Improving Candidate Retrieval with Entity Profile Generation for Wikidata Entity Linking

Anonymous ACL submission

Abstract

There is little work on entity linking (EL) over Wikidata, even though it is the most extensive crowdsourced knowledge base. The scale of Wikidata can open up many new real-world applications, but its massive number of entities also makes EL challenging. To effectively narrow down the search space, we propose a novel candidate retrieval paradigm based on entity profiling. Wikidata entities and their textual fields are first indexed into a text search engine (e.g., Elasticsearch). During inference, given a mention and its context, we use a sequence-to-sequence (seq2seq) model to generate the profile of the target entity, which consists of its title and description. We use the profile to query the indexed search engine to retrieve candidate entities. Our approach complements the traditional approach of using a Wikipedia anchor-text dictionary, enabling us to further design a highly effective hybrid method for candidate retrieval. Combined with a simple cross-attention reranker, our complete EL framework achieves state-of-the-art results on three Wikidata-based datasets and strong performance on TACKBP-2010[1].

1 Introduction

Entity linking (EL) is the task of mapping entity mentions in a document to standard referent entities in a target knowledge base (KB) (Dill et al., 2003; Cucerzan, 2007; Mihalcea and Csomai, 2007; Milne and Witten, 2008; Ji et al., 2010). EL systems have found applications in many tasks such as question answering (Li et al., 2020) and knowledge base population (Dredze et al., 2010). In general, the task is challenging because the same word or phrase can be used to refer to different entities. At the same time, the same entity can be referred to by different words or phrases.

Given the importance of EL, researchers have introduced a plethora of EL methods, ranging from using hand-crafted features (Ratinov et al., 2011; Pan et al., 2015) to using deep language models (Agarwal and Bikel, 2020; Cao et al., 2021; Botha et al., 2020). The vast majority of these studies have focused on linking mentions to Wikipedia or Wikipedia-derived KBs such as DBpedia (Auer et al., 2007) or YAGO (Suchanek et al., 2007). As of November 2021, there are about 6.4 million articles in English Wikipedia. However, many entities are still missing from Wikipedia (Redi et al., 2020).

On the other hand, Wikidata, the most extensive general-interest KB, has much broader coverage than Wikipedia (Vrandečić and Krötzsch, 2014). Wikidata contains more than 40 million entities with English titles, about seven times more than the number of articles in English Wikipedia. Every entity in Wikipedia has an equivalent entry in Wikidata, but not vice versa. The scale of Wikidata can open up many new real-world applications. When a disaster happens, many people rush to social media to share updates about the event (Ashktorab et al., 2014). Using an EL system to extract critical information (e.g., affected locations and donor agencies) can aid in monitoring the situation (Zhang et al., 2018). However, many entities may not be well-known, and these entities are likely to be present in Wikidata than in Wikipedia (Geiß et al., 2017).

Despite the potential of Wikidata becoming a universal hub of real-world entities, there exists little in-depth research on EL over Wikidata (Möller et al., 2021). The massive number of entities in Wikidata makes it challenging to find the correct entity for an input mention. Many previous EL methods for Wikipedia use a dictionary built from anchor texts to reduce the original search space to a small list of candidate entities (Han et al., 2011; Shen et al., 2015; Phan et al., 2017). This dictionary-based approach is not directly applicable to Wikidata, since the description of each entity in Wikidata does not contain any anchor text.

In this work, we propose a novel candidate re-

[1] The code and data will be made publicly available.

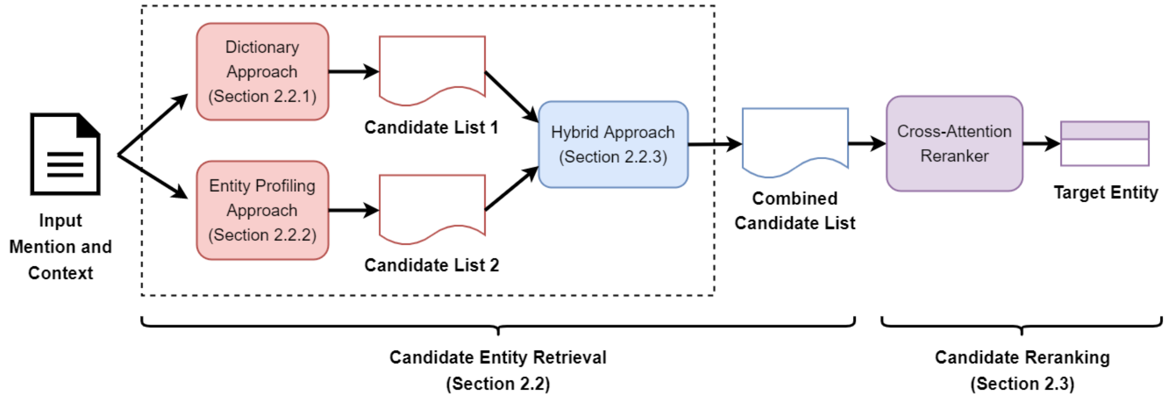

Figure 1: An overview of EPGEL, our entity linking framework.

trieval paradigm for Wikidata based on entity profile generation. Wikidata entities and their textual fields are first indexed into a text search engine (e.g., Elasticsearch). Given an entity mention and its context, we use a seq2seq model to generate the profile of the target entity, which consists of its title and description. The profile is then used to query the indexed search engine to retrieve candidate entities. Our technique is applicable to virtually any KB, not just Wikipedia or Wikidata. It also complements the dictionary-based approach, enabling us to further design an effective hybrid method for candidate retrieval. Combined with a simple cross-attention reranker, our complete EL framework achieves state-of-the-art (SOTA) results on three Wikidata-based datasets and strong performance on the standard TACKBP-2010 dataset.

In summary, our main contributions are: (1) a novel candidate retrieval paradigm based on entity profiling and (2) a new EL framework for Wikidata. Extensive experiments on four public datasets verify the effectiveness of our framework. We refer to our framework as **EPGEL**, which stands for **E**ntity **P**rofile **G**eneration for **E**ntity **L**inking.

2 Methods

2.1 Overview

Problem Formulation Given a set of mentions $M = \{m_1, ..., m_N\}$ in a document and a knowledge base $\mathcal{E}$, the task is to find a mapping $M \rightarrow \mathcal{E}$ that links each mention to a correct entity in $\mathcal{E}$. We assume that entity mentions are already given, e.g., identified by some mention extraction module.

