

SELECTED: Selective LLM Refinement for Entity Disambiguation

Susanna Rücker Alexander Reinicke Alan Akbik
Humboldt-Universität zu Berlin
susanna.ruecker@hu-berlin.de

Abstract

Efficient architectures for Entity Disambiguation (ED) typically pair a dual encoder for candidate retrieval with a cross encoder for re-ranking. While large language models (LLMs) have recently emerged as powerful alternatives for the ranking stage, their application remains constrained by two primary factors: only a limited number of candidate entities can be provided within the model’s input context, and invoking LLMs for every entity mention leads to prohibitive computational cost.

In this work, we propose SELECTED¹, a confidence-guided hybrid framework for efficient entity disambiguation. Our method uses a dual encoder as a fast primary model and selectively invokes an LLM only for uncertain mentions. A central focus is the systematic study of selective LLM refinement, including uncertainty estimation, thresholding, and prompt design choices for balancing accuracy and efficiency. Experiments on the ZELDA benchmark show that selective prompting improves accuracy by 4.3%, establishing a new best result among directly comparable ZELDA-trained systems while requiring LLM inference for only about 20% of mentions. Moreover, SELECTED surpasses full LLM prompting, suggesting that selective refinement can avoid LLM-induced errors on high-confidence mentions.

1 Introduction

Entity disambiguation (ED) is a fundamental task in natural language processing, requiring models to disambiguate mentions in text by linking them to the correct entry in a knowledge base (KB). Accurate entity disambiguation is necessary for a wide range of downstream applications, including information retrieval, question answering, and knowledge base population, where even small disambiguation errors can significantly degrade sys-

tem performance (Yamada et al., 2016; Ganea and Hofmann, 2017).

Formally, given an input document with detected mentions $m_1, \dots, m_n$ and a knowledge base $\mathcal{E}$, ED assigns each mention m_i to the corresponding entity $e_i \in \mathcal{E}$. Following the standard ED evaluation setting, we assume that mentions are given and focus on the disambiguation step. In retrieval-based ED, a first-stage model typically produces a ranked candidate list $C_i = (c_{i,1}, \dots, c_{i,k})$ for each mention. In our setting, the top-ranked candidate $c_{i,1}$ is accepted directly for high-certainty mentions, while the full top- k list C_i is exposed to the LLM only for mentions selected as uncertain.

Retrieval-based ED systems commonly use neural models such as dual encoders or cross encoders to rank candidate entities based on contextual similarity (Gillick et al., 2019; Wu et al., 2020; Procopio et al., 2023; Wang et al., 2024). These approaches achieve strong performance in many settings but remain brittle when contextual cues are weak or when multiple candidate entities are highly similar, causing retrieval-based models to select incorrect labels (Rücker and Akbik, 2025; Li et al., 2025).

Large language models (LLMs) offer a complementary set of strengths. Their ability to reason over context, incorporate world knowledge, and resolve subtle ambiguities makes them attractive for ED (Xiao et al., 2023a; Ding et al., 2024; Xin et al., 2025; Vollmers et al., 2025; Hou et al., 2025). However, applying LLMs exhaustively to all entity mentions is impractical for large-scale or latency-sensitive systems due to their high inference cost and slow generation speed (Glentis et al., 2025). Moreover, LLMs are not naturally suited to tasks that require selecting an exact label from extremely large candidate sets, as ED typically involves knowledge bases containing thousands or even millions of entities (Xin et al., 2025).

Hybrid approaches. Consequently, recent work has explored hybrid approaches that combine

¹<https://github.com/flairNLP/SelectED>

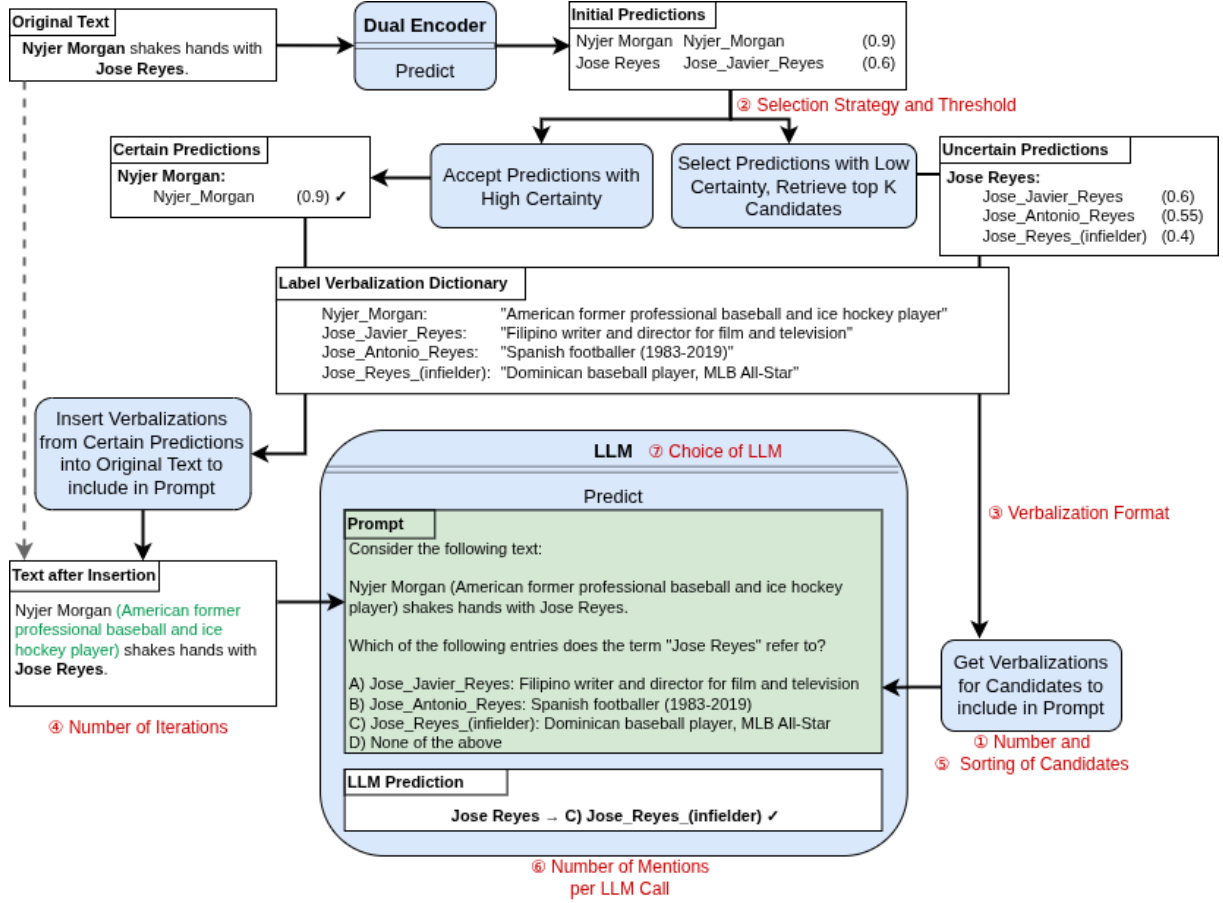


Figure 1: Overview of SELECTED. All mentions are processed by the dual encoder to obtain initial predictions. High-certainty predictions are accepted directly and used to enrich the document context, while low-certainty mentions are routed to the LLM for candidate-based refinement. Numbered red annotations indicate the main design choices studied in this work, including selection strategy, candidate set size and ordering, verbalization format and LLM choice.

retrieval-based models with LLMs. In these systems, LLMs are typically used in one of two ways: Either they are applied as deciders that select the correct entity from a small candidate set provided by some retrieval model or traditional candidate lists. Or they are used as tools for enriching the context of mentions (Ding et al., 2024; Liu et al., 2024; Xin et al., 2025; Vollmers et al., 2025).

While these strategies can improve overall disambiguation accuracy, applying LLMs to all mentions remains costly when applied to large datasets with many entity mentions. Further, as we show in this paper, LLMs may in fact worsen performance for straightforward entities that a base dual encoder already correctly resolves.

Contributions. We propose SELECTED, a hybrid entity disambiguation approach that integrates LLMs *selectively*, i.e. only for uncertain mentions rather than all predictions (see Fig. 1). Our method uses a dual encoder as a fast primary model to

generate candidate entities and certainty estimates, which are then used to decide whether a mention is resolved directly or refined by an LLM.

Our goal is not to train a new ED model, but to study how a strong existing dual-encoder backbone can be improved at inference time through selective LLM refinement. In particular, we systematically investigate design choices that determine the accuracy–efficiency trade-off: certainty estimation, thresholding strategies, candidate set size and ordering, verbalization format, batching multiple mentions per prompt, and the number of refinement iterations.

Selective refinement offers two key benefits: it reduces computational cost compared to full LLM prompting and can even outperform it, as LLMs may degrade predictions for high-certainty mentions already handled well by the dual encoder.

We conduct extensive experiments on the ZELDA benchmark (Milich and Akbik, 2023) and three ad-

ditional unseen ED benchmarks. Refining only about one-fifth of mentions achieves a new best result among directly comparable ZELDA-trained systems, and yields consistent improvements over the backbone on the additional datasets while requiring substantially fewer LLM calls than full prompting and other reported LLM-based ED systems with available routing statistics.

In summary, our contributions are:

- We introduce a certainty-guided hybrid entity disambiguation (ED) framework that uses LLMs only *selectively* for secondary ranking.
- We present a systematic study of selective LLM refinement for ED, including confidence estimation strategies, thresholding methods, and efficiency-aware prompt construction.
- We evaluate our approach on diverse ED benchmarks and achieve a strong accuracy-efficiency trade-off, outperforming full prompting while requiring LLM inference for only a small fraction of mentions.

2 SELECTED: A Hybrid Strategy

Our approach, illustrated in Fig. 1, follows a candidate-based ED setup. For each mention, a dual encoder first retrieves and scores the top- k candidate entities. SELECTED then either accepts the dual encoder’s top prediction directly or forwards the mention and its candidate set to an LLM for refinement, depending on a selection criterion. The main design choices, marked in red in Fig. 1, are therefore: (i) how many candidates to provide, (ii) how to decide whether a mention is uncertain, (iii) how to verbalize candidates and context, and (iv) how to batch and iterate LLM calls.

2.1 Dual Encoder for Primary Prediction

We use the *VerbalizED* dual encoder (Rücker and Akbik, 2025) as our primary entity disambiguation model. It consists of a mention encoder, which embeds a mention with its context, and a label encoder, which embeds knowledge base entries. Both are projected into a shared embedding space, enabling efficient inference via precomputed label embeddings and similarity scoring. The base model uses Euclidean distance as similarity measure. To obtain a more intuitive score where higher values indicate better matches, we instead use its negative.

We use this released backbone without additional retraining. This isolates the effect of

inference-time selective refinement: Any gains over the backbone come from routing and LLM-based re-ranking rather than from training a stronger retriever.

