Evaluating Design Decisions for Dual Encoder-based Entity Disambiguation

Susanna Rücker
Humboldt-Universität zu Berlin
susanna.ruecker@hu-berlin.de

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

Abstract

Entity disambiguation (ED) is the task of linking mentions in text to corresponding entries in a knowledge base. Dual Encoders address this by embedding mentions and label candidates in a shared embedding space and applying a similarity metric to predict the correct label. In this work, we focus on evaluating key design decisions for Dual Encoder-based ED, such as its loss function, similarity metric, label verbalization format, and negative sampling strategy. We present the resulting model VERBALIZED, a document-level Dual Encoder model that includes contextual label verbalizations and efficient hard negative sampling. Additionally, we explore an iterative prediction variant that aims to improve the disambiguation of challenging data points. To support our analysis, we first conduct comprehensive ablation experiments on specific design decisions using AIDA-Yago, followed by large-scale, multi-domain evaluation on the ZELDA benchmark.

1 Introduction

Entity disambiguation (ED) is the task of resolving ambiguous mentions of named entities in text to their corresponding entries in a predefined knowledge base (KB), such as Wikipedia or Wikidata. The ability to correctly link mentions of entities (e.g., "Einstein" or "Princeton") to their respective KB entries is crucial for downstream tasks such as knowledge graph construction, question answering, and information retrieval. Formally, for a given set of entity mentions $\mathcal{M} = \{m_1, \ldots, m_T\}$ in corpus $\mathcal{D}$, entity disambiguation aims to link each mention m_t to its corresponding gold entity e_t from a set of possible entities $\mathcal{E} = \{e_1, \ldots, e_{|\mathcal{E}|}\}$.

Traditional ED systems often operate on lexical similarity, link popularity, hand-crafted features, and candidate lists (Ganea and Hofmann, 2017; Yamada et al., 2016) or simple classification-based approaches (Broscheit, 2019; Févry et al., 2020) that involve fine-tuning a classification head on top of a pre-trained language model. More recent approaches are dense retrieval based, with the Dual Encoder (Gillick et al., 2019; Wu et al., 2020; Procopio et al., 2023; Wang et al., 2024) as one of the most popular architectures. It encodes mentions and KB entries into a shared vector space for similarity-based matching.

However, despite its simplicity, the Dual Encoder architecture involves key design choices that can greatly influence its ability to properly disambiguate entities. For instance, key questions include how to best verbalize labels and how to model similarity. Furthermore, training is greatly affected by decisions on negative sampling, loss functions and efficient label embedding strategies.

Contributions. Our work aims at evaluating those key design decisions for Dual Encoder models systematically. We further introduce VERBALIZED as a resulting system, which integrates document-level processing, refined label verbalizations, hard negative sampling and efficient label embedding updates[1]. Additionally, we conduct exploratory experiments with an iterative prediction and verbalization strategy that leverages already predicted neighboring mentions. This approach aims to improve contextual understanding and address challenging ambiguous cases.

In more detail, our contributions are:

- We present VERBALIZED, a dual encoder architecture for Entity Disambiguation that uses label verbalizations and efficient hard negative sampling, without relying on candidate lists.
- We conduct extensive experiments to evaluate our design choices on the AIDA-Yago benchmark and compare the resulting model to several other ED models on the much larger ZELDA benchmark.

[1]The code for VERBALIZED and label verbalizations are available at https://github.com/flairNLP/VerbalizED.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 15685–15701

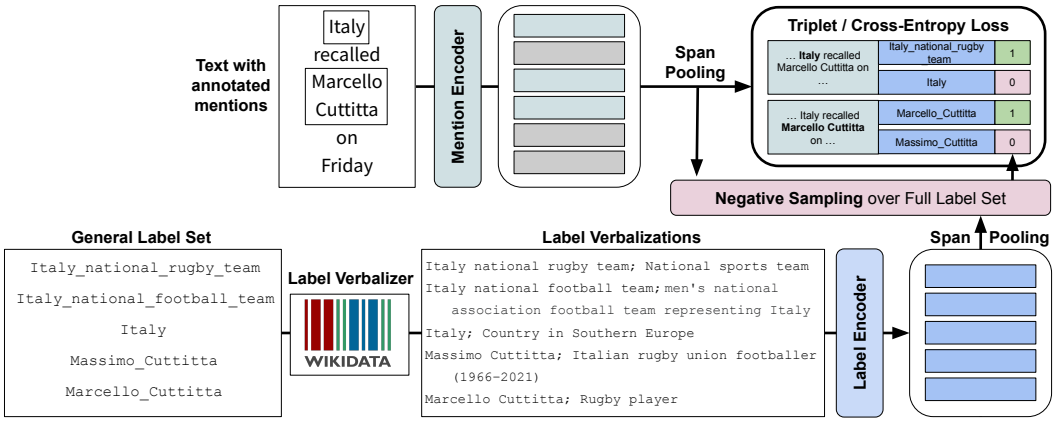

Figure 1: Overview of VERBALIZED during training: The Mention Encoder produces an embedding for each entity mention in a given text (here "Italy" and "Marcello Cuttitta"). The Label Encoder similarly produces an embedding for each unique target in the General Label Set (spanning entities such as "Italy" and "Italy_national_football_team"), by embedding their respective verbalizations. The purpose of training is to learn an embedding space in which mention embeddings lie close to the embeddings of the correct target verbalization. Training uses a Negative Sampling strategy which leverages embeddings to find hard negatives.

- We introduce a variant of VERBALIZED that predicts mentions iteratively, leveraging early disambiguations to assist in resolving challenging cases and show qualitative insights.

2 The VERBALIZED Architecture

This section outlines the Dual Encoder architecture and its key techniques and design decisions.

2.1 Dual Encoder Basics

Figure 1 gives an overview of the Dual Encoder and its main components during training:

Mention Encoder. The Mention Encoder processes the textual context surrounding an entity mention. For instance, in a document or sentence containing the mention "Italy", the encoder considers the surrounding words to generate a contextually rich embedding for the mention. Our approach leverages the entire document as context during mention encoding, ensuring richer semantic information for each mention.

Label Encoder. The Label Encoder generates embeddings for all entities using metadata such as descriptions and KB relations for accurate representation. The label set may e.g. include "Italy" (the country), "Italy_national_rugby_team" and "Italy_national_football_team". The Label Verbalizer produces short descriptions for each, like verbalizing "Italy" as "Country in Southern Europe", which is then encoded into an entity embedding.

Similarity Computation. The embeddings from the Mention and Label Encoder are pooled to obtain span representations and then compared using a similarity metric (e.g., cosine similarity or Euclidean distance). The entity whose embedding is most similar to the mention embedding is selected as the predicted label. The purpose of training is thus to learn an embedding space in which mention embeddings lie close to the embedding of the correct target verbalization.

Sampling Negatives for Training. To improve disambiguation and training robustness, we incorporate negative sampling over the label pool, so the model learns to distinguish correct labels from hard negative candidates. For example, for the mention "Italy" with gold label "Italy_national_rugby_team", a close but incorrect label like "Italy" (country) serves as negative label.

Inference. Inference involves embedding mentions and entities, comparing embeddings in the shared space and selecting the most similar match.

2.2 Design Decisions for the Dual Encoder

The effectiveness of the Dual Encoder model relies on several key design choices, which we summarize here before providing detailed discussion and evaluation for each in Sect. 3.

