Improving Entity Disambiguation by Reasoning over a Knowledge Base

Tom Ayoola* **Joseph Fisher*** **Andrea Pierleoni**
Amazon Alexa AI
Cambridge, UK
{tayoola, fshjos, apierleo}@amazon.com

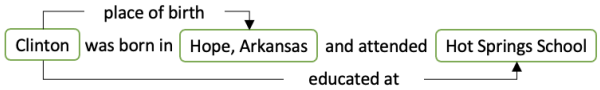
Abstract

Recent work in entity disambiguation (ED) has typically neglected structured knowledge base (KB) facts, and instead relied on a limited subset of KB information, such as entity descriptions or types. This limits the range of contexts in which entities can be disambiguated. To allow the use of all KB facts, as well as descriptions and types, we introduce an ED model which links entities by reasoning over a symbolic knowledge base in a fully differentiable fashion. Our model surpasses state-of-the-art baselines on six well-established ED datasets by 1.3 F1 on average. By allowing access to all KB information, our model is less reliant on popularity-based entity priors, and improves performance on the challenging ShadowLink dataset (which emphasises infrequent and ambiguous entities) by 12.7 F1.

1 Introduction

Entity disambiguation (ED) is the task of linking mentions of entities in text documents to their corresponding entities in a knowledge base (KB). Recent ED models typically use a small subset of KB information (such as entity types or descriptions) to perform linking. These models have strong performance on standard ED datasets, which consist mostly of entities that appear frequently in the training data.

However, ED performance deteriorates for less common entities, to the extent that many recent models are outperformed by outdated feature engineering-based ED systems on datasets that focus on challenging or rare entities (Provatorova et al., 2021). This suggests models over-rely on prior probabilities, which are either implicitly learned or provided as features, rather than make effective use of the mention context. One reason for this is that the subset of KB information used by the models is not enough to discriminate between similar entities in all contexts, meaning the model has to fall back on predicting the most popular entity. Another explanation for the performance drop is that less common entities are prone to missing or inconsistent KB information (e.g. they may not have a description), which is problematic for models which rely on a single source of information. To illustrate, we find that 21% of the 25% least popular[1] entities in Wikidata have neither an English description nor any entity type[2], leaving no mechanism for models which rely on these two sources of information alone to disambiguate them (other than their label).[3] Over half of these entities have at least one KB fact (e.g. [Cafe Gratitude], [headquarters location], [San Francisco]); so by including KB facts the percentage of the least popular entities with no information aside from a label drops from 21% to 8%.



Entity	Description	Types
Hillary Clinton	American politician	Human, politician
Bill Clinton	42nd president of the US	Human, politician

Figure 1: Example of a sentence where fine-grained KB information is required for entity disambiguation.

In light of this, we introduce an ED model which has access to entity types and descriptions, and all KB facts. By using a larger variety of information, our model is more robust to missing KB information, and is able to disambiguate entities in a broader range of contexts without relying on entity priors. Figure 1 shows an example sentence where there is insufficient information in the entity

*Tom and Joseph contributed equally to this work.

[1]We use the number of KB facts where the entity is the subject entity as a proxy for popularity, and only consider entities with an English Wikipedia page.

[2]See for example Q5017238.

[3]Conversely, 100% of the 25% most popular entities in Wikidata have either a description or type.

descriptions and types to disambiguate the mention, *Clinton*. Fine-grained KB information, such as facts about the birthplace or education of candidate entities, is required.

To incorporate KB facts, our model begins by reranking candidate entities using descriptions (Wu et al., 2019) and predicted entity types (Raiman and Raiman, 2018). We then predict, using the document context, the relations which exist between every pair of mentions in the document. For example, given the sentence in Figure 1, the model may predict that the [place of birth] relation exists between the mention *Clinton* and the mention *Hope, Arkansas*.[4] For this, we introduce a novel "coarse-to-fine" document-level relation extraction (RE) module, which increases accuracy and reduces inference time relative to the standard RE approach. Given the relation predictions, we query the KB (Wikidata in our case) for facts which exist between any of the candidate entities for the mention *Clinton* and for the mention *Hope, Arkansas*. In this case we would find the Wikidata fact [Bill Clinton], [place of birth], [Hope], and would correspondingly boost the scores of both the [Bill Clinton] and [Hope] entities. We implement this mechanism with the KB stored in a one-hot encoded sparse tensor, which makes the architecture end-to-end differentiable.

Our model surpasses state-of-the-art (SOTA) baselines on well-established ED datasets by 1.3 F1 on average, and significantly improves performance on the challenging ShadowLink dataset by 12.7 F1. In addition, the model predictions are interpretable, in that the facts used by the model to make predictions are accessible.

Our contributions are summarised as follows:

1. We empirically show that using KB facts for ED increases performance above SOTA methods, which generally rely on a single source of KB information.

2. We introduce a scalable method of incorporating symbolic information into a neural network ED model. To our knowledge, this is the first time an end-to-end differentiable symbolic KB has been used for ED.

3. We introduce a novel document-level relation extraction (RE) architecture which uses coarse-to-fine predictions to obtain competitive accuracy with high efficiency.

[4]We use square brackets to denote relations and entities in the KB, and italics to represent mentions in the input text.

2 Related Work

Recent work on ED has primarily focused on feature-based approaches, whereby a neural network is optimised so that the representation of the correct KB entity is most similar to the mention representation, and each mention is resolved independently. The way in which the KB entities are represented varies between work. Initial work (Ganea and Hofmann, 2017) learned entity embeddings directly from training examples, which performed well for entities seen during training, but could not resolve unseen entities. More recent work improved performance on common datasets by enabling linking to entities unseen during training by using a subset of KB information to represent entities, such as entity descriptions (Logeswaran et al., 2019; Wu et al., 2020) or entity types (Raiman and Raiman, 2018; Onoe and Durrett, 2020).