Each prediction thus yields a similarity score used as a *certainty* signal to distinguish high- from low-confidence cases, determining whether it is accepted or forwarded to the LLM. As shown in Fig. 1, “Nyjer Morgan” is resolved with high certainty, while “Jose Reyes” yields similarly scoring candidates (see *Initial Predictions* and *Uncertain Predictions*), marking it as uncertain and triggering LLM refinement.

2.2 LLM for Selective Refinement

As dual encoders handle straightforward cases well, we invoke an LLM only for low-certainty cases.

Selective Invocation. We split dual encoder predictions into *certain* and *uncertain* cases. Certain predictions are accepted directly, while uncertain ones are forwarded to the LLM. The notion of *certainty* and the selection criterion is a key design choice. In Fig. 1, a fixed similarity threshold is used. We also explore multiple alternatives such as top-candidate score gaps in Sect. 3.3.

Candidate-based Prediction. The LLM receives a prompt² (Fig. 1, green *Prompt* box) containing the document text and a small candidate set for each selected mention. The candidate set consists of the top- k predictions by the dual encoder. Each candidate label is accompanied by a descriptive verbalization from the *label verbalization dictionary* which is derived from the KB. For instance, the entity “Jose_Javier_Reyes” is verbalized as “Filipino writer and director for film and television”. The task is framed as multiple choice. While the LLM is only queried for uncertain mentions, we enrich the input with context from high-confidence predictions by inserting their verbalizations into the document (Fig. 1, *Original Text* → *Text after Insertion*). As shown in App. E, this yields a small average improvement. The LLM then selects the best candidate (or abstains), producing the final prediction. Hyperparameters such as candidate size and verbalization choice are discussed in Sect. 3.

Since the LLM is restricted to the retrieved candidate set, the approach is bounded by the recall of the dual encoder: If the correct entity is absent from the top- k candidates, the LLM cannot select it and can only abstain. This is a general limitation

²The prompt template is shown in Table 17.

of retrieve-and-rank ED systems, and we analyze such cases in App. G.

Iterative Refinement. Uncertain mentions are processed iteratively, grouped by certainty and resolved from higher to lower confidence. After each iteration, resolved mentions are verbalized and inserted into the document, progressively enriching context and improving disambiguation of more challenging cases.

The original *VerbalizED* model (Rücker and Akbik, 2025) also introduced an iterative variant that re-predicts uncertain mentions after inserting verbalizations from already resolved mentions. SELECTED adopts the general idea of using resolved mentions as additional context, but differs in the refinement step: VerbalizED reuses the same dual encoder with a fixed heuristic, whereas SELECTED routes selected mentions to an external LLM. Beyond this architectural difference, our work systematically studies different routing and prompting choices.

Studied Design Decisions. The resulting design choices, highlighted by the red numbers in Fig. 1, include candidate set size, candidate ordering, verbalization format, routing criterion, selection threshold, number of mentions per prompt, and number of refinement iterations. We study these choices in Sect. 3.

3 Systematic Study of Design Choices

The central challenge in our selective ED approach is determining when LLM refinement is beneficial and how candidate information can be presented efficiently. We therefore study the design choices introduced in Sect. 2 and derive a configuration that aims to balance accuracy and computational cost. We first describe the tuning setup, including datasets, evaluation metrics, and efficiency measures, before analyzing each design choice.

3.1 Experimental Setup

Tuning Procedure. Instead of an exhaustive grid search of all hyperparameters – which is prohibitively expensive due to repeated LLM inference – we adopt a sequential tuning strategy. Hyperparameters are ordered by expected impact and optimized one at a time, while keeping remaining parameters fixed at either their default or previously selected best values. All experiments in this section use the Gemini 2.5 Flash-Lite model (Google DeepMind, 2025a) as the LLM component. We

report the most relevant results and decisions in the main text, with detailed analyses in App. A.

Dataset. We evaluate on the ZELDA benchmark (Milich and Akbik, 2023), which standardizes training data and entity vocabularies to address inconsistencies in prior ED setups. It contains 95,000 Wikipedia paragraphs for training (2.6M mentions across ~822k entities) and nine test splits covering diverse domains and text styles. These include AIDA-B (Hoffart et al., 2011) (news articles), TWEEKI (Harandizadeh and Singh, 2020) (tweets), REDDIT-POSTS and REDDIT-COMMENTS (Botzer et al., 2021), WNED-WIKI and WNED-CWEB (Guo and Barbosa, 2017), as well as the three SHADOWLINKS datasets (Provatorova et al., 2021), which evaluate robustness to *overshadowing*: TOP contains easy cases where the most frequent entity is correct, SHADOW contains difficult overshadowed mentions, and TAIL focuses on rare entities.

ZELDA does not provide a dedicated development set. Since the underlying dual encoder was already trained on ZELDA, using parts of the training data for tuning was undesirable. We therefore select hyperparameters on five of the nine ZELDA test splits: SHADOWLINKS-SHADOW, SHADOWLINKS-TAIL, AIDA, REDDIT-COMMENTS, and TWEEKI. We acknowledge that this is not a fully held-out tuning setup and treat it as a limitation. The final evaluation (Sect. 4) applies the selected configuration to all ZELDA test sets. To assess transfer beyond the tuning subsets, we additionally evaluate the fixed configuration without further tuning on three external datasets: MSNBC (Cucerzan, 2007), AQUAINT (Milne and Witten, 2008), and ACE2004 (Ratinov et al., 2011). We further test the same parameter setting across multiple LLMs (Sect. 3.6) and use a different LLM for the final evaluation than for parameter selection.

Evaluation Metrics. We report mention-level accuracy, i.e., the proportion of correctly linked mentions, per dataset and as a macro-average. For comparability, we use the same ZELDA-derived label set as Rücker and Akbik (2025). Efficiency is measured via total LLM token usage (input + output) as a model-agnostic proxy for inference cost.

3.2 Candidate Set Size

This parameter controls how many candidate labels the LLM receives per mention, taking the top- k predictions from the dual encoder. Increasing k can improve accuracy by including the correct en-

k	Accuracy %		Efficiency	
	SELECTED	LLM (Ref.)	Tokens	Rel. Δ
2	80.59	56.61	476,632	—
3	80.86	57.89	476,653	0.0
5	81.29	59.95	488,002	2.3
7	81.35	60.22	508,285	6.6
10	81.47	60.83	539,288	13.1
15	81.47	60.83	598,443	25.6
20	<u>81.42</u>	<u>60.59</u>	655,963	37.6

Table 1: Mean (across datasets) accuracy and accuracy on LLM-refined cases for **varying candidates** (k). Bold and underlined indicate 1st and 2nd best. Efficiency shows total token count and increase relative to $k = 2$.

tity when it is not top-ranked, but may also introduce more incorrect options. We evaluate several top- k values. The effect varies: higher k helps on challenging datasets, but can slightly reduce performance on easier datasets where the top prediction is usually correct.

Overall, the differences remain modest. As shown in Table 1, the best average accuracy is obtained with top- $k = 10$ or top- $k = 15$ (81.5% overall accuracy, 60.8% on the refined cases). However, as the latter requires substantially more tokens, we fix a top- k value of **10 candidates**.

3.3 Selection Strategy and Threshold

The selection strategy and threshold determine which mentions are sent to the LLM, trading off accuracy and efficiency: More mentions increase potential for resolving challenging cases, while fewer reduce cost but may leave ambiguities unresolved. We evaluate the following strategies:

- **Top₁ Similarity**: Selection is based on the similarity score of the top candidate (i.e., negative Euclidean distance between mention and predicted label embeddings). We test a "relative" threshold (marking the lowest $n\%$ of spans as uncertain) and an "absolute" threshold (selecting spans below a fixed score t).
- **Candidate Similarity Difference**: Rather than considering the similarity of the top-ranked candidate alone, we consider the top 2: If both candidates are nearly equally likely options, we interpret this as a sign of uncertainty. Selection is based on the Top₁–Top₂ similarity difference, using either a relative threshold (smallest $n\%$ of differences) or an absolute threshold (below a fixed value).
- **Combination**: Select spans with either low Top₁ similarity or small candidate difference.

Selection Strategy	Type	Best t	Acc. %	Spans %
Top-1 Sim.	Rel.	0.7	82.76	30.00
	Abs.	-22	83.30	25.50
Top ₁ – Top ₂	Rel.	0.75	<u>83.75</u>	<u>25.00</u>
	Abs.	1	84.83	19.36
Combination	Rel.	0.75	83.26	<u>25.00</u>

Table 2: Threshold results for selection strategies. Bold and underlined values indicate 1st and 2nd best results.

We performed a grid search to evaluate all selection strategies at various threshold values. We summarize the best configurations in Table 2. The absolute thresholds outperform the relative ones, suggesting that allowing the number of prompted spans to vary across datasets – prompting more spans for challenging datasets and fewer for easier ones – leads to better overall performance³. Moreover, the score difference between top candidates is a more reliable certainty signal than the Top₁ score alone. Best results are obtained with an **absolute candidate difference** threshold of **1** (84.83% accuracy), which also forwards the fewest spans (19.4%) to the LLM, indicating effective identification of cases where refinement is beneficial⁴.

3.4 Verbalization Strategy

The verbalization strategy defines what information about candidate entities is shown to the LLM and what information from already resolved entities is inserted into the document (see *Text after Insertion* in Fig. 1). We evaluate options from simple entity names to richer descriptions and category information from the KB (example in Table 11).

Full results are reported in App. A.2 and Table 12. The highest accuracy (83.83%) is achieved using both descriptions and categories, while categories alone perform nearly identically (83.80%). However, the richer setting requires substantially more tokens. Given our focus on efficiency, we thus adopt the **categories-only** strategy, which reduces token usage with negligible impact on accuracy.

3.5 Additional Hyperparameters

Number of Iterations. Our iterative refinement processes uncertain mentions in multiple passes, allowing context to accumulate. We evaluate different numbers of iterations (App. A.3, Table 13). While accuracy improves slightly with more itera-

³Full results per strategy are reported in App. A.1.

⁴A dataset-level threshold sensitivity plot for this final strategy is shown in App. A.1, Fig. 3.

tions, gains beyond two are minimal but increase runtime. We select **two iterations** as the best accuracy–efficiency trade-off.

Candidate Ordering. Candidate order can influence LLM decisions (Pezeshkpour and Hruschka, 2024; Zheng et al., 2024). We compare descending similarity with random ordering. Sorting improves accuracy by 2.5 points (App. A.4, Table 14), indicating that the LLM benefits from the ranking signal. We therefore use **similarity-based order**.

Number of Spans per Prompt. To reduce LLM calls, we group multiple mentions into a single prompt, enabling shared context reuse. While larger prompts increase complexity, this reduces invocation overhead. Best accuracy is achieved with five mentions per prompt (83.83%; App. A.5, Table 15), with similar results for 2–3 mentions. Since five also minimizes token usage (79.43% vs. single-mention prompts), we select **five mentions per prompt**.

3.6 Choice of LLM

After determining the hyperparameter configuration using Gemini 2.5 Flash-Lite (Google DeepMind, 2025a), we evaluate several alternative LLMs for the refinement module, including both API-based models (OpenAI and Google) and locally hosted models. Full results are reported in App. B and Table 16. Overall, stronger LLMs tend to improve disambiguation accuracy. Gemini 2.5 Pro achieves the highest average accuracy, closely followed by GPT-5 mini, but requires substantially more tokens. Considering accuracy and efficiency, we select **GPT-5 mini** (OpenAI, 2025).

