ExtEnD: Extractive Entity Disambiguation

Edoardo Barba[*] and **Luigi Procopio**[*] and **Roberto Navigli**
Sapienza NLP Group, Sapienza University of Rome
{edoardo.barba, luigi.procopio, roberto.navigli}@uniroma1.it

Abstract

Local models for Entity Disambiguation (ED) have today become extremely powerful, in most part thanks to the advent of large pre-trained language models. However, despite their significant performance achievements, most of these approaches frame ED through classification formulations that have intrinsic limitations, both computationally and from a modeling perspective. In contrast with this trend, here we propose EXTEND, a novel local formulation for ED where we frame this task as a text extraction problem, and present two Transformer-based architectures that implement it. Based on experiments in and out of domain, and training over two different data regimes, we find our approach surpasses all its competitors in terms of both data efficiency and raw performance. EXTEND outperforms its alternatives by as few as 6 F_1 points on the more constrained of the two data regimes and, when moving to the other higher-resourced regime, sets a new state of the art on 4 out of 6 benchmarks under consideration, with average improvements of 0.7 F_1 points overall and 1.1 F_1 points out of domain. In addition, to gain better insights from our results, we also perform a fine-grained evaluation of our performances on different classes of label frequency, along with an ablation study of our architectural choices and an error analysis. We release our code and models for research purposes at `https://github.com/SapienzaNLP/extend`.

1 Introduction

Being able to associate entity mentions in a given text with the correct entity they refer to is a crucial task in Natural Language Processing (NLP). Formally referred to as Entity Disambiguation (ED), this task entails, given a mention m occurring in a text c_m, identifying the correct entity e^* out of a set of candidates $e_1, \ldots, e_n$, coming from a reference knowledge base (KB). First introduced by Bunescu and Paşca (2006), ED aims to identify the actors involved in human language and, as such, has shown potential in downstream applications like Question Answering (Yin et al., 2016), Information Extraction (Ji and Grishman, 2011; Guo et al., 2013), Text Generation (Puduppully et al., 2019) and Semantic Parsing (Bevilacqua et al., 2021; Procopio et al., 2021).

Since the advent of Deep Learning within the NLP community, this task has mostly been framed as a multi-label classification problem (Shahbazi et al., 2019; Broscheit, 2019), especially leveraging the bi-encoder paradigm (Humeau et al., 2020; Wu et al., 2020). However, although simple and yet powerful enough to push scores past 90% *inKB Micro* F_1 on standard benchmarks, this formulation suffers from a number of downsides. First, the actual disambiguation is only modeled through a dot product between independent mention and entity vectors, which may not capture complex mention-entity interactions. Second, from a computational perspective, entities are represented through high-dimensional vectors that are cached in a pre-computed index. Thus, classifying against a large KB has a significant memory cost that, in fact, scales linearly with respect to the number of entities. Besides this, adding a new entity also requires modifying the index itself. To address these issues, De Cao et al. (2021b) have recently proposed an auto-regressive formulation where, given mentions in their context, models are trained to generate, token-by-token, the correct entity identifiers.[1]

While this approach has addressed the aforementioned issues effectively, it requires an auto-regressive decoding process, which has speed implications, and, what is more, does not let the model see its possible output choices, something

[*] Equal contribution.

[1] i.e. a textual description of the entity; in De Cao et al. (2021b), they use the titles of Wikipedia articles, since their reference KB is Wikipedia.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 2478 - 2488
May 22-27, 2022 ©2022 Association for Computational Linguistics

that has shown significant potential in other semantic tasks (Barba et al., 2021a). In this work, we focus on these shortcomings and, inspired by this latter research trend, propose Extractive Entity Disambiguation (ExtEnD), the first entity disambiguator that frames ED as a text extraction task. Given as input a context c_m in which a mention m occurs, along with a text representation for each of the possible candidates $e_1, \dots, e_n$, a model has to extract the span associated with the text representation of the entity that best suits m. We implement this formulation through 2 architectures: i) a Transformer system (Vaswani et al., 2017; Devlin et al., 2019) that features an almost identical modeling power to that of previous works, and ii) a variant that relaxes the computational requirements of our approach when using common Transformer-based architectures. Evaluating our two systems over standard benchmarks, we find our formulation to be particularly suited to ED. In particular, when restricting training resources to the AIDA-CoNLL dataset (Hoffart et al., 2011) only, ExtEnD appears to be significantly more data-efficient than its alternatives, surpassing them by more than 6 *inKB Micro* F_1 points on average across *in-domain* and *out-of-domain* datasets. Furthermore, when pre-training on external ED data as in De Cao et al. (2021b), our system sets a new state of the art on 4 out of 6 benchmarks under consideration, with average improvements of 0.7 overall and 1.1 when moving out of domain. Finally, we also perform a thorough investigation of our system performances, providing insights and pinpointing the reasons behind our improvements via a fine-grained evaluation on different label-frequency classes.