Enriching representations. One of the most crucial aspects is creating high-quality entity representations. Expressive **label verbalizations**, like descriptions or structured KB data, enrich entity em-

Component	Albert_Einstein	Wembley_Stadium
Title	Albert Einstein	Wembley Stadium
Description	German-born theoretical physicist (1879–1955)	football stadium in London, England
Categories	occupation: physicist, scientist	instance of: multi-purpose sports venue; country: United Kingdom
Paragraph	Albert Einstein was a German-born theoretical physicist who is best known for developing the theory of relativity. [...]	Wembley Stadium is an association football stadium in Wembley, London. It opened in 2007 on the site of the original Wembley Stadium [...]

Table 1: Examples for different components for creating label verbalizations.

Verbalization	F1
Title	63.68 ± 0.05
Title + Categories	64.00 ± 0.13
Title + Description	64.48 ± 0.10
Title + Description + Categories	**65.01 ± 0.08**
Title + Paragraph (100)	64.30 ± 0.02
Title + Paragraph (500)	63.49 ± 0.50

Table 2: Comparing verbalizations formats.

Loss	**Pooling**	F1
Triplet	Mean	64.48 ± 0.10
	First-last	**66.25 ± 0.40**
Cross-Entropy	Mean	65.84 ± 0.22
	First-last	**66.66 ± 0.09**

Table 3: Comparing span pooling methods.

beddings and help disambiguate context-sensitive mentions. These work alongside document-level mention representations to ensure rich semantic understanding. Both encoders require an effective **pooling strategy**, such as mean pooling, to obtain concise representations per mention or label.

Training Dynamics. Additionally, a suitable **similarity metric** is essential, as is a **loss function** to optimize the embedding space, i.e. pulling positive mention-entity pairs closer while pushing negatives away. Both require **negative samples**: Hard negatives – incorrect entities similar to the mention – help improve fine-grained differentiation, while in-batch negatives offer computational efficiency. For negative sampling, cached entity embeddings must be efficiently updated to reflect model changes, with the **update frequency** managed to balance accuracy and computation.

3 Evaluating Design Choices

We conduct several ablation experiments to assess the impact of the design choices sketched in Sect 2.2. Due to limited computational resources, these experiments were performed using the smaller AIDA-CoNLL-Yago (Hoffart et al., 2011) as train set, while evaluation was carried out on the diverse, out-of-domain ZELDA test sets (Milich and Akbik, 2023). The only exception is the ablation for label update frequency (Sect. 3.5), where we also trained on ZELDA. We report the mean F1 over ZELDA-test with standard deviation of three runs. Unless otherwise specified, the default experimental setup uses Title + Description for verbalization, triplet loss, Euclidean distance, hard negative mining with a dynamic factor, and mean pooling, while varying the options of the respective design choice.

3.1 Label Verbalization Formats

Design Choices. We evaluate different verbalization formats and use Wikidata, similar to prior work (Procopio et al., 2023; Atzeni et al., 2023). Table 1 illustrates the verbalizations we consider: (1) The entity's **Title**, (2) a short **Description**, or (3) more structured **Categories** using the *instance_of*, *subset_of*, *country*, and *occupation* relations from Wikidata. We further experiment with (4) using the first Wikipedia **Paragraph**. We test various component combinations, using a semicolon after the title and commas as separators. Verbalizations have a soft 50-character limit, splitting at the next punctuation. For paragraphs, we allow lengths of 100 or 500 characters.

Experimental Analysis. Table 2 shows the results. Using only the title performs worst but remains competitive, suggesting that individual missing descriptions do not severely impact performance. Descriptions slightly outperform categories, but combining all three yields the best results. Descriptions add detail, while categories provide structure for better generalization. Since some entities have only descriptions (2.1%) or categories (3.3%), both are essential for full coverage. Wikipedia paragraphs perform worse, especially at the longer length of 500 characters.

Loss	Similarity	F1
Triplet	Cosine	50.65 ± 0.20
	Dot Product	64.43 ± 0.05
	Euclidean	64.48 ± 0.10
Cross-Entropy	Cosine	34.34 ± 0.25
	Dot Product	64.52 ± 0.04
	Euclidean	**65.84 ± 0.22**

Table 4: Comparing loss and similarity metrics.

Loss	Negatives	F1
Triplet	In-Batch, dyn	54.39 ± 0.08
	Hard, 1	64.46 ± 0.06
	Hard, dyn	**64.48 ± 0.10**
Cross-Entropy	In-Batch, dyn	54.06 ± 0.14
	Hard, 1	65.78 ± 0.17
	Hard, dyn	**65.84 ± 0.22**

Table 5: Comparing negative sampling methods.

3.2 Span Pooling Method

Design Choices. For both mention and label span representations, we evaluate two pooling methods: Taking the **mean** of the token embeddings within the mention span or label, or concatenating the embeddings of the **first and last** tokens of the mention span or label. For label verbalizations, we use only title tokens, computing either their mean or concatenating the first and last token, while treating the rest (e.g. the description) as context, mirroring mention token processing.

Experimental Analysis. Table 3 shows that concatenating the embeddings of the first and last span token consistently performs better than averaging all span tokens. The first and last tokens often encapsulate critical boundary information of the span, which can be especially helpful for disambiguation.

3.3 Similarity Metric and Loss

Design Choices. The choice of similarity metric significantly impacts model effectiveness. We experiment with three options: **Cosine similarity**, which measures the angle between vectors and works well for normalized embeddings; **Euclidean Distance**, measuring the straight-line distance between vectors, used as a negative since our model is similarity-based; and **Dot Product**, which directly computes unnormalized similarity.

For optimizing the embedding space, we explore two loss functions: **Triplet Loss** pulls positive mention-label pairs closer while maintaining a margin from a given negative; **Cross-Entropy Loss** adapts the classification objective to entity disambiguation by aligning mention embeddings with correct labels and penalizing incorrect associations.

Experimental Analysis. Table 4 shows that cross-entropy loss combined with Euclidean similarity achieves best performance. For triplet loss, both dot product and Euclidean distance yield similar results. Cosine distance performs worse.

3.4 Negative Sampling Methods

Design Choices. Training with all possible entity candidates is computationally prohibitive, so we use negative sampling. For each mention, we either sample **in-batch negatives** (labels from other mentions in the batch) or **hard negatives** (incorrect entities most similar to the mention). The number of negatives per mention is another design choice. As re-encoding all labels at each step is infeasible, negatives are retrieved from periodically refreshed cached embeddings, while gold and selected negatives are freshly embedded for loss calculation.

Experimental Analysis. The results in Table 5 show that hard negative sampling significantly outperforms in-batch sampling. We compare using 1 negative label per positive sample and a *dynamic* approach, where number of negatives is maximized based on GPU memory capacity for each batch, leading to marginal improvements.

3.5 Frequency of Label Embedding Updates

Design Choices. Updating label embeddings is crucial for accurate representation, but re-encoding all labels every step is infeasible. To balance accuracy and efficiency, we cache embeddings and refresh them periodically, either **after each epoch** or at **more frequent** intervals. Additionally, labels actively used in a batch (positive or negative) are updated **on-the-fly**, keeping frequently used labels up to date without full re-encoding.

Experimental Analysis. We validate the effectiveness of frequently and dynamically updating cached label embeddings, rather than updating them only after each epoch (see Table 6). Note that this ablation was conducted training on ZELDA, as the update frequency is more impactful for larger datasets where once every epoch is not enough. The results show that more frequent label embedding updates are crucial for performance.