3.7 Final Configuration

To summarize, we fix the configuration as in Table 3 and use it for all experiments in Sect. 4.

Parameter	Value
Candidate set size	Top ₁₀
Selection strategy	Absolute Candidate Sim. Diff. @1.0
Verbalizations	Categories only
Nr. of iterations	2
Candidate order	Sorted
Spans per prompt	5
LLM	GPT-5 mini

Table 3: Final configuration selected from the systematic design study.

4 Experimental Results and Analysis

We evaluate SELECTED using the fixed configuration from Sect. 3, comparing against prior state-of-the-art methods on ZELDA and additional datasets, and provide ablations of individual components.

4.1 Experimental Setup

Comparable ZELDA-trained Systems. We compare our approach against several previously reported systems on the ZELDA benchmark to ensure direct comparability. Specifically, we include FEVRY (Févy et al., 2020), a direct classification-based approach, LUKE (Yamada et al., 2022), a global entity disambiguation model, and GENRE (De Cao et al., 2021), a model that directly generates the target entity title, restricted by mention-specific candidate lists. We further report the numbers for the BLINK dual encoder approach (Wu et al., 2020), retrained on ZELDA by Rücker and Akbik (2025), and FusionED (Wang et al., 2024), an encoder–decoder architecture that integrates entity descriptions with candidate representations.

We include the base and iterative Verbalized variants (Rücker and Akbik, 2025) as well as report results for the checkpoint used as our backbone (Dual Enc. (Verbalized)) to better compare gains from our selective LLM refinement (SELECTED).

Reference Numbers. We further compare against prominent ED systems that were trained on different datasets and only report on a subset of ZELDA’s benchmarks. This includes the original dual and cross encoder variants of BLINK (Wu et al., 2020), ChatEL (Ding et al., 2024) (a full LLM prompting approach), and ARTER (Li et al., 2025), which trains a routing model to classify mentions as “easy” or “hard” and invokes an LLM only for the latter. As these models differ in training data and entity vocabularies (i.e., the set of available labels), results should be interpreted with caution.

4.2 Results

Results are shown in Table 4. On average, SELECTED achieves 84.5% accuracy, establishing a new best result among directly comparable ZELDA-trained systems. It outperforms both the base Verbalized model and its iterative variant. While the latter also revisits uncertain mentions, incorporating verbalizations from already predicted ones, it relies solely on repeated dual encoder predictions rather than SELECTED’s selective LLM refinement. SELECTED also surpasses both the

Method	Trained on	AIDA	TWEE.	RED.- POSTS	RED.- COMM.	CWEB	WNED- WIKI	SLINKS TAIL	SLINKS SHAD.	SLINKS TOP	Avg.
<i>Models Trained on ZELDA (Directly Comparable)</i>											
BLINK _{dual encoder only} Z	Z	65.5	72.1	83.1	79.1	58.1	73.1	96.3	41.8	42.6	68.0
FEVRY	Z	79.5	76.9	89.0	86.5	70.3	84.5	87.6	31.9	47.7	72.7
LUKE	Z	81.2	77.9	81.5	78.5	70.3	76.5	98.0	22.5	51.8	71.0
GENRE	Z	78.6	80.1	<u>92.8</u>	<u>91.5</u>	73.6	88.4	99.6	37.3	52.8	77.2
FusionED	Z	80.1	<u>81.4</u>	93.9	92.3	73.6	89.0	98.3	41.5	57.9	78.7
VerbalizedED	Z	82.6	<u>78.9</u>	89.2	86.2	69.8	91.4	98.6	65.3	<u>67.0</u>	81.0
+ iter. prediction	Z	88.2	78.9	92.2	88.4	<u>71.5</u>	90.8	98.2	<u>66.3</u>	65.9	<u>82.3</u>
<i>Proposed Approach</i>											
Dual Enc. (VerbalizedED)	Z	80.2	77.2	89.0	85.2	69.4	91.0	98.6	65.3	65.8	80.2
SELECTED GPT-5 mini	Z	<u>84.4</u>	81.6	90.0	88.0	<u>71.5</u>	92.1	<u>98.7</u>	77.0	77.0	84.5
<i>Other Training Data / Partial Evaluations (Not directly Comparable)</i>											
BLINK _{dual encoder only}	B	80.6	77.3	90.8	87.8	68.2	79.8	97.9	50.1	57.3	76.6
BLINK _{dual+cross encoder}	B	84.2	82.4	92.8	91.2	77.3	82.3	99.2	64.8	74.2	83.2
ChatEL _{GPT-3.5 Full}	B+CL	82.1	—	—	—	71.1	77.1	—	—	—	—
ARTER _{Claude 3.5 Son. Sel.}	Wiki+A	85.5	—	—	—	70.4	83.6	—	—	—	—
ARTER _{Claude 3.5 Son. Full}	Wiki+A	86.3	—	—	—	70.7	83.5	—	—	—	—

Table 4: ZELDA benchmark results (Accuracy %). Shaded rows denote our methods. The bottom section reports results not directly comparable due to different training data and label sets (B: BLINK; B+CL: ChatEL uses BLINK and candidate lists; Wiki+A: ARTER uses ReFinED pretrained on Wikipedia and fine-tuned on AIDA). In the top section, bold and underlined values indicate 1st and 2nd best results among directly comparable ZELDA-trained models. The bottom section is included for context: These systems differ in training data, label sets, and/or refinement model, and therefore cannot be compared directly.

Metric	Variant	AIDA	TWEE.	RED.- POSTS	RED.- COMM.	CWEB	WNED WIKI	SLINKS TAIL	SLINKS SHAD.	SLINKS TOP	Avg.
Acc. %	Selective	84.62	81.81	90.47	88.38	71.60	91.55	98.59	72.20	75.22	83.83
	Full-Prompting	82.78	79.38	84.14	84.06	67.57	<u>85.40</u>	94.22	72.78	77.21	80.84
Tokens	Selective	713k	120k	71k	70k	3.5M	530k	10k	186k	199k	607k
	Full-Prompting	2.2M	451k	358k	325k	9.1M	3.3M	526k	539k	544k	1.9M

Table 5: Comparison of accuracy and token usage between selective and full prompting. Selective LLM usage significantly reduces cost while maintaining or improving performance on most datasets. Bold and underlined values indicate 1st and 2nd best accuracy. Token counts are rounded for readability.

original and the ZELDA-retrained BLINK dual encoder. Although FusionED remains highly competitive and achieves the best scores on 3 of 9 datasets, SELECTED shows clear advantages on the more challenging SHADOWLINKS subsets.

Although these systems are not evaluated under the directly comparable ZELDA-trained setup, the reported ChatEL and ARTER results provide additional context on the datasets where all systems report results (AIDA, CWEB, WNED-WIKI). This comparison is notable because ChatEL prompts all mentions and ARTER prompts approximately 35.5%, while SELECTED prompts only 19.3% of mentions⁵.

Additional comparisons on MSNBC, AQUAINT, and ACE2004 are provided in App. F, Table 20. These datasets are not used during parameter selection and therefore serve as robustness tests for

the fixed SELECTED configuration. Across all three datasets, SELECTED consistently improves over its VerbalizedED backbone. The results are in a similar range to reported ChatEL and ARTER numbers, while SELECTED prompts only 24% of spans compared to 100% and 44%, respectively. However, we do not claim state-of-the-art performance on these datasets. The strong BLINK cross-encoder, which relies on routing all mentions to an expensive cross-encoding step, remains best-performing. Direct comparisons are affected by differences in training data, label sets, and refinement architectures.

For further insights into LLM-induced changes and qualitative examples of corrections and errors, see App. G and Table 23.

4.3 Ablation: Comparison to Base Model

The gray rows in Table 4 show our base dual encoder and the full system with selective LLM re-

⁵The row ARTER_{Claude-3.5 Son. Full} reports their reference for full LLM usage.

finement. Across datasets, SELECTED improves average accuracy by 4.3 points over the base model. Additional details are provided in App. D and Table 18, which per dataset reports token usage, the share of spans forwarded to the LLM, and accuracy gains on refined cases. On this subset, improvements are substantially larger (14.5 points on average). Overall, only 20% of mentions are forwarded to the LLM, showing that selective refinement improves performance while limiting LLM usage.

4.4 Ablation: Selective vs. Full Prompting

We analyze the impact of selectively prompting the LLM only for *uncertain* dual encoder predictions by comparing it to a variant that prompts the LLM for *all* mentions⁶.

Selective prompting outperforms full prompting.

Table 5 reports accuracy and token usage for both settings. Surprisingly, selective prompting outperforms full prompting on average (83.83 vs. 80.84) while using substantially fewer tokens. Given the stark disparity in model size between the LLM and the encoder model (220M parameters⁷), this finding may be considered counterintuitive. To better understand this effect, we group mentions by the same criterion used for selection (candidate similarity difference) and compare LLM vs. dual encoder performance across certainty buckets. Figure 2 plots the resulting accuracy differences between the LLM and the dual encoder across certainty buckets, both overall and for three representative datasets.

Dual encoder beats LLM on high-certainty mentions. The LLM improves performance on ambiguous mentions (low-certainty buckets), with the largest gains for the most uncertain cases. However, it degrades performance for high-confidence dual encoder predictions, where correct predictions are sometimes changed to incorrect ones in the multiple-choice setting. A similar effect is reported by Xin et al. (2025), who find that using an LLM as a universal reranker degrades the performance of their underlying base ED model. This highlights that selective routing is crucial and explains why selective prompting outperforms full prompting.

5 Related Work

Entity disambiguation (ED) is a long-standing problem in natural language processing. Early ap-

⁶We use Gemini 2.5 Flash-Lite instead of GPT-5 mini in this ablation to avoid high cost when prompting all mentions.

⁷Both encoders use bert-base-uncased (Devlin et al., 2019) as the backbone, each consisting of 110M parameters.



Figure 2: Accuracy difference between the LLM and the dual encoder (DE) across certainty levels for three ZELDA test sets. Spans are grouped by the similarity difference between the top two candidates (smaller values indicate higher ambiguity). Positive values indicate higher LLM accuracy, negative values dual encoder superiority. The average is computed over all 9 test sets.

proaches relied on heuristic features, graph-based methods, and global inference across documents (Ganea and Hofmann, 2017; Le and Titov, 2018; Yang et al., 2018).

Retrieval-based ED. More recently, research has shifted toward neural architectures, particularly retrieval-based models that scale effectively to large knowledge bases (Wu et al., 2020; De Cao et al., 2021; Yamada et al., 2022; Ayoola et al., 2022; Atzeni et al., 2023). Many modern ED systems follow a two-stage pipeline consisting of candidate retrieval followed by candidate ranking (Yang et al., 2019; Wu et al., 2020; Yamada et al., 2022; Procopio et al., 2023). To reduce the search space, many systems rely on fast retrieval models or candidate lists or alias tables that map mentions to a small set of potential entities (Ganea and Hofmann, 2017; Le and Titov, 2018; Yamada et al., 2022; Procopio et al., 2023; Wang et al., 2024).