Our contributions are therefore as follows:

- We propose a new framing of ED as a text extraction task;
- We put forward two architectures that implement our formulation, whose average score across different benchmarks surpasses all previous works in both data regimes we consider;
- We perform a thorough analysis of our systems' performances, evaluating their behavior over different label-frequency classes.

We release our code and models for research purposes at `https://github.com/SapienzaNLP/extend`.

2 Related Work

Entity Disambiguation (ED) is the task of identifying, given a mention in context, the most suitable entity among a set of candidates stored in a knowledge base (KB). Generally the last step in an Entity Linking system (Broscheit, 2019), coming immediately after mention detection and candidate generation, this task has been the object of a vast and diverse literature, with approaches typically clustered into two groups, depending on how they model co-occurring mentions in the same document. On the one hand, *global models* strive to enforce a global coherence across the disambiguations within the same document, leveraging different techniques and heuristics to approximate this objective[2] (Hoffart et al., 2011; Moro et al., 2014; Yamada et al., 2016; Ganea and Hofmann, 2017; Le and Titov, 2018; Yang et al., 2018).

On the other hand, *local models* disambiguate each mention independently of the others, conditioning the entity choice only on the mention and its context. Thanks to the advent of large pretrained language models, this group has recently witnessed a significant improvement in performances, which are nowadays on par with, or even above, those achieved by state-of-the-art global systems (Shahbazi et al., 2019). These approaches usually frame ED as a multi-label classification problem (Broscheit, 2019) and a diverse set of formulations have been proposed. Among these, the bi-encoder paradigm (Bromley et al., 1994; Humeau et al., 2020) has been particularly successful (Gillick et al., 2019; Tedeschi et al., 2021; Botha et al., 2020): here, two encoders are trained to learn vector representations in a shared space for mentions in context and entities, respectively. Classification of a given mention is then performed by retrieving the entity whose representation is closest according to some metric (e.g. cosine similarity).

Although remarkably powerful, these formulations present a number of disadvantages, such as their large memory footprint (each entity in the KB needs to be represented by a high-dimensional vector) and the fact that the actual disambiguation process is only expressed via a dot product of independently computed vectors, potentially neglecting mention-entity interactions. While a number of works (Logeswaran et al., 2019; Wu et al., 2020) attempt to address the latter issue via multi-stage

[2] Approximation is necessary as the exact computation of coherence objectives is NP-hard (Le and Titov, 2018).

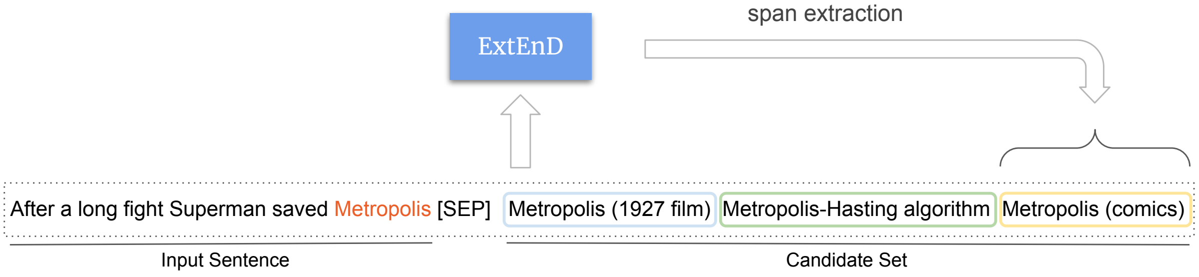

Figure 1: Illustration of ExtEnD on the example sentence *After a long fight Superman saved Metropolis*. The model takes as input a sentence with the target mention to disambiguate, *Metropolis*, explicitly marked (for better visualization, we resort here to highlighting with a different color rather than surrounding it with special tokens) along with the text representation of each candidate. As in our experiments, the knowledge base here is Wikipedia and the candidate text representations are Wikipedia page titles. Then, the model performs the disambiguation by indicating the start and end token of the span containing the predicted entity representation.

approaches where a cross-encoder is stacked after an initial bi-encoder[3] or other retrieval functions, an interesting alternative direction that tackles both problems was recently presented by De Cao et al. (2021b): the authors frame ED as a generation problem and, leveraging an auto-regressive formulation, train a sequence-to-sequence model to generate the correct entity identifier for a given mention and its context.

Nevertheless, while this approach can model more complex interactions, some of these can only occur indirectly inside the backtracking of their beam search. Furthermore, the disambiguation involves an auto-regressive decoding that, although mitigated by later efforts (De Cao et al., 2021a), has intrinsic speed limitations. In contrast, here we propose an extractive formulation, where a model receives as input the mention, its context and the text representation of each candidate, and has to extract the span corresponding to the representation of the entity that best matches the (mention, context) pair under consideration. Note that this differs from the aforementioned cross-encoder formulations (Logeswaran et al., 2019; Wu et al., 2020) where, instead, each entity was encoded together with the (mention, context) pair, but independently from all the other entities. With our schema, complex mention-entity and entity-entity interactions can be explicitly modeled by the neural system, as all the information is provided in input.