2.1 ED with KB context

Mulang' et al. (2020) and Cetoli et al. (2019) incorporate KB facts into ED models by lexicalising KB facts and appending them to the context sentence, then using a cross-encoder model to predict whether the facts are consistent with the sentence. Our model differs from this approach as we resolve entities in the document collectively rather than independently; enabling pairwise dependencies between entity predictions to be captured. Another potential limitation of the cross-encoder method is the high computational cost of encoding the long sequence length of every fact appended to the document context. By accessing KB facts from sparse tensors, we are able to avoid this bottleneck and scale to a larger volume of facts (Cohen et al., 2020).

2.2 ED with knowledge graph embeddings

Graph neural networks (GNN) have been used to represent KB facts to inform ED predictions (Sevgili et al., 2019; Ma et al., 2021). These approaches can potentially access the information in all KB facts, but are reliant on the quality of the graph embeddings, which may struggle to represent many basic semantics (Jain et al., 2021) particularly for unpopular entities (Mohamed et al., 2020).

2.3 Global ED

There has been a series of papers which aim to optimise the global coherence of entity choices

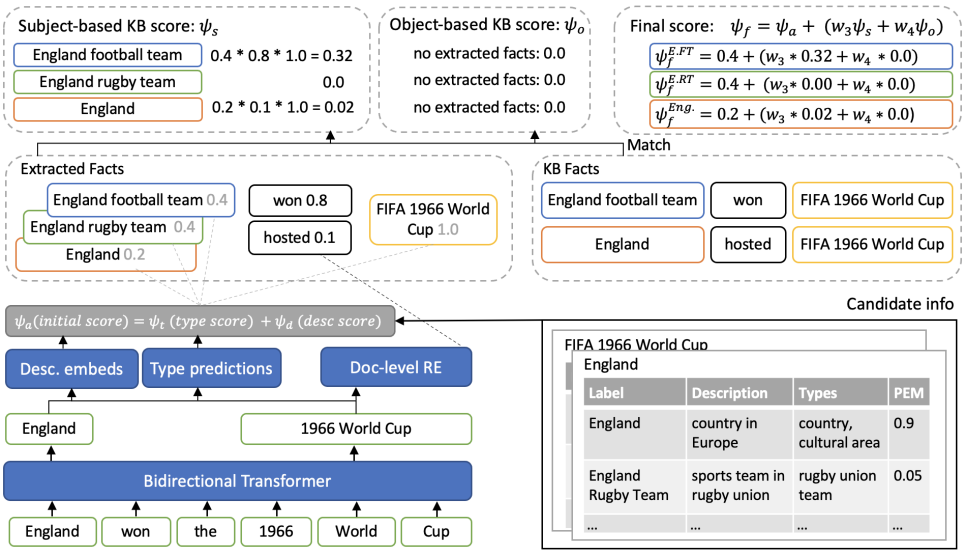


Figure 2: Our model architecture shown for a document with two mentions, *England* and *1966 World Cup*. The model disambiguates all entity mentions in a single pass; making use of the KB facts connecting the candidates of each mention.

across the document (Hoffart et al., 2011; Cheng and Roth, 2013; Moro et al., 2014; Pershina et al., 2015). Our model differs from previous approaches in that the model predicts the relations which exist between mentions based on the document text and weights the coherence scores by these predictions, rather than considering coherence independently of document context. We also limit the model to pairwise coherence between mentions as opposed to global coherence for computational efficiency.

2.4 ED with multiple modules

The most similar work to ours is Orr et al. (2021), which achieves strong results on tail entities by introducing an ED model which uses entity embeddings, relation embeddings, type embeddings, and a KB module to link entities. A key difference to our model is the way in which KB facts are used for disambiguation. In their work, KB facts are encoded independently of the document context in which the candidate entities co-occur, whereas our model is able to leverage the relevant KB facts for the document context.

3 Proposed Method

3.1 Task formulation

Given a document X with mentions, $M = \{m_1, m_2, ...m_{|M|}\}$, a KB with a set of facts $G = \{(s, r, o) \subset E \times R \times E\}$ which express relations $r \in R$ between subject $s \in E$ and object entities $o \in E$, and a description d_k for each KB entity e_k, the goal of ED is to assign each mention $m \in M$ the correct corresponding KB entity $e \in E$.

3.2 Overview

Figure 2 shows a high-level overview of our model. We use a transformer model to encode all mentions in the document in a single-pass. We use these mention embeddings both to generate initial candidate entity scores for each mention using the entity types and descriptions of KB entities and to predict relations between every pair of mentions in the document. We retrieve KB facts for every pair of mentions in the document, for each combination of candidate entities. We weight the retrieved KB facts by multiplying the initial candidate entity score for the subject entity, the predicted score for the relation, and the initial candidate entity score for the object entity. Then we generate KB scores by summing the weighted facts for each candidate entity. The final score used for ranking entities is a weighted sum of the initial score and KB score.

3.3 Mention representation

We encode the tokens in the document X using a transformer-based model, giving contextual token embeddings $H = \{\mathbf{h_1}, \mathbf{h_2}, ..., \mathbf{h_N}\}$.[5] We obtain mention embeddings $\mathbf{m_i}$ for each mention m_i by average pooling the contextualised token embeddings of the mention from the final transformer layer. This allows all mentions M in the document

[5] We use bold letters for vectors throughout our paper.

X to be encoded in a single forward pass.

3.4 Initial entity score $\psi_\mathbf{a}$

Initially, we score candidate entities using entity typing and description scores. We combine the two with a learned weighted sum ψ_a:

$$\psi_a(c_{ik}) = w_1\psi_t(c_{ik}) + w_2\psi_d(c_{ik}) \quad (1)$$

where c_{ik} is the mention-entity pair (m_i, e_k), ψ_t is a scoring function based on candidate entity types, and ψ_d is a scoring function based on candidate entity descriptions.