Many systems further apply a cross encoder to rank retrieved candidates by jointly encoding mention context and candidate descriptions (Wu et al., 2020; Liu et al., 2023). Cross encoders capture richer interactions but are costly for large candidate sets. Alternatives like FusionED (Wang et al., 2024) use encoder-decoder interactions. Our work builds on the efficient dual encoder paradigm while addressing low-certainty cases.

LLMs for ED. Recent work leverages LLMs for entity disambiguation, using their reasoning and world knowledge (Xu et al., 2023; Liu et al., 2024;

Zhou et al., 2024; Ding et al., 2024). A common approach frames ED as a multiple-choice task where the LLM selects the correct entity from a candidate list (Ding et al., 2024, 2025; Hou et al., 2025; Wang et al., 2025). Other work uses LLMs to augment mention context rather than make the final prediction (Xin et al., 2025; Vollmers et al., 2025). While effective in ambiguous cases, these methods increase computational cost and risk hallucinations. Fully LLM-based ED is also hard to scale due to high inference costs (Glentis et al., 2025). In contrast, our approach invokes LLMs selectively for uncertain predictions.

Hybrid and Multi-Stage Approaches. Recent work integrates LLMs as rerankers or for context augmentation (Xin et al., 2025; Vollmers et al., 2025). Interestingly, Xin et al. (2025) observe that full LLM reranking often performs poorly.

A closely related framework is ARTER (Li et al., 2025), which routes mentions by predicted difficulty and forwards harder cases to an LLM. We follow the same broad principle, but our approach differs in two ways. First, it does not train an additional router. Instead, it uses uncertainty signals already produced by the dual encoder. Second, our focus is a systematic design study of selective LLM refinement, including routing criteria, thresholding, candidate set size and ordering, verbalization format, batching, iterations, and LLM choice. This makes our work complementary to ARTER and to earlier hybrid systems.

Contextual, Document-Level ED. Document context is crucial for resolving ambiguities. Prior work leverages global inference, coherence signals, or iterative methods across multiple mentions, using previously resolved ones as input for later predictions (Ganea and Hofmann, 2017; Le and Titov, 2018; Yang et al., 2018, 2019; Provatorova et al., 2022; Yamada et al., 2022; Oba et al., 2022; Xiao et al., 2023b; Rücker and Akbik, 2025). We achieve a similar effect via iterative contextual verbalization: high-certainty predictions are inserted into the document to aid disambiguation of uncertain mentions. The LLM prompt always includes the full document, providing access to global context.

6 Conclusion

We introduced SELECTED, a certainty-guided hybrid framework for entity disambiguation that combines a fast dual encoder with selective LLM refinement. By invoking the LLM only for low-certainty

mentions, our approach addresses limitations of retrieval-based models in ambiguous cases while remaining computationally efficient. We further conducted a systematic study of key design choices for efficient selective refinement, including routing strategies and prompt design aspects such as candidate ordering and mention batching.

Extensive experiments on ZELDA show that SELECTED improves accuracy by 4.3%, establishing a new best result among directly comparable ZELDA-trained systems, while requiring LLM prompting for only around 20% of mentions. Additional experiments on MSNBC, AQUAINT, and ACE2004 show that the selected configuration transfers beyond the tuning subsets, improving over the dual-encoder backbone while prompting only a minority of mentions. Compared to full prompting, selective refinement achieves higher accuracy with substantially lower token usage.

Limitations

While our approach demonstrates significant improvements in entity disambiguation, several limitations remain that offer avenues for future work.

Fixed backbone and no fine-tuning. Our experiments use the released VerbalizedED dual encoder as the backbone and focus on inference-time selective LLM refinement without additional model training. This setup allows us to isolate the effect of routing and LLM refinement on top of a strong ED model, but our conclusions may partly depend on the backbone’s similarity signal and candidate recall. We leave evaluations with additional dual encoders and task-specific fine-tuning to future work. Such extensions could include smaller or fine-tuned LLMs, learned confidence estimators, jointly trained routing models, or end-to-end systems integrating retrieval, routing, and disambiguation.

Partial hyperparameter selection on benchmark test sets. Hyperparameter selection is performed on subsets of the ZELDA test sets due to the absence of a dedicated development split. While we mitigate this by evaluating the final setting on all test sets as well as on additional unseen benchmarks, and experimenting with different LLMs using the same fixed parameters, this setup may still introduce some overfitting to the selected subsets.

Constrained parameter search. Due to computational cost, we do not perform an exhaustive combinatorial search over all design choices (e.g., num-

ber, verbalization and ordering of candidates, selection threshold). Instead, we use a sequential tuning strategy on ZELDA. Although not exhaustive, the resulting configuration transfers well across multiple LLMs and yields competitive results on additional benchmarks (App. F) while reducing token usage. We did not explore additional signals from the dual encoder directly into the prompt, such as raw similarity scores or candidate score margins.

Potential LLM training data overlap. As with most LLM-based systems, the training data of the underlying LLM is not fully transparent and may overlap with common ED benchmarks. This limitation affects all LLM-based methods considered in this work equally.

Problems of comparability. Some compared systems are not fully aligned with the standardized ZELDA setup, including differences in training data, label sets, architectures, and refinement modules. We therefore treat Table 4 primarily as an accuracy comparison under comparable benchmark conditions for the ZELDA-trained systems. The reported percentages of refined spans provide a proxy for second-stage usage, but they are not a direct inference-cost comparison across systems with different refinement modules.

English-only evaluation. All experiments are conducted on English datasets, without evaluating multilingual or cross-lingual settings. While we cover diverse English benchmarks (nine ZELDA splits plus three additional datasets), further work is needed to assess cross-lingual generalization. Future directions include extending selective routing to multilingual ED and evaluating whether uncertainty signals behave similarly across languages.

Retriever recall, abstention, and NIL predictions. Since SELECTED refines only the candidates retrieved by the dual encoder, its maximum accuracy is bounded by retriever recall: If the gold entity is absent from the top- k candidates, the LLM cannot select it. This is a standard limitation of retrieve-and-rank ED systems. SELECTED should be viewed as an accuracy–efficiency improvement over a strong retriever rather than a general solution to all difficult ED cases.

We provide a limited analysis of these cases by reporting how often the gold label is absent from the top- k candidates and how often the model selects the “None of the above” (NOTA) option (Appendix G, Tables 21, 22, and 24). However,

our benchmarks do not contain genuine NIL cases, where no valid label exists in the underlying label set. A systematic study of true NIL prediction, abstention strategies, and uncertainty calibration therefore remains important future work.

Ethical Considerations

While SELECTED aims to improve entity disambiguation efficiently, several ethical considerations remain. Like any method leveraging LLMs, it inherits limitations from both the model and the underlying data. The LLM itself may reproduce societal biases, such as gender, nationality, or occupation stereotypes, and favor frequent entities over less common ones (Chen et al., 2021; Provatorova et al., 2021). Similarly, human-written label descriptions and knowledge bases may contain errors or biases, which can disproportionately affect rare or underrepresented entities.

For all systems that involve LLM API calls, environmental impact should be considered. A key advantage of SELECTED is its selective LLM usage: only the most uncertain mentions are prompted, reducing token usage, lowering environmental impact, and limiting exposure to potential biases. Nevertheless, caution is required in high-stakes or sensitive domains, such as medical texts where confidential information is present. Sending such data to external LLMs risks privacy breaches, and misclassifications or biased predictions could have serious real-world consequences.

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-based generative tools that assisted with language editing and improving the clarity and presentation. These tools were used solely to refine phrasing and readability. The research ideas, experimental design, results, and interpretations presented in this paper are entirely our own.

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.
- Akhiad Bercovich, Itay Levy, Izik Golan, Mohammad Dabbah, Ran El-Yaniv, Omri Puny, Ido Galil, Zach Moshe, Tomer Ronen, Najeeb Nabwani, Ido Shahaf, Oren Tropp, Ehud Karpas, Ran Zilberstein, Jiaqi Zeng, Soumye Singhal, Alexander Bukharin, Yian Zhang, Tugrul Konuk, ..., and Chris Alexiuk. 2025. [Llama-Nemotron: Efficient Reasoning Models](#). *ArXiv*, abs/2505.00949.
- Nicholas Botzer, Yifan Ding, and Tim Weninger. 2021. [Reddit entity linking dataset](#). *Information Processing Management*, 58(3):102479.
- 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.
- DeepSeek-AI. 2025. [DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning](#). *ArXiv*, abs/2501.12948.
- 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, Amrit Poudel, Qingkai Zeng, Tim Weninger, Balaji Veeramani, and Sanmitra Bhattacharya. 2025. [EntGPT: Entity linking with generative large language models](#). *ArXiv*, abs/2402.06738.
- 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](#). *ArXiv*, 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 Baldrige, 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.
- Athanasios Glentis, Jiaxiang Li, Qiulin Shang, Andi Han, Ioannis Tsaknakis, Quan Wei, and Mingyi Hong. 2025. [Scalable parameter and memory efficient pretraining for LLM: Recent algorithmic advances and benchmarking](#). *ArXiv*, abs/2505.22922.
- Google DeepMind. 2025a. Gemini 2.5 Flash-Lite. <https://deepmind.google/models/gemini/flash-lite/>.
- Google DeepMind. 2025b. Gemini 2.5 Pro. <https://ai.google.dev/gemini-api/docs/models/gemini-2.5-pro?hl=en>.
- 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 Fürstenauf, 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.
- Jiajun Hou, Chenyu Zhang, and Rui Meng. 2025. [Harnessing deep LLM participation for robust entity linking](#). *ArXiv*, abs/2511.14181.
- 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.
- Yajie Li, Albert Galimov, Mitra Datta Ganapaneni, Pujitha Thejaswi, De Meng, Priyanshu Kumar, and Saloni Potdar. 2025. [Leveraging the power of large language models in entity linking via adaptive routing and targeted reasoning](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 871–882, Suzhou (China). Association for Computational Linguistics.
- 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.
- Yi Liu, Yuan Tian, Jianxun Lian, Xinlong Wang, Yanan Cao, Fang Fang, Wen Zhang, Haizhen Huang, Weiwei Deng, and Qi Zhang. 2023. [Towards better entity linking with multi-view enhanced distillation](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9729–9743, Toronto, Canada. 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.
- OpenAI. 2024. GPT-4o mini. <https://developers.openai.com/api/docs/models/gpt-4o-mini>.
- OpenAI. 2025. GPT-5 mini. <https://developers.openai.com/api/docs/models/gpt-5-mini>.
- Pouya Pezeshkpour and Estevam Hruschka. 2024. [Large language models sensitivity to the order of options in multiple-choice questions](#). In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2006–2017, Mexico City, Mexico. 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](#). *ArXiv*, abs/2210.06164.
- Lev Ratnov, 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.
- Red Hat AI and Meta. 2025. [Llama-4-Scout-17B-16E-Instruct Quantized Model](#). Hugging Face model repository.
- Susanna Rücker and Alan Akbik. 2025. [Evaluating design decisions for dual encoder-based entity disambiguation](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15685–15701, Vienna, Austria. Association for Computational Linguistics.
- Daniel Vollmers, Hamada Zahera, Diego Moussallem, and Axel-Cyrille Ngonga Ngomo. 2025. [Contextual augmentation for entity linking using large language models](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 8535–8545, Abu Dhabi, UAE. Association for Computational Linguistics.
- Fang Wang, Zhengwei Tao, Ming Wang, Minghao Hu, and Xiaoying Bai. 2025. [AELC: Adaptive entity](#)

- linking with LLM-driven contextualization. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 4313–4327, Suzhou, China. 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, Ming Gong, Jie Wu, Xingyao Zhang, Linjun Shou, and Daxin Jiang. 2023a. [Instructed language models with retrievers are powerful entity linkers](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2267–2282, Singapore. Association for Computational Linguistics.
- Zilin Xiao, Linjun Shou, Xingyao Zhang, Jie Wu, Ming Gong, and Daxin Jiang. 2023b. [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.
- Amy Xin, Yunjia Qi, Zijun Yao, Fangwei Zhu, Kaisheng Zeng, Xu Bin, Lei Hou, and Juanzi Li. 2025. [LL-MAEL: Large language models are good context augmenters for entity linking](#). *ArXiv*, abs/2407.04020.
- 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.
- Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2024. [Large language models are not robust multiple choice selectors](#). *ArXiv*, abs/2309.03882.
- Kang Zhou, Yuepei Li, Qing Wang, Qiao Qiao, and Qi Li. 2024. [GenDecider: Integrating “none of the candidates” judgments in zero-shot entity linking re-ranking](#). 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.