Glancing over other related tasks in the area of semantics, arguably closest to our work is ESC (Barba et al., 2021a), where the authors propose a new framing of Word Sense Disambiguation (WSD) as an extractive sense comprehension task. Yet, differently from their work, we propose here a new framing for ED, i.e. focus on entity descriptions rather than word sense definitions, present a baseline system that implements it and devise an additional architecture that deals with the computational challenges that arise from such implementation.

3 Model

We now introduce ExtEnD, our proposed approach for ED. We first present the formulation we adopt (Section 3.1) and, then, describe the two architectures that implement it (Section 3.2).

3.1 Formulation

Inspired by recent trends in other semantic tasks (Barba et al., 2021a), we formulate Entity Disambiguation as a text extraction problem: given a query x_q and a context x_c, a model has to learn to extract the text span of x_c that best answers x_q. Formally, let m be a mention occurring in a context c_m and denote by $Cnd(m) = \{cnd_1, \ldots, cnd_n\}$ the set of n text representations associated with each candidate of m. Then, we formulate ED as follows: we treat the tuple (m, c_m) and the concatenation of $cnd_1, \ldots, cnd_n$ as the query x_q and the context x_c, respectively, and train a model to extract the text span from x_c associated with the correct $cnd^* \in Cnd(m)$; the overall process is illustrated in Figure 1. This formulation helps to better model the input provided, with the possible

[3] This bi-encoder, rather than performing the actual classification, is tasked to generate a filtered set of candidates.

candidates of m included in the contextualization process, while also disposing of large output vocabularies as in De Cao et al. (2021b) and, yet, not resorting to auto-regressive decoding strategies.

3.2 Architectures

To implement our formulation, we consider two Transformer-based architectures. For both of these, the input is composed of the concatenation of the query x_q and the context x_c, subword-tokenized and separated by a *[SEP]* special symbol. Since x_q is a tuple in our formulation, whereas Transformer models only support text sequences as input, we convert x_q into a string $\hat{x}_q$ by taking only c_m and surrounding the text span where m occurs with the special tokens *<t>* and *</t>*. Additionally, to better separate entity candidate representations and ease their full span identification, we add a trailing special symbol *</ec>* to each of them; henceforth, we denote this resulting modified context by $\hat{x}_c$.

As our first architecture, we use two independent classification heads on top of BART (Lewis et al., 2020) computing, respectively, for each word w in $\hat{x}_c$, whether w is the start or end of the correct entity representation cnd^*. We train the model with a cross-entropy criterion over the start and end of cnd^*. At inference time, we select the entity candidate representation $cnd^{'} \in Cnd(m)$ whose joint probability over the 2 heads is highest.

However, framing ED as we propose here implies that the length of the input to the model scales linearly with the number of output choices m. Taking into account that the attention mechanism of Transformer architectures has quadratic complexity and that several pre-trained models actually support inputs only up to a fixed maximum length,[4] this might pose significant computational limitations depending on the dataset and knowledge base under consideration. To cope with these technical challenges, we consider a second system, similar to the previous one but for two main differences. First, we change the underlying Transformer model, replacing BART with a pre-trained Longformer model (Beltagy et al., 2020), a Transformer architecture with an attention mechanism that is linear with respect to the input length and that can handle longer sequences. This linear complexity is achieved by essentially applying a sliding attention window over each token but for a few pre-selected ones (e.g. *[CLS]*), which instead feature a symmetric global attention: they attend upon and are attended by all the other tokens in the input sequence. This global mechanism is intended to be task-specific and enables the model to learn representations potentially close to those standard fully-attentive Transformers would learn, while still maintaining the overall attention complexity linear with respect to the input size. Therefore, as our second modification, we adapt this global pattern to our setting, activating it on the *[CLS]* special token and on the first token of each $cnd_i \in Cnd(m)$; this allows to better mimic the original quadratic mechanism where different entity candidate representations can also attend upon each other. Furthermore, differently from Beltagy et al. (2020), we disable the global attention mechanism on the tokens in the query $\hat{x}_q$. In Section 5, we report and discuss the impact of these modifications. We illustrate the proposed architecture in Figure 2.

4 Entity Disambiguation Evaluation

We now assess the effectiveness of ExtEnd on Entity Disambiguation. We first introduce the experimental setup we consider (Section 4.1). Then, we present the results achieved by ExtEnd both in terms of raw performances (Section 4.2) and via a breakdown of its behavior on different classes of label frequency (Section 4.3). For ease of readability, we focus here only on the Longformer-based architecture, which we consider as our main model. We defer the comparison with the BART-based system to Section 5.