3.4.1 Entity typing score $\psi_\mathbf{t}$

We construct a fixed set of types $T = \{(r, o) \subset R \times E\}$ by taking relation-object pairs (r, o) from the KB G; for example (instance of, song). We predict an independent unnormalised score for each type $t \in T$ for every mention in the document by applying a linear layer FF_1 to the mention embedding $\mathbf{m_i}$. To compute entity scores ψ_t using the predicted types, we calculate the dot product between the predicted types and the candidate entity's types binary vector $\mathbf{t_k}$.[6] Additionally, we add a $P(e|m)$ (PEM score) which expresses the probability of an entity given the mention text only, and is obtained from hyperlink count statistics as in previous work (Raiman and Raiman, 2018):

$$\psi_t(c_{ik}) = (FF_1(\mathbf{m_i}) \cdot \mathbf{t_k}) + P(e_k|m_i) \quad (2)$$

3.4.2 Entity description score $\psi_\mathbf{d}$

We use a bi-encoder architecture similar to (Wu et al., 2019) but altered to encode all mentions in a sequence in a single forward pass, as opposed to requiring one forward pass per mention. 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 refer to this as d_k. To compute entity scores ψ_d using entity descriptions, we use a separate transformer model TR_1 to encode d_k, taking the final layer embedding for the [CLS], and calculate the dot product between this embedding and the contextual mention embedding $\mathbf{m_i}$ projected by linear layer FF_2:

$$\psi_d(c_{ik}) = FF_2(\mathbf{m_i}) \cdot TR_1(d_k) \quad (3)$$

[6] We use 1 to indicate the presence of an entity type and 0 the absence of an entity type for our binary vector. We also follow this convention for the KB facts binary vector.

3.5 Relation extraction

Our relation extraction layer outputs a relation score vector $\hat{\mathbf{r}}_\mathbf{ij} \in \mathbb{R}^{|R|}$ for each pair of mentions m_i and m_j in the document, where R is the subset of relations chosen from the KB. To calculate $\hat{\mathbf{r}}_\mathbf{ij}$ we begin by passing $\mathbf{m_i}$ and $\mathbf{m_j}$ through a bilinear layer B with output dimension 1, to predict the likelihood $\hat{r}_{ij}^{coarse}$ that a relation exists between mentions m_i and m_j.

$$\hat{r}_{ij}^{coarse} = \sigma(B(\mathbf{m_i}, \mathbf{m_j})) \quad (4)$$

Note that $\hat{r}_{ij}^{coarse}$ is a scalar, denoting the likelihood that **any** relation exists between mention m_i and m_j.

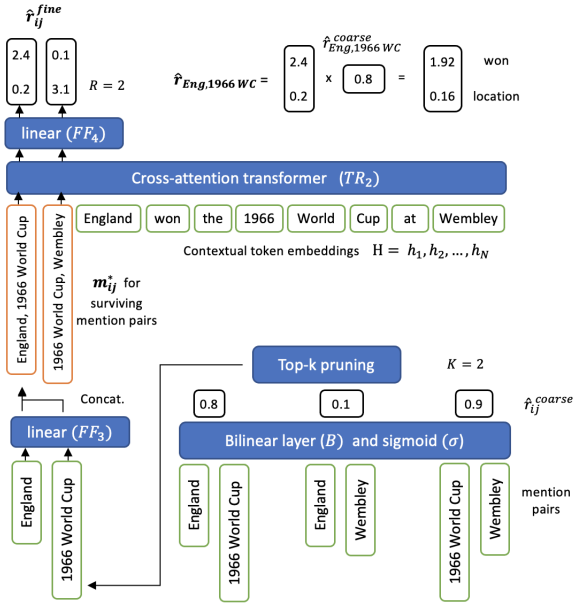


Figure 3: The model component for document-level relation extraction. $\hat{r}_{ij}^{coarse}$ denotes the predicted probability that any relation exists between mentions i and j. R denotes the number of relations we include in the model - set to 2 in the Figure for illustration purposes only.

We then take the top-k mention pairs with the highest values of $\hat{r}_{ij}^{coarse}$ (in similar style to Lee et al. (2018) who introduce a coarse-to-fine approach for coreference resolution), illustrated with $K = 2$ in Figure 3. These are the pairs of mentions which the model predicts have the highest likelihood of having a relation connecting them. For the surviving mention pairs, we pass each of the two mention embeddings individually through a linear layer, FF_3, to reduce their dimension by a factor of two. This ensures that when we concatenate the two representations back together we get a

representation of the mention pair $\mathbf{m^*_{ij}}$ of the same dimension as the contextual token embeddings H.

$$\mathbf{m^*_{ij}} = concat(FF_3(\mathbf{m_i}), FF_3(\mathbf{m_j})) \quad (5)$$

We then pass the resulting embedding $\mathbf{m^*_{ij}}$ through a series of transformer layers TR_2, where they can attend to the contextual embeddings of the original input tokens, $H = \{\mathbf{h_1}, \mathbf{h_2}, ..., \mathbf{h_N}\}$. The mention-pair embeddings from the final transformer layer are passed through a linear layer FF_4 with output dimension $|R|$ to give the score that each relation exists between this mention-pair, $\mathbf{\hat{r}^{fine}_{ij}}$.

$$\mathbf{\hat{r}^{fine}_{ij}} = FF_4(TR_2(\mathbf{m^*_{ij}}, H))) \quad (6)$$

Finally, to get $\mathbf{\hat{r}_{ij}}$ we multiply the coarse layer score $\hat{r}^{coarse}_{ij}$ with the fine layer score $\mathbf{\hat{r}^{fine}_{ij}}$, ensuring that gradients are propagated through the coarse layer during training, despite only the top-k mention pairs being passed to the fine layer.

$$\mathbf{\hat{r}_{ij}} = \hat{r}^{coarse}_{ij} * \mathbf{\hat{r}^{fine}_{ij}} \quad (7)$$

For all mention pairs outside the top-k pairs, we set $\mathbf{\hat{r}_{ij}}$ to a vector of 0s.