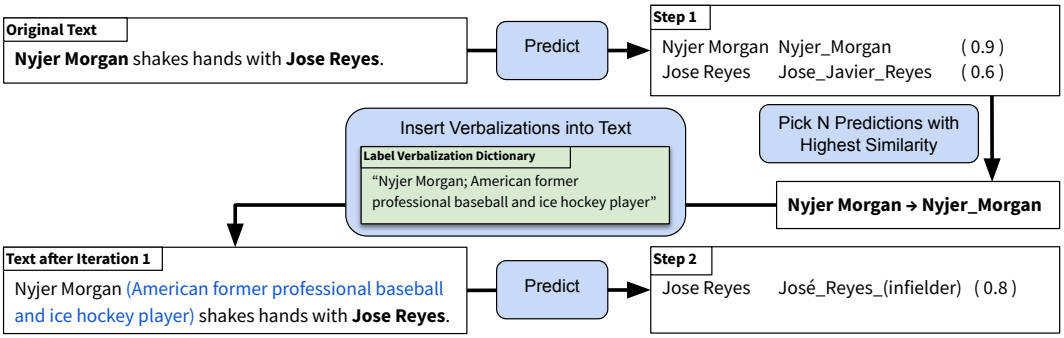

Figure 2: Iterative variant: All mentions are initially predicted, the top-N with highest similarity are selected for text insertion. The enriched text is then re-embedded, and the remaining mentions are re-predicted.

Label Embedding Updates	F1
Once after Epoch	76.17 ± 0.04
Frequent + On-the-Fly	**82.32 ± 0.10**

Table 6: Comparing label embedding update frequency. Trained on **ZELDA**.

3.6 Takeaways

We summarize the key findings from the above ablations. The best label verbalization format combines title, description, and categories, underlining the importance of semantic richness as well as coverage. For span pooling, concatenating the first and last token embeddings surpasses mean pooling. Among similarity metrics and loss functions, cross-entropy loss with Euclidean distance achieved the highest performance. Hard negatives consistently outperformed in-batch negatives. For large training data, frequent label embedding updates are crucial.

4 Iterative Prediction

We experiment with an iterative prediction variant of our base architecture that aims to particularly help challenging, ambiguous cases.

4.1 Enriching Context with Label Insertions

In this approach, illustrated in Figure 2, after predicting all mentions in a document, the N predictions with highest similarity scores are selected, and their label verbalizations are inserted into the text in brackets after their respective mentions. The modified text is re-embedded, and the remaining mentions are re-predicted. This process repeats until all mentions are resolved[2]. The goal is to incrementally enrich the context with entity descriptions, improving subsequent predictions.

In the example from Figure 2, the original text contains the mentions "Nyjer Morgan" and "Jose Reyes". In the first prediction step, the model identifies possible entities, assigning similarity scores (e.g., "Nyjer_Morgan" with 0.9 and "Jose_Javier_Reyes" with 0.6). The most confident prediction, "Nyjer_Morgan", is selected for text insertion, thus its verbalization is added to the text. In the next prediction step, this additional context helps the model correctly disambiguate "Jose Reyes" to "José_Reyes_(infielder)" with higher confidence (0.8) instead of the incorrect previous prediction, a film director. More examples of these insertions are shown in Tables 10 and 11.

4.2 Modified Training

Iterative insertion and prediction can, in theory, be directly applied during inference due to its natural language format. However, to reinforce its utility, we adapt the training process by incorporating label verbalizations for a random subset of mentions[3]. Mentions selected for verbalization are excluded from the loss calculation of the current batch, as their target label is already present in the text.

5 Experiments on ZELDA benchmark

We compare our resulting system VERBALIZED to different baseline and competitive models.

5.1 Experimental Setup

Data. The ZELDA benchmark (Milich and Akbik, 2023) addresses inconsistencies in previous ED setups by unifying training data and entity vocabularies. It includes 95,000 Wikipedia paragraphs (2.6M mentions, ~822,000 unique entities) for training and nine test splits across diverse domains to evaluate model generalizability across

[2]N is set to one-third of the mentions in a batch. We only overwrite previous labels if the similarity score is higher.

[3]Gold labels are used initially (10% randomly corrupted), later in training we switch to confident predictions.

Method	Train data	AIDA-B	TWEEKI	REDDIT-POSTS	REDDIT-COMM	WNED-CWEB	WNED-WIKI	SLINKS-TAIL	SLINKS-SHAD.	SLINKS-TOP	AVG
Classification											
FEVRY$_{ALL}$	Z	79.2	71.8	88.5	84.1	68.0	84.3	63.8	43.4	53.1	70.7
FEVRY$_{CL}$	Z	79.5	76.9	89.0	86.5	70.3	84.5	87.6	31.9	47.7	72.7
LUKE$_{PRE}$	Z	79.3	73.8	76.1	69.9	66.8	68.4	97.7	20.4	50.8	67.0
LUKE$_{FT}$	Z	81.2	77.9	81.5	78.5	70.3	76.5	98.0	22.5	51.8	71.0
Generative											
GENRE$_{ALL}$	Z	72.4	75.9	88.8	83.9	66.5	85.2	95.3	38.7	43.5	72.2
GENRE$_{CL}$	Z	78.6	80.1	92.8	91.5	**73.6**	88.4	**99.6**	37.3	52.8	77.2
Dense Retrieval											
FUSIONED	Z	80.1	**81.4**	**93.9**	**92.3**	**73.6**	89.0	98.3	41.5	57.9	78.7
Proposed											
VERBALIZED	Z	82.6	78.9	89.2	86.2	69.8	**91.4**	98.6	65.3	67.0	81.0
+ *iter. prediction*	Z	84.4	78.4	89.5	87.1	70.4	91.2	98.7	65.3	**67.2**	81.4
+ *iter. training*	Z	**88.2**	78.9	92.2	88.4	71.5	90.8	98.2	**66.3**	65.9	**82.3**

Table 7: Comparison between VERBALIZED and other SoTA models on the ZELDA benchmark.

varying text lengths and contexts. These are:

AIDA-B (Hoffart et al., 2011), the test split of the AIDA-Yago dataset, consisting of 231 news articles. **TWEEKI** (Harandizadeh and Singh, 2020), a collection of 500 tweets. **Reddit-POSTS** and **Reddit-COMMENTS**, a collection of posts and comments from the Reddit forum (Botzer et al., 2021). **WNED-WIKI** and **WNED-CWEB**, a collection of Wikipedia articles vs. web pages (Guo and Barbosa, 2017). Finally, the **Shadowlinks** corpora, three datasets with different levels of difficulty in terms of *overshadowing* (Provatorova et al., 2021). **TOP** includes the easiest cases (the correct entity is the most frequent one), **SHADOW** the most difficult (the correct entity is overshadowed by a more frequent one) and **TAIL** includes generally rare though mostly not overshadowed entities.

In addition to the main experiments on ZELDA, we further evaluate on **MSNBC** (Cucerzan, 2007), **AQUAINT** (Milne and Witten, 2008) and **ACE2004** (Ratinov et al., 2011), see Sect. A.3.

General Label Set. Comparing ED approaches is challenging due to differences in pretraining, reliance on candidate lists and selected label set (Shavarani and Sarkar, 2023; Milich and Akbik, 2023; Yamada et al., 2022; Wang et al., 2024; Ong et al., 2024). For consistency, as a fixed label set we use all 821,402 unique ZELDA candidates and no mention specific candidate list. Further training details and hyperparameters are in Appendix A.1.

Baselines and Competitive Models. We compare to the models reported by the ZELDA authors (Milich and Akbik, 2023): Their reimplementation of the classification baseline **FEVRY** (Févry et al., 2020) with and without using the mention-candidate lists (FEVRY$_{CL}$ and FEVRY$_{ALL}$). A global ED model **LUKE** (Yamada et al., 2022) in two variants: Pre-training in which entity embeddings are learned (LUKE$_{PRE}$), and an optional final epoch of fine-tuning with frozen entity embeddings (LUKE$_{FT}$). The generative model **GENRE** (De Cao et al., 2021), which generates the target label title, restricting either to the full (GENRE$_{ALL}$) or to the mention's specific candidate pool (GENRE$_{CL}$). We add **FUSIONED** (Wang et al., 2024), a novel encoder-decoder architecture to fuse entity descriptions and candidate embeddings which is also trained on ZELDA.