Entity Linking Framework Figure 1 shows an overview of EPGEL. At a high level, similar to many previous methods (Shen et al., 2015), EPGEL consists of two main stages: (1) candidate entity retrieval (2) candidate reranking. Given an entity mention, the role of the candidate retrieval module is to retrieve a small list of candidate entities (Sec. 2.2). Our candidate retrieval approach is a combination of both the traditional dictionary-based approach (Sec. 2.2.1) and our profiling-based approach (Sec. 2.2.2). In the second stage, each candidate entity is reranked by a simple Transformer-based cross-attention reranker (Sec. 2.3).

2.2 Candidate Entity Retrieval

2.2.1 Dictionary-based Candidate Retrieval

Overview Dictionary-based techniques are the dominant approaches to candidate retrieval of many previous Wikipedia EL systems (Guo et al., 2013; Ling et al., 2015; Fang et al., 2020). The basic idea is to estimate the mention-to-entity prior probability $\hat{p}(e|m)$. For example, both the technology company Amazon and the Amazon river could be referred to by "Amazon". However, when people mention "Amazon", it is more likely that they mean the company rather than the river.

Prior Estimation The anchor texts in Wikipedia are frequently used for estimating the prior probability:

$$\hat{p}(e|m) = \frac{\text{count}(m, e)}{\text{count}(m)} \quad (1)$$

where $\text{count}(m)$ is the total number of anchor texts having the entity mention m as the surface form in Wikipedia; $\text{count}(m, e)$ denotes the number of anchor texts with the surface form m pointing to the entity e. Even though this approach is highly effective for EL over Wikipedia (Ganea and Hofmann, 2017), it is not directly applicable to Wikidata. A dictionary built from Wikipedia anchor texts will never return entities that are in Wikidata but not

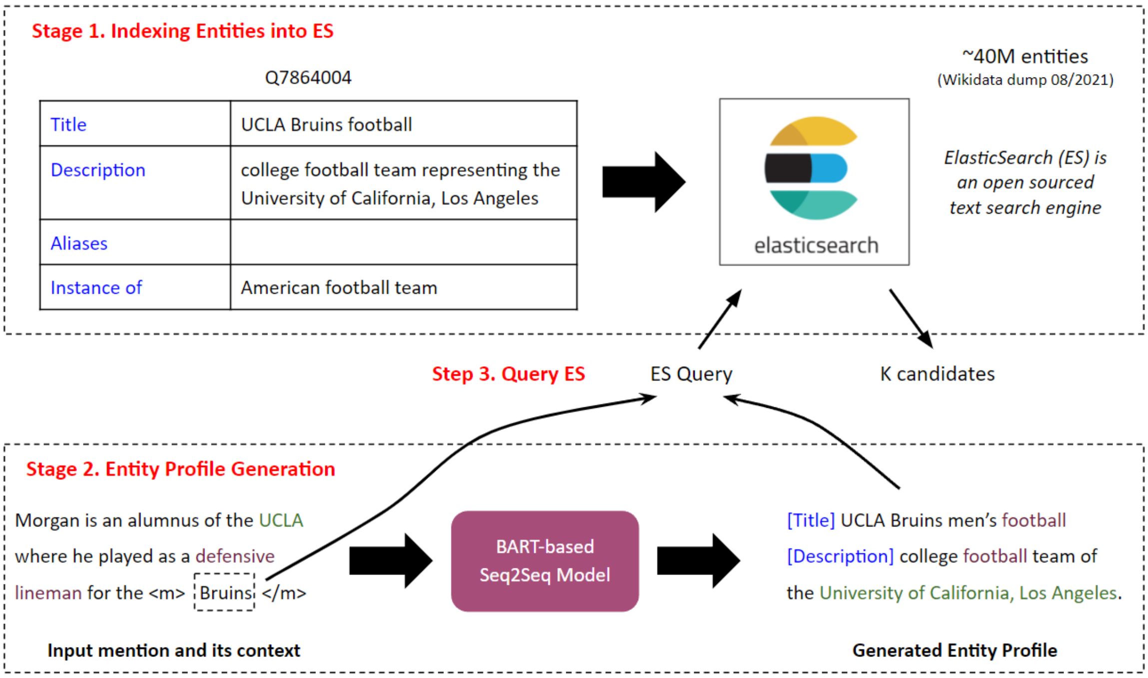

Figure 2: Candidate retrieval based on entity profiling.

in Wikipedia. Furthermore, in Wikidata, the textual description of each entity is typically short and does not contain any anchor text. Therefore, it is not possible to build a dictionary specifically for Wikidata using the same approach. Below, we propose a new approach that is applicable to Wikidata.

2.2.2 Entity Profiling for Candidate Retrieval

Overview We propose a more general paradigm for candidate retrieval (Figure 2). We first index all useful entities from Wikidata into Elasticsearch (ES), an open-source text search engine. During inference, given an entity mention and its context, we use a sequence-to-sequence (seq2seq) model to generate the profile of the target entity. We then use the original mention and the generated profile as the basis for formulating the ES query. This candidate retrieval approach based on entity profiling is applicable to virtually any KB. At the very least, each entity in a KB typically has a textual title.

Entity Profile Generation Model A straightforward approach to query ES is to directly use the literal string of the input mention (Sakor et al., 2020; Kannan Ravi et al., 2021). However, without any contextual information, the literal mention text is not informative and discriminative enough. In the example shown in Figure 2, one can simply ask ES to search for entities whose *title* field or *aliases* field contains the word "Bruins". However, there is an ice hockey team based in Boston named "Bruins" (Q194121), and there is also a college basketball team with the same name (Q3615392). Neither of these entities is the correct target entity (a football team of UCLA). In the input context, the phrase "defensive lineman" implies that the mention refers to a football team. Also, as UCLA is a common acronym abbreviating the University of California, Los Angeles, a well-trained generation model can generate a description that closely resembles the target entity's actual description (Figure 2).

To this end, we train a conditional generation model for generating the profile of the target entity, where the condition is the mention and its context:

$$\text{[s] ctx}_{\text{left}} \text{ [m] } \textit{mention} \text{ [/m] ctx}_{\text{right}} \text{ [/s]}$$

Here, *mention*, ctx_{left}, $\text{ctx}_{\text{right}}$ are the tokens of the entity mention, context before and after the mention respectively. $[m]$ and $[/m]$ are used to separate the original mention from its context. $[s]$ and $[/s]$ are special tokens denoting the start and the end of the entire concatenated input, respectively. The target output is a concatenation of the target entity's title and its description (Figure 2).

