020) and EntQA (Zhang et al., 2021c), the representations of the generative-based and matching-based methods, to be compared with our HOSMEL, as none of them need the pre-defined mentions, which is the same as the proposed HOSMEL.

Evaluation Metrics. We use the topic entities’ top-1 accuracy as the evaluation metric for EntQA and HOSMEL. Since GENRE is different by returning zero or multiple entities without scores, we use recall for it instead of top-1 accuracy.

The details of the experimental settings can be found in Section A.2 in Appendix.

Table 1: Performance of all the models on the two benchmarks, where EntQA and HOSMEL report the top-1 accuracy and GENRE reports the recall rate.

	KgCLUE	Hand-crafted
GENRE	25.28	14.97
EntQA	80.99	60.34
HOSMEL	89.48	77.40
HOSMEL + Relation	94.50	81.15

Table 2: The recall rate of the mention detection result by HOSMEL and the bi-encoder result by EntQA.

	KgCLUE	Hand-crafted
EntQA	89.36	82.02
HOSMEL	99.46	95.12

5.2 Experimental Results

Time and Space Efficiency. Figure 3 shows the time and space cost of GENRE, EntQA, and HOSMEL. GENRE needs a full training of the entire 16 million training data to memorize the knowledge of all the entities in its parameters. EntQA also needs a full training of the dataset because of its bi-encoder. Besides, EntQA needs an additional 60GB storage space for the learned entity embeddings for quick retrieval while HOSMEL and GENRE only need 3.5GB for the Trie tree. The results demonstrate that HOSMEL is quite efficient in both time and space. In advance, we measured the average time used for inference. GENRE takes 21.39 seconds to process each test case, whereas EntQA and HOSMEL respectively only need 0.32 seconds and 0.26 seconds to output the results.

Accuracy Performance. Table 1 shows the performance of all the models on the two benchmarks, KgCLUE and hand-crafted. EntQA, GENRE, and HOSMEL are all trained on the basic training data with only the (text, mention, entity subtitle) tuples. GENRE presents a particularly poor performance, as the search space of the decoder in character-based languages like Chinese is much larger than the word-based languages like English. EntQA also performs worse than HOSMEL because the dense bi-encoder in EntQA is proved to remember the robust representations for common entities but struggles to differentiate rarer entities (Sciavolino et al., 2021). On the contrary, the proposed HOSMEL builds on a sparse Trie tree and a dense mention detection model for retrieving the entity candidates, which can attend to both the common and rare entities. The top-45 recall rates of HOSMEL and EntQA are also reported in Table 2.

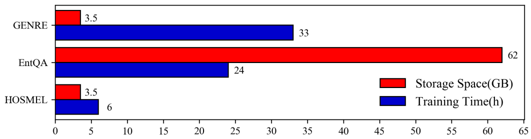


Figure 3: Time and space cost.

Additional Feature's Performance. We additionally evaluate the hot-swappable ability of HOSMEL. We find that when adding a new feature, we can easily train an additional MacBERT on relations and the resultant HOSMEL+Relation further improves $3.71 - 5.02\%$ accuracy. On the contrary, GENRE is unable to leverage the relation features. EntQA is also prevented as it needs to be retrained on the new training data where both the entity subtitle and relation are annotated.

6 Related Work

EL has attracted lots of attention, and many methods have been studied. Among them, matching-based methods (Logeswaran et al., 2019; Yamada et al., 2019; Wu et al., 2020; Jin et al., 2019b; Ferragina and Scaiella, 2012) and generation-based methods (Nicola De et al., 2020; De Cao et al., 2021) are two mainstreams. The former ones usually use a dense retriever based on the maximum inner-product search (MIPS) to retrieve entity candidates, followed by a cross-encoder to re-rank them. The later ones frame EL as a seq2seq model to autoregressively generate the text annotated with the entities' identifiers, such as their subtitles. However, both put all the features together for training, increasing the difficulty of adjusting for specific downstream tasks. Since the downstream tasks usually require specific filtering or additional information invocation for better EL results, these EL models need the newly annotated dataset for retraining, which is costly and inefficient. In addition, most of them are designed for English, and only a few (Jin et al., 2019b; Ferragina and Scaiella, 2012) have been released as a ready-to-use toolkit. HOSMEL is an EL toolkit for Chinese that can easily adjust to downstream tasks.