4.1 Experimental Setup

Data To evaluate ExtEnd on Entity Disambiguation, we reproduce the same setting used by De Cao et al. (2021b). Specifically, we adopt their same candidate sets, which were originally proposed by Le and Titov (2018),[5] use Wikipedia titles (e.g. *Metropolis (comics)*) as the text representation for entities and perform training, along with *in-domain* evaluation, on the AIDA-CoNLL dataset (Hoffart et al., 2011, **AIDA**); similarly, we use their cleaned version of **MSNBC**, **AQUAINT**, **ACE2004**, WNED-CWEB (**CWEB**) and WNED-WIKI (**WIKI**) (Guo and Barbosa, 2018; Evgeniy et al., 2013) for *out-of-domain* evaluation.

[4] For instance, the implementation of BART available in HuggingFace Transformers (Wolf et al., 2020) supports inputs only up to 1024 subwords.

[5] These candidate sets were generated relying upon count statistics from Wikipedia, a large Web corpus and the YAGO dictionary.

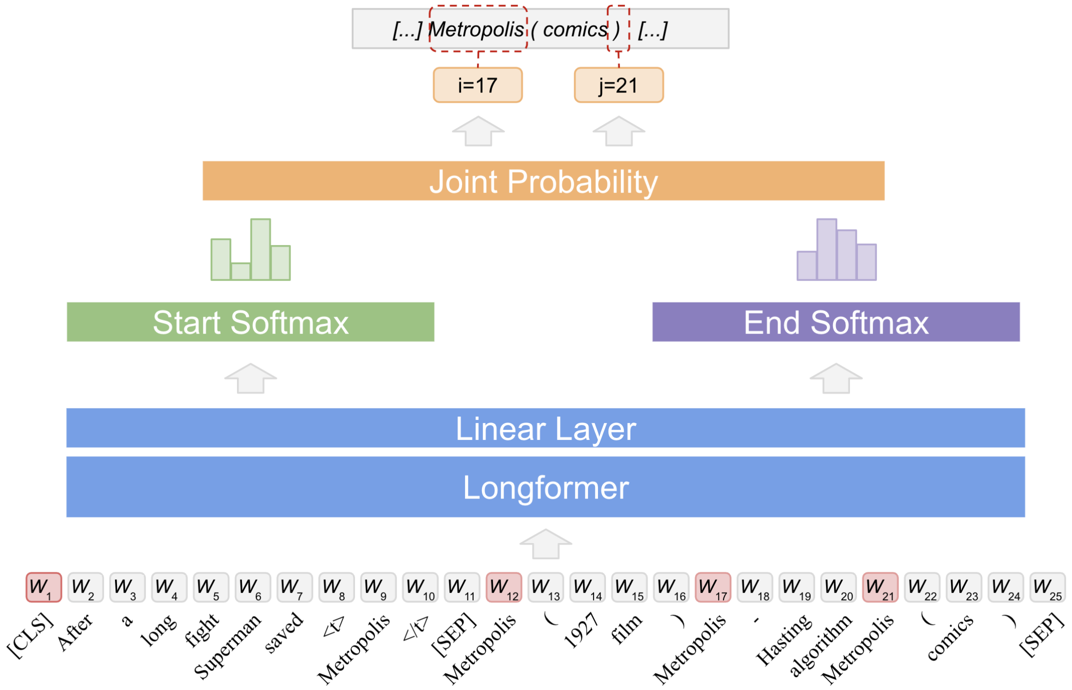
[...] Metropolis (comics) [...]
i=17
j=21
Joint Probability
Start Softmax
End Softmax
Linear Layer
Longformer
W_1 W_2 W_3 W_4 W_5 W_6 W_7 W_8 W_9 W_{10} W_{11} W_{12} W_{13} W_{14} W_{15} W_{16} W_{17} W_{18} W_{19} W_{20} W_{21} W_{22} W_{23} W_{24} W_{25}
[CLS] After a long fight Superman saved <t> Metropolis </t> [SEP] Metropolis (1927 film) Metropolis - Hasting algorithm Metropolis (comics) [SEP]

Figure 2: Longformer-based architecture for ExtEnD. The input context and the candidate textual representations are fed to the model in the same sequence separated by a *[SEP]* special token. The mention is surrounded by two special tokens <t> and </t> and, for the sake of readability, we omit the trailing special tokens </ec>. We highlight in red the tokens with global attention. Best seen in colors.

