ENTQA: ENTITY LINKING AS QUESTION ANSWERING

Anonymous authors
Paper under double-blind review

ABSTRACT

A conventional approach to entity linking is to first find mentions in a given document and then infer their underlying entities in the knowledge base. A well-known limitation of this approach is that it requires finding mentions without knowing their entities, which is unnatural and difficult. We present a new model that does not suffer from this limitation called **EntQA**, which stands for **Ent**ity linking as **Q**uestion **A**nswering. EntQA first proposes candidate entities with a fast retrieval module, and then scrutinizes the document to find mentions of each candidate with a powerful reader module. Our approach combines progress in entity linking with that in open-domain question answering and capitalizes on pretrained models for dense entity retrieval and reading comprehension. Unlike in previous works, we do not rely on a mention-candidates dictionary or large-scale weak supervision. EntQA achieves strong results on the GERBIL benchmarking platform.

1 INTRODUCTION

We consider the most general form of entity linking (EL) in which a system, given a document, must both extract entity mentions and link the mentions to their corresponding entries in a knowledge base (KB). EL is a foundational building block in automatic text understanding with applications to question answering (QA) (Ferrucci, 2012), information retrieval (Xiong et al., 2017; Hasibi et al., 2016; Balog et al., 2013; Reinanda et al., 2015), and commercial recommendation systems (Yang et al., 2018; Slawski, 2015).

The output space in EL is intractably large. Any subset of all possible spans in the document linked to any KB entries (typically in the order of millions) can be a system output. To get around the intractability, existing methods decompose EL into mention detection (MD) and entity disambiguation (ED) and tackle them with varying degrees of independence. In all cases, however, the order of these two subproblems is MD followed by ED: first the system identifies potential entity mentions, and then the mentions are resolved to KB entries. Previous works either assume that mentions are given (Gupta et al., 2017), run an off-the-shelf named-entity recognition (NER) system to extract mentions and resolve them by ED (MD→ED pipeline) (Hoffart et al., 2011; Ling et al., 2015; van Hulst et al., 2020), or train an end-to-end model that jointly performs MD→ED by beam search (Kolitsas et al., 2018; Cao et al., 2021).

A limitation of performing MD before ED is that it requires finding mentions of unknown entities. By definition, a mention needs an entity (i.e., a mention of what?). Existing methods suffer from the dilemma of having to predict mentions before what they refer to, which is unnatural and difficult. For example, the MD→ED pipeline heuristically extracts mentions from spans of named entities found by a third-party NER system, and the performance bottleneck is often errors in MD propagated to ED. End-to-end models alleviate the problem of error propagation, but the search is only approximate and the dilemma, albeit to a lesser degree, remains.

In this work, we propose flipping the order of the two subproblems and solving ED before MD. We first find candidate entities that might be mentioned in the given document, then for each candidate find its mentions if possible. Our key observation is that while finding mentions is difficult without the knowledge of relevant entities, finding relevant entities is easy without the knowledge of their specific mentions. This simple change fundamentally solves the dilemma above since identifying mentions of a particular entity is well defined.

We cast the problem as *inverted* open-domain QA. Specifically, given a document, we use a dual encoder retriever to efficiently retrieve top-K candidate entities from the KB as "questions". Then

we apply a deep cross-attention reader on the document for each candidate to identify mentions of the candidate in the document as "answer spans". Unlike in standard QA, the model must predict an unknown number of questions and answers. We present a simple and effective solution based on thresholding. We call our model **EntQA**, standing for **Ent**ity linking as **Q**uestion **A**nswering.

Beyond conceptual novelty, EntQA also offers many practical advantages. First, EntQA allows us to piggyback on recent progress in dense entity retrieval and open-domain QA. For instance, we warm start EntQA with the BLINK entity retriever (Wu et al., 2020a) and ELECTRA finetuned on a QA dataset (Clark et al., 2019) to obtain an easy improvement. Second, EntQA has no dependence on a hardcoded mention-candidates dictionary which is used in previous works to reduce the search space and bias the model (Ganea & Hofmann, 2017; Kolitsas et al., 2018; Cao et al., 2021). The dictionary is typically constructed using a large KB-specific labeled corpus (e.g., Wikipedia hyperlinks), thus having no dependence on it makes our approach more broadly applicable to KBs without such resources. Third, training EntQA is data efficient and can be done with an academic budget, in contrast with GENRE (Cao et al., 2021) which requires industry-scale pretraining by weak supervision.

EntQA achieves strong performance on the GERBIL benchmarking platform (Röder et al., 2018). The in-domain F_1 score on the test portion of the AIDA-CoNLL dataset is 85.8 (2.1 absolute improvement). The macro-averaged F_1 score across 8 evaluation datasets is 60.2 (2.0 absolute improvement).[1] We analyze EntQA and find that its retrieval performance is extremely strong (over 98 top-100 recall on the validation set of AIDA), verifying our hypothesis that finding relevant entities without knowing their mentions is easy. We also find that the reader makes reasonable errors such as accurately predicting missing hyperlinks or linking a mention to a correct entity that is more specific than the gold label.

