ReFinED: An Efficient Zero-shot-capable Approach to End-to-End Entity Linking

Tom Ayoola, Shubhi Tyagi, Joseph Fisher, Christos Christodoulopoulos, Andrea Pierleoni
Amazon Alexa AI
Cambridge, UK
{tayoola, tshubhi, fshjos, chrchrs, apierleo}@amazon.com

Abstract

We introduce ReFinED, an efficient end-to-end entity linking model which uses fine-grained entity types and entity descriptions to perform linking. The model performs mention detection, fine-grained entity typing, and entity disambiguation for all mentions within a document in a single forward pass, making it more than 60 times faster than competitive existing approaches. ReFinED also surpasses state-of-the-art performance on standard entity linking datasets by an average of 3.7 F1. The model is capable of generalising to large-scale knowledge bases such as Wikidata (which has 15 times more entities than Wikipedia) and of zero-shot entity linking. The combination of speed, accuracy and scale makes ReFinED an effective and cost-efficient system for extracting entities from web-scale datasets, for which the model has been successfully deployed. Our code and pre-trained models are available at https://github.com/alexa/ReFinED.

1 Introduction

Entity linking (EL) is the task of recognising mentions of entities in unstructured text documents and linking them to the corresponding entities in a Knowledge Base (KB), such as Wikidata. EL is commonly a first stage in systems for question answering (Wang et al., 2021), automated KB population (Hoffmann et al., 2011), and relation extraction (Baldini Soares et al., 2019).

Currently, EL systems use deep learning methods to learn representations for entities and mentions (Ganea and Hofmann, 2017; Le and Titov, 2018). Initial techniques learned representations from text alone, which relied on entities appearing in similar contexts in the training data and meant models were only able to link mentions to entities that appeared in the training data. This is problematic both as KBs are continuously growing, and as it is infeasible to build an EL dataset containing all entities in a large KB (such as Wikidata with over 90 million entities). The largest public EL dataset is Wikipedia (using internal hyperlinks as labels), which covers just 3% of the entities in Wikidata.

Recent models addressed this problem by producing entity representations from a subset of KB information, e.g., entity descriptions (Wu et al., 2020; Logeswaran et al., 2019) or fine-grained entity types (Onoe and Durrett, 2020; Raiman and Raiman, 2018), allowing linking to entities not present in the training data or added to the KB after training; termed "zero-shot" in the EL literature.[1]

However, existing zero-shot-capable EL approaches are an order of magnitude more computationally expensive than non-zero-shot models (van Hulst et al., 2020) as they either require numerous entity types (Onoe and Durrett, 2020), multiple forward passes of a large-scale model to encode mentions and descriptions (Wu et al., 2020), or regeneration of the input text autoregressively (Cao et al., 2020). This makes large-scale processing expensive and thus makes it difficult to benefit from many advantages of zero-shot EL, e.g. the ability to keep up-to-date with new or updated KBs.

In this paper, we propose an efficient end-to-end zero-shot-capable EL model, ReFinED[2], which uses fine-grained entity types and entity descriptions to perform entity linking or entity disambiguation (ED; where entity mentions are given). We show that combining information from entity types and descriptions in a simple transformer-based encoder yields performance which is stronger than more complex architectures, surpassing state-of-the-art (SOTA) on 4 ED datasets and 5 EL datasets, and improving overall EL performance by 3.7 F1 points on average across 8 datasets. Importantly, ReFinED performs mention detection, fine-grained entity typing, and entity disambiguation for all men-

[1] Note the difference to "zero-shot" in the language-model literature, which refers to using no training data for the task.

[2] ReFinED stands for Representation and Fine-grained typing for Entity Disambiguation.

Proceedings of NAACL-HLT 2022: Industry Track Papers, pages 209 - 220
July 10-15, 2022 ©2022 Association for Computational Linguistics

tions within a document in a single forward pass, making it comparable in terms of inference speed to non-zero-shot models. It is 6 times faster than the most efficient zero-shot-capable baseline (which has 9 F1 points lower performance), and more than 60 times faster than more accurate systems (which come within 3 F1 points of ReFinED's average ED performance).

As opposed to previous EL models which primarily use Wikipedia as the target KB, ReFinED targets Wikidata, which enables it to link to 15 times more entities. This is because prior work uses information (e.g. titles, categories, first sentences) from Wikipedia to perform linking. It is unclear whether prior work could be expanded to Wikidata without a drop in performance because entity descriptions are less informative and there are fewer types per entity (Weikum et al., 2021).

The combination of high accuracy, scalability (with respect to KB size) and fast inference speed makes ReFinED a strong choice for a "web-scale"[3] EL system, in which cost scales approximately linearly with inference speed. We have successfully deployed ReFinED to production in a real-world application and share the lessons learned in Section 6.