While we use this AIDA-only training scenario, which we refer to as *AIDA*, to test the data efficiency of ExtEnD, most ED systems actually make use of additional data and information originating from Wikipedia at training time. We denote this additional training scenario where Wikipedia is part of the training resources as *Wikipedia+AIDA*. Specifically, as our system is a supervised neural classifier, we follow De Cao et al. (2021b) and utilize BLINK data (Wu et al., 2020) for ED pretraining in this setting. A brief description of each dataset follows:

i) **AIDA**: one of the largest manually annotated corpora for Entity Linking and Disambiguation. It contains 388 articles from the Reuters Corpus with 27,724 labeled mentions. The training set contains 18,448 instances, while the validation and test sets feature 4791 and 4485 samples, respectively.

ii) **MSNBC**: a small news corpus with 20 articles from MSNBC on 10 different topics. It contains 656 annotated instances.

iii) **AQUAINT**: a news corpus composed of 50 documents with news coming from the Xhinua News Service, the New York Times and the Associated Press. It contains 727 annotated instances.

iv) **ACE2004**: a manually annotated subset of the ACE co-reference data set (Doddington et al., 2004). It contains 257 annotated instances.

v) **CWEB**: a dataset automatically extracted from the ClueWeb corpus[6] by Guo and Barbosa (2018) containing English Websites, consisting of 11,154 annotated instances.

vi) **WIKI**: an automatically extracted corpus comprised of Wikipedia pages released by Evgeniy et al. (2013), with 6821 annotated instances.

vii) **BLINK**: a dataset made up of 9 million (document, entity, mention) triples automatically extracted from Wikipedia.

For each of these resources,[7] we use the preprocessed datasets, along with the mention candidate sets, made available by De Cao et al. (2021b) in the authors' official repository.[8]

[6] https://lemurproject.org/clueweb12

[7] Which are all freely available for research purposes.

[8] https://github.com/facebookresearch/

		In-domain	Out-of-domain					Avgs	
	Model	AIDA	MSNBC	AQUAINT	ACE2004	CWEB	WIKI	Avg	Avg$_{OOD}$
Wikipedia + AIDA	Ganea and Hofmann (2017)	92.2	93.7	88.5	88.5	77.9	77.5	86.4	85.2
	Guo and Barbosa (2018)	89.0	92.0	87.0	88.0	77.0	84.5	86.2	85.7
	Yang et al. (2018)	**95.9**	92.6	89.9	88.5	**81.8**	79.2	88.0	86.4
	Shahbazi et al. (2019)	93.5	92.3	90.1	88.7	78.4	79.8	87.1	85.9
	Yang et al. (2019)	93.7	93.8	88.2	90.1	75.6	78.8	86.7	85.3
	Le and Titov (2019)	89.6	92.2	90.7	88.1	78.2	81.7	86.8	86.2
	Fang et al. (2019)	94.3	92.8	87.5	91.2	78.5	82.8	87.9	86.6
	De Cao et al. (2021b)	93.3	94.3	89.9	90.1	77.3	87.4	88.8	87.8
	ExtEnD$_{Large}$ + BLINK	92.6	**94.7**	**91.6**	**91.8**	77.7	**88.8**	**89.5**	**88.9**
AIDA	De Cao et al. (2021b)	88.6	88.1	77.1	82.3	71.9	71.7	79.5	78.2
	Tedeschi et al. (2021)	**92.5**	89.2	69.5	**91.3**	68.5	64.0	79.2	76.5
	ExtEnD$_{Base}$	87.9	92.6	84.5	89.8	74.8	74.9	84.1	83.3
	ExtEnD$_{Large}$	90.0	**94.5**	**87.9**	88.9	**76.6**	**76.7**	**85.8**	**84.9**

Table 1: Results (*inKB Micro* F_1) on the *in-domain* and *out-of-domain* settings when training on the AIDA training split only (bottom) and when using additional resources coming from Wikipedia (top). We mark in **bold** the best scores and underline the second best.

Evaluation Following common practice in ED literature, results over the evaluation datasets are expressed in terms of *inKB Micro* F_1. Furthermore, to better highlight the performance on the *out-of-domain* datasets, we report both the average score over those and AIDA (Avg) and over those alone (Avg$_{OOD}$), that is, when the result on AIDA is excluded from the average.

Comparison Systems In order to contextualize ExtEnD performances within the current landscape of Entity Disambiguation, we evaluate our approach against recent state-of-the-art systems in the literature. Specifically, we consider:

- Global Models: Ganea and Hofmann (2017); Guo and Barbosa (2018); Yang et al. (2018, 2019); Le and Titov (2019); Fang et al. (2019);
- Local Models: Shahbazi et al. (2019) and Tedeschi et al. (2021);
- The auto-regressive approach proposed by De Cao et al. (2021b).