2 MODEL

Let $\mathcal{E}$ denote the set of entities in a KB associated with a text title and description. Let $\mathcal{V}$ denote the vocabulary and $\mathcal{X} = \left\{x \in \mathcal{V}^T : 1 \leq T \leq T_{\max}\right\}$ the set of all documents up to length $T_{\max}$. EL is the task of mapping $x \in \mathcal{X}$ to $y \in \mathcal{P}(\mathcal{Y}(x))$ where $\mathcal{Y}(x) = \{(s, t, e) : 1 \leq s \leq t \leq |x|, e \in \mathcal{E}\}$ is the set of all possible linked spans in x and $\mathcal{P}$ is the power set. The size of the output space is $O(2^{T_{\max}^2 |\mathcal{E}|})$ where $|\mathcal{E}|$ is typically very large (e.g., around 6 million in Wikipedia) and $T_{\max}$ can also be large (e.g., > 3000 in AIDA), ruling out any naive exhaustive search as a feasible approach.

EntQA decomposes EL into two subproblems: entity retrieval and question answering. More specifically, given a document $x \in \mathcal{X}$,

1. The **retriever** module retrieves top-K candidate entities that might be mentioned in x.
2. The **reader** module extracts mentions of each candidate entity in x (or rejects it), then returns a subset of globally reranked labeled mentions as the final prediction.

Our approach bears superficial similarities to a standard framework in open-domain QA that pipelines retrieval and span finding (Karpukhin et al., 2020, *inter alia*), but it has the following important differences. First, instead of retrieving passages given a question, it retrieves questions (i.e., candidate entities) given a passage. Second, even when considering a single question, there can be multiple answer spans (i.e., mentions) instead of one. Both the number of gold entities present in a document and the number of mentions of each gold entity are unknown, making this setting more challenging than standard QA in which we only need to find a single answer span for a single question on a passage.

Input representation. Both the retriever and the reader work with text representations of documents and entities, thus applicable to a zero-shot setting (e.g., linking to a new KB at test time by reading entity descriptions). We use the title $\phi_{\text{title}}(e) \in \mathcal{V}^+$ and the description $\phi_{\text{desc}}(e) \in \mathcal{V}^+$ to represent an entity $e \in \mathcal{E}$. Since a document $x \in \mathcal{X}$ is generally too long to encode with a Transformer encoder which has a quadratic dependency on the input length, we break it down in $m_x \in \mathbb{N}$ overlapping passages $p_1(x) \ldots p_{m_x}(x) \in \mathcal{V}^L$ of length L with stride S (e.g., $L = 32$ and $S = 16$) and operate at the passage-level similarly as in QA (Alberti et al., 2019). When a document is long,

[1] We will release code to reproduce our experiments upon publication.

individual passages may lose global information. For long documents, we find it beneficial to carry a document-level topical text $\psi_{\text{topic}}(x) \in \mathcal{V}^+$ across passages in that document (e.g., first sentence). We emphasize that we do *not* use any extra information outside the document. In our experiments we simply set $\psi_{\text{topic}}(x) = x_1 \in \mathcal{V}$ (i.e., the first token in the document).

Notation. We write $\mathbf{enc}_S^\theta : \mathcal{V}^T \to \mathbb{R}^{d \times T}$ to denote a Transformer encoder that maps any token sequence to the same-length sequence of corresponding contextual embeddings; the symbol S is used to distinguish different encoders. We assume the usual special tokens in the input popularized by BERT (Devlin et al., 2019): `[CLS]` to represent the whole input and `[SEP]` to indicate an input boundary. We write $\oplus$ to denote the text concatenation (we insert a special symbol to represent the concatenation). We write $M_i \in \mathbb{R}^d$ to denote the i-th column of matrix $M \in \mathbb{R}^{d \times T}$.

2.1 Retriever

Given a passage $p \in \mathcal{V}^+$ in document x and an entity $e \in \mathcal{E}$, the retriever computes

$$\begin{aligned} P &= \mathbf{enc}_P^\theta(\texttt{[CLS]}\, p \oplus \psi_{\text{topic}}(x)\, \texttt{[SEP]}) \\ E^e &= \mathbf{enc}_E^\theta(\texttt{[CLS]}\, \phi_{\text{title}}(e) \oplus \phi_{\text{desc}}(e)\, \texttt{[SEP]}) \\ \text{score}_{\text{retr}}^\theta(p, x, e) &= P_1^\top E_1^e \end{aligned}$$

At inference time, we precompute $E^e \in \mathbb{R}^d$ for each $e \in \mathcal{E}$ and use Faiss (Johnson et al., 2019) for fast top-K retrieval.