Our contributions are as follows:

1. We build a simple and efficient zero-shot capable end-to-end EL model using entity descriptions and entity typing, which outperforms previous approaches on standard-EL datasets by 3.7 F1 points on average.
2. We demonstrate our model is more than 6 times faster than existing low-accuracy zero-shot capable systems, and 60 times faster than higher-accuracy systems, whilst also being capable of disambiguating against Wikidata-scale entity sets. The combination of accuracy, speed and scale makes the model suitable for web-scale information extraction.
3. We release our code and models.

2 Related work

Single architecture for entity linking EL consists of two main tasks, mention detection (MD) and ED. MD involves recognising mentions of entities in text, and ED assigns a KB entity to each mention. We follow (Kolitsas et al., 2018) in training a joint model for MD and ED.

Entity disambiguation with fine-grained entity typing In Onoe and Durrett (2020) and Raiman and Raiman (2018) ED is formulated as an entity typing problem. A fine-grained entity typing model is trained on a distantly-supervised dataset consisting of over 10k types derived from Wikipedia categories (e.g. movies released in a specific year). The entity typing model is then used to link entities. We extend their approach to Wikidata, by using a subset of Wikidata triples for providing types instead of Wikipedia categories.

Entity disambiguation with entity descriptions Several recent works have used entity descriptions for ED (Wu et al., 2020; Logeswaran et al., 2019). Typically, descriptions are sourced from Wikipedia by joining the entity's title with the first sentence of the Wikipedia article. Entities are ranked by concatenating mention context and entity description, then passing each mention-entity pair to a cross-encoder. Wu et al. (2020) shows a cross-encoder outperforms a bi-encoder, with the latter missing many fine-grained interactions between context and description. In our work, we find that a bi-encoder is sufficient to achieve SOTA performance when combined with fine-grained entity typing, and generalise the approach from Wikipedia (6M entities) to Wikidata (90M entities).[4]

3 Proposed method

3.1 Task Formulation

Given a KB[5] with a set of entities $E = \{e_1, e_2, \ldots, e_{|E|}\}$, let $X = [x_1, x_2, \ldots, x_{|X|}]$ be a sequence of tokens in the document, and $M = \{m_1, m_2, ...m_{|M|}\}$ be a set of entity mentions. The goal of ED is to create a function $\mathcal{M} : M \rightarrow E$ which assigns each mention the correct entity label. In EL, both the mention spans and entity labels need to be predicted. We only consider mentions with a valid gold entity in the KB during evaluation.

3.2 Overview

We propose an end-to-end EL model which is jointly optimised for mention detection, fine-grained entity typing, and entity disambiguation for all mentions within a document in a single forward pass. In this section, we describe the components of our model, depicted in Figure 1.

[3] We refer to corpora with more than 1 billion documents as "web-scale".

[4] We replace Wikipedia titles with Wikidata labels, and Wikipedia sentences with Wikidata entity descriptions.

[5] We assume entities in the KB have a textual description and a collection of facts.

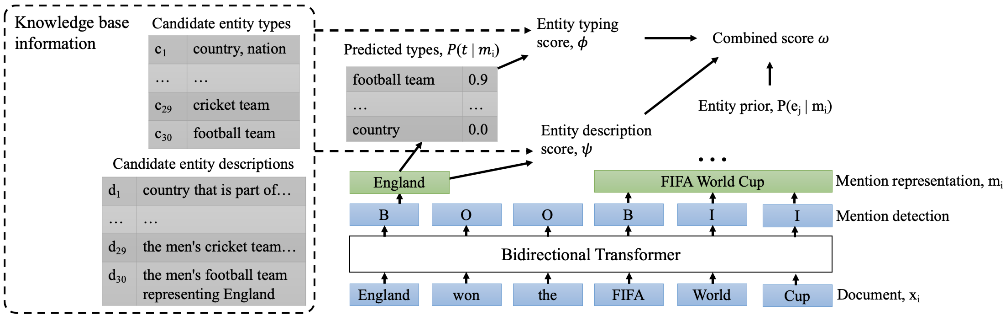

Figure 1: Our model architecture shown for a document with two mentions, *England* and *FIFA World Cup*. The model performs mention detection, entity typing, and entity disambiguation for all mentions in a single pass.

3.3 Context representation

We encode the tokens x_i in the input text document using a Transformer model. We use the contextualised token embeddings from the final layer, denoted as $\mathbf{h_i}$ for the token x_i.[6]

3.4 Mention detection

Entity linking requires entity mentions to be predicted. We encode mentions using the BIO tagging format (Ramshaw and Marcus, 1995) with 3 labels which indicate whether a token is at the beginning, inside of, or outside of a mention. We train a linear layer to perform token classification from the contextualised token embeddings $\mathbf{h_i}$ using cross-entropy loss $\mathcal{L}_m$ with respect to the gold token labels.