Our conditional generation model is an encoder-decoder language model (e.g., BART (Lewis et al., 2020a) and T5 (Raffel et al., 2020)). The generation process models the conditional probability of selecting a new token given the previous tokens and the input to the encoder.

$$p(\mathbf{Y}_{1:n}|c) = \prod_{i=1}^{n} p(\mathbf{Y}_i \,|\, \mathbf{Y}_{<i}, c) \quad (2)$$

where $\mathbf{Y}_{1:n}$ denotes the target output sequence and c denotes the condition (i.e., the input mention and its context).

Elasticsearch Query Construction We directly use the original mention and the generated profile as the basis for formulating the ES query. We ask ES to score each entity based on the following criteria: (1) The similarity between the *title* and *alias* fields and the literal mention text. (2) The similarity between the *title* and *alias* fields and the generated title (3) The similarity between the *description* field and the generated description. More details are in the appendix due to space constraints.

2.2.3 Hybrid Approach to Candidate Retrieval

Overview Our main goal is to perform EL to Wikidata. However, a source document often contains entity mentions that can be linked to Wikipedia since Wikipedia still covers many fields and areas of interest. In addition, every entity in Wikipedia can be automatically mapped to an equivalent entity in Wikidata. As such, we propose a hybrid approach that combines both the dictionary-based technique (Section 2.2.1) and our profiling-based retrieval technique (Section 2.2.2). We first combine the lists produced by these two methods into one single candidate list. We then use a Gradient Boosted Tree (GBT) model (Friedman, 2001) to assign a score to every candidate. Finally, the combined list is sorted based on the candidates' computed scores.

Combining Candidate Lists For a mention m, let $C_d(m)$ be the set of candidates retrieved by a Wikipedia-based dictionary. Let $C_e(m)$ be the set of candidates retrieved by querying ES using generated entity profiles. We train a GBT model that assigns a score to every candidate in the combined set $C_d(m) \cup C_e(m)$. We use two groups of features: string-based and ranking-based features.

For string-based features, we use several similarity metrics: (1) Levenshtein ratios (Levenshtein, 1965), Jaro–Winkler distances (Jaro, 1989), and numbers of common words between the mention's surface form and the candidate entity's name and aliases (2) Numbers of common words between

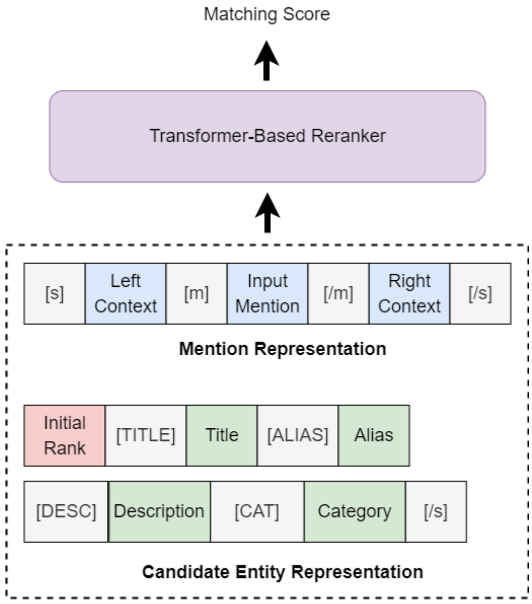

Figure 3: An illustration of the cross-attention reranker.

the mention's context and the entity's name and aliases (3) Numbers of common words between the mention's surface form and context and the entity's description and category.

We also use features that indicate the initial rankings of a candidate entity. For $C_d(m)$, each candidate is initially ranked by its corresponding prior probability (Eq. 1). For $C_e(m)$, each candidate is automatically assigned a score by ES. For a candidate c, let $r_d(c)$ indicate its rank in $C_d(m)$ (if $c \notin C_d(m)$ then $r_d(c) = \infty$). Similarly, let $r_e(c)$ indicate the rank of c in $C_e(m)$. The features to be fed to GBT are:

$$a_d(c) = \begin{cases} 1/r_d(c), & \text{if } c \in C_d(m) \\ 0, & \text{Otherwise} \end{cases}$$
$$a_e(c) = \begin{cases} 1/r_e(c), & \text{if } c \in C_e(m) \\ 0, & \text{Otherwise} \end{cases} \quad (3)$$

2.3 Cross-Attention Reranker

Overview We model the reranking problem as a binary classification problem and fine-tune a basic Transformer-based reranker for the task (Figure 3).

Input Representations The input to the reranker is the concatenation of the mention representation and the candidate entity representation (Figure 3). The mention representation is similar to the one described in Section 2.2.2. Each entity's representation consists of its initial rank (Section 2.2.3), title, alias, description, and category. To denote the initial rank, we define new tokens in the Transformer's vocabulary. For example, [rank1] represents rank 1, [rank2] indicates rank 2, and so on. If an entity has multiple aliases, we select the one with the highest string similarity to the input mention. The special tokens [TITLE], [ALIAS], [DESC], and [CAT] are used to indicate the locations of the entity's title, alias, description, and category (respectively). If any fields are missing, we simply exclude the missing fields and their corresponding special tokens from the entity representation.

Cross-Attention Reranker Given a mention m and a candidate entity e, the reranker computes a matching score $s_{m,e}$ indicating their relevance. The reranker consists of a Transformer-based encoder and a feedforward network:

$$\begin{aligned} \mathbf{h}_{m,e} &= \text{reduce}(T_{\text{cross}}(\tau_{m,e})) \\ s_{m,e} &= \text{FFNN}_s(\mathbf{h}_{m,e}) \end{aligned} \tag{4}$$

where $\tau_{m,e}$ is the concatenation of the mention representation and the entity representation. T_{cross} is a Transformer encoder (Devlin et al., 2019; Liu et al., 2019), and reduce(.) is a function that returns the final hidden state of the Transformer that corresponds to the first token (i.e., the [s] token). FFNN_s is a feedforward network. By taking $\tau_{m,e}$ as input, the Transformer encoder T_{cross} can have deep cross-attention between the mention's context and the entity's information from the KB.

In practice, a mention may not have any corresponding entity in the target KB. For predicting unlinkable mentions, we employ a simple thresholding method. If the score $s_{m,e_{top}}$ of the top-ranked candidate entity e_{top} is smaller than a threshold, we predict the mention m as unlinkable.