Training. We train the retriever by a multi-label variant of noise contrastive estimation (NCE). Given a passage p in document x, we have a set of multiple gold entities $\mathcal{E}(p) \subset \mathcal{E}$ that are mentioned in the passage and optimize the per-example objective

$$\max_\theta \sum_{e \in \mathcal{E}(p)} \log \left(\frac{\exp\left(\text{score}_{\text{retr}}^\theta(p, x, e)\right)}{\exp\left(\text{score}_{\text{retr}}^\theta(p, x, e)\right) + \sum_{e' \in \mathbf{N}(\mathcal{E}, p)} \exp\left(\text{score}_{\text{retr}}^\theta(p, x, e')\right)} \right) \tag{1}$$

where $\mathbf{N}(\mathcal{E}, p) \subset \mathcal{E} \backslash \mathcal{E}(p)$ is a set of negative examples that excludes *all* gold entities $\mathcal{E}(p)$. The objective effectively constructs $|\mathcal{E}(p)|$ independent NCE instances, each of which treats a gold entity as the only correct answer while ensuring that other gold entities are not included in negative examples. We obtain 90% of $\mathbf{N}(\mathcal{E}, p)$ by sampling entities uniformly at random from $\mathcal{E} \backslash \mathcal{E}(p)$ and 10% by hard negative mining (i.e., using highest-scoring incorrect entities under the model), which is well known to be beneficial in entity retrieval (Gillick et al., 2019; Wu et al., 2020a; Zhang & Stratos, 2021).

2.2 Reader

Let $e_{1:K} = (e_1 \ldots e_K) \in \mathcal{E}^K$ denote K candidate entities for a passage p in document x. For each $k \in \{1 \ldots K\}$, the reader computes a joint encoding of (p, x, e_k) by

$$H^k = \mathbf{enc}_H^\theta(\texttt{[CLS]}\, p \oplus \psi_{\text{topic}}(x)\, \texttt{[SEP]}\, \phi_{\text{title}}(e_k) \oplus \phi_{\text{desc}}(e_k)\, \texttt{[SEP]})$$

then defines a conditional distribution over mention spans of e_k in p by

$$\begin{aligned} p_{\text{start}}^\theta(s|p, x, e_k) &= \frac{\exp\left(w_{\text{start}}^\top H_s^k\right)}{\sum_{i=1}^{|p|+1} \exp\left(w_{\text{start}}^\top H_i^k\right)} && \forall s \in \{1 \ldots |p|+1\} \\ p_{\text{end}}^\theta(t|p, x, e_k) &= \frac{\exp\left(w_{\text{end}}^\top H_t^k\right)}{\sum_{i=1}^{|p|+1} \exp\left(w_{\text{end}}^\top H_i^k\right)} && \forall t \in \{1 \ldots |p|+1\} \\ p_{\text{span}}^\theta(s, t|p, x, e_k) &= p_{\text{start}}^\theta(s|p, x, e_k) \times p_{\text{end}}^\theta(t|p, x, e_k) && \forall s, t \in \{1 \ldots |p|+1\} \end{aligned}$$

where $w_{\text{start}}, w_{\text{end}} \in \mathbb{R}^d$ are additional parameters. The reader also multitasks reranking: it uses $w_{\text{rerank}} \in \mathbb{R}^d$ to define a conditional distribution over candidate entities by

$$p_{\text{rerank}}^\theta(e_k|p, x, e_{1:K}) = \frac{\exp\left(w_{\text{rerank}}^\top H_s^k\right)}{\sum_{k'=1}^{K} \exp\left(w_{\text{rerank}}^\top H_s^{k'}\right)} \qquad \forall k \in \{1 \ldots K\}$$

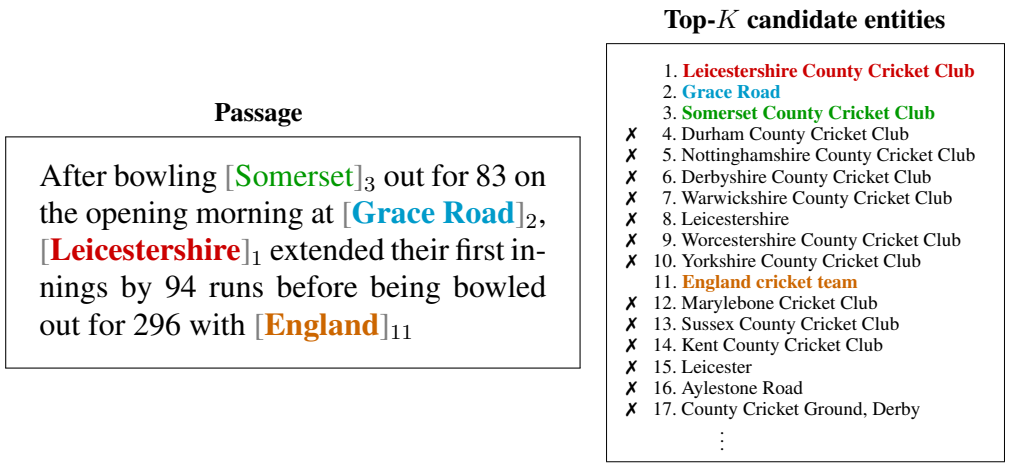

Figure 1: Example prediction by EntQA taken from AIDA-A. Given a passage, the retriever module ranks K candidate entities, then the reader module finds mentions of each entity or rejects it (marked by ✗). Both modules use entity descriptions (not shown). In this example, it predicts the span "England" for the 11th candidate `England cricket team` but rejects the 35th candidate `England` (the country).