The relation extraction layer is trained end-to-end using the signal from the entity disambiguation loss only, and is not pretrained with any task-specific relation extraction data. To validate the effectiveness of the architecture, we include results with the RE module trained in isolation on the DOCRED RE dataset in Appendix D.

3.6 KB score ψ_b

We retrieve KB facts[7] for every mention-entity pair in the document and represent it as a 5-dimensional tensor $\mathbf{r}$, where $\mathbf{r_{ij,kn}}$ is a binary vector indicating the relations that exist in the KB between the two entities (e_k and e_n) for mention-entity pair c_{ik} and c_{jn}.[8] We weight KB facts $\mathbf{r}$ based on initial entity scores ψ_a and relation predictions $\mathbf{\hat{r}}$, according their relevance to the document. To compute the KB score ψ_b for a mention-entity pair, we sum KB facts where the entity (from the mention-entity pair) is the subject entity to give score ψ_s and sum the KB facts where the entity is the object entity to give score ψ_o:

$$\psi_s(c_{ik}) = \dot{\psi}_a(c_{ik}) \sum_{j=1}^{j\leq|M|} \sum_{n=1}^{n\leq|E|} (\mathbf{\hat{r}_{ij}} \cdot \mathbf{r_{ij,kn}}) \dot{\psi}_a(c_{jn}) \quad (8)$$

$$\psi_o(c_{ik}) = \dot{\psi}_a(c_{ik}) \sum_{j=1}^{j\leq|M|} \sum_{n=1}^{n\leq|E|} (\mathbf{\hat{r}_{ji}} \cdot \mathbf{r_{ji,nk}}) \dot{\psi}_a(c_{jn}) \quad (9)$$

where $\dot{\psi}_a$ is the initial entity scoring function ψ_a followed by the softmax function applied over the candidate entities for the given mention. We then combine the two scores with a weighted sum giving ψ_b:

$$\psi_b(c_{ik}) = w_3\psi_s(c_{ik}) + w_4\psi_o(c_{ik}) \quad (10)$$

Note that for computational efficiency, this scoring mechanism considers the coherence of entity predictions between pairs of mentions only, in contrast to methods which consider global coherence Hoffart et al. (2011).

3.7 Optimisation and inference

To obtain final entity scores ψ_f, we add the KB scores ψ_b to the initial entity scores ψ_a.

$$\psi_f(c_{ik}) = \psi_a(c_{ik}) + \psi_b(c_{ik}) \quad (11)$$

We train our model on entity linked documents using cross-entropy loss. Our model is fully differentiable end-to-end, with the training signal propagating through all modules, including the relation extraction module. During ED inference, we take the candidate entity with the highest final entity score for each mention.

4 Experiments

4.1 Standard ED

We evaluate our model on the following well-established standard ED datasets: AIDA-CoNLL (Hoffart et al., 2011), MSNBC (Cucerzan, 2007), AQUAINT (Milne and Witten, 2008), ACE2004 (Ratinov et al., 2011), CWEB (Gabrilovich et al., 2013) and WIKI (Guo and Barbosa, 2018). We train our model on Wikipedia hyperlinks and report *InKB* micro-F1 (which only considers entities with a non-NIL entity label). To ensure fair comparisons to baselines, we use the same method to generate candidates as previous work (Cao et al., 2021; Le and Titov, 2018). Concretely, we use the top-30 entities based on entity priors (PEM) obtained by mixing hyperlink count statistics from Wikipedia hyperlinks, a large Web corpus, and YAGO.

[7] Facts are efficiently retrieved by indexing into a sparse tensor.

[8] The dimensions of tensor $\mathbf{r}$ are: [n_mentions (M), n_mentions (M), n_candidates, n_candidates, n_relations (R)]

4.2 Long-tail and ambiguous ED

We use the ShadowLink ED dataset (Provatorova et al., 2021) to evaluate our model on long-tail and ambiguous examples.[9] The dataset consists of 3 subsets. SHADOW where the correct entity is overshadowed by a more popular entity; TOP where the correct entity is the most popular entity; and TAIL where the correct entity a long-tail entity.[10] All examples in SHADOW and TOP are ambiguous, whereas TAIL has some unambiguous examples, as it is a representative sample of long-tail entities. The original dataset consists of short text snippets from Web pages, which often only include one or two mentions of entities. This limits the ability of our model to use its document-level RE module, and reason over the relationships between entities. We therefore also evaluate on the full-text version of the SHADOW and TOP subsets, referred to as SHADOW-DOC and TOP-DOC in the results tables.[11] The dataset consists of 1 annotated entity per document, so we run spaCy ("en_core_web_lg" model) (Honnibal and Montani, 2017) to identify additional mentions to allow our model and baselines to utilise other mentions to disambiguate the annotated entity mention.

4.3 Model details

We use Wikidata (July 2021) as our KB, restricted to entities with a corresponding English Wikipedia page. This results in 6.2M entities. We use this data to generate lookups for entity types, entity descriptions, and KB facts. We select a fixed set of 1400 relation-object pairs, based on usefulness for disambiguation, to use as our entity types (Appendix A). For the KB facts, we represent the top 128 relations as separate classes and collapse the remaining relations into a single class we refer to as *OTHER*. Additionally, we add a special relation which exists between every entity and itself. We refer to this relation as the *SAME AS* relation, and the idea behind this is to enable the model to implicitly learn coreference resolution.

4.4 Training details

We use Wikipedia hyperlinks (July 2021) with additional weak labels as our training dataset, which consists of approximately 100M labelled mentions. We limit candidate generation to top-30 entities based on entity priors obtained from Wikipedia hyperlink statistics.[12] Our model operates at the document-level and is trained using multiple mentions simultaneously. We initialise the mention embedding Transformer model weights from the RoBERTa (Liu et al., 2019) model and train our model for 1M steps with a batch size of 64 and a maximum sequence length of 512 tokens. This requires approximately 4 days when using 8 V100 GPUs. For additional details, see Appendix B.