3.5 Mention representation

A fixed-length embedding $\mathbf{m_i}$ for each mention m_i is obtained by average pooling the contextualised tokens embeddings of the mention. All mentions M in a document X are encoded in a single forward pass, which improves efficiency relative to previous work that require a forward-pass for each mention (Wu et al., 2020; Orr et al., 2021).

3.6 Entity typing score ϕ

Given a fixed set of types $t \in T$ from a KB, where t is a relation-object pair (r, o) (e.g. (instance of, song)), we predict an independent probability for each type t for each mention by applying a linear layer f_1 followed by a sigmoid activation to the mention embedding $\mathbf{m_i}$. To score mention-entity pairs using predicted types, we calculate the Euclidean distance (L2 norm) between predicted types and the candidate entity's types $\mathbf{c_j}$ binary vector[7]:

$$\phi(e_j, m_i) = \|\sigma(f_1(\mathbf{m_i})) - \mathbf{c_j}\|_2 \qquad (1)$$

We follow Onoe and Durrett (2020) by training the entity typing module on distantly-supervised type labels from the gold entity using binary cross-entropy loss $\mathcal{L}_t$. See Appendix A for details on the choice of types T.

3.7 Entity description score ψ

We use a bi-encoder architecture similar to the work of Wu et al. (2020) but modified to encode all mentions m_i in a document simultaneously (as explained in Section 3.5). We represent KB entities as:

[CLS] label [SEP] description [SEP]

where label and description are the tokens of the entity label and entity description in the KB. We use a separate Transformer model (trained jointly with our mention transformer) to encode the representation of KB entities e_j into fixed dimension vectors (description embeddings) $\mathbf{d_j}$ by taking final layer embedding for the [CLS] token. We apply linear layers f_2 and f_3 to the mention embeddings $\mathbf{m_i}$ and entity description embeddings $\mathbf{d_j}$ respectively to project them to a shared vector space. We calculate the dot product between the two projected embeddings to compute the entity scores:

$$\psi(e_j, m_i) = f_2(\mathbf{m_i}) \cdot f_3(\mathbf{d_j}) \qquad (2)$$

We train this module using cross-entropy loss $\mathcal{L}_d$, with respect to gold entity label.

3.8 Combined score ω

We compute a combined score ω by applying a linear layer (with output dimension 1) f_4 on top

[6] We use bold letters for vectors throughout our paper, and treat m_i and $\mathbf{m_i}$ as different terms.

[7] We use 1 to indicate the presence of an entity type and 0 the absence of an entity type for our binary vector. Note that a single entity can have multiple entity types.

of the concatenation of entity typing score, entity description score, and a global entity prior $\hat{P}(e|m)$. The global entity prior is obtained from a corpus (Hoffart et al., 2011) or a popularity metric (Diefenbach and Thalhammer, 2018). We include $\hat{P}(e|m)$ to improve results for cases where context is limited (e.g. short question text). In addition, we add a special candidate for the NIL entity with an unnormalised score of 0, which indicates none of the candidate entities are correct.

$$\omega(e_j, m_i) = f_4(\psi(\mathbf{e_j}, \mathbf{m_i}); \phi(\mathbf{e_j}, \mathbf{m_i}); \hat{P}(\mathbf{e_j}|\mathbf{m_i})) \tag{3}$$

We train this module using cross-entropy loss $\mathcal{L}_c$ with respect to the gold entity label.

3.9 Optimisation and inference

We optimise the model using a weighted sum of the module-specific losses with fixed weights, which are tunable hyperparameters. At training time, we use the provided mention spans instead of the predicted mention spans and train mention detection alongside the other tasks:

$$\mathcal{L} = \lambda_1 \mathcal{L}_m + \lambda_2 \mathcal{L}_t + \lambda_3 \mathcal{L}_d + \lambda_4 \mathcal{L}_c \tag{4}$$

For EL inference, we use the predicted mention spans and take the KB entity (or NIL) with the highest combined score. For ED inference, we use the provided gold mention spans.

3.10 Zero-shot ED

Our proposed method is able to link to zero-shot (unseen during training) entities because it scores entities based on types and descriptions. New entities can be introduced by updating entity lookups.

4 Experiments

4.1 Entity disambiguation

Non-zero-shot ED We evaluate our model on the ED task using the same experimental setting as previous work (Ganea and Hofmann, 2017; Le and Titov, 2018; Cao et al., 2020). We pretrain on Wikipedia, then use the AIDA-CoNLL dataset (Hoffart et al., 2011) to fine-tune and evaluate. We measure out-of-domain performance on the datasets MSNBC (Cucerzan, 2007), AQUAINT (Milne and Witten, 2008), ACE2004 (Ratinov et al., 2011), WNED-CWEB (CWEB) (Gabrilovich et al., 2013) and WNED-WIKI (WIKI) (Guo and Barbosa, 2018). We report *InKB* micro-F1 (Röder et al., 2018). We also evaluate on AIDA-CoNLL using the candidate list generated by Pershina et al. (2015), known as PPRforNED, for the sake of comparison with previous SOTA results.