Training. We obtain candidates $e_{1:K}$ from a fully trained retrieval module to make training consistent with test time. During training, we always include all gold entities as candidates (i.e., $\mathcal{E}(p) \subset e_{1:K}$). Let $\mathcal{M}(p, e)$ denote the set of gold mention spans of $e \in \mathcal{E}$ in p; if e is not present in p, we define $\mathcal{M}(p, e) = \{(1, 1)\}$. We optimize the per-example objective

$$\max_{\theta} \sum_{k=1}^{K} \mathbb{1}(e_k \in \mathcal{E}(p)) \log p^{\theta}_{\text{rerank}}(e_k|p, x, e_{1:K}) + \sum_{(s,t)\in\mathcal{M}(p,e_k)} \log p^{\theta}_{\text{span}}(s, t|p, x, e_k) \quad (2)$$

where $\mathbb{1}(A)$ is the indicator function equal to one if A is true and zero otherwise. Note that the reader is trained to predict the `[CLS]` span for incorrect entities.

2.3 Inference

At test time, we process a new document $x \in \mathcal{X}$ in passages $p \in \mathcal{V}^L$ independently as follows:

1. Retrieve top-K highest scoring entities $e_{1:K}$ under $\text{score}^{\theta}_{\text{retr}}(p, x, e)$.
2. For each candidate k, extract top-P most likely mention spans $(s^k_1, t^k_1) \dots (s^k_P, t^k_P)$ under $p^{\theta}_{\text{span}}(s, t|p, x, e_k)$ while discarding any span less probable than $(1, 1)$.
3. Return a subset of the surviving labeled mentions (s, t, e_k) with $p^{\theta}_{\text{rerank}}(e_k|p, x, e_{1:K}) \times p^{\theta}_{\text{span}}(s, t|p, x, e_k) > \gamma$ as the final prediction.

We do not apply any further processing to combine passage-level predictions other than merging duplicate labeled spans (s, t, e) in the overlapping sections. This inference scheme is simple yet effective. For each candidate entity, the reader scrutinizes the passage with deep cross-attention to see if there are any mentions of the entity and has a chance to reject it by predicting $(1, 1)$. The reader delays its final decision until it has processed all candidates to globally reconsider labeled mentions with ranking probabilities. Figure 1 shows a successful prediction on a passage from the validation portion of AIDA.

3 Experiments

We evaluate EntQA on the GERBIL benchmarking platform (Röder et al., 2018), which offers reliable comparison with state-of-the-art EL methods on numerous public datasets.

3.1 SETTING

Datasets. We follow the established practice and report the InKB Micro F_1 score on the in-domain and out-of-domain datasets used in Cao et al. (2021). Specifically, we use the AIDA-CoNLL dataset (Hoffart et al., 2011) as the in-domain dataset: we train EntQA on the training portion of AIDA, use the validation portion (AIDA-A) for development, and reserve the test portion (AIDA-B) for in-domain test performance. We use seven out-of-domain test sets: MSNBC, Derczynski (Der) (Derczynski et al., 2015), KORE 50 (K50) (Hoffart et al., 2012), N3-Reuters-128 (R128), N3-RSS-500 (R500) (Röder et al., 2014), and OKE challenge 2015 and 2016 (OKE15 and OKE16) (Nuzzolese et al., 2015). We refer to Table 6 in Kolitsas et al. (2018) for the datasets' statistics. For the KB, we use the 2019 Wikipedia dump provided in the KILT benchmark (Petroni et al., 2021), which contains 5.9 million entities.

Model details. We initialize the passage encoder $\mathbf{enc}_P^\theta$ and the entity encoder $\mathbf{enc}_E^\theta$ in the retriever module with independent BLINK retrievers pretrained on Wikipedia hyperlinks (Wu et al., 2020a) and optimize the NCE objective (1) with hard negative mining. We initialize the joint encoder $\mathbf{enc}_H^\theta$ in the reader module with ELECTRA-large (Clark et al., 2019) finetuned on SQuAD 2.0 (Rajpurkar et al., 2018) and optimize the reader objective (2). We break up each document $x \in \mathcal{X}$ into overlapping passages of length $L = 32$ with stride $S = 16$ under WordPiece tokenization. For each passage in x, we concatenate the input with the first token of the document $\psi_{\text{topic}}(x) = x_1$, which corresponds to the topic in AIDA but not in other datasets. We use 64 candidate entities in training for both the retriever and the reader; we use 100 candidates at test time. We predict up to $P = 3$ mention spans for each candidate entity. We use $\gamma = 0.05$ as the threshold in all experiments, chosen after trying values 0.01, 0.1, and 0.05 on the validation set. For optimization, we use Adam (Kingma & Ba, 2015) with learning rate 2e-6 for the retriever and 1e-5 for the reader; We use a linear learning rate decay schedule with warmup proportions 0.06 for the retriever (4 epochs) and 0.2 for the reader (2 epochs). The batch size is 4 for the retriever and 2 for the reader. The retriever is trained on 4 GPUs (A100) for 24 hours; the reader is trained on 2 GPUs for 12 hours.