5 Results

5.1 Standard ED

The results in Table 1 show our model (KBED) achieves the highest average performance across the datasets by a margin of 1.3 F1, reducing errors by 11.5%. The ablation results indicate the majority of the improvements across the datasets are attributable to our novel KB module. We observe the largest improvement of 3.0 F1 on the WIKI dataset, which is likely due to the documents having high factual density, enabling our model to leverage more KB facts (see Section 6.1 for relation analysis). Despite our model only be trained on Wikipedia, we obtain competitive results on out-of-domain datasets, such as MSNBC news articles, which implies the patterns learned from Wikipedia are applicable to other domains. In addition, the results demonstrate that our 3 modules (entity typing, entity descriptions, and KB facts) are complementary; when any module is used in isolation it reduces performance, demonstrating the benefits of a multifaceted approach to ED. Surprisingly, when our KB module is used in isolation it performs on par with the TagMe baseline, which suggests there is reasonable overlap between KB facts and the facts predicted from documents. Note that the AIDA results in Table 1 contain a mixture of models fine-tuned on this dataset (denoted with **) and trained on Wikipedia only (as in our case), so the numbers are not directly comparable.

5.2 Long-tail and ambiguous ED

Our model achieves an average F1 score of 70.1 on the original ShadowLink dataset (Table 2) which substantially outperforms (+16.5 F1) embeddings-based models (GENRE, REL) and moderately outperforms (+4.0 F1) the Bootleg model (Orr et al.,

[9] A long-tail entity is an entity that is linked to less than 56 times from other Wikipedia pages.

[10] E.g. if the candidates and PEM scores for the mention *England* were ([England (country)], 0.92) and ([England football team], 0.08) then [England (country)] would be a TOP entity, and [England football team] would be a shadow entity.

[11] Details in Appendix E.

[12] We add weak labels by labelling spans that match the title of the page with the entity for the page.

Method	AIDA	MSNBC	AQUAINT	ACE2004	CWEB	WIKI	AVG
AIDA (Yosef et al., 2011)	78.0	79.0	56.0	80.0	58.6	63.0	69.1
TagMe 2 (Ferragina and Scaiella, 2012)	70.6	76.0	76.3	81.9	68.3	-	-
REL (van Hulst et al., 2020)	89.4	90.7	84.1	85.3	71.9	73.1	82.4
GENRE (Cao et al., 2021)	93.3**	94.3	89.9	90.1	77.3	87.4	88.7
Bootleg* (Orr et al., 2021)	80.9	80.5	74.2	83.6	70.2	76.2	77.6
WNEL (Le and Titov, 2019)	89.7	92.2	90.7	88.1	78.2	81.7	86.8
RLEL (Fang et al., 2019)	**94.3****	92.8	87.5	91.2	78.5	82.8	87.9
DCA-RL (Yang et al., 2019)	93.7**	93.8	88.3	90.1	75.6	78.8	86.7
BiBSG (Yang et al., 2018)	93.0**	92.6	89.9	88.5	**81.8**	79.2	87.5
KBED	90.4	**94.8**	**92.6**	**93.4**	78.2	**90.4**	**90.0**
Model Ablations							
w/o KB	87.5	94.4	91.8	91.6	77.8	88.7	88.6
KB only	80.3	88.9	83.0	85.0	69.7	80.8	81.3
Entity types only	85.7	91.8	91.8	89.8	74.3	86.1	86.6
Entity descriptions only	84.8	90.5	91.8	90.8	74.1	87.7	86.6
Bilinear RE layer	86.5	94.4	91.4	93.6	77.5	90.9	89.1

Table 1: Entity disambiguation InKB micro F1 scores on test sets. The best value (excl. model ablations) is **bold** and second best is underlined. *We produced results using the code released by the authors. **Indicates the model was trained on both AIDA and Wikipedia hyperlinks.

2021) which is optimised for tail-performance and also uses entity types and KB facts. On the original dataset, the impact of our KB module is negligible because the limited document context reduces the chances of KB-related entities co-occurring; the strong performance is therefore largely due to the combination of entity types and descriptions. However, we see a notable average improvement of 12.7 F1 on the document-level version of the dataset, with the KB module having a considerable impact especially on the overshadowed entity subset where it contributes 6.7 F1. The performance margin between our model and Bootleg is greater when document-level context is used likely because Bootleg is designed for short contexts and has limited control over which KB facts to use for disambiguation, as all facts are weighted uniformly. We include a more extensive model ablation study in Appendix C.

5.3 Relation extraction module

To analyse the impact of the doc-level RE architecture introduced in Section 3.5 we present results in Tables 1 and 2 of performance with a standard bilinear RE layer (Xu et al., 2021). Our RE architecture leads to an average increase of 0.9 F1 on the standard ED datasets, of 1.3 F1 on the standard ShadowLink splits, and of 1.5 F1 on the ShadowLink doc-level splits. In addition, by avoiding the quadratic complexity bilinear layer, we achieve an increase in inference speed of approximately 2x, as measured on AIDA documents. We include doc-level RE results for our architecture on the DOCRED (Yao et al., 2019) dataset in Appendix D.

5.4 Error Analysis

In Table 3 we show the results from annotating 50 examples in which the model made an incorrect prediction for both the AIDA test split and the ShadowLink SHADOW-DOC split. **Gold not in cands.** refers to cases in which the gold entity was not in the top-30 candidates from the PEM table; **Missing KB fact** are cases where the model correctly predicted a relation connecting two mentions, but the corresponding fact was not in the KB; **Dominant PEM** is when the initial PEM score for one candidate was high (> 0.8), and the model fails to override this score; **Incorrect RE pred.** are cases in which the model makes an incorrect RE prediction between two mentions, and where this wrong prediction leads to the wrong choice of entity; **Ambiguous ann.** refers to gold annotations that are either incorrect or ambiguous.[13]

The results in Table 3 indicate that the largest source of error is the gold entity not being present in the top-30 candidates. This is particularly true for the ShadowLink SHADOW-DOC split, as this split contains a larger number of tail entities which

[13]Note that some examples may contain more than one source of error (or contain an error not clearly in any category), so the sum of the rows will not necessarily be 50.