3 Experiments

3.1 Data and Experiments Setup

Target Knowledge Base In this work, we downloaded the complete Wikidata dump dated August 2021. Wikidata currently contains over 95 million items. However, many of these items are noisy or correspond to Wikimedia-internal administrative entities (i.e., not entities we want to retain). Therefore, we apply several heuristics to filter out unhelpful Wikidata items[2]. At the end, our final knowledge base contains 40,239,259 entities with English titles, substantially more than any other task settings we have found. We use this KB as the target KB for every EL experiment we conduct.

Evaluation Datasets (Wikidata) We use three manually annotated English datasets for evaluating EL over Wikidata: **RSS-500** (Röder et al., 2014), **ISTEX-1000** (Delpeuch, 2020), and **TweekiGold** (Harandizadeh and Singh, 2020). More details of these datasets are in the appendix. Some previous studies on EL over Wikidata also use other datasets such as LC-QuAD 2.0 (Dubey et al., 2019) and T-REx (ElSahar et al., 2018). However, these datasets were created semi-automatically or automatically instead of manually, thus less reliable.

Training Data We use Wikipedia anchor texts and their corresponding Wikidata entities as the supervision signals. We create a training set of 6 million paragraphs and a validation set of 1000 paragraphs. We refer to this dataset as **WikipediaEL**. We train our models (i.e., the generation model and the reranker) using this dataset. We do not fine-tune our models on any of the evaluation datasets.

Baselines For comparison, we choose a set of systems that were previously evaluated on the same evaluation datasets: AIDA (Hoffart et al., 2011), Babelfy (Moro et al., 2014), End-to-End (Kolitsas et al., 2018), OpenTapioca (Delpeuch, 2020), Tweeki (Harandizadeh and Singh, 2020), and KG Context (Mulang et al., 2020).

We also compare our approach to BLINK (Wu et al., 2020) and GENRE (Cao et al., 2021), SOTA methods for EL over Wikipedia or Wikipedia-derived KBs. We evaluated these methods by using their public code and model checkpoints. We implemented a converter to map each returned entity to its corresponding Wikidata entry.

CHOLAN (Kannan Ravi et al., 2021) is a related study, but its open-sourced code lacks running instructions[3]. Furthermore, the authors have not fully disclosed the splits of the dataset they used for evaluating EL over Wikidata. As a result, we did not directly compare CHOLAN and EPGEL.

Hyperparameters Our generation model is initialized with the BART model (bart-base) (Lewis et al., 2020b). For the reranker, we use RoBERTa (roberta-base) as the Transformer encoder (Liu et al., 2019). The maximum numbers of candidates are set to be 100, 100, and 50 for the dictionary-based, profiling-based, and hybrid approaches (respectively). More details are in the appendix.

[2] More details are in the appendix.

[3] https://tinyurl.com/el-cholan

Methods	RSS-500 (test)			ISTEX-1000 (test)			TweekiGold (test)			WikipediaEL (dev)		
	R@1	R@25	R@50	R@1	R@25	R@50	R@1	R@25	R@50	R@1	R@25	R@50
Simple Query	41.06	72.19	74.17	36.42	79.10	90.15	31.02	73.96	82.52	51.19	81.85	85.86
Wikipedia Dictionary	59.60	74.83	76.82	84.93	91.49	91.49	70.60	88.08	88.77	85.11	93.60	93.95
Profiling-based Query												
◆ Title	49.00	73.51	76.82	43.28	82.69	93.28	39.81	79.86	87.03	54.77	88.19	92.13
◆ Title + Desc	60.26	73.51	75.50	87.61	97.31	98.06	71.30	88.77	91.55	80.87	94.26	95.03
Hybrid Approach	**66.89**	**85.43**	**86.09**	**91.34**	**98.51**	**98.66**	**74.54**	**95.14**	**95.60**	**90.25**	**98.95**	**99.23**

Table 1: Overall candidate retrieval results. Recall scores (%) are shown.

Methods	RSS-500 (test)	ISTEX-1000 (test)	TweekiGold (test)	WikipediaEL (dev)
EPGEL	**76.4**	**92.7**	**69.3**	**92.3**
Effects of Candidate Retrieval Strategy				
◆ Simple Query	66.4	87.6	66.0	81.9
◆ Wikipedia Dictionary	71.2	91.6	68.8	89.8
◆ Profiling-Based Query [Title + Desc]	68.4	92.6	69.1	88.4
Previous Methods				
GENRE ⋆ (Cao et al., 2021)	68.2	88.4	62.4	86.3
BLINK ⋆ (Wu et al., 2020)	73.5	88.5	65.9	90.5
KG Context † (Mulang et al., 2020)	-	92.6	-	-
Tweeki (Harandizadeh and Singh, 2020)	-	-	65.0	-
OpenTapioca (Delpeuch, 2020)	46.5	91.6	29.1	-
End-to-End (Kolitsas et al., 2018)	-	-	49.4	-
Babelfy (Moro et al., 2014)	58.1	64.0	25.1	-
AIDA (Hoffart et al., 2011)	56.1	50.4	38.5	-

Table 2: Overall entity linking results. *InKB* micro F1 scores (%) are shown. The symbol "-" denotes results not reported in previous papers. The symbol "⋆" indicates systems that we evaluated by ourselves using their public code and model checkpoints. † KG Context is reported to have an F1 score of 92.6 on ISTEX-1000 (Mulang et al., 2020). However, the work uses a simplified setting where each mention's candidate pool is assumed to consist of the correct entity and only one negative entity. This setting is much easier and less practical than our setting.

3.2 Evaluation of Candidate Entity Retrieval

Table 1 compares the performance of various candidate retrieval approaches. [Simple Query] refers to querying ES using only the literal string of the input mention. This approach is quite similar to what is done in several previous studies on EL over Wikidata (Sakor et al., 2020; Kannan Ravi et al., 2021). As the target KB is huge, many entities have the same titles or aliases. Naively using only the surface form of the mention is not sufficient.

The performance of using a Wikipedia dictionary (Section 2.2.1) is much better than that of [Simple Query]. Although the dictionary-based approach also does not consider the context of the input mention, it computes the conditional probabilities using all anchor texts in the entire Wikipedia. In addition, most target entities in the evaluation datasets can still be found in Wikipedia. As such, this approach still performs reasonably well overall. However, note that for mentions whose linked entities are in Wikidata but not in Wikipedia, the recall score of the Wikipedia dictionary will always be 0.

For our profiling-based approach (Section 2.2.2), we experiment with two variants: (1) The entity profile is only the generated title (2) The entity profile consists of the generated title and the generated description. The latter achieves much better performance. It also achieves comparable or better scores than the Wikipedia dictionary most of the time.

