ZELDA: A Comprehensive Benchmark for Supervised Entity Disambiguation

Marcel Milich
Humboldt-Universitat zu Berlin
marcelmilch@gmx.de

Alan Akbik
Humboldt-Universitat zu Berlin
alan.akbik@hu-berlin.de

Abstract

Entity disambiguation (ED) is the task of disambiguating named entity mentions in text to unique entries in a knowledge base. Due to its industrial relevance, as well as current progress in leveraging pre-trained language models, a multitude of ED approaches have been proposed in recent years. However, we observe a severe lack of uniformity across experimental setups in current ED work, rendering a direct comparison of approaches based solely on reported numbers impossible: Current approaches widely differ in the data set used to train, the size of the covered entity vocabulary, and the usage of additional signals such as candidate lists. To address this issue, we present ZELDA, a novel entity disambiguation benchmark that includes a unified training data set, entity vocabulary, candidate lists, as well as challenging evaluation splits covering 8 different domains. We illustrate its design and construction, and present experiments in which we train and compare current state-of-the-art approaches on our benchmark. To encourage greater direct comparability in the entity disambiguation domain, we open source our benchmark at https://github.com/flairNLP/zelda.

1 Introduction

Entity disambiguation (ED) is the task of disambiguating textual mentions of entities to a corresponding unique entry in a knowledge base. For instance, the entity mention "NBA" might refer to one of several organizations with this abbreviation, such as "National Basketball Association" or "National Boxing Association". ED resolves these ambiguities and creates links between a knowledge base of unique entities and the various ways an entity may be referred to in text. It is the core component in the larger task of entity linking (EL), which includes the identification of entity mentions in text, often handled by a named entity recognition (NER) system.

Recent progress in the field is driven by advances in large language models (Shen et al., 2021; Sevgili et al., 2022), pushing the scores on standard evaluation datasets to new heights. These models are typically trained in a supervised manner. Unlike many other NLP tasks with relatively few target classes, such as sentiment analysis or part-of-speech tagging, ED may have millions of target classes, since each entity in a knowledge base is modeled as a distinct class. Accordingly, most current state-of-the-art ED approaches are trained over very large amounts of annotated text data that often is automatically derived from Wikipedia.

Lack of uniformity in experimental setup. However, while a number of standard evaluation datasets exist to measure final ED accuracy, such as the AIDA-B test split of the popular AIDA dataset for newswire data (Hoffart et al., 2011), we observe that no such standardization exists for the data used to train ED systems. To illustrate this disparity, refer to Table 1 for an overview of current state-of-the-art approaches, published numbers and their respective training setups.

As Table 1 shows, approaches use different amounts of training data (ranging from 2 to 20 million "snippets" of annotated text), sourced from different Wikipedia versions using different sampling methodologies, and in some cases augmented with weak labels. Importantly, there is a stark difference in the size of the entity vocabulary for which approaches are trained, ranging from models that disambiguate a few thousand entities to models that handle over 6 million. Approaches also typically leverage so-called "candidate lists" that contain all possible disambiguation targets for textual mentions and so greatly narrow the search space. Prior work (see Section 2) has shown that each of these factors greatly influences the accuracy of an otherwise identical ED system (Broscheit, 2019; Wu et al., 2020; Févry et al., 2020; Orr et al., 2021; De Cao et al., 2021).

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 2061–2072

Approach	Training data	Weak labels?	Data source	Additional features	Entity vocab	Candidate lists	AIDA-B
Yamada et al. (2022)	∼10M snippets	no	Wikipedia (Dec, 2018)				
- LUKE$_{GH}$					128k	GH	92.4
- LUKE$_{PPR}$					128k	PPR	94.6
- LUKE$_{GH+AIDA}$				+AIDA-TRAIN	128k	GH	95
- LUKE$_{PPR+AIDA}$				+AIDA-TRAIN	128k	PPR	97.1
Barba et al. (2022)	9M snippets†	no	Wikipedia (May, 2019)				
- EXTEND				+AIDA-TRAIN	1.5M	GH	92.6
Ayoola et al. (2022)	∼20M snippets	yes	Wikipedia (July, 2021)	+KB +descriptions +types			
- AYOOLA					6.2M	GH	90.4
De Cao et al. (2021)	9M snippets†	no	Wikipedia (May, 2019)				
- GENRE					1.5M	GH	89.3
- GENRE$_{+AIDA}$				+AIDA-TRAIN	1.5M	GH	93.3
- GENRE$_{+AIDA-NOC}$				+AIDA-TRAIN	1.5M	*none*	91.2
Orr et al. (2021)	5.7M sentences	yes	Wikipedia (Nov, 2019)	+KB +types			
- BOOTLEG				+AIDA-TRAIN	3.3M	PPR+*custom*	96.7
Févry et al. (2020)	17.5M snippets	no	Wikipedia (Apr, 2019)	+AIDA-TRAIN			
- FEVRY$_{HF}$					5.7M	HF+*custom*	92.5
- FEVRY$_{PPR}$					5.7M	PPR+*custom*	96.7
Broscheit (2019)		yes	Wikipedia (June, 2017)	+AIDA-TRAIN			
- BROSCHEIT$_{700k}$	8.8M snippets				700k	*none*	78.8
- BROSCHEIT$_{500k}$	2.4M snippets				500k	*none*	87.9