Method	SHADOW	TOP	TAIL	AVG	SHADOW-DOC	TOP-DOC	DOC-AVG
AIDA (Yosef et al., 2011)	35	56	67	52.7	-	-	-
TagMe 2 (Ferragina and Scaiella, 2012)	29	57	83	56.3	-	-	-
GENRE* (Cao et al., 2021)	26	42	93	53.7	40.9	59.2	50.1
REL (van Hulst et al., 2020)	21	54	91	55.3	-	-	-
Bootleg* (Orr et al., 2021)	44.5	60.0	93.7	66.1	46.9	62.7	54.8
KBED	**47.6**	**64.2**	**98.5**	**70.1**	**60.8**	**74.2**	**67.5**
Model Ablations							
w/o KB	46.4	64.2	98.3	69.6	54.1	72.0	63.1
KB only	26.9	45.7	98.4	57.0	41.9	60.0	51.0
Entity descriptions only	42.1	54.7	97.8	64.9	52.6	65.0	58.8
Entity types only	39.6	55.6	98.5	64.6	47.3	62.1	54.7
Bilinear RE layer	47.1	61.5	97.7	68.8	59.2	72.7	66.0

Table 2: Entity disambiguation InKB micro F1 scores on ShadowLink test sets. SHADOW-DOC and TOP-DOC refers to the extended version of the dataset which includes the full-text of the document to use as additional context. The best value is **bold**. *We produced results using the code released by the authors.

	AIDA	SHADOW-DOC
Gold not in cands.	18	32
Missing KB fact	2	6
Dominant PEM	0	1
Incorrect RE pred.	1	2
Ambiguous ann.	24	4
Total	50	50

Table 3: Counts per error category from 50 annotations on AIDA-CoNLL and ShadowLink-Shadow datasets.

are less likely to be mentioned on Wikipedia. For the AIDA dataset, there are also many cases which are in some sense ambiguous.[14] There are 8 cases in total where the model predicts a relation which it expects to be in the KB, but which is not in fact present. This is largely in the ShadowLink split, where tail entities are likely to be less well represented in Wikidata. The model is generally good at not depending on entity priors; despite every gold candidate in the Shadowlink SHADOW-DOC split being "overshadowed" by a more popular entity in the PEM table, there is only one example where the model fails to override this. Although the model often "over-predicts" relations between mentions, it rarely gets penalised for doing so, as in general the extra facts it predicts are not in the KB, meaning the **Incorrect RE pred.** count is low.

To further explore the role of missing candidates, Table 4 shows the percentage of the gold entities present in the top-30 candidates we pass to the model, representing a hard upper-bound on the recall our model can achieve. The results vary from a high coverage of 99.5 for the MSNBC dataset, which largely contains head entities, to a lower coverage for the ShadowLink SHADOW (75.3) and TOP (83.6) splits. Table 4 also shows the coverage if we pass all PEM candidates to the model. For some datasets, such as WIKI, this increases the coverage significantly. However, for the ShadowLink SHADOW split, the coverage is still below 80%, indicating that better candidate generation strategies are an interesting avenue for future research.

Main datasets test splits						
n	**AIDA**	**MSNBC**	**AQUAINT**	**ACE2004**	**CWEB**	**WIKI**
30	97.8	99.5	95.1	90.9	95.9	93.7
All	1.0	99.5	95.8	92.9	97.0	98.1
ShadowLink splits						
	SHADOW		**TOP**	**TAIL**		
30	75.3		83.6	98.6		
All	76.8		84.3	98.7		

Table 4: Percentage of gold entities in top-n candidates by dataset. We set n=30 for this paper.

6 Analysis

6.1 Relation predictions

To understand the relations which the model utilises to make predictions, Table 5 displays for the WIKI dataset the number of KB (Wikidata) facts which exist between gold annotated mentions in the documents (**Gold**), the number of facts between mentions our model predicts with a score above 0.5 (**Predicted**) and the percentage of gold facts which our model also predicts (**Recall**).[15]

[14] These are often cases with national sports teams, such as "Little will miss Australia's fixture..." where "Australia" could refer to the country Australia or the Australian rugby team.

[15] Note that as the RE predictions are continuous, the **quantity** of facts our model predicts depends entirely on the choice of this threshold.

The *SAME AS* relation is used extensively by the model, demonstrating that using coreferences to other (potentially easier to disambiguate) mentions of the same entity in the document is a powerful addition for ED. We leave evaluation of the model on the coreference-specific task to future work. The *OTHER* relation is also commonly predicted, suggesting the long tail of relations in Wikidata still hold useful information. The other widely used relations are generally either geographical or sports related, which is expected given the large number of sports entities in Wikidata.

The recall numbers appear low, although this is expected behaviour in that the existence of a **Gold** fact does not necessarily imply that the text in the document infers this fact. For example, the text "Donald Trump visited New York" would include the gold fact [Donald Trump] [place of birth] [New York] but making this prediction for all sentences of this form would likely harm performance.

	Gold	**Predicted**	**Recall**
sport	1083	1028	0.53
shares border with	1012	5211	0.68
OTHER	1011	10077	0.29
SAME AS	940	9666	0.36
country	890	285	0.10
located in the a.t.e	709	4912	0.66
contains a.t.e	278	319	0.26
instance of	151	1241	0.23
country of citizenship	120	90	0.08
subclass of	90	2430	0.26
genre	83	204	0.29
part of	80	2154	0.44
follows	69	1104	0.67
followed by	68	1312	0.71
member of sports team	63	449	0.83

Table 5: Analysis of relation predictions for WIKI dataset with threshold 0.5. 320 documents with 6772 entity mentions.