ExtEnD Setup As previously mentioned, we use the Longformer model (Beltagy et al., 2020) as our reference architecture and retrieve the pre-trained weights, for both its *base* and *large* variants, from the HuggingFace Transformers library (Wolf et al., 2020); we refer to these variants as ExtEnD$_{Base}$ (139M parameters) and ExtEnD$_{Large}$ (435M parameters). Following standard practice, we use the last encoder output for the representation of each token and a simple linear layer on top of it to compute the start and end tokens probability distributions. We use a 64-token attention window and fine-tune the whole architecture using the Rectified Adam (Liu et al., 2020) optimizer with 10^{-5} learning rate for at most 100,000 steps. We use 8 steps of gradient accumulation and batches made of a maximum of 1024 tokens. We evaluate the model on the validation dataset every 2000 steps, enforcing a patience of 15 evaluation rounds. We train every model for a single run on a GeForce RTX 3090 graphic card with 24 gigabytes of VRAM. Due to computational constraints, we do not perform any hyperparameter tuning, except for the attention window where we try [32, 64, 128], and select the other hyperparameters following previous literature. We implement our work in PyTorch (Paszke et al., 2019), using *classy*[9] as the underlying framework.

4.2 Results

We report in Table 1 (top) the *inKB Micro* F_1 score ExtEnD and its comparison systems attain on the evaluation datasets in the *Wikipedia+AIDA* setting.

Arguably the most interesting finding we report is the improvement ExtEnD achieves over its comparison systems. ExtEnD$_{Large}$ + BLINK, that is, ExtEnD$_{Large}$ pre-trained on BLINK[10] and

GENRE

[9] https://github.com/sunglasses-ai/classy

[10] We note that, due to computational and hardware constraints, we were unable to match the training configuration

then fine-tuned on AIDA, sets a new state of the art on 4 out of 6 datasets, with the only exceptions being *in-domain* AIDA and CWEB, where we fall short compared to the global model of Yang et al. (2018). On the Avg score, ExtEnD pushes performances up by 0.7 points, and this improvement becomes even more marked when considering Avg$_{OOD}$ (+1.1). These results suggest that our approach is indeed well-suited for ED and, furthermore, is particularly effective when scaling out of domain.

Additionally, we also evaluate ExtEnD on the AIDA-only training setting and compare against De Cao et al. (2021b) and Tedeschi et al. (2021), the only systems available in this setting. As shown in Table 1 (bottom), ExtEnD behaves better, with both ExtEnD$_{Base}$ and ExtEnD$_{Large}$ achieving higher Avg scores. In particular, ExtEnD$_{Base}$, which features only $149M$ parameters, fares better (by almost 5 points) than De Cao et al. (2021b), whose model parameters amount to $406M$ (2.7×). Moreover, the Avg$_{OOD}$ results, which are also higher, further confirm our previous hypothesis as regards the benefits of our approach in *out-of-domain* scalability. Paired together, these results highlight the higher data efficiency that our formulation achieves, in comparison to its alternatives.

4.3 Fine-grained Results

Inspired by standard practices in the evaluation of Word Sense Disambiguation systems (Blevins and Zettlemoyer, 2020; Barba et al., 2021a), we perform a fine-grained analysis where we break down the performances of our model into different classes of label frequency. To this end, we partition both the AIDA test set and the concatenation of all the *out-of-domain* datasets in three different subsets: i) **MFC**, containing all the instances in the test set where the target mention is associated with its most frequent candidate in the training corpus (i.e. the AIDA training split).; ii) **LFC**, containing all the instances in the test set annotated with a least frequent candidate of the target mention that appeared at least once in the training corpus; iii) **Unseen**, containing all the instances in the test set whose mention was never seen in the training corpus.

We then evaluate all the systems of the *AIDA* setting, except for De Cao et al. (2021b) for which the original model is unavailable, on these six test sets. To put the results in perspective, we introduce a simple baseline (PEM-MFC) that consists in always predicting the most frequent candidate for each mention, taking mention-candidate frequencies from Le and Titov (2018).

	In-domain			**Out-of-domain**		
Model	MFC	LFC	UNS	MFC	LFC	UNS
PEM-MFC	79.2	12.6	74.0	82.2	37.1	66.1
Tedeschi et al. (2021)	**95.8**	60.9	89.0	91.1	43.0	61.7
ExtEnD $_{Base}$	94.2	53.2	87.1	94.0	43.9	75.0
ExtEnD $_{Large}$	94.8	**62.4**	**89.1**	**94.3**	**48.1**	**77.0**

Table 2: Results (*inKB Micro* F_1) when training on the AIDA training split only, on the MFC, LFC and UNS (Unseen) partitions for both *in-domain* and *out-of-domain* settings. We mark in **Bold** the best scores.

As we can see from Table 2, PEM-MFC is a rather strong baseline, confirming the skewness of the distribution with which each mention is annotated with one of its possible candidates towards the most frequent ones. Indeed, the gap between the performances of all the models on the MFC split and the LFC split is rather large, with a difference of almost 50 points in the *out-of-domain* setting. While future works should investigate the performances on these splits more in depth, here we can see that ExtEnD$_{Base}$ and especially ExtEnD$_{Large}$ outperform their competitors in the LFC and Unseen splits, in both the *in-domain* and *out-of-domain* settings. This highlights the strong generalization capabilities of our proposed approach, which is able to better handle rare or unseen instances at the cost of only 1 point in F_1 score on the MFC of the *in-domain* setting.