Appendix

This appendix provides additional experimental details and results. Sect. **A** reports the complete results for the hyperparameters that were summarized in Sect. **3**. Sect. **B** presents the full comparison of LLMs discussed in Sect. **3.6**, and Sect. **C** includes the prompt used for our LLM component.

Sect. **D** provides a per-dataset comparison between the base dual encoder and selective LLM refinement, including accuracy gains, the proportion of spans forwarded to the LLM and token usage. Sect. **E** contains an additional ablation study analyzing the impact of verbalizations. Sect. **F** includes more results for the comparison to BLINK, ChatEL and ARTER on a different set of benchmarks.

Finally, Sect. **G** reports further performance insights, such as the amount of positive and negative label changes introduced by the LLM, the dual encoder accuracy on *certain* (i.e. not LLM-refined) cases, the amount of cases in which the gold entity is (not) contained in the candidate set as well as where the LLM selected the “None of the above” answer. Qualitative examples are presented in Table **23**.

A Design Parameter Selection – Details

We present detailed results concerning the design choices which were only summarized in Sect. 3.

A.1 Selection Strategy and Threshold

This parameter determines which spans are determined as *uncertain* and thus, forwarded to the LLM. Since a key goal of our approach is to minimize LLM usage, we aim to set the threshold so that the least amount of spans possible are forwarded, while maintaining strong overall accuracy. We evaluate the following strategies for span selection:

- **Relative threshold on Top₁ similarity:** Select the worst $n\%$ of spans according to the top candidate similarity score.
- **Absolute threshold on Top₁ similarity:** Select all spans where the top candidate’s similarity score falls below a certain threshold.
- **Relative threshold on candidate similarity difference:** Select the $n\%$ of spans with smallest similarity difference between Top₁ and Top₂ candidate, indicating low certainty.
- **Absolute threshold on candidate similarity difference:** Select all spans where the difference between the Top₁ and Top₂ candidate falls below a certain threshold.
- **Combined:** Select based on both low Top₁ similarity and small candidate difference.

Due to the large number of possible threshold values and combinations, and the computational cost of repeated LLM inference, we conduct this evaluation on a representative subset of the ZELDA datasets rather than the full benchmark⁸.

Relative threshold on Top₁. This strategy selects the worst $n\%$ of spans according to the similarity score of the top candidate. While simple, it does not directly reflect model certainty and instead assumes that a fixed proportion of spans should be refined by the LLM. As a result, it may overuse the LLM on straightforward cases or underuse it when many spans are challenging. Performance typically peaks at an intermediate threshold: prompting too many spans can reduce accuracy when the dual encoder is already reliable, whereas prompting too few leaves challenging cases unresolved.

Table 6 reports averaged results. The highest overall accuracy (82.76%) is achieved with a threshold of 0.7 (30% of spans are prompted). However, 0.8 yields very similar accuracy (82.72%) while reducing token usage by nearly 30%. We therefore select **0.8** for comparison with other strategies.

LLM accuracy (calculated only on the spans that are refined) decreases steadily as the threshold increases. This does not indicate worse model behavior. Instead, higher thresholds forward fewer but increasingly *difficult* spans to the LLM. These spans correspond to highly ambiguous cases where both models make more errors, which lowers the measured LLM accuracy.

Absolute threshold on Top₁. This strategy selects spans based on the absolute similarity score rather than a fixed percentage. Spans whose top candidate similarity falls below the threshold are forwarded to the LLM, while spans above the threshold are accepted directly. Consequently, the number of prompted spans varies across datasets: straightforward documents trigger fewer LLM calls, while ambiguous ones trigger more.

We evaluate several score thresholds. Table 7 summarizes the results. Intermediate thresholds generally perform best, balancing LLM refinement and reliance on the dual encoder. The best overall accuracy is achieved with a threshold of **-22** (euclidean distance), yielding 83.30% accuracy while prompting 25.50% of spans and using 215,826 tokens (-54.2%). Compared to the relative threshold strategy, this approach achieves higher accuracy with a similar prompting rate, indicating that certainty-based selection is more effective than forwarding a fixed percentage of spans.

Relative threshold on candidate difference. This strategy selects spans based on the score difference between the Top₁ and Top₂ candidates predicted by the dual encoder. A small difference indicates uncertainty because the model assigns similar scores to multiple candidates. We prompt the $n\%$ of spans with the smallest candidate differences. As with the relative Top₁ threshold, this approach routes a fixed proportion of spans to the LLM, regardless of the absolute score difference.

Table 8 reports the averaged results. Performance again peaks at an intermediate threshold. The best configuration is a threshold of **0.75**, prompting 25% of spans and achieving an average overall accuracy of 83.75%. This setting is used for comparison with the other strategies.

⁸These were: SLINKS-SHADOW/ -TAIL, AIDA, REDDIT-COMMENTS, TWEET.

Rel. Threshold Top ₁	Accuracy %		Efficiency	
	SELECTED	LLM (Ref.)	Tokens	Rel. Δ
0.0	81.39	81.39	721 994	—
0.1	81.51	80.03	664 668	-7.9
0.2	81.51	78.28	603 277	-16.4
0.3	81.31	76.28	540 773	-25.1
0.4	81.72	74.89	476 428	-34.0
0.5	82.22	73.42	405 759	-43.8
0.55	82.14	72.16	371 107	-48.6
0.6	82.38	71.37	333 454	-53.8
0.65	82.69	70.17	301 894	-58.2
0.7	82.76	68.40	265 357	-63.2
0.75	82.51	65.53	225 203	-68.8
0.8	<u>82.72</u>	64.72	190 973	-73.5
0.85	82.59	61.04	149 981	-79.2
0.9	82.22	56.05	111 908	-84.5
0.95	81.67	46.57	65 015	-91.0

Table 6: Average overall accuracy and on the cases refined by the LLM for **different relative Top₁ thresholds**. We also report required token counts along with the the relative decrease compared to when prompting all spans (threshold at 0.0). The bold and underlined value indicates the best and second-best overall accuracy.

Abs. Threshold Top ₁	Accuracy %		Efficiency		
	SELECTED	LLM (Ref.)	Spans prompted %	Tokens	Tokens Rel. Δ
-18	82.43	75.70	62.39	471 672	—
-19	82.53	73.45	53.02	412 982	-12.4
-19.5	82.78	73.26	48.42	380 900	-19.2
-20	<u>83.09</u>	72.58	43.73	349 299	-25.9
-20.5	82.89	70.47	39.28	316 138	-33.0
-21	83.06	69.46	34.65	284 316	-39.7
-22	83.30	66.61	25.50	215 826	-54.2
-23	83.05	61.35	17.26	148 369	-68.5
-24	82.61	52.75	10.65	90 079	-80.9
-25	82.06	41.32	5.47	48 974	-89.6

Table 7: Average overall accuracy and on the cases refined by the LLM for **different absolute Top₁ thresholds**. We also report the amount of spans given to the LLM. Relative token decrease compared to the setting where most spans are prompted.

Absolute threshold on candidate difference.

This strategy also uses an absolute threshold: All spans whose score difference between the Top₁ and Top₂ candidates falls below a predefined value are forwarded to the LLM. This allows the number of prompted spans to vary across datasets.

Table 9 summarizes the results. The best performance is obtained with a threshold of **1**, reaching 84.83% average accuracy while prompting 19.36% of spans and using 190,138 tokens (-73.8%). Compared to the best relative-threshold strategies, this improves accuracy by more than one point while requiring fewer tokens. Figure 3 complements the averaged results with a dataset-level view. Very low thresholds prompt too few mentions, while very high thresholds approach full prompting and may reduce accuracy by overriding correct high-certainty dual encoder predictions. Intermediate thresholds therefore provide the strongest trade-off,

motivating our final choice of 1.

The figure also shows dataset-level variation: SHADOWLINKS-SHADOW benefits from higher thresholds, suggesting that more aggressive refinement helps in highly ambiguous cases, whereas SHADOWLINKS-TAIL remains strong across many thresholds but degrades when too many mentions are prompted. Overall, this supports using an absolute rather than relative threshold, as it allows the number of prompted spans to vary with each dataset’s certainty distribution.

Combination of Top₁ and candidate difference.

Our final approach combines the two previously evaluated criteria. For each dataset, we create two rankings of spans: one based on the Top₁ similarity score and one based on candidate difference. These rankings are then merged by alternately selecting spans from each list while avoiding dupli-

Candidate Diff. (rel.)	Accuracy %		Efficiency	
	SELECTED	LLM (Ref.)	Tokens	Rel. Δ
0.0	81.17	81.17	725 276	—
0.1	81.44	79.65	666 935	-8.0
0.2	81.29	77.27	604 933	-16.6
0.3	81.65	75.17	543 759	-25.0
0.4	82.34	73.80	478 058	-34.1
0.5	82.89	71.40	406 233	-44.0
0.55	82.76	69.45	377 058	-48.0
0.6	83.02	68.00	340 381	-53.0
0.65	83.36	66.56	308 382	-57.5
0.7	83.41	64.12	270 359	-62.7
0.75	83.75	63.06	230 374	-68.2
0.8	83.48	60.90	195 520	-73.0
0.85	83.27	58.23	153 380	-78.9
0.9	82.84	56.54	115 318	-84.1
0.95	82.29	55.26	67 125	-90.7

Table 8: Average overall accuracy and on the cases refined by the LLM for **different relative candidate difference thresholds**. Relative token decrease compared to prompting for all spans (threshold 0.0).