Table 1: Differences in the signal used to train current state-of-the-art ED approaches and their reported accuracy on the AIDA-B evaluation dataset. Differences include: the number of snippets used to train each approach, the definition of what constitutes a "snippet", the Wikipedia version the data is sourced from, and -importantly- the size of the entity vocabulary and quality of the candidate lists used ("HF" are lists by Hoffart et al. (2011), "GH" lists by Ganea and Hofmann (2017), and "PPR" lists by Pershina et al. (2015)).

Lack of direct comparability. With this paper, we argue that these differences in training setup impair our ability to directly compare approaches based solely on published numbers on evaluation datasets. For instance, Table 1 shows that LUKE (Yamada et al., 2022) slightly outperforms the comparatively simple approach by FEVRY (Févry et al., 2020) on AIDA-B; but since FEVRY is trained to cover a much larger set of entities, we cannot know if the difference in evaluation score is due to algorithmic differences in both approaches, or simply a function of the signal used to train them.

Contributions. We argue that -much like in most other NLP tasks- we require a uniform experimental setup to evaluate large ED models. To this end, we present ZELDA, a comprehensive benchmark for supervised entity disambiguation. The benchmark consists of 95k full text paragraphs from Wikipedia, annotated with mention boundaries and disambiguation targets, and integrates 8 existing ED datasets from various domains as evaluation splits. ZELDA defines a fixed entity vocabulary of 822k entities, together with fixed candidate lists and entity descriptions. In this paper:

1. We analyze training setups in recent state-of-the-art ED approaches, and derive desiderata for a uniform training benchmark (Section 2).
2. We present the ZELDA benchmark, the design goals that inform our sampling methodology to create it, and its properties (Section 3).
3. We compute evaluation scores for baselines and three state-of-the-art approaches to present standardized scores and illustrate the usefulness of our benchmark (Section 4).
4. We make our benchmark available to the research community as an open source project.

We hope that the public release of ZELDA will encourage future ED works to leverage our benchmark, and thus facilitate greater direct comparability of future ED approaches.

2 Analysis of Training Setups

ED approaches employ large language models to embed an entity mention, its textual context and additional features. To decode, approaches typically either use variants of softmax classification (Broscheit, 2019; Févry et al., 2020; Yamada et al., 2017; Orr et al., 2021; Ayoola et al., 2022), generative decoding (De Cao et al., 2021) or retrieval-based models that compute the pairwise similarity of an embedded mention and a textual description for each target entity (Ravi et al., 2021).

Rather than focus on the algorithmic differences of these approaches, this section analyzes current state-of-the-art approaches from the point of view of their respective training setups.

2.1 Training Data

Size of training dataset. Approaches are typically trained over short snippets of text with annotated entity mentions, derived from Wikipedia page links. The number and length of these snippets varies greatly across approaches. For instance, as Table 1 shows, Ayoola et al. (2022) train their model with 20 million snippets of 512 tokens length, while De Cao et al. (2021) train with 9 million snippets of 100 token length. Orr et al. (2021) train on single sentences only, and use a comparatively small set of 5.7 million snippets. The sampling methodology to derive these snippets from Wikipedia is seldom described in detail and bespoke to each paper.

While we could find few ablation experiments in prior work, Broscheit (2019) presents an experimental evaluation in which he trains his proposed approach over two different datasets sampled from Wikipedia, one with 8.8 million and one with 2.4 million snippets. The difference in dataset size is due to a threshold parameter for frequent entities in his sampling method. Surprisingly, he finds that the model trained on the smaller dataset yields significantly better results on AIDA-B. He believes this may be because his computational resources limited training on the large dataset to only 4 epochs, whereas on the small dataset he could train for 14.

Single-mention vs multi-mention data. The number of snippets is only partly illustrative of the training signal, as the number of mentions dramatically differs per setup. In "single-mention" data as used by Barba et al. (2022) and De Cao et al. (2021), each snippet only contains a single annotated entity mention (indicated with a *dagger* asterisk in Table 1). In "multi-mention" data on the other hand, more than one mention might be annotated, thus potentially greatly increasing the training signal.

Optional augmentation with weak labels. One particularity of Wikipedia text is that within an article, usually only the first mention of an entity is marked with a page link. In fact, Orr et al. (2021) estimate that 68% of mentions are unlabeled. For this reason, many works use "weak labeling" methods to annotate unlabeled mentions (Orr et al., 2021; Ayoola et al., 2022; Broscheit, 2019) in Wikipedia articles. These methods dramatically increase the number of labeled mentions per text snippet, but may introduce errors into the training data. This naturally impacts the performance of ED: for instance, Orr et al. (2021) find that their model performs better on rare and unseen entities but worse on frequent ones when using weak labels.