5 Model Ablation

While the above-mentioned experiments showed our approach to be rather effective, we only focused on the Longformer-based architecture, to which we resorted owing to the computational challenges we mentioned in Section 3.2. We now investigate this model choice, evaluating first how the BART-based system fares. Then, we ablate the attention pattern we propose for the Longformer and, finally, discuss the trade-off between our two proposed architectures.

BART Strictly speaking, the results we reported in the previous Section are not exactly conclusive as to whether or not our formulation is beneficial. Indeed, while it is true that we use a new formulation,

of De Cao et al. (2021b) and our pre-training performed a significantly smaller number of updates. The scores reported here are therefore likely to be higher.

we also rely upon a Transformer model that none of our comparison systems considered. Therefore, to better pinpoint the origin of the improvements, we train our BART-based architecture in the *AIDA* setting; we refer to this model as BART. Note that the underlying Transformer is identical to that of De Cao et al. (2021b), except for the final classification heads.[11] As shown in Table 3, BART with our extractive formulation attains significantly better performances. This finding suggests that the overall improvement does indeed originate from our extractive formulation. Furthermore, as the two systems are entirely identical except for the framing adopted, this finding further underlines the data efficiency of our approach.

Longformer Ablations We now compare our chosen global attention strategy with two standard alternatives. First, we consider the schema originally proposed by Beltagy et al. (2020) for question-answering tasks, where all the tokens in the input query (i.e. the text containing the mention) have a global attention (Longformer$_{query}$). Then, we compare against an EXTEND variant where the only token with global attention enabled is the start of sequence token (i.e. *[CLS]*). Table 3 shows how the three systems behave, reporting both their *in-domain* and *out-of-domain* scores, along with the average percentage of tokens in the input sequence with global attention enabled (GA%). From these results, we can see that i) our approach fares the best and that ii) Longformer$_{CLS}$ achieves performances almost in the same ballpark, making it a viable option for more computationally limited scenarios.

BART and Longformer Finally, we compare our two architectures. As we can see from Table 3, BART performs better in the *in-domain* dataset, whereas the Longformer outperforms it in the *out-of-domain* setting. Nevertheless, neither of these differences is very significant and, thus, this result confirms our initial hypothesis that using our second architecture is a valid approximation of the standard quadratic attention strategy for the extractive Entity Disambiguation task.

Model	**In-domain**	**Out-of-domain**	**GA%**
De Cao et al. (2021b)	88.6	78.2	100.0
EXTEND	90.0	**84.9**	21.1
Longformer$_{query}$	89.2	84.1	43.3
Longformer$_{CLS}$	88.8	84.3	0.8
BART	**90.4**	84.5	100.0

Table 3: Results (*inKB Micro* F_1) of the ablation study for the *in-domain* and *out-of-domain* settings along with the percentage of global tokens (GA%). We mark in **Bold** the best scores.

6 Error Analysis

To further investigate the generalization capabilities of EXTEND, we performed a black-box testing (Ribeiro et al., 2020) of our system leveraging the available test sets. Apart from the problem of label frequencies (e.g. unseen entities), we discovered two additional main classes of errors, namely i) *insufficient context*, and ii) *titles alone might not be enough*.

Insufficient Context Since the average number of candidates for each mention is roughly 50, the probability of having multiple valid candidates given the input context is far from negligible. For instance, let us consider the following example: *"In the last game Ronaldo scored two goals despite coming from a bad injury."*. In this sentence, the mention *Ronaldo* can refer both to Cristiano Ronaldo, the Portuguese player, and to Ronaldo de Lima, the Brazilian player. While this particular problem holds for several instances in the test sets, the performance drop is, in fact, mitigated by the labels skewness towards the most frequent candidates. Indeed, the model appears always to predict the most frequent candidate for this kind of instance, therefore being right in the majority of cases.

Titles might not be enough For both comparability and performance purposes, the text representation we use for a given entity in this work is simply its Wikipedia title. While article titles in Wikipedia are rather informative, in several circumstances they do not contain enough information to make them sufficiently distinguishable from other candidates. For example, several pages describing "Persons" are entitled just with their respective names and surnames. This kind of identifier is especially ineffective if the mentions taken into consideration were not present in the training dataset, or were rare or unseen during the underlying Transformer pre-

[11] The model of De Cao et al. (2021b) has a single head on the whole output vocabulary, whereas we have two (start and end).

training. To this end, we strongly believe that future research might benefit from focusing on enriching entities' identifiers by adding a small description of the articles (summary) or at least some keyword representing the domain the entity belongs to.