Zero-shot ED To compare our method to previous work, we measure zero-shot ED performance using the WikiLinksNED Unseen Mentions dataset (Eshel et al., 2017; Onoe and Durrett, 2020). The dataset contains a diverse set of ambiguous entities spanning multiple domains. We train our model on the provided training data and evaluate accuracy on the test set for seen and unseen (zero-shot) entities.

4.2 Entity linking

Non-zero-shot EL Following previous work (Kolitsas et al., 2018; Cao et al., 2020), we use the GERBIL platform (Röder et al., 2018) to evaluate EL. We evaluate *InKB* micro-F1 with strong matching (predictions must match exactly the gold mention boundaries). Similarly to the non-zero-shot ED experiment, we pretrain on Wikipedia, then use the AIDA-CoNLL dataset for fine-tuning and evaluation. For out-of-domain performance evaluation we use MSNBC (Cucerzan, 2007), OKE-2015, OKE-2016 (Nuzzolese et al., 2015), N3-Reuters-128 (R128), N3-RSS-500 (Röder et al., 2014), Derczynski (Derczynski et al., 2015), KORE50 (Hoffart et al., 2012).

4.3 Inference speed

We compare the computational efficiency of our model to three high-performing EL systems (Wu et al., 2020; Cao et al., 2020; Orr et al., 2021) for which code is available. We benchmark both modes of BLINK (Wu et al., 2020); the bi-encoder (encodes mention and entities independently) and the more accurate cross-encoder (encodes mention and entities jointly).[8] We measure the time to perform end-to-end EL inference on the AIDA-CoNLL test dataset using a single V100 GPU. The dataset consists of 231 documents and 4464 mentions.

4.4 Training details

Candidate generation We follow Le and Titov (2018) by selecting the top-30 candidate entities using entity priors.[9] For training, we only keep 5 candidates, 1 gold candidate, 2 candidates with the highest $\hat{p}(e_j|m_i)$ and 2 random candidates. When the gold entity is not in the candidate list during training, we use NIL as the correct label.

[8]We use a max context length of 128 tokens and precomputed entity embeddings for the bi-encoder. For the cross-encoder, we use max context length of 32 tokens.

[9]Derived from Wikipedia hyperlink count statistics, YAGO, a large Web corpus and Wikidata aliases.

	Method	AIDA	MSNBC*	AQUAINT*	ACE2004*	CWEB*	WIKI*	Avg.
	Yang et al. (2018)	93.0	92.6	89.9	88.5	81.8	79.2	87.5
	Yang et al. (2019)	93.7	93.8	88.3	90.1	75.6	78.8	86.7
	Fang et al. (2019)	**94.3**	92.8	87.5	91.2	78.5	82.8	87.9
	Wu et al. (2020)†	86.7	90.3	88.9	88.7	**82.6**	86.1	87.2
	Cao et al. (2020)	93.3	94.3	89.9	90.1	77.3	87.4	88.7
	Orr et al. (2021)**	80.9	80.5	74.2	83.6	70.2	76.2	77.6
	ReFinED (Wikipedia)	87.5	**94.4**	**91.8**	**91.6**	77.8	**88.7**	88.6
	ReFinED (fine-tuned)	93.9	94.1	90.8	90.8	79.4	87.4	**89.4**
Ablations	w/o entity priors (Wikipedia)	86.3	93.7	86.0	92.8	76.0	88.3	87.2
	w/o entity types (Wikipedia)	82.2	92.6	91.1	90.1	76.5	87.0	86.6
	w/o descriptions (Wikipedia)	85.7	93.9	89.5	91.2	76.1	84.3	86.8
	w/o pretraining (fine-tuned)	88.2	92.3	86.8	90.6	75.1	74.5	84.6

Table 1: ED InKB micro F1 scores on in-domain and out-of-domain test sets. The best value in **bold** and second best is underlined. †Normalised accuracy is reported. *Out-of-domain datasets. **Result obtained using code released by authors.

Wikipedia pretraining We use Wikidata as our KB (i.e. for entity types and descriptions). To make comparisons reliable, we restrict to the set of entities in English Wikipedia (total of 6.2M). We build a training dataset from the 2021-02-01 dump of Wikipedia and Wikidata and use hyperlinks as entity labels. To increase entity label coverage, we add weak labels to mentions of the article entity (Orr et al., 2021; Broscheit, 2019; Cao et al., 2020).[10] The dataset consists of approximately 100M mention-entity pairs. We use entity labels to generate entity type labels, as in Onoe and Durrett (2020). In addition, we follow Févry et al. (2020) by adding mention labels to unlinked mentions using a named entity recogniser to provide additional mention detection signal.