Wikipedia version of training data. Table 1 also shows that the data is sourced from different Wikipedia versions. This poses problems as the entity set covered by Wikipedia grows significantly over time (Gillick et al., 2019). Further, the information contained in Wikipedia is constantly updated (such as which person currently holds which political office), potentially giving advantages to models trained on a Wikipedia version from a similar point in time as the evaluation data.

2.2 Entity Vocabulary

As Table 1 shows, published approaches also differ in their entity vocabulary, i.e. the number of unique entities they can resolve, ranging from 128k (Yamada et al., 2017) to about 6 million entities (Févry et al., 2020; Ayoola et al., 2022). While a very large entity set is desirable for a general-purpose ED system, a smaller vocabulary tuned to an evaluation dataset will likely result in better evaluation numbers. This intuition is supported by experiments by Wu et al. (2020) who found that a model trained to handle an entity set specific to their evaluation dataset outperforms a general-purpose model trained to handle 5.9M Wikipedia entities.

2.3 Candidate Lists

Most state-of-the-art ED approaches employ *candidate lists* that contain for each mention string a set of sensible entity candidates (Sevgili et al., 2022).

	Domain	Doc. Type	# Docs	∅ length	# Entities	# Mentions
Evaluation Splits						
AIDA-B	news	articles	231	201 tokens	1,538	4,485
TWEEKI	tweets	short texts	500	16 tokens	639	860
REDDIT-POSTS	forum posts	short texts	377	20 tokens	524	705
REDDIT-COMMENTS	forum comments	short texts	360	41 tokens	483	638
WNED-WIKI	Wikipedia	articles	318	315 tokens	5,293	6,747
WNED-CWEB	web	pages	320	1,433 tokens	4,467	11,116
SLINKS-TOP	web	short texts	904	35 tokens	899	904
SLINKS-SHADOW	web	short texts	904	35 tokens	902	904
SLINKS-TAIL	web	short texts	902	35 tokens	902	902
Training Split						
ZELDA-TRAIN	Wikipedia	paragraphs	95k	527 tokens	822k	2.6M

Table 2: Descriptive statistics of the training and evaluation splits of ZELDA. Note that the statistics reported for evaluation splits may differ slightly from previous literature as a consequence of our normalization procedure.

The model then classifies over the small list rather than the whole entity set. The advantage of this approach is that it greatly narrows the search space and speeds up computation. A drawback however is that these candidate lists must be created separately and that incomplete lists lower the upper bound of what an ED approach can achieve: if the correct entity is not included in the candidate list of a mention, correct classification is not possible.

Prior work showed that the choice of candidate lists significantly impacts overall results. For instance, the lists of Pershina et al. (2015) were found to be exceptionally well-tailored to the AIDA-B evaluation dataset, with high recall and low ambiguity for its entities (Yang et al., 2018). As Table 1 shows, both Févry et al. (2020) and Yamada et al. (2022) find that their models improve significantly when using these lists instead of the more generic lists by Hoffart et al. (2011) and Ganea and Hofmann (2017) respectively. Unfortunately, some approaches such as Févry et al. (2020) also employ custom lists that are not released.

2.4 Domain-Specific Features

It is possible to tailor ED systems to achieve better results on individual domains.

Page titles. Févry et al. (2020) and Orr et al. (2021) disambiguate entities in news articles, and present a custom approach for constructing snippets: instead of only taking a token window around an entity mention, they also add the title and first two sentences of the article as additional context, reasoning that these texts contain salient information that pertains to the whole article. However, such custom contexts can only be defined for individual domains (e.g. tweets for instance do not have titles) and are therefore challenging to integrate for general-purpose ED systems.

Domain-specific data. In addition to large Wikipedia-derived datasets, many works also incorporate domain-specific data into their training or fine-tuning. While many available evaluation datasets are limited to small test splits, some popular datasets also define training splits. A well-known example is AIDA, which next to AIDA-B defines a train split consisting of 20k sentences and covering 30k entities. Table 1 shows (in column "additional features") that all models either present ablations in which AIDA-TRAIN is included, or include this data by default. The numbers clearly show that including domain-specific data improves overall results. However, prior works have shown that fine-tuning to a particular domain degrades performance on other datasets (Yamada et al., 2022; De Cao et al., 2021; Le and Titov, 2019).

2.5 Additional Features

Some ED approaches leverage additional sources of information (Shen et al., 2021; Sevgili et al., 2022). In particular, *entity descriptions* are concise textual summaries of the "meaning" of each entity, and a core component of all ED approaches that follow a retrieval-based approach (Ravi et al., 2021). *Entity type information* equips each entity with semantic type as additional signal. Finally, some approaches employ *knowledge base (KB)* information to decode multiple mentions in a text paragraph such that overall entity relatedness is observed (Ayoola et al., 2022; Orr et al., 2021).

