Find the Funding: Entity Linking with Incomplete Funding Knowledge Bases

Gizem Aydin
Radboud University
gizemaydin96@gmail.com
Seyed Amin Tabatabaei
Elsevier
s.tabatabaei@elsevier.com
Giorgios Tsatsaronis
Elsevier
g.tsatsaronis@elsevier.com
Faegheh Hasibi
Radboud University
f.hasibi@cs.ru.nl

Abstract

Automatic extraction of funding information from academic articles adds significant value to industry and research communities, including tracking research outcomes by funding organizations, profiling researchers and universities based on the received funding, and supporting open access policies. Two major challenges of identifying and linking funding entities are: (i) sparse graph structure of the Knowledge Base (KB), which makes the commonly used graph-based entity linking approaches suboptimal for the funding domain, (ii) missing entities in KB, which (unlike recent zero-shot approaches) requires marking entity mentions without KB entries as NIL. We propose an entity linking model that can perform NIL prediction and overcome data scarcity issues in a time and data-efficient manner. Our model builds on a transformer-based mention detection and a bi-encoder model to perform entity linking. We show that our model outperforms strong existing baselines.

1 Introduction

Entity Linking (EL) aims to annotate text with corresponding entity identifiers from a Knowledge Base (KB) and is a building block for different tasks, such as document ranking (Xiong et al., 2017), entity retrieval (Hasibi et al., 2016), and question understanding in conversations (Shang et al., 2021). Recent years have witnessed the flourishing of entity linking approaches for zero-shot (Wu et al., 2020; Li et al., 2020) and open-domain setups (van Hulst et al., 2020; Cao et al., 2021). While zero-shot entity linking can generalize to new specialized domains and entity dictionaries, existing approaches cannot perform NIL prediction; i.e., identifying entity mentions without a target entity in a knowledge base and assigning them to NIL. Open-domain entity linkers, on the other hand, build on the availability of rich entity relations and descriptions in KBs. This makes existing EL approaches suboptimal for real-world applications of entity linking in domains with incomplete knowledge bases, where both in-KB and out-of-KB entities should be identified.

In this paper, we aim to address entity linking in the funding domain (Dai et al., 2021; Alexander and de Vries, 2021), which is essential for funding organizations to track the outcome of the research they funded (Kayal et al., 2019) and also helps to comply with open access rules (Dai et al., 2021). Knowledge bases of funding organizations, either proprietary or open access (e.g., the funding KB Crossref\(^{1}\)), contain brief information about entities (e.g., official name and acronym). They also have extremely sparse graph structure with large amount of missing entities that need to be found from research articles. This implies that EL in the funding domain requires detecting mentions with out-of-KB entities while handling sparse entity relations and descriptions. The approach, should also be able to operate with limited training data, as large public datasets are rarely available for domain-specific applications.

We propose a two-step EL approach, where we first identify entity mentions using task adaptive pre-training (Gururangan et al., 2020) of BERT (Devlin et al., 2019) and then perform Entity Disambiguation (ED) by utilizing a bi-encoder model to learn dense entity and mention representations. Our bi-encoder model and training approach using negative sampling are specifically designed to operate with out-of-KB entities. The bi-encoder is followed by a modest feature-based model to map the entities to an entity in KB or NIL. We create two new datasets for EL in the funding domain and compare our mention detection and ED approaches with strong neural and feature-based models. We show that our model improves over existing baselines for both entity disambiguation and end-to-end entity linking.

\(^{1}\)https://www.crossref.org

1937

Proceedings of the 29th International Conference on Computational Linguistics, pages 1937–1942
October 12–17, 2022.
We denote \( E \) which represents all possible mentions linked to an entity in the KB. For mention detection, we adapt BERT (Devlin et al., 2020) and open domain entity linking (van Hulst et al., 2019) to the funding domain. Domain Adaptation Pre-Training (DAPT), while being effective, requires large amount of training data which is not feasible for the funding domain (Gerritse et al., 2022; Nogueira et al., 2019). We therefore utilize the Task-Adaptive Pre-Training (TAPT) (Gururangan et al., 2020), which requires a far smaller but more task-relevant training corpus and is proven to be more effective than DAPT. We train BERT with the Masked Language Model objective on acknowledgments of research papers. We refer to this model as BERT\textsubscript{TAPT}. We then fine-tune BERT\textsubscript{TAPT} for the mention detection task using IOB tags.