Finally, we see that our profiling-based approach complements the dictionary-based approach. Our hybrid technique (Section 2.2.3) is highly effective, outperforming all other methods.

3.3 Overall Entity Linking Results

Table 2 shows the overall entity linking results. Our complete framework (i.e., EPGEL) uses the hybrid

Methods	P@1
Neural Cross-Lingual EL (Sil et al., 2018)	87.4
DeepType (Raiman and Raiman, 2018)	90.9
Neural Collective EL (Cao et al., 2018)	91.0
DEER (Gillick et al., 2019)	87.0
BLINK (Wu et al., 2020)	90.9
RELIC (Ling et al., 2020)	89.8
Attribute-sep. (Vyas and Ballesteros, 2021)	84.9
EPGEL	90.9

Table 3: In-KB accuracy scores (%) of different models on TACKBP-2010. Note that our Wikidata-based target KB is much larger than the ones used by previous studies (e.g., the TAC Reference KB).

candidate retrieval approach (Section 2.2.3) and the cross-attention reranker (Section 2.3). EPGEL outperforms a variety of SOTA techniques across all datasets. For example, EPGEL achieves better results than GENRE (Cao et al., 2021) on the tested datasets. GENRE is an autoregressive system that directly retrieves entities by generating the entity names conditioned on the context. In theory, GENRE does not require a candidate retrieval step to work. However, as detailed in the original paper (Cao et al., 2021), GENRE achieves the best performance when high-quality candidate lists are available. Therefore, having an effective candidate retrieval method can still be helpful even during this era of large language models.

Table 2 also shows the results of using different candidate retrieval strategies. There is a positive correlation between the candidate retrieval performance and the final EL performance. This is expected, as the recall from the candidate retrieval step provides an upper bound on the entire EL framework's recall. Also, even if EPGEL uses only the profiling-based approach (without relying on the Wikipedia dictionary), it can still achieve good results compared to the baselines.

3.4 Results on TACKBP-2010

Even though our focus is EL over Wikidata, we also use the TACKBP-2010 dataset (Ji et al., 2010) for evaluation since it is a standard dataset used by many previous studies. There are 1,020 annotated mention/entity pairs in total for evaluation. All the entities are from the TAC Reference KB, containing only 818,741 entities. To evaluate EPGEL, we use our large-scale Wikidata-based KB as the target KB. Also, we do not fine-tune EPGEL on the training set of TACKBP-2010. Overall, the performance of EPGEL is comparable to previous state-of-the-art systems (Table 3), even though EPGEL needs to map mentions to entities in a large-scale KB.

3.5 Qualitative Analysis

Table 4 shows some examples of our conditional generation model's predictions.

In the first example, as the model has seen the mention "Christmas truce" with similar context during training, the model generates the exact title and description for the target entity. In fact, using this accurate profile, ES already ranks the target entity in the top 1 even without using the reranker.

In the second example, the model has not come across the mention "Kevin Colbert" during training. However, because of the phrases "National Football League" and "general manager", the model infers that the mention refers to an "American football executive". The generated description is quite close to the actual description, "American football player and executive". This generated profile helps ES rank the target entity higher than the entity Q91675515 (a researcher named Kevin Colbert).

The last example presents a failure case of our generation model. The target entity is a baseball team, but the model incorrectly infers that the mention "Baltimore" refers to a city. We will discuss this failure case in more detail in next section. Nevertheless, if the hybrid approach is used, we can still recover from this error since the target entity is in the Wikipedia dictionary.

3.6 Remaining Challenges

In this section, we will discuss some major categories of the remaining errors made by EPGEL.

Generation model's popularity bias When encountering an input mention whose literal form has already appeared in the training set, the generation model sometimes ignores the context entirely and generates the most common entity profile for that literal form. In the last example in Table 4, the mention Baltimore refers to a sports team. However, our model mistakenly generates the most common profile for the mention (a city in Maryland). A possible approach to tackle the challenge is to randomly mask out the input mention during training. This would encourage the generation model to pay more attention to the surrounding context and not rely too much on the mention's literal form.

Need to optimize global coherence Entities within the same document are generally related;

Input Mention	Generated Profile	Target Entity
... They had an only son, Arthur, a British Army officer who played a leading role in the 1914 Christmas truce.	[Title] Christmas truce | [Description] unofficial cease fire in Western Front during World War I	Q163730
... and as a member of the National Football League. It also marked the 14th season under leadership of general manager Kevin Colbert and the seventh under head ...	[Label] Kevin Colbert | [Description] American football executive	Q6396037
... Baltimore beat Josh Beckett and the Red Sox 7-1 Tuesday night ...	[Title] Baltimore | [Description] Independent city in Maryland, United States	Q650816

Table 4: Example outputs from our conditional generation model.

however, our reranker disambiguates each mention independently. Therefore, it sometimes makes mistakes that can be easily avoided if the global coherence among entities is considered. For example, given the following tweet, *"Syracuse and Pitt in the #ACC ... its gonna be a long year for Maryland."*, EPGEL correctly infers that "Syracuse" and "Pitt" are basketball teams. However, for "Maryland", the reranker ranks a football team higher than the actual target entity (a basketball team). This shows that EPGEL may benefit from utilizing more global information for collective inference.

4 Related Work

4.1 Candidate Entity Retrieval

Dictionary-based techniques are the dominant approaches to candidate retrieval of many previous Wikipedia EL systems (Shen et al., 2012; Gattani et al., 2013; Shen et al., 2013; van Hulst et al., 2020). The structure of Wikipedia provides a set of useful features for building an offline name dictionary between various names and their possible mapped entities. For example, many previous studies build such name dictionaries by mining anchor texts of Wikipedia pages (Han et al., 2011; Phan et al., 2017; Zeng et al., 2018). Even though this approach is highly effective for EL over Wikipedia (Ganea and Hofmann, 2017), it is not directly applicable to Wikidata as previously discussed.

4.2 Entity Linking over Wikidata

Compared to Wikipedia, there are relatively fewer studies on EL over Wikidata (Möller et al., 2021). Recently, Cetoli et al. (2019) proposed a neural EL approach for Wikidata. The setting used in their work is that each mention comes with one correct entity candidate and one incorrect candidate. This setting is much less challenging and realistic than ours. Sakor et al. (2020) proposed Falcon 2.0, a rule-based system for entity and relation linking over Wikidata. Its candidate retrieval approach is to query ES using the literal string of the input mention. This method is much less effective than our profiling-based approach (Sec. 3.2). OpenTapioca is another attempt that performs EL over Wikidata by utilizing two main features: local compatibility and semantic similarity (Delpeuch, 2020). For the social media domain, Tweeki (Harandizadeh and Singh, 2020) is an unsupervised approach for linking entities in tweets to Wikidata. EPGEL outperforms both OpenTapioca and Tweeki (Sec. 3.3).