Baselines. We compare with state-of-the-art EL systems that represent a diverse array of approaches. Hoffart et al. (2011) and van Hulst et al. (2020) use the MD→ED pipeline; despite the limitation of pipelining MD with ED, the latter achieve excellent performance by solving MD with a strong NER system (Akbik et al., 2018). Kolitsas et al. (2018) use an end-to-end model that sequentially performs MD and ED; to make the problem tractable, they drastically prune the search space with a mention-candidates dictionary and the model score. Cao et al. (2021) propose GENRE, a sequence-to-sequence model for EL. The model conditions on the given document and autoregressively generates a labeled version of the document by at each position either copying a token, starting or ending a mention span, or, if the previous generation was the end of a mention m, generating the entity title associated with m token by token. At inference time, GENRE critically relies on a prefix tree (aka. trie) derived from Wikipedia to constrain the beam search so that it produces a valid entity title in the KB.

3.2 RESULTS

Table 1 shows the main results. EntQA achieves the best in-domain test F_1 score for AIDA (+2.1) and is also performant on out-of-domain datasets (+7.1 on KORE 50 and +7.4 on N3-Reuters-128, close second-best on Derczynski and N3-RSS-500). The performance is lower on OKE15 and OKE16 for the same reason pointed out by Cao et al. (2021): these datasets are annotated with coreference (i.e., they contain pronouns and common nouns linked to entities) which our model is not trained for, while many other systems have a component in their pipelines to handle these cases. We hypothesize that the performance on MSNBC is lagging because it has long documents (544 words per document on average) which are processed in relatively short passages under EntQA due to our computational constraints. Overall, EntQA achieves the best macro-averaged F_1 score across the 8 evaluation datasets (+2.0).

We note that there is an issue of using different editions of Wikipedia between the systems. For instance, Hoffart et al. (2011) use the 2010 dump, van Hulst et al. (2020) and we use the 2019 dump, whereas Kolitsas et al. (2018) and Cao et al. (2021) use the 2014 dump (even though the latter use the 2019 dump for pretraining). Thus there is a concern that differences in performance are due to

Table 1: InKB Micro F_1 on the in-domain and out-of-domain test sets on the GERBIL benchmarking platform. For each dataset, **bold** indicates the best model and underline indicates the second best.

	In-domain	Out-of-domain							
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
Steinmetz & Sack (2013)	42.3	30.9	26.5	46.8	18.1	20.5	46.2	46.4	34.7
Moro et al. (2014)	48.5	39.7	29.8	55.9	23.0	29.1	41.9	37.7	38.2
Kolitsas et al. (2018)	82.4	72.4	34.1	35.2	50.3	38.2	61.9	52.7	53.4
Broscheit (2019)	79.3	-	-	-	-	-	-	-	
Martins et al. (2019)	81.9	-	-	-	-	-	-	-	
van Hulst et al. (2020)	80.5	72.4	41.1	50.7	49.9	35.0	**63.1**	**58.3**	56.4
Cao et al. (2021)	83.7	**73.7**	**54.1**	60.7	46.7	40.3	56.1	50.0	58.2
EntQA	**85.8**	71.0	53.5	**67.8**	**54.1**	41.7	58.9	49.6	**60.2**

different snapshots of Wikipedia. While we consider it out of scope in our work to fully address this concern, we find that using different editions of Wikipedia does not fundamentally change the performance of EntQA, which is consistent with GERBIL's intent of being KB-agnostic. For instance, we obtained the same validation F_1 on AIDA with our model trained on either the 2014 or 2019 dump. We use the KILT edition of Wikipedia mainly for convenience.

3.2.1 Other Practical Highlights

No dictionary. EntQA has no dependence on a mention-candidates dictionary. All previous works rely on a dictionary $\mathcal{D} : \mathcal{V}^+ \to \mathcal{P}(\mathcal{E})$ that maps a mention string m to a small set of candidate entities $e \in \mathcal{E}$ associated with empirical conditional probabilities $\hat{p}_{e|m} > 0$ (Hoffart et al., 2011, *inter alia*). For instance, it is an essential component of the search procedure in the end-to-end model of Kolitsas et al. (2018). While not mentioned in the paper or on the GitHub repository, GENRE (Cao et al., 2021) also uses the dictionary from Kolitsas et al. (2018) in their prefix tree to constrain the beam search (personal communication with one of the authors of the paper). Constructing such a dictionary typically assumes the existence of a large KB-specific labeled corpus (e.g., internal links in Wikipedia). EntQA is thus more broadly applicable to KBs without such resources (e.g., for small domain-specific KBs).