Candidate Diff. (abs.)	Accuracy %		Efficiency		
	SELECTED	LLM (Ref.)	Spans prompted %	Tokens	Tokens Rel. Δ
0.1	82.3	51.44	3.53	40 666	-94.4
0.25	83.03	48.17	7.09	78 495	-89.2
0.5	84.08	52.88	12.34	126 692	-82.5
0.75	84.59	54.01	15.93	161 573	-77.7
1	84.83	55.29	19.36	190 138	-73.8
1.25	84.81	55.75	22.52	214 410	-70.4
1.5	84.65	58.44	25.42	237 722	-67.2
2	84.54	62.61	30.77	278 986	-61.5
3	83.74	67.04	41.49	345 866	-52.3
4	82.99	70.82	51.73	415 153	-42.7
5	82.52	74.20	61.34	479 017	-33.9
7	81.68	78.14	78.63	583 310	-19.6
9	81.24	79.93	90.96	664 210	-8.4
(All spans)	81.19	81.19	100.00	725 084	—

Table 9: Average overall accuracy and on the cases refined by the LLM for **different absolute candidate difference thresholds**. Relative token decrease compared to prompting all spans.

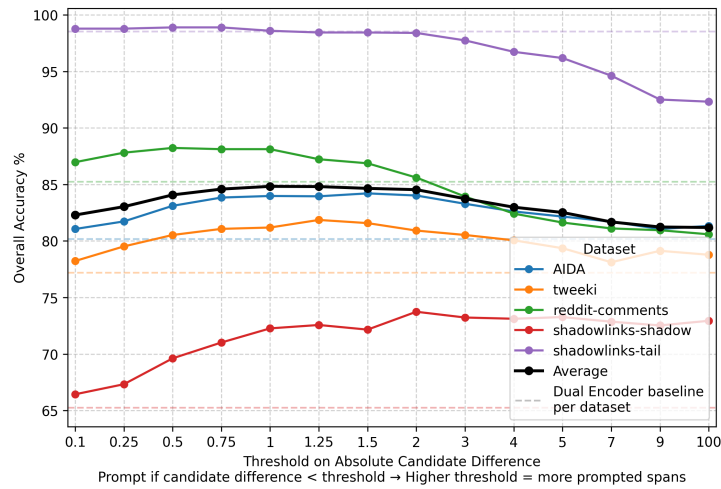


Figure 3: Threshold sensitivity for the **absolute candidate difference** strategy. A mention is routed to the LLM when the similarity difference between the top two Dual Encoder candidates is below the threshold. Thus, higher thresholds result in more prompted spans. Solid lines show overall accuracy for each dataset and their macro-average. Faded dashed lines indicate the corresponding dual encoder baseline accuracy without LLM refinement.

Relative Threshold	Accuracy %		Efficiency	
	SELECTED	LLM (Ref.)	Tokens	Rel. Δ
0.0	81.32	81.32	724 153	—
0.1	81.14	79.53	662 794	-8.5
0.2	81.29	77.82	603 605	-16.6
0.3	81.79	76.12	543 747	-24.9
0.4	81.79	73.76	478 858	-33.9
0.5	82.53	72.23	405 456	-44.0
0.55	82.28	70.44	375 401	-48.2
0.6	82.74	69.97	337 148	-53.4
0.65	82.94	68.59	304 236	-58.0
0.7	83.06	66.47	266 682	-63.2
0.75	83.26	64.91	227 515	-68.6
0.8	83.05	61.92	195 277	-73.0
0.85	82.89	58.26	151 699	-79.1
0.9	82.58	55.22	114 299	-84.2
0.95	82.05	49.82	62 543	-91.4

Table 10: Average overall accuracy and on the cases refined by the LLM when using a **combination** of Top₁ similarity and score difference of top two candidates. Relative token decrease compared to prompting all spans.

cates, ensuring that spans by either criterion are selected. We apply a relative threshold on the combined ranking to select the worst $n\%$ of spans for LLM refinement. Absolute thresholds are not used because defining separate cutoffs for both criteria would complicate the procedure.

Results are reported in Table 10. The best accuracy is achieved at a threshold of 0.75, reaching 83.26% overall accuracy with 227,515 tokens. The second-best result is obtained at 0.7 (83.06%), with similar performance at 0.8. Higher or lower thresholds reduce accuracy while providing only modest token savings, we therefore select **0.75** (prompting 25% of spans) as best for this strategy.

Conclusion. In this evaluation we compared both different strategies for identifying uncertain spans and corresponding threshold values. Table 2 in Sect. 3.3 summarizes the best configuration per strategy. The highest overall accuracy is achieved by using the **absolute candidate difference** with a threshold of **1**, resulting in **84.83%** accuracy. This result exceeds all other strategies by more than one percentage point. Notably, it also prompts fewer spans (**19.36%**) and uses fewer tokens (**190,138**) than the competing approaches, indicating that this strategy most effectively identifies challenging spans where LLM usage is helpful.

A.2 Verbalization Strategy

The verbalization strategy determines what information about the candidates is provided to the LLM and also, what information about already predicted mentions is included as additional context.

We experiment with the following different formats, an example can be found in Table 11:

- **None:** The LLM does not receive any additional information for a label candidate or an already predicted span. As candidates, the LLM will only receive the label name, which generally corresponds to the title of the Wikipedia entry. Predicted spans in the text are not verbalized.
- **Description:** This approach provides the LLM with a short description for each candidate or already predicted span, obtained from Wikidata. Usually a one-sentence overview.
- **Categories:** Each entry in Wikidata is part of relations. We extract a specific subset of these and provide their values. For example, the label “Apple_Inc.” is categorized as a “business” (*instance_of*). We provide the label name and the category key-value pairs to the LLM.
- **Description + Categories:** This approach combines the previous two and provides the name of the label, a short description, and the categories. While this approach conveys the most information, it also significantly increases the length of the prompt.

Table 12 shows average overall and LLM accuracy (on the refined cases) along with token counts. Overall, the best overall accuracy is achieved with Description + Categories (83.83%), closely followed by Categories only (83.80%). Looking at LLM accuracy, using Categories performs

Strategy	Eiffel_Tower
None	Eiffel Tower
Description	Eiffel Tower - tower located on the Champ de Mars in Paris, France
Categories	Eiffel Tower - instance of: lattice tower, observation tower, tourist attraction, architectural landmark ; part of: banks of the Seine ; country: France
Description + Categories	Eiffel Tower - tower located on the Champ de Mars in Paris, France ; instance of: lattice tower, observation tower, tourist attraction, architectural landmark ; part of: banks of the Seine ; country: France

Table 11: Different verbalization strategies illustrated with the label Eiffel_Tower.

slightly better (55.08%) than the combined approach (54.51%). Using only Descriptions generally decreases both overall and LLM accuracy, and in some cases, providing descriptions even harms performance for the refined spans.

Considering token efficiency, Categories only requires significantly fewer tokens than the combined approach while achieving nearly identical overall accuracy. Given the marginal accuracy difference and the substantial token savings, we select **Categories only** as the preferred verbalization strategy.

A.3 Number of Iterations

This parameter controls the number of LLM prediction iterations: selected spans are split into batches, according to the selection criterion, and processed sequentially, starting with the most certain spans. After each iteration, verbalizations of the accepted predictions are inserted into the text, providing additional context for subsequent batches and helping the LLM resolve more challenging spans.

We evaluated different numbers of iterations with respect to accuracy, token usage, and execution time (Table 13). Accuracy is largely unaffected by the number of iterations while token usage and execution time increase: token counts grow because fewer spans can be combined per prompt, causing the document text to be repeated more often, while execution time increases roughly linearly since iterations are processed sequentially.

Considering the trade-off between accuracy, token cost, and execution time, we recommend **two iterations**. This yields a modest accuracy gain over a single iteration while keeping resource requirements low. Additional iterations provide only marginal improvements and are not justified given

the substantially higher token and time costs.

A.4 Candidate Ordering

Within the prompt, the LLM is presented with a set of candidate labels, which are, by default, sorted by their similarity score from the dual encoder. To evaluate the impact of candidate order, we also test a randomized one. The results in Table 14 show that sorting candidates with decreasing similarity consistently outperforms random order by nearly 2.5 percentage points on average (LLM accuracy). These findings suggest that the LLM benefits from the implicit ordering, so we adopt **similarity-based sorting** as the preferred strategy.

A.5 Number of Spans per Prompt

This parameter controls how many mentions (predictions) are combined into a single LLM prompt. Combining spans saves input tokens, since the document text only needs to be included once. However, it may reduce accuracy, as the LLM must handle more tasks simultaneously. The results in Table 15 show that accuracy is largely unaffected. The token saving plateaus as the number of spans per prompt increases beyond five⁹.

Considering both accuracy and token efficiency, we find that using **five spans per prompt** provides the best trade-off. It achieves the highest average accuracy (83.83%) while reducing token usage to 79.43% relative to using a single span per prompt.

B Comparing different LLMs – Results

As briefly summarized in Sect. 3.6, we experimented with several LLMs for the LLM module of SELECTED. Our evaluation includes both widely used cloud-based LLMs accessed via API and a selection of locally installed models. For cloud-based LLMs, we primarily consider models from Google and OpenAI. We compare the following LLMs:

- **GPT-4o mini** and **GPT-5 mini** (OpenAI, 2024, 2025) (via API)
- **Gemini 2.5 Flash Lite** and **Gemini 2.5 Pro** (Google DeepMind, 2025a,b) (via API)
- **DeepSeek R1 Distill Qwen 32B**¹⁰ (DeepSeek-AI, 2025) (locally)

⁹Token savings are strongly dependent on the dataset: combining spans yields substantial reductions for datasets that commonly include multiple mentions per document, while datasets with few such documents see little or no reduction.

¹⁰Available at <https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-32B>

Verbalization Strategy	Accuracy %		Efficiency	
	SELECTED	LLM (Ref.)	Tokens	Rel. Δ
Description	83.65	53.86	577 439	+7.6
Description + Categories	83.83	<u>54.51</u>	772 110	+43.9
Categories	<u>83.80</u>	55.08	684 771	+27.7
None	83.68	54.33	536 433	—

Table 12: Average overall accuracy and on the cases refined by the LLM for using **different verbalization strategies**. Relative token increase compared to using no verbalizations.

Iteration Count	Accuracy %		Efficiency			
	SELECTED	LLM (Ref.)	Tokens	Tokens Rel. Δ	Execution Time (s)	Time Rel. Δ
1	83.60	54.24	522 410	—	29.88	—
2	83.82	54.85	607 856	+16.4	61.54	+106.0
3	83.85	55.65	684 223	+31.0	81.53	+172.9
4	<u>83.86</u>	55.31	757 192	+44.9	106.72	+257.2
5	83.90	<u>55.34</u>	819 022	+56.8	118.32	+296.0

Table 13: Average overall accuracy and on the cases refined by the LLM for **different numbers of LLM iterations**. We also report execution time along with relative increase compared to one single iteration.