5 Conclusions and Future Work

This paper has proposed a novel profiling-based paradigm to candidate retrieval for EL. The technique is highly generalizable and complementary to the traditional dictionary-based approach, enabling the design of an effective hybrid candidate retrieval method. Together with a cross-attention reranker, our complete EL framework achieves strong performance on four public datasets. We plan to explore a broader range of properties and information about the target entity that can be extracted from the mention's context. For example, type-based features can be helpful for EL (Onoe and Durrett, 2020); as such, we aim to make our generation model generate the target entity's type. Also, in this work, we use a local model for candidate reranking. We plan to explore the use of a more global model for collective EL (Yang et al., 2018; Phan et al., 2019).

References

Oshin Agarwal and D. Bikel. 2020. Entity linking via dual and cross-attention encoders. *ArXiv*, abs/2004.03555.

Zahra Ashktorab, Christopher Brown, Manojit Nandi, and Aron Culotta. 2014. Tweedr: Mining twitter to inform disaster response. In *ISCRAM*.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. 2007. Dbpedia: A nucleus for a web of open data. In *ISWC/ASWC*.

Jan A. Botha, Zifei Shan, and Daniel Gillick. 2020. Entity Linking in 100 Languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7833–7845, Online. Association for Computational Linguistics.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *International Conference on Learning Representations*.

Yixin Cao, Lei Hou, Juan-Zi Li, and Zhiyuan Liu. 2018. Neural collective entity linking. In *COLING*.

A. Cetoli, Stefano Bragaglia, Andrew D. O'Harney, Marc Sloan, and Mohammad Akbari. 2019. A neural approach to entity linking on wikidata. In *ECIR*.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP*.

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*.

Antonin Delpeuch. 2020. Opentapioca: Lightweight entity linking for wikidata. *ArXiv*, abs/1904.09131.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Stephen Dill, Nadav Eiron, David Gibson, Daniel Gruhl, R. Guha, Anant Jhingran, Tapas Kanungo, Sridhar Rajagopalan, Andrew Tomkins, John A. Tomlin, and Jason Y. Zien. 2003. Semtag and seeker: Bootstrapping the semantic web via automated semantic annotation. In *Proceedings of the 12th International Conference on World Wide Web*, WWW '03, page 178–186, New York, NY, USA. Association for Computing Machinery.

Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin. 2010. Entity disambiguation for knowledge base population. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 277–285, Beijing, China. Coling 2010 Organizing Committee.

Mohnish Dubey, Debayan Banerjee, Abdelrahman Abdelkawi, and Jens Lehmann. 2019. Lc-quad 2.0: A large dataset for complex question answering over wikidata and dbpedia. In *SEMWEB*.

Hady ElSahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon S. Hare, Frédérique Laforest, and Elena Paslaru Bontas Simperl. 2018. T-rex: A large scale alignment of natural language with knowledge base triples. In *LREC*.

Zheng Fang, Yanan Cao, Ren Li, Zhenyu Zhang, Yanbing Liu, and Shi Wang. 2020. High quality candidate generation and sequential graph attention network for entity linking. In *Proceedings of The Web Conference 2020*, pages 640–650.

Jerome H. Friedman. 2001. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232.

Yi Fung, Christopher Thomas, Revanth Gangi Reddy, Sandeep Polisetty, Heng Ji, Shih-Fu Chang, Kathleen McKeown, Mohit Bansal, and Avi Sil. 2021. InfoSurgeon: Cross-media fine-grained information consistency checking for fake news detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1683–1698, Online. Association for Computational Linguistics.

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep joint entity disambiguation with local neural attention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2619–2629, Copenhagen, Denmark. Association for Computational Linguistics.

Abhishek Gattani, Digvijay S. Lamba, Nikesh Garera, Mitul Tiwari, Xiaoyong Chai, Sanjib Das, Sri Subramaniam, Anand Rajaraman, Venky Harinarayan, and AnHai Doan. 2013. Entity extraction, linking, classification, and tagging for social media: A wikipedia-based approach. *Proc. VLDB Endow.*, 6(11):1126–1137.

Johanna Geiß, Andreas Spitz, and Michael Gertz. 2017. Neckar: A named entity classifier for wikidata. In *GSCL*.

Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning dense representations for entity retrieval. *ArXiv*, abs/1909.10506.

Stephen Guo, Ming-Wei Chang, and Emre Kiciman. 2013. To link or not to link? a study on end-to-end tweet entity linking. In *Proceedings of the 2013 Conference of the North American Chapter of the*

Association for Computational Linguistics: Human Language Technologies, pages 1020–1030, Atlanta, Georgia. Association for Computational Linguistics.

Xianpei Han, Le Sun, and Jun Zhao. 2011. Collective entity linking in web text: a graph-based method. *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*.

Bahareh Harandizadeh and Sameer Singh. 2020. Tweeki: Linking named entities on Twitter to a knowledge graph. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 222–231, Online. Association for Computational Linguistics.

C. Harris, K. J. Millman, S. Walt, Ralf Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, Nathaniel J. Smith, R. Kern, Matti Picus, S. Hoyer, M. Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fern'andez del R'io, Mark Wiebe, P. Peterson, Pierre G'erard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, Christoph Gohlke, and T. E. Oliphant. 2020. Array programming with numpy. *Nature*, 585 7825:357–362.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Matthew A. Jaro. 1989. Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. *Journal of the American Statistical Association*, 84:414–420.

Heng Ji, Ralph Grishman, Hoa Trang Dang, Kira Griffitt, and Joe Ellis. 2010. Overview of the tac 2010 knowledge base population track. In *In Third Text Analysis Conference (TAC)*.

Manoj Prabhakar Kannan Ravi, Kuldeep Singh, Isaiah Onando Mulang', Saeedeh Shekarpour, Johannes Hoffart, and Jens Lehmann. 2021. CHOLAN: A modular approach for neural entity linking on Wikipedia and Wikidata. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 504–514, Online. Association for Computational Linguistics.

Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. 2018. End-to-end neural entity linking. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 519–529, Brussels, Belgium. Association for Computational Linguistics.

Vladimir I. Levenshtein. 1965. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics. Doklady*, 10:707–710.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020b. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *ArXiv*, abs/1910.13461.