3 The ZELDA Benchmark

We create ZELDA to enable analysis and direct comparison of large ED models. Refer to Table 2 for an overview. We start by selecting and normalizing appropriate evaluation datasets (Section 3.1), upon

which we define a methodology to sample training data that satisfies several objectives (Section 3.2). To ensure broad applicability, we also produce candidate lists and entity descriptions (Section 3.3).

3.1 Selection of Evaluation Splits

Desiderata. Our analysis of Section 2 showed that there are many ways to tailor the training setup to a specific evaluation dataset: one might employ domain-optimized candidate lists, include domain-specific features, use an optimized entity vocabulary, or optimize the process of sampling Wikipedia for training data. With ZELDA, we seek to minimize opportunities for such domain-specific engineering to place greater focus on evaluating algorithmic rather than engineering components. We also seek an evaluation setup that not only produces a single score, but facilitates more granular analysis of the capabilities of large ED models.

We therefore sought evaluation datasets that both span a broad range of domains (web pages, newswire text, social media) as well as isolate specific challenges in ED. To facilitate distribution, we limited our search to freely available datasets.

Selected datasets (Table 2). We chose the following 8 datasets for inclusion:

- AIDA-B is the test split of AIDA, the most commonly used ED dataset. It contains 231 manually annotated Reuters news articles.
- TWEEKI (Harandizadeh and Singh, 2020) is a collection of 500 randomly selected and hand-annotated tweets.
- Two datasets from Botzer et al. (2021), referred to as REDDIT-COMMENTS and REDDIT-POSTS respectively, that consist of top-scoring posts and comments from the internet forum Reddit. We use the "gold" subset of this dataset, i.e. all annotations in which all three annotators agreed.
- Two datasets from Guo and Barbosa (2018), referred to as WNED-WIKI and WNED-CWEB respectively, that cover the domains of Wikipedia articles and web pages. We include these datasets because they include annotation of the difficulty of each document on a scale from 0 to 1. This enables analyses of approaches as a function of estimated difficulty, as we show in Section 4.
- Three datasets from Provatorova et al. (2021), created specifically to analyze three classes of mention ambiguities: (1) SLINKS-TOP contains only easy cases in which the correct disambiguation is the most frequent sense of a mention. (2) SLINKS-SHADOW is the opposite and contains only difficult cases in which the correct disambiguation of a mention is "overshadowed" by a more popular entity. (3) SLINKS-TAIL contains only "long tail" entities that are very rare in Wikipedia.

Normalization. We unify these datasets in two ways: First, since these datasets were created at different times, we update entity annotations to the most recent version of Wikipedia (October, 2022). Second, as datasets are provided in various formats, we convert them into two commonly used standard formats, namely CoNLL and JsonL.

3.2 Training Data

Desiderata. We define a sampling methodology to create training data to balance two objectives: our first goal is to evaluate entity disambiguation for "broad entity coverage" approaches that derive large-scale training data from Wikipedia. However, if the training data is too large, model training becomes to costly for thorough analyses of design choices and hyperparameters; with the exception of the work by Févry et al. (2020), we find such analyses to be rare in current literature. For this reason, our second goal is to limit the overall size of the training dataset.

Sampling process (Algorithm 1). To balance these two goals, our sampling process starts from the vocabulary of all entities in the evaluation splits, which we refer to as the test entity set E_t. Our sampling seeks to find at least a minimal number of training examples for these entities, set by the *threshold* parameter. Following prior analyses by Vasilyev et al. (2022), we set the threshold to 10, meaning that each entity in the test set should appear at least 10 times in the training data. However, this is only possible for entities that do appear this often in the source Wikipedia data; for long-tail entities that appear fewer times, we select as many examples as possible.

Our sampling selects entire Wikipedia paragraphs for inclusion into the training dataset. The reason for choosing paragraphs as atomic document type is threefold: (1) Unlike fixed-length token windows that center on one particular entity,

paragraphs typically consist of multiple full sentences that provide natural context for entity mentions. (2) By choosing random paragraphs instead of full articles, we limit overall dataset size and introduce more textual variety as opening paragraphs of Wikipedia articles were observed to often have similar wording (Le and Titov, 2019). (3) Paragraphs contain mentions to many other entities outside of E_t. These entities are naturally skewed and added to the overall entity vocabulary of ZELDA.

Data preprocessing. We leverage the Kensho Derived Wikimedia Dataset[1], derived from the Wikipedia dump of December 2019. This dataset is preprocessed such that redirect-, disambiguation- and list-pages are removed, the text is cleaned and articles are divided into sections. Here, each section corresponds to one paragraph of text. We discard common section types that typically contain little text (such as the "Bibliography" and "External Links" sections common to Wikipedia articles). To ensure that all annotations are consistent with our evaluation splits, we update entity annotations to the most recent version of Wikipedia and discard those for which no article exists anymore.