Model details We divide the documents into chunks of 300 tokens and subsample 40 mentions per chunk during pretraining. The model is trained for 2 epochs on Wikipedia and the transformers are initialised with RoBERTa (Liu et al., 2019) base weights. The description transformer has 2 layers. BERT-style masking (Devlin et al., 2019) is applied to mentions during pretraining. During fine-tuning and evaluation, we increase the sequence length to 512 and set the maximum candidate entities to 30.

5 Results

5.1 Entity disambiguation

Non-zero-shot ED We report *InKB* micro-F1 (with and without fine-tuning on AIDA) and compare it with SOTA ED models in Table 1. Our model performs strongly across all datasets, surpassing the previous average F1 across the 6 datasets by 0.7 F1 points. We observe the model achieves SOTA performance on 4 out of the 6 datasets without fine-tuning, suggesting it is able to learn patterns from Wikipedia that transfer well to other domains. Nonetheless, fine-tuning on the AIDA-CoNLL dataset leads to a substantial improvement (+6.4 F1 points) which can be attributed to the model learning peculiarities of the dataset (e.g. cricket score tables).

The ablations in Table 1 show entity types and entity descriptions are complementary (+2.0 F1 points when combined). This is explained by increased robustness to partially missing entity information (e.g. KB entities without descriptions) and different knowledge being expressed. Entity priors are useful but contribute less than other components of our combined score (Section 3.8). Without priors, F1 falls by 5.0 points on AQUAINT and increases by 1.2 points on ACE2004, which is expected as AQUAINT contains a high proportion of popular entities, and ACE2004 more rare entities. Pretraining has the largest impact on ED performance, particularly on datasets such as WIKI (+12.0 F1) derived from encyclopedia text.

Method	AIDA
Onoe and Durrett (2020)	85.9
Raiman and Raiman (2018)	94.9
Orr et al. (2021)	96.8
ReFinED (Wikipedia)	89.1
ReFinED (fine-tuned)	**97.1**

Table 2: ED accuracy on AIDA-CoNLL using PPRForNED candidates.

Table 2 shows accuracy on the AIDA-CoNLL dataset when we use PPRforNED candidates. ReFinED outperforms purely entity typing approaches

[10]We add weak labels by using simple heuristics such as matching mentions to the page's title.

Method	AIDA	MSNBC*	DER*	K50*	R128*	R500*	OKE15*	OKE16*	Avg.
Hoffart et al. (2011)	72.8	65.1	32.6	55.4	46.4	**42.4**	63.1	0.0	47.2
Kolitsas et al. (2018)	82.4	<u>72.4</u>	34.1	35.2	50.3	38.2	61.9	52.7	53.4
van Hulst et al. (2020)	80.5	<u>72.4</u>	41.1	50.7	49.9	35.0	63.1	58.3	56.4
Cao et al. (2020)	<u>83.7</u>	**73.7**	**54.1**	60.7	46.7	40.3	56.0	50.0	58.2
ReFinED (Wikipedia)	77.8	70.0	49.0	**65.9**	<u>52.6</u>	40.1	**65.0**	**59.5**	<u>60.0</u>
ReFinED (fine-tuned)	**84.0**	71.8	<u>50.7</u>	<u>64.7</u>	**58.1**	<u>42.0</u>	<u>64.4</u>	<u>59.1</u>	**61.9**

Table 3: EL InKB micro F1 scores on in-domain and out-of-domain test sets reported by Gerbil. The best value in **bold** and second best is <u>underlined</u>. *Out-of-domain datasets.

(Raiman and Raiman, 2018; Onoe and Durrett, 2020) by a margin of +2.2% accuracy, due to the addition of entity descriptions.

Zero-shot ED In Table 4, we report ED accuracy on the WikiLinksNED Unseen Mentions test set for seen and unseen entities. Our model outperforms the baseline by 3.0 F1, with, surprisingly, 6.6% higher accuracy for unseen than for seen entities. We find this is partly due to higher top 30 candidate recall for the unseen entity subset (95.0% compared to 91.1% for the seen entity subset) and also because our mention masking strategy reduces the reliance of entities appearing in the training data with similar surface forms. Moreover, ReFinED uses entity types and descriptions to link entities instead of relying on entity memorisation, which means the number of training examples for a given entity will not necessarily correlate with performance. The number of similar entities in the training dataset and the ambiguity of the test examples (Provatorova et al., 2021) will likely have more significant influence on performance.