Belinda Z. Li, Sewon Min, Srinivasan Iyer, Yashar Mehdad, and Wen-tau Yih. 2020. Efficient one-pass end-to-end entity linking for questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6433–6441, Online. Association for Computational Linguistics.

Jeffrey Ling, Nicholas FitzGerald, Zifei Shan, Livio Baldini Soares, Thibault Févry, David Weiss, and Tom Kwiatkowski. 2020. Learning cross-context entity representations from text. *ArXiv*, abs/2001.03765.

Xiao Ling, Sameer Singh, and Daniel S. Weld. 2015. Design challenges for entity linking. *Transactions of the Association for Computational Linguistics*, 3:315–328.

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*, abs/1907.11692.

Rada Mihalcea and Andras Csomai. 2007. Wikify!: linking documents to encyclopedic knowledge. In *CIKM '07*.

David N. Milne and Ian H. Witten. 2008. Learning to link with wikipedia. In *CIKM '08*.

Cedric Möller, Jens Lehmann, and Ricardo Usbeck. 2021. Survey on english entity linking on wikidata. *Semantic Web*.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity Linking meets Word Sense Disambiguation: a Unified Approach. *Transactions of the Association for Computational Linguistics*, 2:231–244.

Isaiah Onando Mulang, Kuldeep Singh, Chaitali Prabhu, Abhishek Nadgeri, Johannes Hoffart, and Jens Lehmann. 2020. Evaluating the impact of knowledge graph context on entity disambiguation models. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*.

Yasumasa Onoe and Greg Durrett. 2020. Fine-grained entity typing for domain independent entity linking. In *AAAI*.

Xiaoman Pan, Taylor Cassidy, Ulf Hermjakob, Heng Ji, and Kevin Knight. 2015. Unsupervised entity linking with abstract meaning representation. In *NAACL*.

Adam Paszke, S. Gross, Francisco Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, Alban Desmaison, Andreas Köpf, E. Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, B. Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*.

Minh C. Phan, Aixin Sun, Yi Tay, Jialong Han, and Chenliang Li. 2017. Neupl: Attention-based semantic matching and pair-linking for entity disambiguation. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*.

Minh C. Phan, Aixin Sun, Yi Tay, Jialong Han, and Chenliang Li. 2019. Pair-linking for collective entity disambiguation: Two could be better than all. *IEEE Transactions on Knowledge and Data Engineering*, 31:1383–1396.

Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *ArXiv*, abs/1910.10683.

Jonathan Raiman and Olivier Raiman. 2018. Deeptype: Multilingual entity linking by neural type system evolution. In *AAAI*.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and global algorithms for disambiguation to Wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1375–1384, Portland, Oregon, USA. Association for Computational Linguistics.

Miriam Redi, Martin Gerlach, Isaac Johnson, Jonathan Morgan, and Leila Zia. 2020. A taxonomy of knowledge gaps for wikimedia projects (second draft). *arXiv preprint arXiv:2008.12314*.

Michael Röder, Ricardo Usbeck, Sebastian Hellmann, Daniel Gerber, and Andreas Both. 2014. N³ - a collection of datasets for named entity recognition and disambiguation in the nlp interchange format. In *LREC*.

Ahmad Sakor, Kuldeep Singh, Anery Patel, and Maria-Esther Vidal. 2020. Falcon 2.0: An entity and relation linking tool over wikidata. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*.

Wei Shen, Jianyong Wang, and Jiawei Han. 2015. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27:443–460.

Wei Shen, Jianyong Wang, Ping Luo, and Min Wang. 2012. Linden: Linking named entities with knowledge base via semantic knowledge. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, page 449–458, New York, NY, USA. Association for Computing Machinery.

Wei Shen, Jianyong Wang, Ping Luo, and Min Wang. 2013. Linking named entities in tweets with knowledge base via user interest modeling. *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*.

Avirup Sil, Gourab Kundu, Radu Florian, and Wael Hamza. 2018. Neural cross-lingual entity linking. In *AAAI*.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *WWW '07*.

Johannes M van Hulst, Faegheh Hasibi, Koen Dercksen, Krisztian Balog, and Arjen P de Vries. 2020. Rel: An entity linker standing on the shoulders of giants. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2197–2200.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Yogarshi Vyas and Miguel Ballesteros. 2021. Linking entities to unseen knowledge bases with arbitrary schemas. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 834–844, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Yi Yang, Ozan Irsoy, and Kazi Shefaet Rahman. 2018. Collective entity disambiguation with structured gradient tree boosting. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 777–786, New Orleans, Louisiana. Association for Computational Linguistics.

Weixin Zeng, Jiuyang Tang, and Xiang Zhao. 2018. Entity linking on chinese microblogs via deep neural network. *IEEE Access*, 6:25908–25920.

Boliang Zhang, Ying Lin, Xiaoman Pan, Di Lu, Jonathan May, Kevin Knight, and Heng Ji. 2018. ELISA-EDL: A cross-lingual entity extraction, linking and localization system. In *Proc. The 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT2018)*.

Section A describes the datasets that we used for evaluation. Section B describes how we preprocessed the original Wikidata dump. Section C presents our reproducibility checklist. Section D describes how we construct an ES query from a generated profile. Finally, Section E discusses the potential risks of our work.

A Evaluation Datasets

We use three different English datasets (Möller et al., 2021) for evaluating the performance of EL over Wikidata:

- **RSS-500** (Röder et al., 2014) is a manually annotated dataset consisting of RSS-feeds (i.e., short formal documents) from major international newspapers. The target KB of the original version of RSS-500 is DBpedia. However, Delpeuch (2020) created a new version of the dataset for evaluating EL over Wikidata.
- **ISTEX-1000** (Delpeuch, 2020) is a dataset of 1,000 author affiliation strings extracted from scientific publications. It was manually annotated to align entity mentions to Wikidata.
- **TweekiGold** (Harandizadeh and Singh, 2020) is a manually annotated dataset for EL over tweets. It has 500 tweets for evaluation but does not have a separate training set.

For RSS-500, ISTEX-1000, and WikipediaEL, the setting is that the gold-standard entity mentions are already given as input, and the task is only to link the input mentions to the correct entities.

For TweekiGold, similar to the study that introduced the dataset (Harandizadeh and Singh, 2020), we do not assume that the mentions are provided. As such, for TweekiGold, we need to do both mention extraction and entity disambiguation. In this work, we simply use an off-the-shelf RoBERTa-based model from HuggingFace for mention extraction (roberta-base-finetuned-ner). Note that we do not fine-tune the mention extractor. In addition, when evaluating BLINK and GENRE on TweekiGold, we also use the same extractor to make the comparison fair.