7 Conclusion and Future Work

We release a Chinese EL toolkit HOSMEL, which has shown to be an effective, efficient, and interpretable method due to the hot-swappable modulized structure. Moreover, for adapting to the downstream tasks, HOSMEL can be easily improved by training on additional features with limited training data. Experiments on two question answering benchmarks have demonstrated the time/space efficiency and the effectiveness compared with the SOTA EL models, as well as the easy task adaptation ability. An English version of the toolkit is planned to be released in the future.

Acknowledgments

This work is supported by the National Natural Science Foundation of China 62076245, and the NSFC for Distinguished Young Scholar 61825602.

References

Bo Chen, Jing Zhang, Jie Tang, Lingfan Cai, Zhaoyu Wang, Shu Zhao, Hong Chen, and Cuiping Li. 2020. Conna: Addressing name disambiguation on the fly. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1.

Bo Chen, Jing Zhang, Xiaokang Zhang, Xiaobin Tang, Lingfan Cai, Hong Chen, Cuiping Li, Peng Zhang, and Jie Tang. 2022. Coad: Contrastive pre-training with adversarial fine-tuning for zero-shot expert linking. In *AAAI 2022*.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 657–668, Online. Association for Computational Linguistics.

Nicola De Cao, Ledell Wu, Kashyap Popat, Mikel Artetxe, Naman Goyal, Mikhail Plekhanov, Luke Zettlemoyer, Nicola Cancedda, Sebastian Riedel, and Fabio Petroni. 2021. Multilingual autoregressive entity linking. *arXiv preprint arXiv:2103.12528*.

Yu Feng, Jing Zhang, Gaole He, Wayne Xin Zhao, Lemao Liu, Quan Liu, Cuiping Li, and Hong Chen. 2021. A pretraining numerical reasoning model for ordinal constrained question answering on knowledge base. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1852–1861.

Paolo Ferragina and Ugo Scaiella. 2012. Fast and accurate annotation of short texts with wikipedia pages. *IEEE Software*, 29(1):70–75.

Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. 2020. A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge and Data Engineering*.

Hailong Jin, Chengjiang Li, Jing Zhang, Lei Hou, Juanzi Li, and Peng Zhang. 2019a. Xlore2: large-scale cross-lingual knowledge graph construction and application. *Data Intelligence*, 1(1):77–98.

Hailong Jin, Chengjiang Li, Jing Zhang, Lei Hou, Juanzi Li, and Peng Zhang. 2019b. Xlore2: large-scale cross-lingual knowledge graph construction and application. *Data Intelligence*, 1(1):77–98.

Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. 2019. Zero-shot entity linking by reading entity descriptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3449–3460, Florence, Italy. Association for Computational Linguistics.

Mauricio Marrone. 2020. Application of entity linking to identify research fronts and trends. *Scientometrics*, 122(1):357–379.

Cao Nicola De, Izacard Gautier, Riedel Sebastian, and Petroni Fabio. 2020. Autoregressive entity retrieval. *arXiv: Computation and Language*. GENRE.

Hadas Raviv, Oren Kurland, and David Carmel. 2016. Document retrieval using entity-based language models. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 65–74.

Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2021. Simple entity-centric questions challenge dense retrievers. *arXiv preprint arXiv:2109.08535*.

Bhavani Thuraisingham. 2020. The role of artificial intelligence and cyber security for social media. In *2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 1–3. IEEE.

Catherine Tucker. 2019. 17. privacy, algorithms, and artificial intelligence. In *The Economics of Artificial Intelligence*, pages 423–438. University of Chicago Press.

MV Wilkes. 1974. The art of computer programming, volume 3, sorting and searching. *The Computer Journal*, 17(4):324–324.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407.

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, Yanting Li, Yina Patterson, Zuoyu Tian, Yiwen Zhang, He Zhou, Shaoweihua Liu, Zhe Zhao, Qipeng Zhao, Cong Yue, Xinrui Zhang, Zhengliang

Yang, Kyle Richardson, and Zhenzhong Lan. 2020. CLUE: A Chinese language understanding evaluation benchmark. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4762–4772, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Ikuya Yamada, Koki Washio, Hiroyuki Shindo, and Yuji Matsumoto. 2019. Global entity disambiguation with pretrained contextualized embeddings of words and entities. *arXiv preprint arXiv:1909.00426*.