## 2 Method

In this section, we provide a formal definition of the task, followed by the description of our Mention Detection (MD) approach and Funding entity Disambiguation model, referred to as FunD.

### 2.1 Task Definition

We denote \( \mathcal{E} \) as the set of entities in a knowledge base, where each entity \( e \in \mathcal{E} \) is accompanied by a textual description. Let \( m = (s, t) \in M \) denote an entity mention with start and end positions \( s \) and \( t \). Given a document \( d = \{w_1, w_2, \ldots, w_n\} \), our aim is to generate the list:

\[
\mathcal{L} = \{(s, t, a) | 1 \leq s \leq t \leq n, a \in \mathcal{E} \cup \mathbb{NIL}\},
\]

which represents all possible mentions linked to an entity in the KB (in-KB setup) or NIL (out-of-KB setup). This task is similar to zero-shot (Wu et al., 2020) and open domain entity linking (van Hulst et al., 2020), but different from them, entities do not need to have an entry in the KB.

### 2.2 Mention Detection

For mention detection, we adapt BERT (Devlin et al., 2019) to the funding domain. Domain Adaptive Pre-Training (DAPT), while being effective, requires a large amount of domain-specific text, requires large amount of training data which is not feasible for the funding domain (Gerritse et al., 2022; Nogueira et al., 2019). We therefore utilize the Task-Adaptive Pre-Training (TAPT) (Gururangan et al., 2020), which requires a far smaller but more task-relevant training corpus and is proven to be more effective than DAPT. We train BERT with the Masked Language Model objective on acknowledgments of research papers. We refer to this model as BERT\textsubscript{TAPT}. We then fine-tune BERT\textsubscript{TAPT} for the mention detection task using IOB tags.

## 2.3 Entity Disambiguation

### Candidate Entity Selection

To obtain the likelihood of an entity being a target link of a mention, we employ a bi-encoder model (Wu et al., 2020) for encoding a mention (with its context) and an entity. Our bi-encoder model utilizes two BERT encoders for generating entity and mention representations. The entity encoder takes the structured entity description as the input:

\[
x_e = \text{BERT}_{[CLS]}(\text{[CLS]} \ \text{val}_{A_1}[\text{E}_A] \ldots [\text{E}_A] \ \text{val}_{A_n} [\text{E}_B] \ \text{val}_{B} \ \text{[SEP]}),
\]

where \([\text{E}_A]\) and \([\text{E}_B]\) are two word-piece tokens, selected among the unused tokens of BERT, and \(\text{val}_{A_i}\) and \(\text{val}_{B}\) denote values for entity attributes \(A\) and \(B\). Here \(A\) corresponds to names of entities, which is a multi-valued attribute, and \(B\) is the country of the funding organization. For the mention encoder, we follow Wu et al. (2020) and obtain mention representations by:

\[
x_m = \text{BERT}_{[CLS]}(\text{[CLS]} \ \text{ctxt}_{left} \ [M_e] \ \text{mention} \ [M_e] \ \text{ctxt}_{right} \ \text{[SEP]}),
\]

where \(\text{ctxt}_{left}\) and \(\text{ctxt}_{right}\) represent context words before and after the mention. In this work, both BERT models are initialized with BERT\textsubscript{TAPT}. The mention-entity score is then obtained by:

\[
f(m, e) = \mathbf{W}^T (x_m \odot x_e),
\]

where \(\odot\) refers to element-wise multiplication of mention representation \(x_m\) and entity representation \(x_e\), and \(\mathbf{W} \in \mathbb{R}^{BERT \times 2}\) represents learnable weights. The binary cross-entropy loss \(L\) is used to train the model:

\[
L = - \frac{1}{N} \sum_{i=1}^{N} l_i \log(f(m_i, e_i)) - (1 - l_i) \log(1 - f(m_i, e_i)),
\]
where \(N\) is the number of training examples and \(l_i\) is a binary value that is set to 1 if \(e_i\) is the correct entity for mention \(m_i\).

**Negative Sampling** Following Gillick et al. (2019), we perform training in rounds, where the model obtained in each round is used to produce hard negatives for the next round. Contrary to (Gillick et al., 2019), we do not use in-batch random negative sampling, as it provides less diverse random negatives for sparse domain-specific applications compared to open-domain EL.