Method	Seen	Unseen	Total
Cao et al. (2020)	64.3	63.2	63.5
ReFinED	61.6	68.2	66.5

Table 4: ED accuracy on WikiLinksNED Unseen Mentions test.

5.2 Entity linking

EL results are shown in Table 3. ReFinED outperforms other models on all but 3 datasets, often by a considerable F1 point margin (e.g. 7.8 on N3-Reuters-128 and 4.0 on KORE50) and improves the average across all 8 datasets by 3.7 F1 points. EL improves as ED and mention detection can generalise to different datasets due to the model being pretrained on Wikipedia hyperlinks as opposed to only AIDA-CoNLL. We also report results on the ISTEX and WebQSP datasets in Appendix C.

5.3 Inference speed

Table 5 shows the time taken to run inference on the AIDA-CoNLL test dataset, alongside the average ED performance. ReFinED is 6 times faster than the BLINK (Wu et al., 2020) bi-encoder, which also has an average F1 which is 9 points lower. Compared to the higher accuracy systems, ReFinED is 60 times faster than the BLINK cross-encoder, and 140 times faster than the autoregressive approach of Cao et al. (2020). This is because ReFinED uses a single forward pass to jointly encode all mentions and candidate KB entities in the document (512 token chunk), and hence requires ≈ 231 forward passes for the full dataset. The bi-encoder model requires ≈ 4464 forward passes as mentions are encoded individually, and the cross-encoder baseline requires ≈ 90k forward passes as each mention is encoded with each candidate. The autoregressive approach suffers from high computational cost due to the deep decoder, which generates a single token at a time. Also, all baselines require a separate model for MD whereas ReFinED performs end-to-end EL using a single model, which improves efficiency and simplifies model deployment.

Method	Time taken (s)	Avg. ED F1
Cao et al. (2020)	2100	88.7
Wu et al. (2020) bi-encoder	93	80.4
Wu et al. (2020) cross-encoder	917	87.2
Orr et al. (2021)	438	77.6
ReFinED	**15**	**89.4**

Table 5: Time taken in seconds for EL inference on AIDA-CoNLL test dataset.

6 Deployment Details

We have successfully deployed the ReFinED EL model in a real-world application, the aim of which is to populate a KB by extracting facts from unstructured text found on web pages with high precision. The application requires running ReFinED on a billion web pages (in which we link 25 billion mentions) multiple times per year. The scale of this deployment highlighted a number of learning points.

Firstly, the entity linking model must be computationally efficient. The inference speed of ReFinED allows the processing of the billion web pages in 27k machine hours (2 days using 500 instances), on machines with a single T4 GPU. Given availability of cloud compute, the cost of processing the same documents with the models evaluated in Section 5.3 would scale approximately linearly with their inference speeds. That is, the BLINK bi-encoder would require 3000 instances for 2 days, or 500 instances for 12 days, implying a roughly 6-fold increase in cost.

Secondly, the scale of the number of pages also brings with it diversity of domains, meaning the model benefits from linking to a large catalogue of entities (over 90 million) - including zero-shot entities.

Thirdly, we observed that deploying an end-to-end self-contained EL model is easier to horizontally scale and has a lower operational cost than deploying multiple systems for each subcomponent (such as candidate generation).

Finally, in real-world data, unlike in ED datasets, there are a large number of cases where the correct entity does not exist in the KB. This meant that we had to train the model on examples where the correct entity was not in the candidate list to reduce overprediction.

7 Conclusion

We propose a scalable end-to-end EL model which uses entity types and entity descriptions to perform linking. Our model achieves SOTA results for both ED (+0.7 F1 points on average across 6 datasets) and EL (+3.7 F1 points on average across 8 datasets) while being 60 times faster than comparatively accurate baselines. We demonstrate our approach scales well to a KB (Wikidata) 15 times larger than Wikipedia while maintaining competitive performance. The combination of accuracy, speed and scale means the system is capable of being deployed to extract entities from web-scale datasets with higher accuracy and an order of magnitude lower cost than existing approaches.

Acknowledgements

The authors would like to thank Clara Vania, Grace Lee, and Amir Saffari for helpful discussions and feedback. We also thank the anonymous reviewers for valuable comments that improved the quality of the paper.

References

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.

Samuel Broscheit. 2019. Investigating entity knowledge in BERT with simple neural end-to-end entity linking. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 677–685, Hong Kong, China. Association for Computational Linguistics.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and F. Petroni. 2020. Autoregressive entity retrieval. *ArXiv*, abs/2010.00904.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 708–716, Prague, Czech Republic. Association for Computational Linguistics.

Antonin Delpeuch. 2020. Opentapioca: Lightweight entity linking for wikidata.