Jing Zhang, Bo Chen, Lingxi Zhang, Xirui Ke, and Haipeng Ding. 2021a. Neural, symbolic and neural-symbolic reasoning on knowledge graphs. *AI Open*, 2:14–35.

Shiyue Zhang, Benjamin Frey, and Mohit Bansal. 2021b. Chrentranslate: Cherokee-english machine translation demo with quality estimation and corrective feedback. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 272–279.

Wenzheng Zhang, Wenyue Hua, and Karl Stratos. 2021c. Entqa: Entity linking as question answering.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.

A Appendix

A.1 Proof of Eq.(1)

$$\begin{aligned}
& s(m_i, e_i) \qquad (2)\\
= \; & \log P(A_i^1, A_i^2, \cdots, A_i^t, m_i|d),\\
= \; & \log(P(A_i^1|A_i^2, \cdots, A_i^t, d, m_i)P(A_i^2, \cdots, A_i^t, m_i|d)),\\
= \; & \log(P(A_i^1|d, m_i)P(A_i^2, \cdots, A_i^t, m_i|d)),\\
= \; & \log P(m_i|d)\prod_{\tau=1}^{t} P(A_i^\tau|d, m_i),\\
= \; & \log P(m_i|d) + \sum_{\tau=1}^{t} \log P(A_i^\tau|d, m_i),\\
= \; & s(m_i) + \sum_{\tau=1}^{t} s(A_i^\tau).
\end{aligned}$$

As shown above, the second equation is obtained according to the Bayes Theorem. The third equation is derived based on the assumption that the probability of attribute A_i^1 is independent against the other attributes. Then we achieve the fourth equation according to the general assumption that the probabilities of attributes are independent against each other. Finally, by changing the log-product to sum-log, we show the score of a (mention, entity) pair is equal to the sum of the mention's score and all the attributes' scores.

A.2 Experimental Settings

Dataset

KB. For creating a concise KB from XLore, We filter out the entities that explain the Chinese characters and select the top 10% popular entities by the edit times. Since we filter the mentions based on the Trie tree created by the titles and alias names of all the entities in XLore, to ensure the recall of unseen mentions in the above popular entities, we keep the top 15 popular entities for each of these unseen mentions. The title, subtitle, and the relations except for the alias name of an entity are available in XLore. To improve the mention's coverage of the Trie tree, we collect the alias name of an entity from its relation named "alias" or the similar meaning.

Test Set. We choose KgCLUE(Xu et al., 2020) and a hand-crafted dataset, which contains real world questions with entities we collected from daily life conversations, as the test sets. Both of them are for one-hop question answering from the general Chinese open domain KB, which respectively contains 1,673 and 1,597 questions labeled with the topic entities. Note the fact that, instead of using classic entity linking data sets, question answering data sets are selected for evaluation because most previous entity linking data sets are closer to hyperlink labels, which makes it easier to achieve a better performance but lacks the connection to real-world applications, which is often less similar to the hyperlink labels. Although KgCLUE contains its own KB, a large number of useful entities are filtered out, and some important features such as the subtitle are unavailable. Thus we replace its KB with our created KB from XLore and align the topic entities in the questions to our KB. The hand-crafted test set contains the questions involving more common entities, which raises the difficulty of EL, as common entities are more likely to have ambiguity. We also provide several formats of the same question to increase the question answering difficulty as well as the topic entity linking difficulty.

Basic Training Data. Instead of collecting the hyperlinks and the corresponding anchor texts as the training data of EL (Logeswaran et al., 2019; Wu et al., 2020), we use the descriptions of the entities to construct a weak-supervised EL training dataset, as the hyperlinks are not always available in some KBs. Specifically, for each entity, we extract the first sentence from its description that explicitly mentions the entity's title or alias name to compose the training data because the title or alias name that occurred in the description is highly probably to mention the entity itself. As a result, we obtain about 16 million (text, mention, entity subtitle) tuples as the basic training data.

Additional Training Data for Question Answering. Since a commonly used feature for question answering is the relation name of an entity, we choose it as the additional feature. For training an additional model based on relations, we need to know the correct relation the input text mentions. KgCLUE's training data, including 18,000 (question, mention, entity relation) tuples, exactly satisfy this demand.