The following strategy is employed for random and hard negative sampling. In the first round, negative entities of each mention are sampled randomly from the entire KB. For the next rounds, both random and hard negatives are used. Hard negatives are entities ranked above the correct entity by the model learned in the previous step. For mentions with out-of-KB entities, the top-K entities are selected as hard negatives (K is set to 10 following (Gillick et al., 2019)). The number of random negatives for each mention is computed based on the number of hard negatives:

\[
Neg_r(m) = \left\lfloor \frac{\sum_{i=1}^{|M|} Neg_h(m)}{|M|} \right\rfloor, \quad (2)
\]

where \(Neg_r()\) and \(Neg_h()\) give number of random and hard negatives, respectively. Using this strategy, we strive a balance between random and hard negatives, while giving hard mentions (i.e., mentions that their correct entities are in low ranks) a larger number of hard negatives.

**Entity or NIL Selection** Once we have obtained candidate entities from our bi-encoder model, we turn to mapping each mention to an entity in the knowledge base or NIL. We employ a feature-based model utilizing Gradient Boosting Machine (GBM) (Friedman, 2001). Our model utilizes five light-weight features: (i) score obtained by the bi-encoder model, (ii) maximum Levenshtein similarity between the mention and the labels of the candidate entity, (iii) link probability of the mention (Balog, 2018) obtained by dividing numbers of times a term appears as a link by total number of occurrences of a term: \(P(link|m) = n_{link}(m)/n(m)\), and (iv) commonness score (Balog, 2018) obtained by \(P(e|m) = n(m,e)/\sum_{e \in \mathcal{E}} n(m,e)\), with \(n(m,e)\) denoting number of times that mention \(m\) is linked to entity \(e\). A mention is linked to the entity with the highest GBM score if higher than threshold \(\tau\).

**3 Experiments**

**Data** We use the Crossref funding registry as our KB, containing information about 25,859 funding organizations. We create two new datasets for the funding domain, ELFund and EDFund (Afzal et al., 2022), which are used for experiments. The datasets are split into training, validation, test, and eval sets, with no overlaps in the training, test, and eval across the two datasets. The validation set is used for searching hyper-parameters. The eval set contains completely unseen production data, used for the final evaluation of the models; see Appendix A and B for more details about the datasets and experimental setup.

**Evaluation Metrics** To evaluate the mention detection step, we use strong matching precision, recall, and F1 score (Tjong Kim Sang and De Meulder, 2003; Usbeck et al., 2019). The ED and EL tasks are evaluated in three settings: (i) In-KB, for mentions linked to entities in KB, (ii) Emerging Entities (EE) for Out-of-KB entities; i.e., mentions linked to no entities, and (iii) All, for in-KB and emerging entities. We evaluate the ED task using micro and macro averaged accuracy for the All setting (Hoffart et al., 2014). For In-KB and EE settings, we report on micro- and macro-averaged precision, recall, and F1 (Usbeck et al., 2019).

**4 Results**

**Mention Detection** Table 1 shows the results for the mention detection step. We compare BERTMD TAPT with Stanford NER (Finkel et al., 2005), FlairNER (Akbik et al., 2018), and BERTNER (Devlin et al., 2019). The results show that FlairNER achieves the highest precision, but the lowest recall compared to the BERT-based models. We also observe that BERTMD TAPT outperforms all baselines with respect to the F1 score, showing the importance of task adaptive pre-training when limited data is available.

<table>
<thead>
<tr>
<th>System</th>
<th>Set</th>
<th>P</th>
<th>R</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stanford NER</td>
<td>Test</td>
<td>73.70</td>
<td>75.10</td>
<td>74.39</td>
</tr>
<tr>
<td>FlairNER</td>
<td>Test</td>
<td><strong>85.83</strong></td>
<td>78.02</td>
<td>81.74</td>
</tr>
<tr>
<td>BERTNER</td>
<td>Test</td>
<td>79.18</td>
<td>86.03</td>
<td>82.46</td>
</tr>
<tr>
<td>BERTMD TAPT</td>
<td>Test</td>
<td>80.28</td>
<td><strong>86.54</strong></td>
<td><strong>83.29</strong></td>
</tr>
<tr>
<td>Stanford NER</td>
<td>Eval</td>
<td>76.17</td>
<td>72.87</td>
<td>74.48</td>
</tr>
<tr>
<td>BERTMD TAPT</td>
<td>Eval</td>
<td><strong>79.08</strong></td>
<td><strong>85.31</strong></td>
<td><strong>82.08</strong></td>
</tr>
</tbody>
</table>

Table 1: Mention detection results on ELFund dataset.
Table 2: Entity disambiguation results on the EDFund dataset. Best results for each set are marked in bold face.