No model-specific pretraining. EntQA does not require model-specific pretraining; it only uses standard pretrained Transformers for initialization and is directly finetuned on AIDA. This is in contrast with GENRE which requires industry-scale pretraining by weak supervision. Specifically, GENRE is trained by finetuning BART (Lewis et al., 2020) on autoregressive EL training examples constructed from all Wikipedia abstract sections on 64 GPUs for 30 hours, followed by finetuning on AIDA. Thus training GENRE from scratch is beyond the means of most academic researchers, making it difficult to make substantial changes to the model. EntQA can be trained with academic resources and outperforms GENRE.

3.3 Ablation Studies

The final form of EntQA in Section 3.2 is the result of empirically exploring various modeling and optimization choices during development. We present an ablation study to illustrate the impact of these choices.

Retriever Table 2 shows an ablation study for the retriever module. We report top-100 recall (R@100) on the validation set of AIDA. The baseline retriever is initialized with BLINK (Wu et al., 2020a), uses the passage representation $p \oplus x_1$, and is trained by optimizing the multi-label variant of NCE (1) that considers one gold entity at a time by excluding others in the normalization term. We see that the baseline retriever has an extremely high recall (98.2), confirming our hypothesis that it is possible to accurately infer relevant entities in a passage without knowing where they are mentioned. We also see that it is very important to use the proposed multi-label variant of NCE

Table 2: Ablation study for the retriever module. Each line makes a single change from the baseline retriever used in Table 1.

Retriever	Val R@100
Baseline	98.2
– Omit excluding other gold entities in the normalization term of NCE	82.7
– Train by optimizing the marginal log-likelihood	83.8
– Initialize with BERT-large	94.4
– Omit hard negatives in NCE (i.e., negative examples are all random)	94.4
– Omit the document-level information x_1 in the passage representation	96.6

Table 3: Ablation study for the reader module. Each line makes a single change from the baseline reader used in Table 1. Candidate entities are obtained from the baseline retriever in Table 2 (except the oracle experiment).

Reader	Val F_1
Baseline	87.5
– Initialize with BERT-large	85.6
– Train by optimizing the marginal log-likelihood	86.9
– Initialize with ELECTRA-large (not finetuned on SQuAD 2.0)	88.4
– Omit the reranking probabilities $p^{\theta}_{\text{rerank}}$ (i.e., only use span probabilities)	87.9
– Omit the document-level information x_1 in the input passage representation	87.5
Oracle experiment: use gold entities as the only candidate entities	94.9

instead of naive NCE that normalizes over all gold entities, which results in a massive decrease in recall (82.7). We consider optimizing the marginal log-likelihood (i.e., the log of the sum of the probabilities of gold entities, rather than the sum of the log), but it yields much worse performance (83.8). It is helpful to initialize with BLINK rather than BERT-large, use hard negatives in NCE, and append x_1 to input passages.

Reader Table 3 shows an ablation study for the reader module. We report F_1 on the validation set of AIDA. The baseline reader is initialized with ELECTRA-large (Clark et al., 2019) finetuned on SQuAD 2.0, uses the joint passage-entity input representation $p \oplus x_1 \texttt{[SEP]} \phi_{\text{title}}(e) \oplus \phi_{\text{desc}}(e)$, and is trained by optimizing (2). Candidate entities are obtained from the baseline retriever in Table 2. We see that BERT is less performant than ELECTRA for reader initialization, consistent with findings in the QA literature (Yamada et al., 2021). Training by optimizing the marginal log-likelihood is comparable to (2). Interestingly, we find that we can fit the reader just as well without using a SQuAD-finetuned ELECTRA, ranking probabilities, or x_1 in passages. However, in our preliminary investigation we found that these variants generalized slightly worse outside the training domain, thus we kept our original choice. Lastly, we conduct an oracle experiment in which we provide only gold entities as candidates to the reader. In this scenario, the reader is very accurate (94.9 F_1), suggesting that the main performance bottleneck is correctly distinguishing gold vs non-gold entities from the candidates. We investigate this issue more in depth in the next section.

3.4 Error Analysis

To better understand the source of errors made by EntQA, we examine passages in the validation set for which the model's prediction is not completely correct. We partition them into three types: (1) over-predicting (i.e., the gold mentions are a strict subset of the predicted mentions), (2) under-predicting (i.e., the predicted mentions are a strict subset of the gold mentions), and (3) neither over- nor under-predicting. Table 4 shows examples of each error type. We find that over-predicting often happens because the model correctly "fills in" entity mentions missing in the gold annotation. Under-predicting happens most likely because the threshold value is too large to catch certain mentions. Finally, many errors that are neither over- nor under-predicting are largely due to annotation noise. For instance, the predicted entity `Headingly Stadium` is a correct and more specific en-

Table 4: Categorizing errors on the validation set passages. The number of passages in each category is given in parentheses. **G** refers to the gold annotation; **P** refers to the predicted annotation.