Resulting training data. This set is randomized and sampled using Algorithm 1, yielding a training data set of 95k paragraphs spanning on average 527 tokens. It contains a total of 2.6 million mentions covering a vocabulary of 825k distinct entities, which we refer to as ZELDA-TRAIN. See Table 2 for descriptive statistics.

3.3 Additional Structured Information

We provide *candidate lists* that we derive with a general approach from the Kensho Wikimedia dataset, the Wikilinks web corpus (Singh et al., 2012) and the "also known as" information from Wikidata. For all mentions in these sources we list and count all entities that they refer to and filter entities from these lists that are not contained in the ZELDA entity vocabulary (details in Appendix A.3). Moreover we derive the most-frequent-sense baseline (MFS) by choosing, for every mention, the entity that this mention refers to the most often.

We also provide standardized *entity descriptions*: For each entity we extract the opening paragraph of its Wikipedia article as its description.

4 Experiments

We showcase the ZELDA benchmark by training a set of baselines and state-of-the-art approaches on ZELDA-TRAIN, and comparatively evaluating them on our evaluation splits.

Algorithm 1: Paragraph sampling

input : Set of sections S, test entity set E_t
output : Filtered list of sections $\hat{S}$

```
threshold ← 10;
counter_e ← 0 for e in E_t;
while E_t ≠ ∅ do
    s = random.sample(S);
    if E_t ∩ s.links ≠ ∅ then
        Ŝ.add(s);
        for e in s.links do
            counter_e ← counter_e + 1;
            if counter_e ≥ threshold then
                E_t.remove(e)
            end
        end
    end
end
return Ŝ
```

4.1 Evaluated Approaches

We compare 8 different models, as listed in Table 3:

Simple baselines. We include two baseline approaches. The first is MFS, a simple most-frequent-sense baseline that assigns each mention to its most commonly observed entity. The second is CL-RECALL, which calculates the upper bound reachable with our provided candidate lists: for each mention, the gold entity is assigned if it is included in the candidate list.

Simple softmax classifier (FEVRY). We include a reimplementation of the approach by Févry et al. (2020) in two variants: FEVRY$_{CL}$ uses our candidate lists, while FEVRY$_{ALL}$ does not use any lists to restrict the search space. The approach leverages a simple softmax classification head trained on top of a transformer model that takes as input a text snippet. Despite its simplicity, it was found to be surprisingly competitive. We reimplemented the approach as Févry et al. (2020) did not release their code. However, since our train set is much smaller we use `bert-base-uncased` instead of just a 4-layer transformer and adapt the hyperparameters to our setting (see Appendix A.1)

LUKE. We train two variants of LUKE (Yamada et al., 2022), the current state-of-the-art approach for several benchmark datasets. It trains a language model with entity embeddings, and employs a global decoding mechanism to decode mentions

[1] https://datasets.kensho.com/datasets/wikimedia

	AIDA-B	TWEEKI	REDDIT-POSTS	REDDIT-COMM.	WNED-CWEB	WNED-WIKI	SLINKS-TAIL	SLINKS-SHADOW	SLINKS-TOP	∅
Baselines										
MFS	0.635	0.723	0.834	0.81	0.612	0.651	0.994	0.149	0.413	0.647
CL-RECALL	0.911	0.94	0.984	0.983	0.924	0.988	0.988	0.567	0.731	0.891
Classification										
FEVRY$_{ALL}$	0.792	0.718	0.885	0.841	0.68	0.843	0.638	**0.434**	**0.531**	0.707
FEVRY$_{CL}$	0.795	0.769	0.89	0.865	0.703	0.845	0.876	0.319	0.477	0.727
LUKE$_{PRE}$	0.793	0.738	0.761	0.699	0.668	0.684	0.977	0.204	0.508	0.670
LUKE$_{FT}$	**0.812**	0.779	0.815	0.785	0.703	0.765	0.980	0.225	0.518	0.710
Generative										
GENRE$_{ALL}$	0.724	0.759	0.888	0.839	0.665	0.852	0.953	0.387	0.435	0.722
GENRE$_{CL}$	0.786	**0.801**	**0.928**	**0.915**	**0.736**	**0.884**	**0.996**	0.373	0.528	**0.772**

Table 3: Results of our experiments. Bold scores indicate the best scores of all the trained models. Underlined scores represent the best scores among the classification-based models.

in a given text snippet by order of confidence. Each classified mention is used as a feature to better classify the remaining mentions in a snippet. For training, they distinguish between *pre-training* in which entity embeddings are learned, and an optional final epoch of *fine-tuning* in which they are frozen. We train one model only with pre-training (LUKE$_{PRE}$) and one with fine-tuning (LUKE$_{FT}$).

We use their publicly available code[2] to train our two models. For direct comparison to the FEVRY model, we use `bert-base-uncased` instead of `bert-large-uncased` (utilized in Yamada et al. (2022)) and slightly adapt their hyperparameters to our setting, see Appendix A.1.