Additional Training Data for Question Answering. Since a commonly used feature for question answering is the relation name of an entity, we choose it as the additional feature. For training an additional model based on relations, we need to know the correct relation the input text mentions. KgCLUE's training data, including 18,000 (question, mention, entity relation) tuples, exactly satisfy this demand.

Baselines. GENRE trains a seq2seq model to translate the input text into the text annotated with the mentions and entity subtitles based on a pre-built Trie tree the same as HOSMEL. EntQA first trains a bi-encoder to retrieve the entity candidates and then trains a machine comprehensive model to extract the mention spans from the input text given the entity candidates. The scores of the two models are summed as the final score of a (mention, entity) pair.

For a fair comparison, all the models are trained on the basic training data. GENRE needs a full training of the entire 16 million training data to encode all the entity information into its parameters. EntQA also requires the use of the full data to improve the recall for the bi-encoder. HOSMEL doesn't suffer the need to train a bi-encoder, thus only needs a small amount of the training data. For training efficiency, we sample and create a multiple-choice-like training dataset from the basic training data with 406,420 (text, mention) pairs for training the mention detection model and 111,648 (text, mention, entity subtitle) tuples for training entity disambiguation by subtitle in our model. Only the proposed HOSMEL is trained on the additional training data because of its adaptation ability. Based on the KgCLUE's training data, we create an additional multiple-choice-like training data with 47,870 (text, mention, entity relation) tuples for training entity disambiguation by relation.

B Ethical Considerations

For years the press has been arguing the use of AI and its pros and cons. One advance could be used in various ways and thus lead to different outcomes. To take a cultural look at how this work and other works in similar tracks will take effect, we would like first to take a brief on how might our work be used in both good and bad ways, then move on to applying our advance and ethical reasons for developing our toolkit, along with privacy issues.

For our demo, the outcome can shift in between justice and harmful outcomes. EL could be viewed as having an expert to extract key concepts from a given text, which means that it could be used in education to help the students find a related term in their reading before they fully understand the field. This could also be used in specialized domains such as biological and pharmaceutical for fast retrieval of useful concepts (Marrone, 2020). However, this could also be used in harmful ways. The chance of EL being used for detecting particular views in social media might be further applied to ban a specific group from expressing opinions, harming freedom of speech and equality. But if we look at it from a different perspective, if such use could be controlled by the users of the social media, potentially people who have difficulties can filter out the harmful languages to them (Thuraisingham, 2020) and find what they wanted faster.

To ensure our work could be used in the right way, we extracted our domain from XLore, where it's only a general KB without the worry of having harmful potential entities. We also separated the features. This raises the challenge to train the ranking model to favor a specific semantic pattern and thus makes it harder to be used against free speech. Our work purely ranks the similarity in context rather than learning the complete set of all entities, this could prevent the linking result from being biased to only popular entities (Sciavolino et al., 2021), yet we still worry that specific context might lead to the linking of only popular entities. As a result, we strongly call for more work conducted to study the context and candidate similarity in retrievals instead of joining the popularity into final performance for better equality. We noticed that in recent years the protection for minor languages has finally drawn more attention (Zhang et al., 2021b), and one of the reasons for conducting our demonstration is the attempt in calling both English and Chinese, the two most popular languages, speakers to better consider the difference in languages and robustness while developing methods not only for the sake of equality between languages but also for better protection of the minor languages because not all languages are like English.

On the other hand, privacy has raised a significant portion of attention (Tucker, 2019). It is essential to discuss how our tool might relate to privacy. Our demo is based on XLore, which means it only uses data publicly available on the internet, but the risk of privacy leakage still remains while being applied to the downstream tasks. During usage, a level of caution to prevent privacy issues should still be kept for the sake of respect. In advance, we suggest usages of our method to be checked before actual deployment in a downstream system.

One overall solution to resolve the release of methods in data science is to discuss and consider the caution of ethics and respect during education. Some might argue the key is to take extra care during the development of such tools but to notice critical factors in a system that might lead to harmful usage requires strong integrity and respect to others. It is only with high ethical standards one could better consider the design and take better consideration of one's system during the design and before releasing.