Error	Examples (text snippets)	
Over (443)	**G**: england fast bowler [martin mccague]$_{\text{Martin McCague}}$	(Fill in missing mentions)
	P: [england]$_{\text{England cricket team}}$ fast bowler [martin mccague]$_{\text{Martin McCague}}$	
	G: duran, 45, takes on little - known [mexican]$_{\text{Mexico}}$	
	P: [duran]$_{\text{Roberto Durán}}$, 45, takes on little - known [mexican]$_{\text{Mexico}}$	
Under (474)	**G**: second innings before [simmons]$_{\text{Phil Simmons}}$ stepped in	(Bad threshold)
	P: second innings before simmons stepped in	
	G: [ato boldon]$_{\text{Ato Boldon}}$ - lpr - [trinidad]$_{\text{Trinidad}}$ - rpr - 20.	
	P: [ato boldon]$_{\text{Ato Boldon}}$ - lpr - trinidad - rpr - 20.	
Neither (378)	**G**: match against yorkshire at [headingley]$_{\text{Headingly}}$	(Ambiguous entity)
	P: match against yorkshire at [headingley]$_{\text{Headingly Stadium}}$	
	G: at the [oval]$_{\text{The Oval}}$, surrey captain chris lewis	(Ambiguous span)
	P: at [the oval]$_{\text{The Oval}}$, surrey captain chris lewis	
	G: scores in [english]$_{\text{England}}$ county championship matches	(Others)
	P: scores in [english county championship]$_{\text{County Championship}}$ matches	

tity for the span "headingley" than the gold entity `Headingly` (a suburb); the predicted span "the oval" is more suitable, or at least as correct as, the gold span "oval" for the entity `The Oval`.

We also consider distinguishing MD errors from ED errors on the validation set. EntQA obtains 87.5 overall F_1. When we only measure the correctness of mention spans (equivalent to treating all entity predictions as correct), we obtain 92.3 F_1. When we only measure the correctness of rejecting or accepting candidate entities, we obtain 64.5 F_1 at the passage level and 89.3 F_1 at the document level (i.e., consider the set of candidates from all passages). The reader's relatively low passage-level F_1 in rejecting or accepting candidates is consistent with the the oracle experiment in Table 3. That is, the main performance bottleneck of EntQA is discriminating gold vs non-gold entities from the candidates, though this should be taken with a grain of salt given the noise in annotation illustrated in Table 4.

4 Related Work

Our work follows the recent trend of formulating language tasks as QA problems, but to our knowledge we are the first to propose reduction to inverted open-domain QA. Most previous works supply questions as input to the system, along with passages in which answer spans are found. They differ only in question formulation, for instance a predicate in semantic role labeling (He et al., 2015), a relation type along with its first argument in KB completion (Levy et al., 2017; Li et al., 2019), an entity category in (nested) NER (Li et al., 2020), an auxiliary verb or a *wh*-expression in ellipsis resolution (Aralikatte et al., 2021), and other task-specific questions (McCann et al., 2018). In contrast, we solve question formulation as part of the problem by exploiting recent advances in dense text retrieval.

A notable exception is CorefQA (Wu et al., 2020b), from which we take direct inspiration. In this approach, the authors formulate coreference resolution as QA in which questions are coreferring spans and answers are the spans' antecedents (i.e., earlier spans that belong to the same coreference cluster). Since coreferring spans are unknown, the authors rely on the end-to-end coreference resolution model of Lee et al. (2017) that produces candidate spans by beam search. In contrast, EntQA handles varying numbers of questions in a simpler framework of text retrieval.

As in this work, some previous works propose methods to handle varying numbers of answer spans for a given question. But their methods are based on one-vs-all classification (i.e., each label is associated with a token-level binary classifier) or reduction to tagging (i.e., spans are expressed as a BIO-label sequence) (Wu et al., 2020b; Li et al., 2019; 2020). We found these methods to be ineffective in preliminary experiments, and instead develop a more effective inference scheme in which the model delays its final prediction to the end for global reranking (Section 2.3).

We discuss pros and cons of EntQA vs other models in practice. While EntQA outperforms GENRE without large-scale weakly supervised pretraining, it involves dense retrieval which incurs a large memory footprint to store and index dense embeddings as pointed out by Cao et al. (2021). But it can be done on a single machine with ample RAM (ours has 252G) which is cheap. Bypassing dense retrieval is a unique strength of the autoregressive approach of GENRE and orthogonal to ours; we leave combining their strengths as future work. Our model requires a threshold γ for inference, but we find that it is easy to pick a good threshold; we also argue that it can be a useful feature in a real-world setting in which the practitioner often needs a customized trade-off between precision and recall. The threshold-based inference implies another unique feature of EntQA not explored in this work: it can naturally handle nested entity mentions. We leave nested linking as future work.