We further compare our approach to **BLINK** (Wu et al., 2020) and **CHATEL** (Ding et al., 2024). As differences in training data and label sets make direct comparison to our and the other models challenging, see Appendix A.3 for the results.

5.2 Results

We train VERBALIZED on ZELDA train and evaluate on its test sets (Table 7). For MSNBC, ACE2004, AQUAINT, and comparison to BLINK and CHATEL, see Tables 13 and 15 in A.3.

5.2.1 Baseline Comparison

Overall Performance. On average, VERBALIZED achieves the highest performance over the ZELDA test sets, outperforming both classification and generative approaches. The highly competitive FUSIONED is surpassed on 5 of the 9 datasets.

Strength on Shadowlinks Corpora. VERBALIZED achieves top results on SHADOW and TOP, where candidate list-based models struggle due to low recall (56.7% on SHAD, 73.1% on TOP). The

	Step 1	Step 2
Input	"Peggy Olson is awesome... one of the best characters on **#madmen**"	"Peggy Olson (fictional character from Mad Men) is awesome... one of the best characters on **#madmen**"
Prediction	Mad_(TV_series) (✗)	Mad_Men (✓)
Input	"Karlsruhe 3 (Reich 29th, Carl 44th, **Dundee** 69th) Freiburg 0. Halftime 2-0."	"Karlsruhe (German sport club) 3 (Reich 29th, Carl 44th, **Dundee** 69th) Freiburg (German football club) 0. Halftime 2-0."
Prediction	Sean_Dundee (✓)	Dundee_F.C. (✗)

Table 8: Positive and negative example of label change after iterative prediction. See more in Tables 10 and 11.

candidate-independent models also face challenges with rare, ambiguous or overshadowed mentions. These results highlight VERBALIZED's ability to effectively leverage subtle contextual cues while also demonstrating its strength in not relying on mention-specific candidates.

Performance on Long-Form Text. On AIDA-B and WNED-WIKI, VERBALIZED tops all models and remains competitive on REDDIT-POSTS – all datasets with long, continuous documents.

Challenges with Short-Form Text. VERBALIZED struggles on REDDIT-COMM and TWEEKI. This shortcoming stems from the reliance on long context, a direct consequence of its document-level encoding approach. Since mention representations include signal from the whole document, short-form social media texts offer fewer cues for disambiguation. Still, VERBALIZED performs relatively well, even surpassing other models on SHADOW, despite its single-sentence documents.

Low Performance on WNED-CWEB. On WNED-CWEB, performance is lower than expected, comparable to classification models. We attribute this to web scraping artifacts and annotation inconsistencies[4]. While all models face these challenges, disjointed documents – composed of unrelated sentences – pose a greater challenge for document-based approaches like ours, which rely on coherent context for effective predictions.

5.2.2 Evaluating the Iterative Variant

For evaluating the iterative variant of VERBALIZED, we compare: (1) the base model without iteration, (2) the base model with iterative prediction (+ *iter. prediction*), and (3) the full iterative variant incorporating both iterative training and prediction (+ *iter. training*). We observe some gains with iterative prediction on AIDA-B, REDDIT-COMM, and WNED-CWEB, while other datasets (TWEEKI, REDDIT-POSTS, WNED-WIKI, SLINKS) show no clear improvement. Adding insertions during training boosts performance. On average, and on five datasets, the iterative outperforms the base approach, though slight declines occur on WNED-WIKI, SLINKS-TAIL, and -TOP, and no difference on TWEEKI. Shadowlinks datasets contain only single-mention documents, while TWEEKI and REDDIT-COMM, based on social media, include only few multi-mention documents. Since the iterative variant requires at least two entity mentions per document to be effective – an assumption often violated in single-sentence Reddit or Tweet-based inputs – no performance gains are visible on some of these datasets (TWEEKI, SLINKS).

Qualitative Inspection. As the iterative variant showed only slight and inconsistent improvements over the base model, we qualitatively analyze iterative prediction steps. Table 8 provides examples for both positive and negative label change. As positive example, disambiguating "Peggy Olson" as a character from Mad Men aids in correctly labeling the series. Conversely, inserting two sports team labels leads to misinterpreting "Dundee" as the sports team "Dundee_F.C." instead of the person "Sean_Dundee", initially correctly predicted. Refer to A.2 for additional examples and discussion of both improved and erroneous predictions after label insertions, along with quantitative counts.

Computational Cost. To better understand the trade-offs between our base and iterative variant, we report training (in hours per ZELDA train epoch) and inference times (in seconds per document) on a single NVIDIA H100 PCIe 80GB for each variant. As shown in Table 9, the iterative variant introduces additional overhead due to the iterative re-embedding of documents after inserting the verbalizations.

We emphasize that domain-specific characteristics such as document length, entity frequency, and annotation noise impact performance and effectiveness. For instance, performance drops on WNED-

[4]E.g., our model links "William Pitt" to a person, while gold is "University_of_Pittsburgh", or "taxpayers" to "Taxpayers" and "Rose" to "Derrick_Rose," which seem more accurate than the gold labels "Tax" and the flower "Rose."

Variant	Training Time	Inference Time
VERBALIZED base	11 h/epoch	0.94 s/doc
+ *iter. prediction*	11 h/epoch	2.1 s/doc
+ *iter. training*	34 h/epoch	2.1 s/doc

Table 9: Training and inference time for model variants. Both include the time for embedding the full label set multiple times (during training) or once (inference).

CWEB are tied to non-coherent text spans and scraping noise, while both the base but even more so the iterative variant of VERBALIZED struggle with short-form text with little context or neighboring mentions.

In summary, although the iterative approach offers advantages in handling underspecified mentions and complex domains, its limitations – vulnerability to linguistic artifacts, error propagation and training overhead – tend to outweigh its modest and inconsistent benefits. Practitioners should carefully consider these trade-offs. Given the cost and the strong performance of the base VERBALIZED architecture, we recommend adopting the latter as the preferred solution.

6 Related Work

Entity Disambiguation (ED) models resolve ambiguous mentions in text to corresponding entities in a knowledge base (KB)[5]. Recent advances use transformers for better contextual representations and zero-shot abilities.

Candidates. ED is often split into candidate retrieval and ranking (Procopio et al., 2023; Wu et al., 2020; Yamada et al., 2022; Yang et al., 2019). Many methods use precompiled candidate lists (alias tables) to map mentions to small sets of potential entities, reducing the search space (Procopio et al., 2023; Yamada et al., 2022; Wang et al., 2024; Yang et al., 2019). Common lists are e.g. by Ganea and Hofmann (2017) or Le and Titov (2018).

Simple Classifiers. Early classification-based ED approaches, like Broscheit (2019) and (Févry et al., 2020), use a softmax classification head on a pretrained transformer. While effective, they are limited by the need for a fixed, small label set and sufficient training data for each label, making adaptation to new domains or labels challenging.

Dense Retrieval and Ranking. To provide more flexibility in label sets, Dense Retrieval has gained popularity, embedding mentions and entity candidates into a shared embedding space. Many models further enrich label representations by incorporating expressive label descriptions (Procopio et al., 2023; Wu et al., 2020; Provatorova et al., 2022).