- **Llama 3.3 Nemotron Super 49B v1.5**¹¹ (Bercovich et al., 2025) (locally)
- **Llama 4 Scout 17B 16E Instruct (Quantized)**¹² (Red Hat AI and Meta, 2025) (locally)

We evaluate both overall accuracy on ZELDA and required token count (input + output tokens) for each dataset. To mitigate LLM randomness, we execute each benchmark multiple times and present averaged results in Table 16.

Across datasets, stronger LLMs improve accuracy, particularly on more challenging, ambiguous datasets. On average, Google’s Gemini 2.5 Pro performs best at 84.55%, with OpenAI’s GPT-5 mini close (84.49%). The remaining models show noticeably lower accuracy, largely due to the strong performance of Gemini and GPT-5 mini on the SHADOWLINKS datasets. On other datasets, alternative LLMs are often competitive. Regarding

¹¹Available at https://huggingface.co/nvidia/Llama-3.3-Nemotron-Super-49B-v1_5

¹²Available at <https://huggingface.co/RedHatAI/Llama-4-Scout-17B-16E-Instruct-quantized.w4a16>

Candidate Order	Accuracy %	
	SELECTED	LLM (Ref.)
Sorted	83.82	55.29
Random	83.31	52.78

Table 14: Average overall accuracy and on the cases refined by the LLM with/without **sorting candidates**.

Spans per Prompt	Accuracy %		Efficiency
	SELECTED	LLM (Ref.)	Tokens Rel. Δ
1	83.68	54.49	100.00
2	83.81	54.74	85.76
3	<u>83.82</u>	54.86	82.14
5	83.83	54.75	79.43
7	83.77	<u>54.81</u>	78.40
10	83.75	54.76	77.83

Table 15: Average overall accuracy and on the cases refined by the LLM for **different number of spans per prompt**. Relative token count compared to one span.

efficiency, Gemini 2.5 Pro requires nearly twice as many tokens as GPT-5 mini. Considering both accuracy and token efficiency, we select OpenAI’s **GPT-5 mini** for our final evaluation in Sect. 4.2.

C Prompt for LLM Prediction

The LLM component of SELECTED, independent of the specific model used, relies on a structured prompt for entity disambiguation. The prompt consists of a system instruction and a user prompt containing the document text (which may include verbalizations of previously linked mentions) together with a list of (verbalized) candidate labels. Each span to be disambiguated is marked with a question tag (e.g., [****Q1****]), enabling the LLM to locate the relevant mention in case of multiple occurrences. Multiple questions can be included in a single prompt to improve efficiency. The full prompt structure is shown in Table 17.

Model	Metric	AIDA	TWEEKI	REDDIT- POSTS	REDDIT- COMMENTS	CWEB	WNED- WIKI	SLINKS- TAIL	SLINKS- SHADOW	SLINKS- TOP	Average
Dual Encoder (No LLM)	Acc. % Tokens	80.18	77.22	89.05	85.24	69.37	91.01	98.55	65.27	65.82	80.19
Gemini 2.5 Flash-Lite	Acc. % Tokens	84.62 712 755	81.81 120 340	90.47 70 976	88.38 69 713	71.60 3 503 224	91.55 592 555	98.59 10 235	72.20 186 222	75.22 199 376	83.83 607 266
Gemini 2.5 Pro	Acc. % Tokens	84.01 1 331 217	81.62 316 365	89.76 210 222	87.86 203 961	71.20 5 679 474	92.48 1 195 673	98.63 35 394	78.28 578 256	77.10 637 695	84.55 1 132 029
GPT-4o mini	Acc. % Tokens	84.44 683 203	81.23 120 077	90.37 70 964	88.33 69 549	71.95 3 479 755	91.11 574 334	98.55 10 540	71.94 181 542	72.57 217 964	83.39 600 881
GPT-5 mini	Acc. % Tokens	84.40 791 332	81.60 135 756	90.04 80 708	88.07 77 672	71.54 3 739 378	92.08 638 093	98.67 12 280	76.99 220 886	77.04 238 942	84.49 659 450
DeepSeek R1 Distill Qwen 32B	Acc. % Tokens	83.52 927 130	80.88 207 269	88.05 124 261	87.02 116 133	70.94 4 329 401	91.07 807 274	98.63 19 843	71.94 340 750	73.89 369 327	82.88 804 599
Llama 3.3 Nemotron Super 49B v1.5	Acc. % Tokens	84.24 1 069 789	81.55 262 089	89.67 161 715	87.86 158 177	71.03 4 843 953	91.85 983 522	98.51 29 480	74.30 530 331	75.15 519 080	83.80 950 904
Llama 4 Scout 17B 16E Instruct (Quantized)	Acc. % Tokens	84.60 677 350	80.95 115 277	90.80 67 920	88.43 68 418	71.96 3 341 565	91.88 567 628	98.55 9 039	70.73 174 836	75.55 188 688	83.72 578 969

Table 16: Comparison of SELECTED accuracy and required token count when plugging in several **different LLMs**. For reference, we also include the base dual encoder. Bold values indicate the best performance on a dataset.

System Instruction: You are a professional entity-disambiguation annotator. For each question in the prompt you must select exactly one answer number. Hard rules (must follow): 1. Read the full text, then answer every question in order (Q1, Q2, ...). 2. For each question, consider ONLY the mention marked with that question tag (e.g., [Q1]) and ONLY the options listed immediately after that question. Do NOT reuse options from other questions. 3. Choose the option that best matches the mention given the context. Prefer contextual relevance over surface-form match when they conflict. Prefer the most specific, contextually accurate entry when multiple options fit. 4. If no option matches, choose '0' (None of the below). 5. Sometimes, the answer numbering may not be continuous. Carefully look at the answer number of your selected option and provide it. 6. Output format: exactly N lines for N questions. Each line must contain exactly one digit (the chosen option number). The first line is the answer for Q1, the second for Q2, etc. No other characters, labels, punctuation, or explanation. No blank lines. 7. Do NOT provide chain-of-thought or any additional text. User Prompt: Consider the following text: <Document Text, possibly including verbalizations of neighboring mentions> Which entry does the mention '' at [**Q1**] refer to? Evaluate all options carefully. (0) None of the answers below (1) <Answer option 1> - <Verbalization> (2) <Answer option 2> - <Verbalization> ... Which entry does the mention '' at [**Q2**] refer to? Evaluate all options carefully. (0) None of the answers below (1) <Answer option 1> - <Verbalization> (2) <Answer option 2> - <Verbalization> ...

Table 17: Structure of the LLM prompt. The model receives a system instruction and a user prompt containing the document text (including inserted verbalizations of resolved mentions). Target mentions are marked (e.g., [****Q1****]) to help locate the span, followed by candidate entities with optional descriptions. Multiple questions may be included in one prompt to reduce token usage.

ZELDA Dataset	Overall Accuracy %			Accuracy on Ref. Cases %			Efficiency	
	SELECTED	Dual Encoder	Δ Acc.%	LLM	Dual Encoder	Δ Acc.%	Tokens	Spans Prompted %
AIDA	84.62	80.18	+4.44	61.66	40.11	+21.55	712 755	20.62
TWEEKI	81.81	77.22	+4.59	55.04	34.87	+20.17	120 340	22.78
REDDIT-POSTS	90.47	89.05	+1.42	55.26	46.49	+8.77	70 976	16.22
REDDIT-COMMENTS	88.38	85.24	+3.14	56.90	39.66	+17.24	69 713	18.21
CWEB	71.60	69.37	+2.23	35.51	27.28	+8.23	3 503 224	27.04
WNED-WIKI	91.55	91.01	+0.54	54.43	49.20	+5.23	592 555	10.18
SLINKS-TAIL	98.59	98.55	+0.04	52.08	50.00	+2.08	10 235	1.78
SLINKS-SHADOW	72.20	65.27	+6.93	57.84	37.09	+20.75	186 222	33.41
SLINKS-TOP	75.22	65.82	+9.40	64.00	37.85	+26.15	199 376	35.95
Average	83.83	80.19	+3.64	54.75	40.28	+14.47	607 266	20.69

Table 18: Comparison of SELECTED and its base dual encoder. We report overall accuracy as well as accuracy on the cases refined by the LLM for which we compare the LLM’s vs. the dual encoder’s performance. We also include the number of required LLM tokens and the amount of mentions prompted per dataset.

D Analysis: Amount of Refined Spans and Accuracy Gains

We analyze both accuracy gains and amount of forwarded spans on the ZELDA test sets after fixing the parameter setting described in Sect. 3. Table 18 reports SELECTED’s overall accuracy as well as accuracy on the spans refined by the LLM¹³, in comparison to the base dual encoder.

Across all datasets, SELECTED consistently outperforms the dual encoder, achieving an average improvement of 3.6% with gains up to 6.9% and 9.4% on the most challenging SHADOWLINKS datasets. When focusing only on spans processed by the LLM, the gains are substantially more pronounced, with an average improvement of 14.5%.

Token usage and the proportion of spans which are forwarded to the LLM vary across datasets, reflecting our relative selection strategy: more spans are prompted for difficult datasets and fewer for easier ones, as illustrated by the SHADOWLINKS datasets. On average, about 20% of spans are forwarded to the LLM. While we use a fixed absolute threshold here, future work could explore dynamically adapting this threshold to dataset difficulty.

E Ablation: Without Verbalizations

We evaluate the impact of adding label verbalizations as additional context for the LLM. Our default implementation verbalizes both already predicted mentions in the text and candidate labels in the

prompt. We compare this to variants where we only verbalize one or none of both options.

Table 19 reports the results. On average, verbalizations provide only a slight accuracy gain, while omitting them can significantly lower token usage. Whether this trade-off is beneficial therefore depends on the application and cost constraints.

Nonetheless, verbalizations remain essential for SELECTED: the candidate retrieval (the initial dual encoder prediction) relies on them, whereas for the LLM, which is able to leverage its inherent world knowledge, verbalizations are less critical.

F MSNBC, AQUAINT, ACE2004 – Results

While Sect. 4 and Table 4 mainly report results on the ZELDA benchmark, we additionally evaluate our approach on three widely used ED datasets: MSNBC (Cucerzan, 2007), AQUAINT (Milne and Witten, 2008), and ACE2004 (Ratinov et al., 2011), for which several prominent systems report results. This additional evaluation helps mitigate the fact that subsets of ZELDA were used during hyperparameter selection, as these three datasets are entirely unseen during tuning. We evaluate using the fixed configuration obtained in Sect. 3 to assess generalizability.

Table 20 reports the results for this comparison, where we also include ZELDA test sets for which the compared systems report results. We again include both BLINK variants, as well as the LLM-based approaches ChatEL (GPT-3.5) and ARTER (with Claude 3.5 Sonnet). We further include the reported results from Verbalized (Rücker and Akbik, 2025) and its specific checkpoint we used as our

¹³This was computed using the Gemini 2.5 Flash-Lite model employed during the hyperparameter search.