GENRE. We also train two variants of GENRE (De Cao et al., 2021), a generative approach that formulates ED as a sequence-to-sequence problem. A given input text with flagged mention boundaries is input, from which an entity title is generated. To ensure that the generated sequence is a valid title, GENRE uses a prefix-tree generated from all entity titles in the data to constrain the generation process. GENRE does not use candidate lists during training but in inference the prefix tree can be derived from the candidate lists. We call this variant GENRE$_{CL}$.

We use their publicly available code[3] to train our two models. Instead of the `bart-large` model we use the `bart-base` version to make the comparison more fair. We adapt the recommended hyperparameters to our setting (see Appendix A.1).

4.2 Results

Table 3 breaks down the accuracy of each model for each of the evaluation splits, and provides a single macro-averaged accuracy score for all data. We make a number of interesting observations:

Different ranking of approaches. Most importantly, we arrive at a starkly different ranking of approaches compared to published numbers on AIDA-B as listed in Table 1 where GENRE is one of the lowest-scoring models. In contrast, in our evaluation the two GENRE models clearly outperform all other considered models in most evaluation splits.

Impact of candidate lists. We note that our general-purpose candidate lists derived from Wikipedia score unevenly across domains. As our CL-RECALL baseline shows, our lists have a high upper bound on evaluation splits covering Wikipedia and social media domains, but a relatively low upper bound on splits from the domains of web pages or news text. We also note that of the classification-based approaches, only FEVRY$_{ALL}$ does not use candidate lists, but scores best.

Moreover, the overall best-scoring approach GENRE is trained without candidate lists. But during prediction, the better-scoring variant GENRE$_{CL}$ employs candidate lists, while GENRE$_{ALL}$ does not. This indicates using candidate lists only for prediction, but not training, may be a worthwhile approach to further explore.

Hard-to-disambiguate entities. On the SLINKS-SHADOW dataset of "overshadowed" entities, FEVRY$_{ALL}$ outperforms GENRE. One possible interpretation is that a generative approach naturally favors decoding into the most prominent sense, as the generated entity title will be most similar to the mention text. On the other hand, classification-based approaches are not influenced by string similarity of entity and mention text, potentially leading to better performance here.

In Table 4, we additionally list the scores on the brackets for WNED-WIKI provided by Guo and Barbosa (2018). The table shows that accuracy scores of all models steadily decrease from left (the

[2] https://github.com/studio-ousia/luke
[3] https://github.com/facebookresearch/GENRE

Approach	1	[1 - 0.9]	[0.9 - 0.8]	[0.8 - 0.7]	[0.7 - 0.6]	[0.6 - 0.5]	[0.5 - 0.4]	[0.4 - 0.3]	∅
CG-recall	0.997	0.972	0.992	0.983	0.983	0.991	0.988	0.996	0.988
FEVRY$_{CL}$	0.948	0.922	0.888	0.872	0.841	0.8	0.76	0.722	0.844
LUKE$_{FT}$	0.915	0.904	0.863	0.803	0.778	0.738	0.622	0.562	0.773
GENRE$_{CL}$	0.971	0.942	0.912	0.856	0.878	0.869	0.869	0.797	0.887

Table 4: Accuracy measured for best approaches on different *difficulty brackets* of WNED-WIKI. The lowest scores are observed for the most difficult bracket "[0.4 - 0.3]".

easiest bracket) to right (the hardest bracket). As we see this is not caused by a the CG-recall on WNED-WIKI which is independent from the brackets. This indicates that there remains much room for improving ED performance on ZELDA even when leveraging candidate lists during prediction.

4.3 Discussion

Our evaluation showed that the generative GENRE approach outperforms all classification-based approaches, and a simple direct classification approach without candidate lists as second-best performing approach overall. This indicates that equalizing the training signal, removing opportunities for domain-specific engineering, and evaluating across diverse evaluation splits may yield more insights into which algorithmic approach is best-suited to train large ED models.

However, we must also caution against overinterpreting this ranking: due to the large training times for each of these models, we did not explore any hyperparameters. Instead, we used default parameters whenever possible, and in the case of LUKE and FEVRY changed the underlying transformer to the same model, for more direct comparability. Models were only trained for as long as our computational resources allowed. Upon publication of the benchmark we anticipate that authors will explore better hyperparameters for their respective approaches, which may change the ranking (see Limitations section).

5 Related Work

Prior work has addressed aspects of evaluating ED.

Standardized evaluation. GERBIL (Röder et al., 2018) standardizes ED evaluation over multiple datasets in a unifying framework, but does not define the training data and thus only focuses on comparing already-trained models. Similarly, a range of prior works have sought to refine and standardize ED evaluation (Waitelonis et al., 2019; Nait-Hamoud et al., 2021; Noullet et al., 2021; Odoni et al., 2019; van Erp and Groth, 2020; Braşoveanu et al., 2018). In contrast, ZELDA defines the full experimental setup, including training data, the entity vocabulary and other training signals.