For the TACKBP-2010 dataset (Ji et al., 2010), there are 1,020 annotated mention/entity pairs in total for evaluation. All the entities are from the TAC Reference KB, containing only 818,741 entities. However, to evaluate EPGEL, we use our large-scale Wikidata-based KB as the target KB.

RSS-500 and ISTEX-1000 can be downloaded from the Github repository of OpenTapioca (Delpeuch, 2020). And OpenTapioca is released under the Apache-2.0 license. TweekiGold is also released under the Apache-2.0 license. The TACKBP-2010 dataset can be downloaded from LDC's website. The license information can be found at `https://catalog.ldc.upenn.edu/LDC2018T16`. Our use of the datasets is consistent with their licenses.

Our work focuses on English entity linking. In addition, we randomly sampled about 10~20 examples for each dataset and then checked whether the examples contained any offensive content. Overall, we did not see any example that had offensive content.

B Wikidata Preprocessing

In this work, we use the complete Wikidata dump dated August 2021. Even though Wikidata currently contains over 95 million items, many of the items are unhelpful (i.e., not entities we want to retain). Therefore, we apply several heuristics to filter out unuseful Wikidata items. First, we remove all entities with no English titles (i.e., entities whose English titles are empty strings). Second, we remove entities that are a subclass (P279) or instance of (P31) the most common Wikimedia-internal administrative entities (Table 5). Finally, we remove entities whose English titles start with "Category:", "Template:", or "Project:".

C Reproducibility Checklist

In this section, we present the reproducibility information of the paper. We are planning to make the code publicly available after the paper is reviewed.

Wikidata ID	Label
Q4167836	Wikimedia category
Q24046192	Wikimedia category of stubs
Q20010800	Wikimedia user language category
Q11266439	Wikimedia template
Q11753321	Wikimedia navigational template
Q19842659	Wikimedia user language template
Q21528878	Wikimedia redirect page
Q17362920	Wikimedia duplicated page
Q14204246	Wikimedia project page
Q21025364	WikiProject
Q17442446	Wikimedia internal item
Q26267864	Wikimedia KML file
Q4663903	Wikimedia portal
Q15184295	Wikimedia module
Q13442814	Scholarly Article

Table 5: Wikidata identifiers used for filtering out items (adapted from (Botha et al., 2020; De Cao et al., 2021))

Implementation Dependencies Libraries Pytorch 1.9.1 (Paszke et al., 2019), Transformers 4.11.3 (Wolf et al., 2020), Numpy 1.19.5 (Harris et al., 2020), CUDA 11.2.

Computing Infrastructure The experiments were conducted on a server with Intel(R) Xeon(R) Gold 5120 CPU @ 2.20GHz and NVIDIA Tesla V100 GPUs. Each GPU's memory is 16G.

Datasets RSS-500 and ISTEX-1000 can be downloaded from `https://github.com/wetneb/opentapioca`. TweekiGold can be downloaded from `https://ucinlp.github.io/tweeki/`. The TACKBP-2010 dataset can be downloaded from `https://catalog.ldc.upenn.edu/LDC2018T16`.

Number of Model Parameters The number of parameters in the conditional generation model is about 140M. The number of parameters in the reranker is about 125M.

Hyperparameters For training the conditional generation model, the batch size is set to be 128, the number of epochs is set to be 3, and the base learning rate is set to be 5e-5. For training the reranker, the batch size is set to be 8 mentions per batch (each mention has at most 50 candidates), the number of epochs is set to be 5, and the base learning rate is 1e-05.

Expected Validation Performance The main paper has the results on the development set of WikipediaEL. We do not fine-tune our trained models on any of the evaluation datasets (i.e., RSS-500, ISTEX-1000, TweekiGold, and TACKBP-2010). For example, in Table 2, for EPGEL, we report the test results of the system with the best score on the development set of WikipediaEL.

D Elasticsearch Query Construction

We use the example shown in Figure 2 as the running example. In this case, the surface form of the input mention is "Bruins", the generated title is "UCLA Bruins men's football", and the generated description is "college football team of the University of California, Los Angeles". Then, the actual query to be fed to ES is shown in Figure 4. Intuitively, the query consists of three main parts:

1. The similarity between the *title* and *alias* fields and the **surface form**.

2. The similarity between the *title* and *alias* fields and the **generated title**.

3. The similarity between the *description* field and the **generated description**.

Note that to reduce the querying latency, we merged the *title* and *alias* fields of each entity into one single field named *title_and_aliases*. In other words, for each entity, its *title_and_aliases* field is an array of strings corresponding to the entity's title and its aliases (if any). The `match` keyword is the standard keyword in ES for invoking a full-text search over a field. We use the `term` keyword to increase the final matching score when an exact match exists between the *title_and_aliases* field and the surface form / the generated title. Overall, our ES query structure is quite basic and does not have many parameters.

E Potential Risks

Our EL system has several potential malicious use cases (e.g., disinformation, generating fake news,

```
{
  "query": {
    "bool": {
      "should": [
        {
          "match": {
            "title_and_aliases": {
              "query": "Bruins",
              "fuzziness": "AUTO"
            }
          }
        },
        {
          "term": {
            "title_and_aliases.raw": {
              "value": "Bruins",
              "boost": 2.0
            }
          }
        },
        {
          "match": {
            "title_and_aliases": {
              "query": "UCLA Bruins men's football",
              "fuzziness": "AUTO"
            }
          }
        },
        {
          "term": {
            "['title_and_aliases'].raw": {
              "value": "UCLA Bruins men's football",
              "boost": 2.0
            }
          }
        },
        {
          "match": {
            "description": {
              "query": "college football team of the University of California, Los Angeles",
              "fuzziness": "AUTO"
            }
          }
        }
      ]
    }
  }
}
```

(1) Surface Form

(2) Generated Title

(3) Generated Description

Figure 4: ES query for the example shown in Figure 2.

surveillance). For example, Fung et al. (2021) introduced a novel approach for fake news generation. The technique works by first taking a genuine news article, extracting a multimedia knowledge graph, and replacing or inserting salient nodes or edges in the graph. To build such a multimedia knowledge graph, the authors do use an EL system. Another example is that our EL system may be used as part of a malicious surveillance system (e.g., automatically tracking the locations of celebrities based on social media posts and online news).