Dual Encoders (or Bi-Encoders) (Gillick et al., 2019; Humeau et al., 2020) embed mentions and entities into a shared space using separate encoders, predicting labels via similarity. While often used for retrieval (Orlando et al., 2024), they can also make final predictions. **Cross Encoders** rank a small set of pre-retrieved candidates by jointly encoding mentions and candidates, capturing deeper interactions for improved ranking. Both are often combined: Efficient candidate retrieval with a Dual Encoder, then ranking via Cross Encoder, like in the BLINK model (Wu et al., 2020). For training the Bi-Encoder, they mainly use in-batch negatives, optionally adding hard ones. While BLINK's Bi-Encoder focuses on candidate retrieval and prioritizes recall, our Dual Encoder directly handles label prediction, omitting the expensive Cross Encoder step. Wang et al. (2024) introduce a novel encoder-decoder architecture FUSIONED, which employs an encoder-decoder architecture to learn interactions between the text and each candidate entity. While using entity descriptions and hard negative sampling is similar to our work, its decoder-based approach introduces more complexity.

Many approaches further enhance dense retrieval models by incorporating structural knowledge from knowledge graphs such as entity types or relations (Ayoola et al., 2022; Atzeni et al., 2023; Bouziani et al., 2024), similarly to Provatorova et al. (2022) and Tedeschi et al. (2021), who use entity type information for filtering candidates.

Global Prediction. Most ED approaches, including our base model, treat mentions individually (local ED). Our iterative variant instead inserts dynamic label verbalizations from neighboring mentions, aligning it with global ED approaches (Yamada et al., 2022; Oba et al., 2022; Xiao et al., 2023; Ganea and Hofmann, 2017; Le and Titov, 2018; Yang et al., 2018; Provatorova et al., 2022; Yang et al., 2019). LUKE (Yamada et al., 2022) proposes a global ED model that resolves mentions within a document sequentially, using previously

[5]While Entity *Disambiguation* (ED) assumes mentions already identified, Entity *Linking* (EL) also detects spans. We focus on ED, though it is often integrated into end-to-end EL or Relation Extraction systems (Shavarani and Sarkar, 2023; Bouziani et al., 2024; Orlando et al., 2024).

resolved entities as additional input tokens. Unlike our verbalization insertions, LUKE appends entity titles and relies on candidate lists.

Alternative Methods. Generative models like GENRE (De Cao et al., 2021) frame ED as sequence-to-sequence to generate label titles from an entity list. Other methods use knowledge graphs (Li et al., 2022), structured prediction (Shavarani and Sarkar, 2023), or span extraction (Barba et al., 2022). Recently, few-shot ED with LLM prompting gained more attention (Liu et al., 2024; Zhou et al., 2024; Xu et al., 2023; Ding et al., 2024).

7 Conclusion

We systematically evaluated key design choices for Dual Encoder-based entity disambiguation, focusing on loss functions, similarity metrics, label verbalization formats, negative sampling and efficient label embedding updates. Our experiments on the AIDA and ZELDA benchmarks provide valuable insights into the impact of these decisions for the effectiveness of Dual Encoder models for ED.

Based on these investigations, we introduced VERBALIZED, a system that integrates document-level processing, contextual label verbalizations, efficient hard negative sampling, and cashed label embeddings and achieves state-of-the-art performance on the ZELDA benchmark. By eliminating the reliance on candidate lists, VERBALIZED offers a scalable and flexible solution.

While our iterative prediction strategy on average improved performance, gains were inconsistent and qualitative analysis revealed some unwanted negative effects. However, with positive cases mostly prevailing, there is potential for further pursuing this variant.

Limitations

While our approach demonstrates significant advancements in entity disambiguation, several limitations remain that offer avenues for future improvement:

Limited Evaluation of Training Configurations. Due to time and resource constraints, we could not run extensive training experiments on the ZELDA dataset across multiple seeds or hyperparameter configurations and had to rely on the ablations conducted on AIDA-Yago. This limits the robustness of our reported results on this benchmark. It is possible, that the difference in size and diversity of ZELDA train, would favor different settings for certain design choices. Furthermore, there are several hyperparameters that we did not evaluate systematically, such as the margin parameter for triplet loss, which may significantly impact its effectiveness depending on the chosen similarity metric or more in-depth analysis of the effect of the number of negative samples per datapoint, length of label verbalizations or, concerning the iterative prediction variant, insertion format.

Dependency on Descriptions for Labels. Our approach relies heavily on the availability of detailed descriptions for most labels in the entity set. This dependence may restrict its applicability to domains or datasets lacking such. On the other hand, our approach *only* relies on descriptions and does not require e.g. candidate lists which also are challenging to collect. Furthermore, the title-only setting still showed reasonable results, suggesting that individual sparse cases pose minimal concern.

Training Complexity of the Iterative Model. The iterative variant of our approach for enriching entity disambiguation with contextual verbalizations shows potential, but a) requires significantly more time and computational resources to train and also for inference, and b) given the inconclusive results concerning its added value, further investigation is needed to fully understand its impact, refine its methodology and determine its usefulness.

Focus on English Datasets. Our experiments were conducted exclusively on English ED datasets, leaving the generalizability of our method to multilingual scenarios unexplored. Unfortunately, while very valuable, this was beyond the scope of this work. Thaid said, applying dual encoder systems to other languages could pose challenges including the quality of multilingual resources (e.g., Wikidata), linguistic differences which might affect span encoding method (e.g., first-last pooling), and the scarcity of large-scale annotated datasets. Addressing these will require robust multilingual embeddings and cross-lingual verbalization strategies to ensure broader applicability.

Ethical Considerations

While our approach to entity disambiguation has few direct ethical concerns, some considerations arise due to inherent limitations in language models and the reliance on external data. Human-written

label descriptions can contain errors, misinformation, and biases. Potentially, rare entities, being less documented, might be more likely to be misclassified, perpetuating disparities in representation. Additionally, the language model may perpetuate biases, such as gender stereotyping in professions or favoring frequent entities (Chen et al., 2021; Provatorova et al., 2021).

Acknowledgements

We thank all reviewers for their valuable comments. Susanna Rücker and Alan Akbik are supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Emmy Noether grant "Eidetic Representations of Natural Language" (project number 448414230). Further, 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).

We acknowledge the use of AI generative tools that assisted with language, presentation, and formulation of this paper. These tools were utilized to refine the phrasing, improve clarity, and ensure the linguistic quality of the document. The ideas, content, and research presented in this paper are entirely our own, and these tools were not involved in generating the scientific concepts or findings discussed. All concepts, methodologies, and interpretations are original and derived from our work.

References

Mattia Atzeni, Mikhail Plekhanov, Frederic Dreyer, Nora Kassner, Simone Merello, Louis Martin, and Nicola Cancedda. 2023. Polar ducks and where to find them: Enhancing entity linking with duck typing and polar box embeddings. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9129–9146, Singapore. Association for Computational Linguistics.

Tom Ayoola, Shubhi Tyagi, Joseph Fisher, Christos Christodoulopoulos, and Andrea Pierleoni. 2022. ReFinED: An efficient zero-shot-capable approach to end-to-end entity linking. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*, pages 209–220, Hybrid: Seattle, Washington + Online. 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.

Nacime Bouziani, Shubhi Tyagi, Joseph Fisher, Jens Lehmann, and Andrea Pierleoni. 2024. REXEL: An end-to-end model for document-level relation extraction and entity linking. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, pages 119–130, Mexico City, Mexico. Association for Computational Linguistics.

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

Anthony Chen, Pallavi Gudipati, Shayne Longpre, Xiao Ling, and Sameer Singh. 2021. Evaluating entity disambiguation and the role of popularity in retrieval-based NLP. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4472–4485, Online. Association for Computational Linguistics.

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

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.

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.

Yifan Ding, Qingkai Zeng, and Tim Weninger. 2024. ChatEL: Entity linking with chatbots. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3086–3097, Torino, Italia. ELRA and ICCL.

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

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. 2017. Robust named entity disambiguation with random walks. *Semantic Web*, 9:1–21.

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, 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: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring.