5 CONCLUSIONS

Existing methods for entity linking suffer from the dilemma of having to predict mentions of unknown entities. We have presented EntQA, a new model that solves this dilemma by predicting entities first and then finding their mentions. Our approach is based on a novel reduction to inverse open-domain QA in which we retrieve an unknown number of questions (candidate entities) and predict potentially multiple answer spans (mentions) for each question. Our solution is a simple pipeline that takes full advantage of progress in text retrieval and reading comprehension. EntQA achieves new state-of-the-art results on the GERBIL benchmarking platform without relying on a KB-specific mention-candidates dictionary or expensive model-specific pretraining.

REFERENCES

Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th international conference on computational linguistics*, pp. 1638–1649, 2018.

Chris Alberti, Kenton Lee, and Michael Collins. A bert baseline for the natural questions. *arXiv preprint arXiv:1901.08634*, 2019.

Rahul Aralikatte, Matthew Lamm, Daniel Hardt, and Anders Søgaard. Ellipsis resolution as question answering: An evaluation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 810–817, 2021.

Krisztian Balog, Heri Ramampiaro, Naimdjon Takhirov, and Kjetil Nørvåg. Multi-step classification approaches to cumulative citation recommendation. In *Proceedings of the 10th conference on open research areas in information retrieval*, pp. 121–128, 2013.

Samuel Broscheit. 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)*, pp. 677–685, 2019.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. Autoregressive entity retrieval. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=5k8F6UU39V`.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*, 2019.

Leon Derczynski, Diana Maynard, Giuseppe Rizzo, Marieke Van Erp, Genevieve Gorrell, Raphaël Troncy, Johann Petrak, and Kalina Bontcheva. Analysis of named entity recognition and linking for tweets. *Information Processing & Management*, 51(2):32–49, 2015.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pp. 4171–4186, 2019.

David A Ferrucci. Introduction to this is watson. *IBM Journal of Research and Development*, 56 (3.4):1–1, 2012.

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

Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. Learning dense representations for entity retrieval. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pp. 528–537, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/K19-1049. URL https://www.aclweb.org/anthology/K19-1049.

Nitish Gupta, Sameer Singh, and Dan Roth. Entity linking via joint encoding of types, descriptions, and context. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2681–2690, 2017.

Faegheh Hasibi, Krisztian Balog, and Svein Erik Bratsberg. Exploiting entity linking in queries for entity retrieval. In *Proceedings of the 2016 acm international conference on the theory of information retrieval*, pp. 209–218, 2016.

Luheng He, Mike Lewis, and Luke Zettlemoyer. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pp. 643–653, 2015.

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

Johannes Hoffart, Stephan Seufert, Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. Kore: keyphrase overlap relatedness for entity disambiguation. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pp. 545–554, 2012.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 2019.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015.

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

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 188–197, 2017.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pp. 333–342, 2017.

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

Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. Entity-relation extraction as multi-turn question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1340–1350, 2019.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. A unified mrc framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5849–5859, 2020.

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

Pedro Henrique Martins, Zita Marinho, and André FT Martins. Joint learning of named entity recognition and entity linking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pp. 190–196, 2019.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*, 2018.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2: 231–244, 2014.

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

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2523–2544, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.200. URL https://aclanthology.org/2021.naacl-main.200.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you dont know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 784–789, 2018.

Ridho Reinanda, Edgar Meij, and Maarten de Rijke. Mining, ranking and recommending entity aspects. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 263–272, 2015.

Michael Röder, Ricardo Usbeck, Sebastian Hellmann, Daniel Gerber, and Andreas Both. N^3-a collection of datasets for named entity recognition and disambiguation in the nlp interchange format. In *LREC*, pp. 3529–3533, 2014.

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

Bill Slawski. How google uses named entity disambiguation for entities with the same names, September 2015. URL https://www.seobythesea.com/2015/09/disambiguate-entities-in-queries-and-pages/. Accessed: 2021-09-27.

Nadine Steinmetz and Harald Sack. Semantic multimedia information retrieval based on contextual descriptions. In *Extended Semantic Web Conference*, pp. 382–396. Springer, 2013.

Johannes M van Hulst, Faegheh Hasibi, Koen Dercksen, Krisztian Balog, and Arjen P de Vries. 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*, pp. 2197–2200, 2020.

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

Wei Wu, Fei Wang, Arianna Yuan, Fei Wu, and Jiwei Li. Corefqa: Coreference resolution as query-based span prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6953–6963, 2020b.

Chenyan Xiong, Jamie Callan, and Tie-Yan Liu. Word-entity duet representations for document ranking. In *Proceedings of the 40th International ACM SIGIR conference on research and development in information retrieval*, pp. 763–772, 2017.

Ikuya Yamada, Akari Asai, and Hannaneh Hajishirzi. Efficient passage retrieval with hashing for open-domain question answering. *arXiv preprint arXiv:2106.00882*, 2021.

Yi Yang, Ozan İrsoy, and Kazi Shefaet Rahman. 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)*, pp. 777–786, 2018.

Wenzheng Zhang and Karl Stratos. Understanding hard negatives in noise contrastive estimation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1090–1101, 2021.