Manually labeled training data. A few existing ED datasets not only define a test set, but also a training split. An example discussed in this paper is the AIDA dataset. Other datasets include TAC-KBP2010 (Ji and Grishman, 2011), which is not available anymore, and a zero-shot dataset from Logeswaran et al. (2019). However, these datasets are too small and cover too few entities for evaluation of large ED approaches.

Deriving training data from Wikipedia. All current state-of-the-art approaches derive their training data from Wikipedia, though the exact process is often not thoroughly described and/or provided to the public. One exception is the BLINK corpus created by Wu et al. (2020) that is used in other works. However, this corpus consists only of single-mention snippets and cannot be used for approaches that train global decoders, like LUKE.

Most similar to our sampling method is from Orr et al. (2021). The authors sample a small Wikipedia subset by sampling for the mentions of the KORE50 benchmark (Hoffart et al., 2012). Unlike our approach, they sample all occurrences of each mention and sample only single sentences, yielding 520k sentences for only 144 mentions.

6 Conclusion

We presented the ZELDA benchmark to unify experimental setups across large ED approaches, and conducted an evaluation of various approaches. We find that given the exact same training signal, approaches compare differently than published numbers suggest. We release the datasets, our sampling and preprocessing scripts and our FEVRY reimplementation to the research community as an open source project available at `https://github.com/flairNLP/zelda`. Additionally, we integrate our benchmark into the open source NLP framework FLAIR (Akbik et al., 2019).

We hope that this will encourage present and fu-

ture ED works to compare algorithmic differences on a more equal setting and thus help generate insights to further advance the field of ED.

7 Limitations

As discussed in Section 4.3, an important limitation of our experimental evaluation is our lack of hyperparameter exploration of published approaches. Given the effort required to train large ED models and the many involved hyperparameters, we believe that only the original authors of their respective approaches can perform a meaningful search of hyperparameters for our benchmark, limiting us to best-effort parameters from prior literature. It is therefore possible if not likely that the respective authors of the approaches we compare might arrive at better numbers than the ones presented here.

Regarding the ZELDA benchmark itself, we note that it is designed to evaluate supervised ED approaches. As we sampled the dataset to contain at least 10 annotations for each entity in the ZELDA test splits whenever possible, it is unclear whether ZELDA is useful for evaluating the currently growing family of zero-shot ED models (Logeswaran et al., 2019; Wu et al., 2020). Finally, our training corpus is relatively small compared to some other Wikipedia corpora used in prior approaches. While we made this design choice purposefully to enable faster training times and hopefully more exploration of hyperparameters by future works, we cannot be certain whether rankings obtained on ZELDA transfer to approaches trained on orders of magnitude of more data.

Acknowledgments

We thank the reviewers for their helpful comments. Marcel Milich is supported the Investitionsbank Berlin through research project "AI Marketeer", confinanced by the European Regional Development Fund (ERDF). Alan Akbik is supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy "Science of Intelligence" (EXC 2002/1, project number 390523135) and the DFG Emmy Noether grant "Eidetic Representations of Natural Language" (project number 448414230).

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota. Association for Computational Linguistics.

Tom Ayoola, Joseph Fisher, and Andrea Pierleoni. 2022. Improving entity disambiguation by reasoning over a knowledge base. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2899–2912, Seattle, United States. Association for Computational Linguistics.

Edoardo Barba, Luigi Procopio, and Roberto Navigli. 2022. ExtEnD: Extractive entity disambiguation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2478–2488, Dublin, Ireland. Association for Computational Linguistics.

Nicholas Botzer, Yifan Ding, and Tim Weninger. 2021. Reddit entity linking dataset. *Information Processing & Management*, 58(3):102479.

Adrian M. P. Braşoveanu, Giuseppe Rizzo, Philipp Kuntschik, Albert Weichselbraun, and Lyndon J. B. Nixon. 2018. Framing named entity linking error types. In *LREC*.

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

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

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

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.

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

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

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

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

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.

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

Phong Le and Ivan Titov. 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.

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.

Mohamed Cherif Nait-Hamoud, Fedoua Lahfa, and Abdellatif Ennaji. 2021. A step further towards a consensus on linking tweets to wikipedia. *Evolutionary Intelligence*, pages 1–16.

Kristian Noullet, Samuel Printz, and Michael Färber. 2021. Clit: Combining linking techniques for everyone. In *ESWC*.

Fabian Odoni, Adrian M. P. Braşoveanu, Philipp Kuntschik, and Albert Weichselbraun. 2019. Introducing orbis: An extendable evaluation pipeline for named entity linking performance drill-down analyses. *Proceedings of the Association for Information Science and Technology*, 56.