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.

Qijia Li, Feng Li, Shuchao Li, Xiaoyu Li, Kang Liu, Qing Liu, and Pengcheng Dong. 2022. Improving entity linking by introducing knowledge graph structure information. *Applied Sciences*, 12(5).

Xukai Liu, Ye Liu, Kai Zhang, Kehang Wang, Qi Liu, and Enhong Chen. 2024. OneNet: A fine-tuning free framework for few-shot entity linking via large language model prompting. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13634–13651, Miami, Florida, USA. Association for Computational Linguistics.

Marcel Milich and Alan Akbik. 2023. ZELDA: A comprehensive benchmark for supervised entity disambiguation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2061–2072, Dubrovnik, Croatia. Association for Computational Linguistics.

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

Daisuke Oba, Ikuya Yamada, Naoki Yoshinaga, and Masashi Toyoda. 2022. Entity embedding completion for wide-coverage entity disambiguation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6333–6344, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Nicolas Ong, Hassan Shavarani, and Anoop Sarkar. 2024. Unified examination of entity linking in absence of candidate sets. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 113–123, Mexico City, Mexico. Association for Computational Linguistics.

Riccardo Orlando, Pere-Lluís Huguet Cabot, Edoardo Barba, and Roberto Navigli. 2024. ReLiK: Retrieve and LinK, fast and accurate entity linking and relation extraction on an academic budget. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14114–14132, Bangkok, Thailand. Association for Computational Linguistics.

Luigi Procopio, Simone Conia, Edoardo Barba, and Roberto Navigli. 2023. Entity disambiguation with entity definitions. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1297–1303, Dubrovnik, Croatia. 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.

Vera Provatorova, Simone Tedeschi, Svitlana Vakulenko, Roberto Navigli, and Evangelos Kanoulas. 2022. Focusing on context is nice: Improving overshadowed entity disambiguation.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and global algorithms for disambiguation to Wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages

1375–1384, Portland, Oregon, USA. Association for Computational Linguistics.

Hassan Shavarani and Anoop Sarkar. 2023. SpEL: Structured prediction for entity linking. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11123–11137, Singapore. Association for Computational Linguistics.

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.

Junxiong Wang, Ali Mousavi, Omar Attia, Ronak Pradeep, Saloni Potdar, Alexander Rush, Umar Farooq Minhas, and Yunyao Li. 2024. Entity disambiguation via fusion entity decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6524–6536, Mexico City, Mexico. 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.

Zilin Xiao, Linjun Shou, Xingyao Zhang, Jie Wu, Ming Gong, and Daxin Jiang. 2023. Coherent entity disambiguation via modeling topic and categorical dependency. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7480–7492, Singapore. Association for Computational Linguistics.

Zhenran Xu, Yulin Chen, Baotian Hu, and Min Zhang. 2023. A read-and-select framework for zero-shot entity linking. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13657–13666, Singapore. 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.

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.

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.

Kang Zhou, Yuepei Li, Qing Wang, Qiao Qiao, and Qi Li. 2024. GenDecider: Integrating "none of the candidates" judgments in zero-shot entity linking reranking. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 239–245, Mexico City, Mexico. Association for Computational Linguistics.

A Appendix

A.1 Hyperparameters for Training

We use bert-base-uncased (Devlin et al., 2019) as the backbone for both the label and mention encoders, each comprising 110 million parameters[6].

Models are trained with a batch size of 32 documents per update step. To handle CUDA memory limitations, especially with long documents in the ZELDA benchmark, we employ minibatching based on maximum chunk length and number of mentions per chunk: When mention count or text length exceeds GPU capacity, documents are split into smaller chunks, each containing up to 100 mentions and 2,800 characters, while maintaining

[6]We chose BERT-base due to its balance between computational efficiency and strong performance. Using more powerful models like BERT-large or RoBERTa could potentially improve results by providing richer contextual embeddings, but these models come with significantly higher computational and memory requirements. Given the scale of training (~300 model runs for our ablations) and the document-level context used in this work, BERT-base was a practical choice. However, we argue that using larger models might not necessarily result in significant gains. LLMs excel on tasks requiring extensive reasoning, but for tasks like ED, where representations rely heavily on structured context and fine-tuning, the improvements might be marginal. Also, in Dual Encoder setups, the simplicity of embedding space operations (like similarity comparisons) may not fully leverage the additional representational power of larger models.

as much contextual continuity as possible. Verbalizations are truncated after a soft threshold of 50 (100 or 500 for the paragraph setting) characters, using heuristics to avoid splitting words or phrases. Label embeddings are computed with a batch size of 128. Training employs the AdamW optimizer with a learning rate of 5e-6. For the design choice evaluations conducted on AIDA-Yago, all models were trained for 20 epochs. For the larger-scale experiments on the ZELDA benchmark, training was performed for 10 epochs to accommodate the dataset's size. For the triplet loss, margin values are set to 0.5 for cosine similarity and 3.0 for euclidean and dot product similarity.

Label embeddings are fully updated at the start of each epoch. For the large ZELDA training set, additional full updates occur after every 160,000 spans to prevent outdated negative samples, as well as the on-the-fly updates (see Sect. 3.5).

In training the iterative variant, we initially insert a random subset of gold labels per batch (partly corrupted). After 30,000 spans, we switch to inserting real predictions instead, aligning the training setup with the procedure during inference.

A.2 Qualitative Insights of Iterative Prediction

In addition to the short discussion in Sect. 5.2.2, we give some more qualitative insights into the model predictions of the iterative VERBALIZED variant.

Examples for Label Change. Table 10 highlights instances where label insertions improved predictions. In the first example, "Penn" was initially mislinked to Pennsylvania State University but corrected to the Penn State football team after inserting "University of Alabama Football Team" from a neighboring mention. Similarly, the context of baseball clarified "Jose Reyes" as a Dominican baseball player rather than a film and television director.

However, the iterative approach did not deliver consistent performance improvements. Table 11 illustrates cases where insertions misled predictions. E.g., when multiple insertions referenced sports teams ("German sport club", "German football club"), the model over-relied on these, favoring sports-related predictions even when the mention referred to a person ("Dundee_F.C" vs. "Sean_Dundee"). Similarly, when many insertions include the terms "actress" and "actor", this leads to the mention "Italian" being linked to "Cinema_of_Italy" instead of the more general "Italy" (country). While both solutions could be argued correct in this instance, in other cases we see real errors due to too much reliance or "mimicking" of the insertions. While we attempted to address this with a modified iterative training setup that includes a mix of mentions with and without disambiguated neighbouring mentions, the results remained mixed, as the positive and negative effects balance out.

Quantitative Counts. In Table 12 we present a quantitative analysis of the iterative variant's impact on prediction quality, comparing the initial and final prediction state. We categorize instances into four key types: "Correct", where both initial and final predictions match the gold label, "Incorrect", where both differ from the gold label, "Incorrect > Correct", where the initial incorrect prediction changes to a correct one after iterative adjustments, and "Correct > Incorrect", where the initial correct prediction later changes to an incorrect one.

We observe that as expected, no changes occur for the Shadowlinks datasets, as they consist solely of single-mention documents, and only a few changes are seen for social-media-based datasets with mostly short documents with few mentions. Although positive changes are generally more common than negative ones, there are still a significant number of negative changes. Overall, the rate of change is low, with most labels remaining unchanged.

These analyses highlight the mixed results and modest benefits of the iterative approach, particularly considering its increased costs and complexity. However, it also suggests that there is potential in using enriched contextual clues, though the negative effects and surprisingly small gains would need to be addressed before confidently proposing this variant as the primary model. Still, we view the successful cases as evidence of the approach's potential and the failures as valuable lessons for developing future global or iterative ED models.