7 Conclusion

We presented a novel ED model, which achieves SOTA performance on well-established ED datasets by a margin of 1.3 F1 on average, and by 12.7 F1 on the challenging ShadowLink dataset. These results were achieved by introducing a method to incorporate large symbolic KB data into an ED model in a fully differentiable and scalable fashion. Our analysis shows that better candidate-generation strategies are an interesting avenue for future research, if results are to be pushed higher on ambiguous and tail entities. Dynamic expansion of the KB by incorporating facts identified by the ED model is also a potentially promising direction.

Acknowledgements

We would like to thank Vera Provatorva for providing us with the extended version of the ShadowLink dataset and Laurel Orr for assisting us with running the Bootleg baseline.

References

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

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

Alberto Cetoli, Stefano Bragaglia, Andrew D. O'Harney, Marc Sloan, and Mohammad Akbari. 2019. A neural approach to entity linking on wikidata. *Advances in Information Retrieval*, page 78–86.

Xiao Cheng and Dan Roth. 2013. Relational inference for wikification. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1787–1796, Seattle, Washington, USA. Association for Computational Linguistics.

William W. Cohen, Haitian Sun, R. Alex Hofer, and Matthew Siegler. 2020. Scalable neural methods for reasoning with a symbolic knowledge base. In *International Conference on Learning Representations*.

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.

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*, 29:70–75.

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, 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 Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Nitisha Jain, Jan-Christoph Kalo, Wolf-Tilo Balke, and Ralf Krestel. 2021. Do embeddings actually capture knowledge graph semantics? In *Eighteenth Extended Semantic Web Conference - Research Track*.

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*.

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.

Phong Le and Ivan Titov. 2019. Boosting entity linking performance by leveraging unlabeled documents. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1935–1945, Florence, Italy. Association for Computational Linguistics.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. *CoRR*, abs/1804.05392.

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.

Jiangtao Ma, Duanyang Li, Yonggang Chen, Yaqiong Qiao, Haodong Zhu, and Xuncai Zhang. 2021. A knowledge graph entity disambiguation method based on entity-relationship embedding and graph structure embedding. *Comput. Intell. Neurosci.*, 2021:2878189.

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

Aisha Mohamed, Shameem Parambath, Zoi Kaoudi, and Ashraf Aboulnaga. 2020. Popularity agnostic evaluation of knowledge graph embeddings. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124 of *Proceedings*

of Machine Learning Research, pages 1059–1068. PMLR.

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 *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.

Laurel J. Orr, Megan Leszczynski, Neel Guha, Sen Wu, Simran Arora, Xiao Ling, and Christopher Ré. 2021. Bootleg: Chasing the tail with self-supervised named entity disambiguation. In *11th Conference on Innovative Data Systems Research, CIDR 2021, Virtual Event, January 11-15, 2021, Online Proceedings*. www.cidrdb.org.

Maria Pershina, Yifan He, and Ralph Grishman. 2015. Personalized page rank for named entity disambiguation. In *NAACL*.

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*.

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.

Özge Sevgili, Alexander Panchenko, and Chris Biemann. 2019. Improving neural entity disambiguation with graph embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 315–322, Florence, Italy. Association for Computational Linguistics.

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.

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. 2019. Zero-shot entity linking with dense entity retrieval. *CoRR*, abs/1911.03814.

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.

Benfeng Xu, Quan Wang, Yajuan Lyu, Yong Zhu, and Zhendong Mao. 2021. Entity structure within and throughout: Modeling mention dependencies for document-level relation extraction. *CoRR*, abs/2102.10249.

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.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. Docred: A large-scale document-level relation extraction dataset. *CoRR*, abs/1906.06127.

Mohamed Amir Yosef, Johannes Hoffart, Ilaria Bordino, Marc Spaniol, and Gerhard Weikum. 2011. Aida: An online tool for accurate disambiguation of named entities in text and tables. *Proc. VLDB Endow.*, 4(12):1450–1453.

A Entity Type Selection

Our entity types are formed from direct Wikidata relation-object pairs and relation-object pairs inferred from the Wikidata subclass hierarchy; for example, (instance of, geographical area) can be inferred from (instance of, city). We only consider types with the following relations: instance of, occupation, country, and 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.

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. Our model has approximately 197M trainable parameters. 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. To reduce GPU memory usage to below 16 GB during training, we subsample 30 mentions per context window, and subsample 5 candidates per mention (subsampling is not required during inference).

Hyperparameter	Value
learning rate	3e-5
batch size	64
max sequence length	512
dropout	0.05
task hidden layer units	768
# training steps	1M
# candidates	30
# relations	128
# entity types	1400
mention transformer init.	roberta-base
# mention encoder layers	12
description transformer init.	roberta-base
# description encoder layers	2
# description tokens	32
RE transformer init.	random
RE coarse-to-fine K	600
# RE transformer layers	4

Table 6: Our model hyperparameters.

C Model Ablation Study

In this section, we measure the contribution of key aspects of our model. For each model ablation, we train our model from scratch on the AIDA-CoNLL training set and evaluate on the development set, keeping hyperparameters constant. Surprisingly, the performance of our model is strong in this limited data setting, which means that our model is not dependent on a large set of training examples when there is a small amount of annotated in-domain data. Note that for "w/o 128 standard relations" we collapse all standard relations into the *OTHER* special relation; and for "w/o RE transformer" we replaced the RE transformer with a single bilinear layer. Our results (Table 7) indicate that all aspects of our model that we measured have a positive impact on performance. Interestingly, the KB module (+2.2 F1) has a greater impact than the entity description (+0.48 F1) and entity typing (+0.9 F1) modules despite weaker performance when used on its own (Table 1). This implies there is less overlap between examples where KB module performs well, and the other modules perform well. We observe, the *SAME AS* relation improves performance by 0.72 F1, which demonstrates that using coreference improves ED. Finally, we find that when the KB module has greater control over how to weight KB facts (based on the context) it leads to better results, for example if we collapse all standard relations into a single relation our performance drops by 0.7 F1.