<table>
<thead>
<tr>
<th>Method</th>
<th>Set</th>
<th>Acc&lt;sub&gt;mic&lt;/sub&gt;</th>
<th>Acc&lt;sub&gt;mac&lt;/sub&gt;</th>
<th>P&lt;sub&gt;mic&lt;/sub&gt;</th>
<th>R&lt;sub&gt;mic&lt;/sub&gt;</th>
<th>F1&lt;sub&gt;mic&lt;/sub&gt;</th>
<th>P&lt;sub&gt;mac&lt;/sub&gt;</th>
<th>R&lt;sub&gt;mac&lt;/sub&gt;</th>
<th>F1&lt;sub&gt;mac&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commonness</td>
<td>Test</td>
<td>83.8</td>
<td>85.81</td>
<td>53.55</td>
<td>66.44</td>
<td><strong>82.2</strong></td>
<td>78.67</td>
<td>78.89</td>
<td><strong>88.25</strong></td>
</tr>
<tr>
<td>GBM&lt;sub&gt;F&lt;sub&gt;26&lt;/sub&gt;&lt;/sub&gt;</td>
<td>Test</td>
<td>91.02</td>
<td><strong>92.84</strong></td>
<td><strong>79.11</strong></td>
<td><strong>81.14</strong></td>
<td><strong>79.25</strong></td>
<td>93.2</td>
<td><strong>93.29</strong></td>
<td>93.25</td>
</tr>
<tr>
<td>FunD</td>
<td>Test</td>
<td>91.15</td>
<td>92.76</td>
<td>77.44</td>
<td>81.14</td>
<td><strong>79.25</strong></td>
<td>93.82</td>
<td>92.99</td>
<td><strong>93.40</strong></td>
</tr>
<tr>
<td>GBM&lt;sub&gt;F&lt;sub&gt;26&lt;/sub&gt;&lt;/sub&gt;</td>
<td>Eval</td>
<td>90.26</td>
<td>90.84</td>
<td><strong>80.03</strong></td>
<td>81.49</td>
<td>80.75</td>
<td>92.69</td>
<td>92.3</td>
<td>92.5</td>
</tr>
<tr>
<td>FunD</td>
<td>Eval</td>
<td><strong>90.66</strong></td>
<td><strong>91.11</strong></td>
<td>79.26</td>
<td><strong>85.45</strong></td>
<td><strong>82.24</strong></td>
<td><strong>93.56</strong></td>
<td>91.86</td>
<td><strong>92.7</strong></td>
</tr>
</tbody>
</table>

Table 3: Entity linking results on the ELFund dataset.

<table>
<thead>
<tr>
<th>MD</th>
<th>ED</th>
<th>Setting</th>
<th>F1&lt;sub&gt;mic&lt;/sub&gt;</th>
<th>F1&lt;sub&gt;mac&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stanford NER</td>
<td>GBM&lt;sub&gt;F&lt;sub&gt;26&lt;/sub&gt;&lt;/sub&gt;</td>
<td>All</td>
<td>68.43</td>
<td>69.34</td>
</tr>
<tr>
<td>BERT&lt;sub&gt;MD&lt;/sub&gt; TAPT</td>
<td>FunD</td>
<td><strong>75.81</strong></td>
<td><strong>76.59</strong></td>
<td></td>
</tr>
<tr>
<td>Stanford NER</td>
<td>GBM&lt;sub&gt;F&lt;sub&gt;26&lt;/sub&gt;&lt;/sub&gt;</td>
<td>EE</td>
<td>43.34</td>
<td>71.01</td>
</tr>
<tr>
<td>BERT&lt;sub&gt;MD&lt;/sub&gt; TAPT</td>
<td>FunD</td>
<td><strong>52.82</strong></td>
<td><strong>73.68</strong></td>
<td></td>
</tr>
<tr>
<td>Stanford NER</td>
<td>GBM&lt;sub&gt;F&lt;sub&gt;26&lt;/sub&gt;&lt;/sub&gt;</td>
<td>In-KB</td>
<td>77.33</td>
<td>74.73</td>
</tr>
<tr>
<td>BERT&lt;sub&gt;MD&lt;/sub&gt; TAPT</td>
<td>FunD</td>
<td><strong>85.14</strong></td>
<td><strong>81.86</strong></td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Efficiency of ED models (in seconds).