A.3 Comparison to BLINK and CHATEL

We add two competitive models to our comparison, which we exclude from our main results (Table 7) due to differences in training and evaluation settings, making direct comparison challenging.

Comparison to BLINK. BLINK (Wu et al., 2020) combines a Bi-Encoder with a subsequent Cross Encoder. The Bi-Encoder retrieves top 100 candidates using dot product similarity, which are

	Step 1	Step 2
Input excerpt	"A few more days until I can watch **Penn** state get bent over and rammed by Alabama lol"	"A few more days until I can watch **Penn** state get bent over and rammed by Alabama (University of Alabama Football Team) lol"
Pred. Verbal.	Pennsylvania_State_University (✗) "Pennsylvania State University; public university in Pennsylvania"	Penn_State_Nittany_Lions_football (✓) "Penn State Nittany Lions football; football team of Penn State University"
Input excerpt	"Nyjer Morgan makes **Jose Reyes** seem tolerable."	"Nyjer Morgan (American former professional baseball and ice hockey player) makes **Jose Reyes** seem tolerable."
Pred. Verbal.	Jose_Javier_Reyes (✗) "José Javier Reyes, Filipino writer and director for film and television"	José_Reyes_(infielder) (✓) "José Reyes (infielder); Dominican baseball player, MLB All-Star"
Input excerpt	"Relations between Clarke, **Major** good - spokesman. LONDON 1996-12-06 Relations between Chancellor of the Exchequer Kenneth Clarke and Prime Minister John Major are good despite media reports"	"Relations between Clarke, **Major** good - spokesman. LONDON (capital and largest city of England and the United Kingdom) 1996-12-06 Relations between Chancellor of the Exchequer Kenneth Clarke (British politician (born 1940)) and Prime Minister John Major (former prime minister of the United Kingdom (born 1943)) are good despite media reports"
Pred. Verbal.	Major (✗) "Major"	John_Major (✓) "John Major; former prime minister of the United Kingdom (born 1943)"
Input excerpt	"Peggy Olson is awesome... one of the best characters on **#madmen**"	"Peggy Olson (fictional character from Mad Men) is awesome... one of the best characters on **#madmen**"
Pred. Verbal.	Mad_(TV_series) (✗) "Mad; American adult animated sketch comedy television series"	Mad_Men (✓) "Mad Men; American period drama television series"
Input excerpt	"the director of **law and order**, his name is Dick Wolf"	"the director of **law and order**, his name is Dick Wolf (American television producer (born 1946))"
Pred. Verbal.	Law_and_order (✗) "Law and Order; Wikimedia disambiguation page"	Law_&_Order (✓) "Law & Order; American police procedural and legal drama television series"

Table 10: Insights to the iterative variant: Examples for successful disambiguation after label insertions.

then ranked by the the Cross Encoder. As the Bi-Encoder is very similar to our Dual Encode (the biggest difference being the verbalizations and negative sampling) and BLINK is currently widely used for ED, comparison remains interesting.

As the BLINK model was trained on different data and with a different label set[7], direct comparison to our approach is challenging. See Table 13 for the following comparison setups, for all of which we used their code base[8]: We first evaluate their final model on the ZELDA test sets, both the Bi-Encoder (BLINK-OG$_{bi}$) and the Cross Encoder (BLINK-OG$_{cross}$). For better comparison, we also train a BLINK Bi-Encoder on ZELDA (BLINK-Z$_{bi}$), where we plug in our general label set (ca. 800K entities) but keep the BLINK verbalizations[9].

Unsurprisingly the original BLINK-OG$_{cross}$ – benefiting from more data and the additional cross-encoder step – beats VERBALIZED on most (not all) datasets and on average, while our iterative variant comes close. Interestingly, our model (also the base variant) significantly outperforms the original BLINK Bi-Encoder. This is particularly impressive given the similar architecture of BLINK's Bi-Encoder and its advantage in training data[10].

For the BLINK Bi-Encoder trained on ZELDA, the results lag behind our Dual Encoder as well as the original BLINK Bi-Encoder[11]. Next to the smaller train data, this performance drop is likely due to differences in BLINK's and our training

[7]The BLINK dataset consist of 9M datapoints with a label dictionary of size 5.9M, created from a 2019 Wikipedia dump (Wu et al., 2020). ZELDA only includes 2.6M datapoints and 822K entities (Milich and Akbik, 2023).

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

[9]Note that ~3% of the labels in our label set do not appear in the BLINK label set. We exclude those from the label set for this experiment as well as exclude the affected datapoints from evaluation.

[10]However we acknowledged that their bigger label set makes accurate disambiguation also more challenging.

[11]We did not have the resources to perform an extensive hyperparameter search for BLINK. Furthermore, based on our observations and reports from an open issue (github.com/facebookresearch/BLINK/issues/31), the code version in the repository appears to rely solely on in-batch negatives, omitting the hard negatives described in Wu et al. (2020). This discrepancy suggests that the original BLINK model may have benefited from a more robust negative sampling strategy, which we could not reproduce.

	Step 1	Step 2
Input excerpt	"Karlsruhe 3 (Reich 29th, Carl 44th, **Dundee** 69th) Freiburg 0. Halftime 2-0. Attendance 33,000"	"Karlsruhe (German sport club) 3 (Reich 29th, Carl 44th, **Dundee** 69th) Freiburg (German football club) 0. Halftime 2-0. Attendance 33,000"
Pred.	Sean_Dundee (✓)	Dundee_F.C. (✗)
Verbal.	"Sean Dundee; South African–German footballer"	"Dundee F.C.; association football club in Dundee, Scotland"
Input excerpt	"West Indies tour manager Clive Lloyd has apologised for Lara's behaviour on Tuesday . He (**Lara**) had told Australia coach Geoff Marsh that ..."	"West Indies (multinational cricket team) tour manager Clive Lloyd has apologised for Lara's behaviour on Tuesday . He (**Lara**) had told Australia (national sports team) coach Geoff Marsh that ..."
Pred.	Brian_Lara (✓)	Australia_national_cricket_team (✗)
Verbal.	"Brian Lara; West Indian cricketer"	"Australia national cricket team; national sports team"
Input excerpt	"We offer the following types of posters; Classic Film posters, movie posters, French movie posters, **Italian** movie posters, cinema posters, affiche de cinema, bogart posters [...] Vintage Movie posters about the greats of their time like Humphrey Bogart, Marylin Monroe, Audrey Hepburn, Brigitte Bardot, Marlene Dietrich, James Dean, Greta Garbo [...]"	"We offer the following types of posters; Classic Film posters, movie posters, French movie posters, **Italian** movie posters, cinema posters, affiche de cinema, bogart posters [...] Vintage Movie posters about the greats of their time like Humphrey Bogart (American actor (1899–1957)), Marylin Monroe, Audrey Hepburn (British actress (1929–1993)), Brigitte Bardot, Marlene Dietrich, James Dean (American actor (1931–1955)), Greta Garbo (Swedish-American actress (1905–1990)) [...]"
Pred.	Italy (✓)	Cinema_of_Italy (✗)
Verbal.	"Italy; country in Southern Europe"	"Cinema of Italy; aspect of history"
Input excerpt	"Alright. ESPN not in **HD** has sound. HD doesn't. Boo."	"Alright. ESPN (American television and radio sports network) not in **HD** has sound. HD doesn't. Boo."
Pred.	High-definition_television (✓)	HD_Radio (✗)
Verbal.	"High-definition television; TV resolution standard"	"HD Radio; digital radio technology"

Table 11: Insights to the iterative variant: Examples for detrimental label insertions resulting in worse prediction.

setup like our dynamic hard negatives, large context, constant label embedding updates and more concise label verbalizations, all of which likely contribute to its superior performance.