Method	AIDA
KBED	**94.37**
w/o KB	92.20
w/o *SAME AS* relation	93.65
w/o *OTHER* relation	94.23
w/o 128 standard relations (collapsed)	93.67
w/o RE transformer	94.06
w/o weighting facts by entity scores	94.23
w/o weighting facts by relation scores	93.44
w/o reflexive RE	93.84
w/o entity descriptions	93.89
w/o entity types	93.47
w/o entity priors	93.63
w/o task hidden layers	94.07
w/o negative relation scores	94.19
with Wikipedia ED pre-training	95.58

Table 7: ED F1 score on AIDA-CoNLL development split for model ablations trained from scratch on AIDA-CoNLL training split using the standard CoNLL candidates (Hoffart et al., 2011). The result is in red when the performance drops by more than 0.7.

D Doc-level RE results on DOCRED

To verify the performance of our document-level RE architecture introduced in Section 3.5, we present results of models trained and evaluated on the DOCRED dataset (Yao et al., 2019). Our baseline implementation uses roberta-base as an encoder and a bilinear output layer. We show two variants in Table 8, a bilinear layer with input dimension 128 and with input dimension 256, which give an F1 score of 57.8 and 58.4 respectively. This compares to a score of 59.5 for an equivalent baseline implemented in (Xu et al., 2021). The difference is explained by our baseline not giving the model access to the gold coreference information, which is allowed in the DOCRED task but which we exclude as it will not be available for our entity linking task.

	F1	Train time (seconds per epoch)
Baseline (bilinear)- SSAN imp.	59.5	-
SSAN (roberta-base)	60.9	-
Baseline (bilinear-128)	57.8	155.7
Baseline (bilinear-256)	58.4	343.2
Coarse to fine (2 layers)	60.4	100.6
Coarse to fine (4 layers)	**61.2**	106.2

Table 8: Document-level relation extraction F1 scores on the DOCRED dev dataset.

Our coarse-to-fine approach, with 4 "fine" transformer layers, pushes the dev-level F1 up by 2.8 F1 to 61.2. This puts it slightly above the roberta-base version of the current state-of-the-art model, SSAN (Xu et al., 2021), which scores 60.9, and additionally has access to the gold coreference labels in the embedding layer of the model. This validates that our document-level RE architecture is capable of producing accurate relation predictions, which we see in the main results table (Table 1) also translates into stronger ED performance.

By avoiding the bilinear layer, our implementation is also faster to train, achieving 106.2 seconds per epoch on the DOCRED dataset on a single Tesla V100 GPU, compared to 155.7 seconds for the baseline model with a 128-dimension bilinear layer, and 343.2 seconds for the more accurate baseline model with a 256 dimension bilinear layer.

E Dataset details

E.1 Dataset statistics

We present the topic, number of documents and number of mentions for each dataset used for evaluation (Table 9). 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.

	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
ShadowLink-ALL	web	2712	2712

Table 9: Dataset statistics for entity disambiguation datasets.

E.2 ShadowLink Full Text versions

The authors of Provatorova et al. (2021) kindly provided us with the full documents from which the shorter text snippets (usually one or two sentences) in the ShadowLink dataset were sourced. We were able to match 596 of the 904 examples in the SHADOW split to its corresponding document, and 530 out of the 904 examples in the TOP split. As some full articles were extremely long we limited the document-length to 10000 characters, centred around the single annotated entity. To validate that the subset of examples we were able to match to full documents were representative of the original dataset splits, we ran our model on the sentence-level versions of these subsets, achieving 47.7 on the SHADOW split (comparable to 47.6 in Table 2) and 63.9 on the TOP split (comparable to 64.2 in Table 2).

F Additional relation analysis

To expand on the analysis in Section 6.1 we also include the number of gold and predicted relations in documents in the AIDA dataset (Table 10). The first clear difference is that there is a far higher count of gold *SAME AS* facts in the AIDA dataset, which is potentially explained by pages on Wikipedia generally having hyperlinks for the first mention of an entity only.

It is also interesting to note that there are lower recall numbers for the AIDA dataset relative to WIKI (Table 5), indicating that the RE module may have "overfit" in some sense to the Wikipedia style of article, and may be less effective on AIDA style news articles.

	Gold	Predicted	Recall
SAME AS	4410	14765	0.31
OTHER	3169	5501	0.05
country	3066	728	0.06
member of sports team	880	2653	0.44
country of citizenship	861	150	0.02
shares border with	628	294	0.01
member of	385	669	0.08
league	380	416	0.23
located in the a.t.e	297	1199	0.09
contains a.t.e	261	11	0.01
headquarters location	252	1924	0.15
country for sport	175	130	0.00
has part	126	532	0.07
place of birth	123	365	0.11
sport	103	32	0.00

Table 10: Analysis of relation predictions for AIDA dataset with threshold 0.5.

G Inference Speed and Scalability

We measure the time taken to run inference on the AIDA-CoNLL test dataset and compare it to SOTA baselines. Table 11 shows the results alongside the average ED performance on the 6 standard ED datasets (used in Table 1). Our model is an order of magnitude faster than the baselines with comparable ED performance.

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
KBED	96	**90.0**
w/o KB	**15**	88.6

Table 11: Time taken in seconds for EL inference on AIDA-CoNLL test dataset.

The most computationally expensive part of our model (accounting for approximately 80% of the inference and training time) is computing the KB score due to the large number of pairwise interactions present in documents. The hyperparameter for coarse-to-fine relation extraction can be lowered to trade-off computation cost with ED performance by reducing the number of pairwise interactions. Alternatively, as computation of the initial entity score $\psi_{\mathbf{a}}$ is computationally cheap relative to the KB score $\psi_{\mathbf{b}}$, candidate entities with low initial entity scores can be pruned to further increase training and/or inference speed. These approaches would also allow scaling of the initial number of candidate entities to more than the 30 used for inference in this paper, if the use case required it.