<table>
<thead>
<tr>
<th>System</th>
<th>With GPU</th>
<th>Without GPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>FunD</td>
<td>9.26 ± 0.47</td>
<td>23.07 ± 0.78</td>
</tr>
<tr>
<td>GBM&lt;sub&gt;F&lt;sub&gt;26&lt;/sub&gt;&lt;/sub&gt;</td>
<td>99.2±0.45</td>
<td>99.2±0.45</td>
</tr>
</tbody>
</table>

5 Conclusions

In this paper, we have introduced an entity linking method for funding domain, where the knowledge base has sparse graph structure and limited information is available about entities. The model builds on BERT to perform mention detection, and a bi-encoder model to conduct the entity disambiguation. We compared our method to strong feature-based and zero-shot models and showed that our model can perform NIL assignments and overcome data scarcity issues more efficiently and effectively than comparable baselines. As future work, we would like to explore the benefit of employing contrastive learning for the highly ambiguous entity mentions, which could provide further robustness to extracting and linking such entities in scientific texts that span across all sciences.
Acknowledgements

We thank Ramadurai Petchiappan, Georgios Cheirmpos, and Efthymios Tsakonas, for their assistance in preparing the final released datasets.

References


Table 5: Overview of ELFund and EDFund datasets. For each split, number of articles, number of mentions used in training the NER models, number of mention-entity pairs used in training the ED models, and the percentage of NIL mentions among those pairs are shown.

<table>
<thead>
<tr>
<th>Set</th>
<th>#Articles</th>
<th>#LinksEL</th>
<th>#LinksED</th>
<th>NIL%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>22,720</td>
<td>67,671</td>
<td>95,761</td>
<td>18.58%</td>
</tr>
<tr>
<td>Val</td>
<td>991</td>
<td>4,333</td>
<td>5,618</td>
<td>15.47%</td>
</tr>
<tr>
<td>Test</td>
<td>3,943</td>
<td>16,355</td>
<td>19,765</td>
<td>15.56%</td>
</tr>
<tr>
<td>Eval</td>
<td>13,851</td>
<td>37,495</td>
<td>52,378</td>
<td>18.83%</td>
</tr>
</tbody>
</table>

The ELFund and EDFund datasets are created based on scientific articles published before 2017. Expert annotators were asked to identify sentences that contain funding organizations of the research (e.g., X was funded by source Y) and link the organizations to entities in the Crossref KB. ELFund further contains sentences that could be also automatically identified by a classifier. Both datasets were annotated by two experts to find the boundary of mentions and their corresponding entities if available. Disagreements were resolved by a third annotator, and mentions with out-of-KB entities are annotated with NIL. We note that ELFund is not a subset of EDFund.

A Dataset Statistics

The ELFund and EDFund datasets are created based on scientific articles published before 2017. Expert annotators were asked to identify sentences that contain funding organizations of the research (e.g., X was funded by source Y) and link the organizations to entities in the Crossref KB. ELFund further contains sentences that could be also automatically identified by a classifier. Both datasets were annotated by two experts to find the boundary of mentions and their corresponding entities if available. Disagreements were resolved by a third annotator, and mentions with out-of-KB entities are annotated with NIL. We note that ELFund is not a subset of EDFund.

B Training Configuration

We train the case-preserving version of BERT<sub>BASE</sub> with 2M sentences containing funding information to obtain the BERT<sub>TAPT</sub> model. Unless indicated otherwise, the hyper-parameters recommended by Gururangan et al. (2020) are used for training. The training is done on an NVIDIA Tesla K80 GPU with 12 GB of memory with a batch size of 2048 through gradient accumulation and for one epoch (1000 steps). We further fine-tune BERT<sub>TAPT</sub> for the mention detection task on the ELFund dataset. The fine-tuning process is done for 3 epochs with batch size of 8. We refer to this model as BERT<sub>MD</sub><sup>TAPT</sup>. For disambiguation, the bi-encoder model is trained on the EDFund dataset with a learning rate of $2 \times 10^{-5}$ and batch size of 16. The training is performed in 4 rounds, each round consisting of 2 epochs. In the first round, 3 random negatives are used for each mention. The score threshold $\tau$ is set to 0.042 using grid search. Following (Wu et al., 2020), the mention and entity representations are limited to 64 and 256 tokens, respectively.