Leon Derczynski, Diana Maynard, Giuseppe Rizzo, Marieke van Erp, Niraj Aswani, Raphaël Troncy, and Kalina Bontcheva. 2015. Analysis of named entity recognition and linking for tweets. *Information Processing and Management, Volume 51, N°2, March 2015, Elsevier*. © Elsevier. Personal use of this material is permitted. The definitive version of this paper was published in Information Processing and Management, Volume 51, N°2, March 2015, Elsevier and is available at : http://dx.doi.org/10.1016/j.ipm.2014.10.006.

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.

Dennis Diefenbach and Andreas Thalhammer. 2018. PageRank and Generic Entity Summarization for RDF Knowledge Bases. In *European Semantic Web Conference , ESWC 2018,*, The Semantic Web 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings, Heraklion, Greece.

Yotam Eshel, Noam Cohen, Kira Radinsky, Shaul Markovitch, Ikuya Yamada, and Omer Levy. 2017. Named entity disambiguation for noisy text. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 58–68, Vancouver, Canada. Association for Computational Linguistics.

Zheng Fang, Yanan Cao, Qian Li, Dongjie Zhang, Zhenyu Zhang, and Yanbing Liu. 2019. Joint entity linking with deep reinforcement learning. In *The World Wide Web Conference*, WWW '19, page 438–447, New York, NY, USA. Association for Computing Machinery.

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

Thibault Févry, Nicholas FitzGerald, Livio Baldini Soares, and T. Kwiatkowski. 2020. Empirical evaluation of pretraining strategies for supervised entity linking. *ArXiv*, abs/2005.14253.

Evgeniy Gabrilovich, Michael Ringgaard, and Amarnag Subramanya. 2013. Facc1: Freebase annotation of clueweb corpora. Preprint.

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.

Zhaochen Guo and Denilson Barbosa. 2018. Robust named entity disambiguation with random walks. *Semantic Web*, Preprint(Preprint):1–21.

Johannes Hoffart, Stephan Seufert, Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. 2012. Kore: Keyphrase overlap relatedness for entity disambiguation. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, page 545–554, New York, NY, USA. Association for Computing Machinery.

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.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 541–550, Portland, Oregon, USA. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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.

Phong Le and Ivan Titov. 2018. Improving entity linking by modeling latent relations between mentions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1595–1604, Melbourne, Australia. Association for Computational Linguistics.

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.

Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

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.

David Milne and Ian H. Witten. 2008. Learning to link with wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, page 509–518, New York, NY, USA. Association for Computing Machinery.

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*.

Andrea Giovanni Nuzzolese, Anna Lisa Gentile, Valentina Presutti, Aldo Gangemi, Darío Garigliotti, and Roberto Navigli. 2015. Open knowledge extraction challenge. In *Semantic Web Evaluation Challenges*, pages 3–15, Cham. Springer International Publishing.

Yasumasa Onoe and Greg Durrett. 2020. Fine-grained entity typing for domain independent entity linking. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8576–8583. AAAI Press.

L. Orr, Megan Leszczynski, Simran Arora, Sen Wu, Neel Guha, Xiao Ling, and C. Ré. 2021. Bootleg: Chasing the tail with self-supervised named entity disambiguation. *ArXiv*, abs/2010.10363.

Maria Pershina, Yifan He, and Ralph Grishman. 2015. Personalized page rank for named entity disambiguation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 238–243, Denver, Colorado. Association for Computational Linguistics.

Vera Provatorova, Samarth Bhargav, Svitlana Vakulenko, and Evangelos Kanoulas. 2021. Robustness evaluation of entity disambiguation using prior probes: the case of entity overshadowing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10501–10510, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*.

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.

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

Michael Röder, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. 2018. GERBIL - benchmarking named entity recognition and linking consistently. *Semantic Web*, 9(5):605–625.

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*, SIGIR '20, page 2197–2200, New York, NY, USA. Association for Computing Machinery.

Zhiguo Wang, Patrick Ng, Ramesh Nallapati, and Bing Xiang. 2021. Retrieval, re-ranking and multi-task learning for knowledge-base question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 347–357, Online. Association for Computational Linguistics.

Gerhard Weikum, Luna Dong, Simon Razniewski, and Fabian Suchanek. 2021. Machine knowledge: Creation and curation of comprehensive knowledge bases.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

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.

Xiyuan Yang, Xiaotao Gu, Sheng Lin, Siliang Tang, Yueting Zhuang, Fei Wu, Zhigang Chen, Guoping Hu, and Xiang Ren. 2019. Learning dynamic context augmentation for global entity linking. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 271–281, Hong Kong, China. 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.

A Entity Type Selection

Our entity types are formed from Wikidata relation-object pairs and relation-object pairs inferred from the Wikidata subclass hierarchy (for example, (instance of, organisation) can be inferred from (instance of, business)). We only consider types with the following relations: instance of, occupation, country, sport. We select types by iteratively adding types that separate (assuming an oracle type classifier) the gold entity from negative candidates for the most examples in our Wikipedia training dataset.