7 Conclusion

In this work we presented ExtEnD, a novel local formulation for ED that frames this task as a text extraction problem: given as input a string containing a marked mention in context and the text representation of each entity in its candidate set, a model has to extract the span corresponding to the text representation of the correct entity. Together with this formulation, we also presented two Transformer models that implement it and, by evaluating them across several experiments, we found our approach to be particularly suited to ED. First, it is extremely data efficient, surpassing its alternatives by more than 6 F_1 points when considering an AIDA-only training setting. Second, pre-training on BLINK data enables the model to set a new state of the art on 4 out of 6 benchmarks under consideration and yield average improvements of 0.7 F_1 points overall and 1.1 F_1 points when focusing only on *out-of-domain* evaluation datasets.

As future work, we plan to relax the requirements towards the candidate set and explore adapting this local formulation to a global one, so as to enforce coherence across predictions. For instance, we believe integrating the feedback loop strategy we proposed in Barba et al. (2021b) would be an interesting direction to pursue.

Acknowledgments

The authors gratefully acknowledge the support of the ERC Consolidator Grant MOUSSE No. 726487.

This work was partially supported by the MIUR under the grant "Dipartimenti di eccellenza 2018-2022" of the Department of Computer Science of Sapienza University.

References

Edoardo Barba, Tommaso Pasini, and Roberto Navigli. 2021a. ESC: Redesigning WSD with extractive sense comprehension. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4661–4672, Online. Association for Computational Linguistics.

Edoardo Barba, Luigi Procopio, and Roberto Navigli. 2021b. ConSeC: Word sense disambiguation as continuous sense comprehension. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1492–1503, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One SPRING to rule them both: Symmetric AMR semantic parsing and generation without a complex pipeline. In *Proceedings of AAAI*.

Terra Blevins and Luke Zettlemoyer. 2020. Moving down the long tail of word sense disambiguation with gloss informed bi-encoders. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1006–1017, Online. Association for Computational Linguistics.

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

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1994. Signature verification using a "siamese" time delay neural network. In *Advances in Neural Information Processing Systems*, volume 6. Morgan-Kaufmann.

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

Razvan Bunescu and Marius Paşca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 9–16, Trento, Italy. Association for Computational Linguistics.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021a. Highly parallel autoregressive entity linking with discriminative correction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7662–7669, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021b. Autoregressive entity retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

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

George R. Doddington, Alexis Mitchell, Mark A. Przybocki, Lance A. Ramshaw, Stephanie M. Strassel, and Ralph M. Weischedel. 2004. The automatic content extraction (ACE) program - tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004, May 26-28, 2004, Lisbon, Portugal*. European Language Resources Association.

Gabrilovich Evgeniy, Ringgaard Michael, and Subramanya Amarnag. 2013. FACC1: Freebase annotation of clueweb corpora, version 1 (release date 2013-06- 26, format version 1, correction level 0).

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 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 438–447. ACM.

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.

Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning dense representations for entity retrieval. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 528–537, Hong Kong, China. Association for Computational Linguistics.

Stephen Guo, Ming-Wei Chang, and Emre Kiciman. 2013. To link or not to link? a study on end-to-end tweet entity linking. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1020–1030, Atlanta, Georgia. Association for Computational Linguistics.

Zhaochen Guo and Denilson Barbosa. 2018. Robust named entity disambiguation with random walks. *Semantic Web*, 9(4):459–479.

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.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Heng Ji and Ralph Grishman. 2011. Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1148–1158, Portland, Oregon, USA. Association for Computational Linguistics.

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

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.

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

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2020. On the variance of the adaptive learning rate and beyond. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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.

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.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Luigi Procopio, Rocco Tripodi, and Roberto Navigli. 2021. SGL: Speaking the graph languages of semantic parsing via multilingual translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 325–337, Online. Association for Computational Linguistics.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with entity modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2023–2035, Florence, Italy. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Hamed Shahbazi, Xiaoli Z. Fern, Reza Ghaeini, Rasha Obeidat, and Prasad Tadepalli. 2019. Entity-aware elmo: Learning contextual entity representation for entity disambiguation. *CoRR*, abs/1908.05762.

Simone Tedeschi, Simone Conia, Francesco Cecconi, and Roberto Navigli. 2021. Named Entity Recognition for Entity Linking: What works and what's next. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2584–2596, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, page 6000–6010. Curran Associates, Inc.

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

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

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016. Joint learning of the embedding of words and entities for named entity disambiguation. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 250–259, Berlin, Germany. Association for Computational Linguistics.

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

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

Wenpeng Yin, Mo Yu, Bing Xiang, Bowen Zhou, and Hinrich Schütze. 2016. Simple question answering by attentive convolutional neural network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1746–1756, Osaka, Japan. The COLING 2016 Organizing Committee.