BLINK's vs. our Label Verbalizations. For direct comparison of the effect of the applied label verbalizations, we trained two BLINK Bi-Encoders on AIDA with our general label set: Once with our verbalizations (Title+Description), and once with the BLINK verbalizations (first Wikipedia paragraph). The results in Table 14 highlight the effectiveness of concise descriptions in otherwise exact settings, moreover when only having access to a small train set like AIDA.

Comparison to CHATEL. We also include a representative of another system type, CHATEL (Ding et al., 2024), a current method that leverages LLM prompting for ED. In this approach, a small set of entity candidates is provided to a LLM. It is first tasked to enhance the mention context by generating auxiliary information from the document and its own knowledge and, in a second step, it is asked to select the correct entity through a multiple-choice formatted prompt.

Refer to Table 15 for a comparison of VERBALIZED to the prompting method CHATEL[12], as well as the BLINK models, on the subset of datasets for which all three models report numbers, including MSNBC (Cucerzan, 2007), ACE2004 (Ratinov et al., 2011) and AQUAINT (Milne and Witten, 2008). As these are not part of the ZELDA splits, it is not guaranteed that all their target labels are included in our employed label set. In fact, we found that ~15% of their target labels are missing, mostly due to changes in article names. For example, one gold label in AQUAINT is the article "Dave_Richardson", an article now redirecting to "David_Richardson", which is a disambiguation page that links to, among others, two different cricket players with the same name. To ensure a fair evaluation, we add the missing labels (along with their verbalizations) to our pool. Keep in mind that while this ensures the inclusion of all labels, this may lead to a) incomplete verbalizations for outdated entity labels, and b) possibly multiple versions of the same entity, decreasing the likelihood

[12]We report the numbers that are presented in the README of the repository (https://github.com/yifding/In_Context_EL), for which they used GPT-3.5.

Method	AIDA-B	TWEEKI	REDDIT-POSTS	REDDIT-COMM	WNED-CWEB	WNED-WIKI	SLINKS-TAIL	SLINKS-SHAD.	SLINKS-TOP
correct	3843	657	642	561	7771	6054	884	599	595
incorrect > correct	97	10	7	2	171	70	0	0	0
correct > incorrect	63	12	2	4	156	79	0	0	0
incorrect	482	177	52	70	3042	562	15	305	309
#Mentions	4485	856	703	637	11140	6765	899	904	904
Accuracy Step 1	87.1	**78.2**	91.6	**88.7**	71.2	**90.7**	98.3	66.3	65.8
Accuracy Last Step	**87.9**	77.9	**92.3**	88.4	**71.3**	90.5	98.3	66.3	65.8

Table 12: Insights to the iterative variant's performance over iteration steps. *Correct*: initial and final prediction align with gold label, *incorrect*: initial and final prediction are different from gold label, *incorrect* > *correct* and *correct* > *incorrect*: prediction changes from initially incorrect to correct and vice versa.

Method	Train data	AIDA-B	TWEEKI	REDDIT-POSTS	REDDIT-COMM	WNED-CWEB	WNED-WIKI	SLINKS-TAIL	SLINKS-SHAD.	SLINKS-TOP	AVG
BLINK-OG$_{bi}$[1]	B	80.6	77.3	90.8	87.8	68.2	79.8	97.9	50.1	57.3	76.6
BLINK-OG$_{cross}$[1]	B	84.2	**82.4**	**92.8**	**91.2**	**77.3**	82.3	**99.2**	64.8	**74.2**	**83.2**
BLINK-Z$_{bi}$[2]	Z	65.5	72.1	83.1	79.1	58.1	73.1	96.3	41.8	42.6	68.0
VERBALIZED	Z	82.6	78.9	89.2	86.2	69.8	**91.4**	98.6	65.3	67.0	81.0
+ *iter. training*	Z	**88.2**	78.9	92.2	88.4	71.5	90.8	98.2	**66.3**	65.9	82.3

Table 13: Comparison between VERBALIZED and BLINK on the ZELDA benchmark. [1]BLINK-OG results come from evaluating the original BLINK on ZELDA test sets (Wu et al., 2020). However, they are not fully comparable since BLINK-OG was trained on significantly more data (B). [2]BLINK-Z: We trained a BLINK Bi-Encoder on ZELDA, using our general label set with the BLINK verbalizations.

of selecting the "correct" one.

On average over all datasets for which we could compare scores, VERBALIZED beats CHATEL and the BLINK Bi-Encoder, while on MSNBC and ACE2004 CHATEL performs better, possibly due to the discrepancy in the label sets. Note that the performance improvements after incorporating additional labels highlight VERBALIZED's ability to adapt to unseen labels without requiring retraining.

A.4 Results with Target Label Set

In Table 16, we report results of our main models trained with the general label set on ZELDA, but restricting the label set to each respective target label set for inference. As expected, this simplification leads to significantly higher accuracy across all datasets. The findings indicate that training with a broader label set does not compromise performance on more specific label sets.

Method	Train data	AIDA-B	TWEEKI	REDDIT-POSTS	REDDIT-COMM	WNED-CWEB	WNED-WIKI	SLINKS-TAIL	SLINKS-SHAD.	SLINKS-TOP	AVG
BLINK_{bi} with											
BLINK verbalizations	A	59.4	47.9	50.9	53.7	44.2	50.5	76.3	**40.3**	38.5	51.3
our verbalizations	A	**61.8**	**57.7**	**77.4**	**73.2**	**49.5**	**51.8**	**80.1**	23.3	**25.3**	**55.6**

Table 14: Training a BLINK Bi-Encoder on AIDA, using our label set with BLINK's vs. our verbalizations.

Method	Train data	AIDA	CWEB	WIKI	MSNBC	ACE2004	AQUAINT	AVG
LLM Prompting								
CHATEL	(B+CL)	82.1	71.1	77.1	86.6	88.4	79.1	80.7
Dense Retrieval								
BLINK-OG_{bi}	B	80.6	68.2	79.8	83.5	84.3	87.2	80.6
BLINK-OG_{cross}	B	84.2	**77.3**	82.3	**97.1**	**98.4**	**98.7**	**89.7**
BLINK-Z_{bi}	Z	65.5	58.1	73.1	67.4	74.0	79.8	69.7
VERBALIZED								
ZELDA labels	Z	82.6	69.8	**91.4**	74.7	75.9	73.0	77.9
+ additional labels	Z	82.6	69.8	**91.4**	80.3	82.5	80.5	81.2
+ iter. training	Z	**88.2**	71.5	90.8	80.8	85.6	84.2	83.5

Table 15: VERBALIZED performance compared to CHATEL and BLINK. We include the datasets for which both models report numbers, including MSNBC, ACE2004, and AQUAINT. *(B+CL)*: CHATEL, as a prompting method, does not use any specific train data, however for candidate generation they use both BLINK as well as candidate lists based on frequency statistics from hyperlinks (Ganea and Hofmann, 2017). *+ additional labels*: About 15% of labels from MSNBC, ACE2004 and AQUAINT are not included in our ZELDA-based label set. To enable fair evaluation, we add those to the label set for inference.

Method	AIDA-B	TWEEKI	REDDIT-POSTS	REDDIT-COMM	WNED-CWEB	WNED-WIKI	SLINKS-TAIL	SLINKS-SHAD.	SLINKS-TOP	AVG
VERBALIZED	94.5	91.0	97.7	95.9	83.1	**96.3**	99.6	97.7	**93.9**	94.4
+ iter. training	**95.7**	**91.1**	**97.9**	**97.3**	**84.0**	95.6	**99.7**	**98.1**	92.6	**94.7**

Table 16: Results on ZELDA, restricting to the respective target dataset's label set for inference.