Type information stored in KBs often varies in granularity between entities (e.g. some capital city entities have the type capital city and others only city), adversely affecting training signal. To mitigate this, we use the class hierarchy to add parent types to entities. This injects class hierarchy information into the model, enabling type granularity to depend on context.

B Training Details

We use the Hugging Face implementation of RoBERTa (Wolf et al., 2019) and optimise our model using Adam (Kingma and Ba, 2015) with a linear learning rate schedule. We ignore the loss from mentions where the gold entity is not in the candidate set. The named-entity recogniser, used to preprocess our Wikipedia training dataset, is a RoBERTa token classification model trained on the AIDA-CoNLL dataset mention boundaries. We add weak entity labels for mentions that match the page's title (or surname for Wikipedia pages about people). We present our main hyperparameters in Table 6. Due to the high computational cost of training the model, we did not conduct an extensive hyperparameter search. Training on Wikipedia took approximately 48 hours on a single machine with 4 V100 GPUs. The model has approximately 154M parameters (123 million in the roberta-base architecture, and 31M for the additional description encoder and output layers).

C Additional results

Wikidata ED experimental setup To measure ED performance on non-Wikipedia entities, we expand our entity set to Wikidata (which has over 90M entities) and evaluate our model on the ISTEX test dataset (Delpeuch, 2020). We add labels and aliases from Wikidata for candidate generation and remove entity priors from our entity scoring (Section 3.8).

Hyperparameter	**Value**
learning rate	3e-5
batch size	64
max sequence length	300
dropout	0.05
description embeddings dim.	300
# training steps	1M
# candidates	30
# entity types	1400
mention transformer init.	roberta-base
# mention encoder layers	12
description transformer init.	roberta-base
# description encoder layers	2
# description tokens	32
$\lambda_1, \lambda_2, \lambda_3, \lambda_4$	(0.01, 1, 0.01, 1)
mention mask prob.	0.7

Table 6: Our model hyperparameters

Wikidata ED results We evaluate ED performance on the ISTEX dataset (which targets Wikidata). Our model outperforms Delpeuch (2020) (92.1 vs 87.0 micro F1) which uses hand-crafted features specifically designed for linking Wikidata entities. This shows that our approach scales to Wikidata and generalises well when there is increased mention ambiguity. Our model performs 0.5 F1 points below the SOTA Mulang' et al. (2020) (92.6 vs 92.1 micro F1) which is likely due to differing candidate entity generation methods.

Entity Linking performance on questions We report results on the WebQSP dataset in Table 7, which shows EL performance on questions. Our model has similar performance to ELQ, which is SOTA on WebQSP and is optimised for questions. Our model is faster than all baselines which can be attributed to using an end-to-end EL model, restricting ED predictions to the predicted mentions only, and using a smaller model (compared to ELQ which uses BERT-large (Devlin et al., 2019)).

Method	**WebQSP**	**#Q/s**
TAGME (Ferragina and Scaiella, 2012)	36.1	2.39
BLINK (Wu et al., 2020) (Wikipedia)	80.8	0.80
ELQ (Li et al., 2020) (Wikipedia)	83.9	1.56
ELQ (Li et al., 2020) (fine-tuned)	89.0	1.56
ReFinED (Wikipedia)	84.1	**2.78**
ReFinED (fine-tuned)	**89.1**	**2.78**

Table 7: Entity linking weak matching InKB micro F1 scores on WebQSP EL dataset (Li et al., 2020). #Q/s is number of questions per second for a single CPU.

D Dataset statistics

We present the topic, number of documents and number of mentions for each dataset used for evaluation. The datasets used cover a variety of sources including wikipedia text, news articles, web text and tweets. Note that the performance of the model outside these domains may be significantly different.

Note also that all datasets used are for English only, allowing comparison to previous work. Our method is extendable to any language for which there is an language-specific version of Wikipedia on which the model could be trained. However, we cannot guarantee the accuracy of the model across these languages without further experimentation.

	Topic	**Num docs**	**Num Mentions**
AIDA	news	231	4464
MSNBC	news	20	656
AQUAINT	news	50	743
ACE2004	news	57	259
CWEB	web	320	11154
WIKI	Wikipedia	320	6821
WikilinksNED	web	10000	10000

Table 8: Dataset statistics for entity disambiguation datasets

	Topic	**Num docs**	**Num Mentions**
AIDA	news	231	4464
MSNBC	news	20	656
DER	tweets	182	242
K50	mixed	50	145
R128	news	128	638
R500	news	500	530
OKE15	Wikipedia	199	1017
OKE16	Wikipedia	254	1402

Table 9: Dataset statistics for entity linking datasets