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

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

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

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

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

Özge Sevgili, Artem Shelmanov, Mikhail Arkhipov, Alexander Panchenko, and Chris Biemann. 2022. Neural entity linking: A survey of models based on deep learning. *Semantic Web*, 13(3):527–570.

Wei Shen, Yuhan Li, Yinan Liu, Jiawei Han, Jianyong Wang, and Xiaojie Yuan. 2021. Entity linking meets deep learning: Techniques and solutions.

Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. 2012. Wikilinks: A large-scale cross-document coreference corpus labeled via links to Wikipedia. Technical Report UM-CS-2012-015.

Marieke van Erp and Paul Groth. 2020. Towards entity spaces. In *LREC*.

Oleg Vasilyev, Alex Dauenhauer, Vedant Dharnidharka, and John Bohannon. 2022. Named entity linking on namesakes.

Jörg Waitelonis, Henrik Jürges, and Harald Sack. 2019. Remixing entity linking evaluation datasets for focused benchmarking. *Semantic Web*, 10:385–412.

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. 2017. Learning distributed representations of texts and entities from knowledge base. *Transactions of the Association for Computational Linguistics*, 5:397–411.

Ikuya Yamada, Koki Washio, Hiroyuki Shindo, and Yuji Matsumoto. 2022. Global entity disambiguation with BERT. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3264–3271, Seattle, United States. Association for Computational Linguistics.

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

A Appendix

A.1 Training Parameters and Times

Our training parameters are informed by recommended parameters of prior works, adapted to the smaller training dataset size of ZELDA-TRAIN. In some cases, we adapted parameters across approaches for greater comparability, for instance by using the same transformer model for both LUKE and FEVRY. Across all approaches, we use the following parameters: We train all models for 6 epochs with a mini-batch size of 64 and a learning rate of 5e-5. The remaining hyperparameters are specific to each model:

FEVRY. The remaining parameters in FEVRY follow the recommendations from the paper, i.e. we use the Adam optimizer (Kingma and Ba, 2015) with a linear warmup for the first 10% of training and gradient clipping. However, the smaller dataset allowed us to look at more context. We split paragraphs of ZELDA into snippets of 400 tokens (Fevry: 256) and use an entity embedding size of 200 (Fevry: 256). Training FEVRY$_{ALL}$ took around 5h per epoch and for FEVRY$_{CL}$ 2.5h per epoch on a single Nvidia 3090ti GPU, respectively.

LUKE. We run experiments with both one-stage and two-stage training in LUKE. In one-stage training, we use for all epochs the same learning rate and do not fix the transformer weights. The entity masking rate is set to 30%. In two-stage training (LUKE$_{CL}$), we do fine-tuning in the last training epoch where we fix the entity embeddings and set the entity masking rate to 90%. To ensure comparability to FEVRY we set the entity embedding size to 200. For the remaining parameters we stick to the ones of the original LUKE which can be found in detail in table 4 and 5 of Yamada et al. (2022). Paragraphs of ZELDA are divided into snippets with ≤ 512 tokens. Training LUKE took roughly 2h per epoch on a single Nvidia 3090ti GPU.

GENRE. Apart from the parameters that we already discussed, we take all default parameters from the original paper. The parameters can best be found in the released code[4]. Since GENRE takes much longer to train than the other models and processes mentions individually we gave the model less context: We split context 500 chars to the left and 500 chars to the right of each mention (a context of roughly 190 tokens). At inference we use a beam size of 10 and a maximum number of 15 decoding steps as in the original paper. Training GENRE took around 16 hours on two Nvidia 3090ti GPUs per epoch.

A.2 Model Parameters

Our models have the following number of parameters: Both LUKE and FEVRY use a `bert-base-uncased` transformer model (110M parameters), a projection layer ($768 \times 200 \approx 153$k parameters) and the entity embedding layer (200×825k ≈ 165M parameters), and thus have about 274M parameters in total. GENRE adds a decoder with $768 \times 51197 \approx 39$M parameters to its underlying transformer and thus has a total of 178M parameters.

A.3 Candidate Lists

We derive the candidate lists with a straightforward approach from three sources: Wikipedia Kensho, WikiLinks and Wikidata. The first two are text corpora with entity annotations derived from page links. As each page link has a mention string (the so-called "anchor text") and a target Wikipedia

[4] https://github.com/facebookresearch/GENRE/blob/main/scripts_genre/train.sh

page, we can simply go through both datasets and collect all mentions and their targets. To ensure that the entity titles are up-to-date, we check with calls to the Wikipedia API if the titles lead to an existing Wikipedia page and discard them if not. This yields a set of [mention, entity] tuples. We aggregate and count these tuples.

Using the Wikidata API, we retrieve for each entity in our vocabulary the corresponding Wikidata page. From this page, we extract aliases from the "also known as" field. We interpret all aliases as additional mentions to an entity, leading to another set of [mention, entity] tuples that we aggregate with the first list. To cover a broader range of mentions we add the lower cased version and version without blanks and special characters of each mention to the tuple set.