Approach	Metric	AIDA	TWEEKI	REDDIT- POSTS	REDDIT- COMMENTS	CWEB	WNED- WIKI	SLINKS- TAIL	SLINKS- SHADOW	SLINKS- TOP	Average
Default Text: ✓ Candidates: ✓	Acc. % Tokens	84.62 712 755	81.81 120 340	90.47 70 976	88.38 69 713	71.60 3 503 224	91.55 529 555	98.59 10 235	72.20 186 222	75.22 199 376	83.83 607 266
No Verbalizations Text: ✗ Candidates: ✗	Acc. % Tokens	83.75 394 420	81.47 79 106	90.56 47 531	87.60 46 846	71.29 2 519 386	91.36 361 671	98.74 7 837	72.16 137 811	75.22 148 213	83.57 415 869
Reduced Verbalizations Text: ✓ Candidates: ✗	Acc. % Tokens	83.84 556 644	81.03 84 350	90.38 50 784	87.76 50 226	71.50 3 007 523	91.22 480 353	98.74 7 837	72.68 138 026	74.63 147 581	83.53 502 592
Reduced Verbalizations Text: ✗ Candidates: ✓	Acc. % Tokens	83.81 548 606	81.62 114 701	90.37 67 514	88.54 65 181	71.35 3 029 351	91.62 472 379	98.67 10 434	72.38 187 101	75.22 200 076	83.73 521 705

Table 19: Comparison of accuracy and token counts for **inserting verbalizations into the Text** for already predicted spans and/or enriching **provided Candidates** in the prompt. Bold values indicate the best accuracy on a dataset.

Method	AIDA	CWEB	WNED- WIKI	MSNBC	ACE2004	AQUAINT	Average Acc. %	% Spans Refined
BLINK _{dual encoder only}	80.6	68.2	79.8	83.5	84.3	<u>87.2</u>	80.6	-
BLINK _{dual+cross encoder}	84.2	77.3	82.3	97.1	98.4	98.7	89.7	100%
ChatEL _{GPT-3.5}	82.1	71.1	77.1	86.6	88.4	79.1	80.7	100%
ARTER _{Claude 3.5 Sonnet}	85.5	70.4	83.6	<u>90.0</u>	86.3	83.9	83.3	-
<i>Spans Prompted</i>	33.0%	40.7%	32.7%	40.5%	49.6%	42.9%	-	39.6%
VerbalizED				+ Adding Labels from MSNBC, ...				
VerbalizED _{base}	82.6	69.8	<u>91.4</u>	80.3	82.5	80.5	81.2	-
SELECTED (Ours)								
Dual Enc. (VerbalizED)	80.2	69.4	91.0	73.1	76.3	72.4	77.1	-
SELECTED _{GPT-5 mini}	84.4	71.5	92.1	81.2	78.9	74.3	80.4	-
<i>Spans Prompted</i>	20.6%	27.0%	10.2%	25.5%	20.2%	21.9%	-	20.9%
SELECTED (Ours)				+ Adding Labels from MSNBC, ...				
Dual Enc. (VerbalizED)	80.2	69.4	91.0	79.9	82.1	80.9	80.6	-
SELECTED _{GPT-5 mini}	84.4	<u>71.5</u>	92.1	88.4	85.9	81.6	<u>84.0</u>	-
<i>Spans Prompted</i>	20.6%	27.0%	10.2%	26.5%	22.6%	21.9%	-	21.5%

Table 20: Evaluation of SELECTED on additional benchmarks. Direct comparisons should be interpreted with caution due to differences in training data. In parallel to VerbalizED, we report results for SELECTED before and after adding the ~15% missing labels from MSNBC, ACE2004, and AQUAINT to our label set (shaded rows). For ChatEL, ARTER, and SELECTED, we report the proportion of spans forwarded to the LLM (*Spans Prompted*), averaged as % Spans refined, where we also note that for BLINK, all spans are forwarded to its cross encoder.

backbone (Dual Encoder (*VerbalizED*)) and finally our proposed approach including selective LLM prompting (SELECTED_{GPT-5 mini}). For systems that involve additional refinement (cross encoder or LLM), we also report the amount of spans that are refined (% Spans Refined).

Since the three additional datasets are not part of ZELDA, our ZELDA-based label set does not cover all gold entities in these datasets. As reported by [Rücker and Akbik \(2025\)](#), approximately 15% of the target entities are missing and therefore cannot be predicted by our system. To account for this limitation, we report results (I) with the original label set, and (II) when incorporating the missing entities to the label set (shaded block). Importantly, no retraining is required: the additional labels and verbalizations are simply added to the label set.

With the original ZELDA-derived label set, both our approach and VerbalizED perform substantially

worse on these benchmarks because some gold entities cannot be predicted. Nevertheless, SELECTED consistently improves over the backbone in this setting. After extending the label set, SELECTED achieves 84.0% average accuracy and outperforms ChatEL and ARTER, while still forwarding only 21.5% of spans to the LLM (compared to 100% for ChatEL and 39.6% for ARTER). However, the BLINK cross-encoder remains the strongest system on these datasets. We therefore interpret these results as evidence that the fixed SELECTED configuration transfers beyond the tuning datasets and improves the backbone, without reaching SOTA on these additional benchmarks.

G Additional Performance Insights

In addition to the results presented in Sect. 4, we investigate specific aspects of the model predictions.

G.1 Analysis of “None of the Above”

Although our benchmarks do not contain true NIL gold labels, Table 21 provides a NIL-style analysis of the LLM’s “None of the above” (NOTA) option. We report the general amount of prompted spans, the number of cases where the gold entity is absent from the top-10 candidate set, and the amount of NOTA predictions. We also calculate NOTA precision, NOTA recall, and false NOTA predictions, i.e., cases where NOTA is selected despite the gold entity being present among the candidates.

On average, 13.7% of the LLM decisions result in NOTA. These decisions are largely appropriate, as reflected by the high NOTA precision: In most cases where NOTA is selected, the gold entity is indeed missing from the top-10 candidates (79.5% of NOTA cases). At the same time, NOTA recall remains limited, with only about half of all cases in which the gold entity is absent being identified through a NOTA prediction. False NOTA predictions are comparatively rare: Among prompted cases where the gold entity is available, the LLM selects NOTA in only 3.4%.

Table 22 examines whether replacing LLM “None of the above” (NOTA) predictions with the Dual Encoder (DE) top-1 candidate can improve overall accuracy. We compare the original setting, where NOTA by default counts as an incorrect prediction on all test sets, with a fallback setting in which each NOTA prediction is replaced by the DE top-1 candidate. For all NOTA cases, we additionally report how often the DE top-1 candidate is correct and how often the gold entity is absent from the top-10 candidate set.

Overall, this fallback yields only a marginal improvement of +0.13% on average. This is consistent with the fact that the DE top-1 candidate is correct in only 10.9% of NOTA cases, while the gold entity is absent from the candidate set in 79.4% of them. The fallback can therefore recover only a small number of predictions, although it is somewhat more beneficial on individual datasets, especially CWEB and WNED-WIKI.

We thus keep NOTA as the final prediction rather than applying the DE fallback. An explicit NOTA or NIL option is generally desirable in settings where the correct entity may genuinely be absent

from the candidate set or knowledge base. Note that the datasets used in this work do not include NIL gold labels, so a systematic evaluation of NOTA/NIL behavior is outside the scope of this work. We leave this aspect to future work.

G.2 Qualitative Examples of LLM Predictions

In Table 23 (table is split in two parts), we present qualitative examples illustrating the LLM refinement step. We include successful cases, where the LLM corrects an incorrect dual encoder prediction selected for refinement, as well as cases where correct dual encoder predictions are preserved, either because the LLM selects the top-ranked candidate or because it is not invoked. We also show failure cases, where the LLM overrides a correct prediction or fails to correct an incorrect one. For reference, we also include results from the iterative Verbalized model (Rücker and Akbik, 2025).

We observe several cases where SELECTED corrects errors that VerbalizedED does not, often requiring world knowledge or reduced reliance on surface-form similarity (e.g., “Grexit”, “Red Planet”). However, there are also cases where both methods fail, as well as instances where SELECTED degrades a correct dual encoder prediction (e.g., *Art_theft* → *Smuggling*), which VerbalizedED preserves. These examples highlight that selective LLM refinement can both reliably correct challenging dual encoder errors and, in many cases, preserve strong initial predictions while offering meaningful gains over iterative dual encoder refinement.

G.3 Analysis of LLM-induced Label Changes

In Table 24, we further report the following counts and metrics:

Correct > Correct Number of spans correctly predicted by both the dual encoder and the LLM.

Incorrect > Correct Predictions that are corrected by the LLM refinement.

Correct > Incorrect Initially correct predictions that the LLM changes to incorrect ones.

Incorrect > Incorrect Spans mispredicted by both the dual encoder and the LLM.

Gold in Top 10 Spans where the gold label appears among the Top₁₀ candidates.

Dataset	Spans Prompted		Gold not in Top-10		NOTA		NOTA Prec.		NOTA Rec.		False NOTA	
	Count	%	Count	%	Count	%	Count	%	Count	%	Count	%
AIDA	925/4485	20.62	189/925	20.43	74/925	8.00	70/74	94.59	70/189	37.04	4/736	0.54
TWEEKI	195/856	22.78	65/195	33.33	40/195	20.51	40/40	100.00	40/65	61.54	0/130	0.00
REDDIT-P.	114/703	16.22	19/114	16.67	10/114	8.77	8/10	80.00	8/19	42.11	2/95	2.11
REDDIT-C.	116/637	18.21	21/116	18.10	14/116	12.07	12/14	85.71	12/21	57.14	2/95	2.11
CWEB	3006/11115	27.04	1229/3006	40.88	857/3006	28.51	699/857	81.56	699/1229	56.88	158/1777	8.89
WNED-WIKI	689/6765	10.18	103/689	14.95	45/689	6.53	29/45	64.44	29/103	28.16	16/586	2.73
SLINKS-TAIL	16/899	1.78	5/16	31.25	2/16	12.50	1/2	50.00	1/5	20.00	1/11	9.09
SLINKS-SH.	302/904	33.41	36/302	11.92	31/302	10.26	20/31	64.52	20/36	55.56	11/266	4.14
SLINKS-TOP	325/904	35.95	72/325	22.15	52/325	16.00	49/52	94.23	49/72	68.06	3/253	1.19
Average		20.69		23.30		13.68		79.45		47.39		3.42

Table 21: Analysis of “None of the above” (NOTA) predictions. For each dataset, we report the number and percentage of prompted spans, cases where the gold label is absent from the top-10 candidate set, and the frequency of NOTA predictions. NOTA precision denotes the amount of NOTA predictions for which the gold label was indeed absent, while NOTA recall denotes the proportion of gold-absent cases which are correctly identified as NOTA. False NOTA denotes cases where NOTA is selected despite the gold label being present among the candidates.

Dataset	Accuracy			NOTA Cases			
	w/ NOTA	w/ Fallback	Δ	DE Top-1 Count	Correct %	Gold not in Top-10 Count	%
AIDA	84.50	84.55	+0.04	2/74	2.70	70/74	94.59
TWEEKI	81.31	81.31	+0.00	0/40	0.00	40/40	100.00
REDDIT-P.	90.33	90.47	+0.14	1/10	10.00	8/10	80.00
REDDIT-C.	88.54	88.54	+0.00	0/14	0.00	12/14	85.71
CWEB	71.70	72.13	+0.42	47/857	5.48	697/857	81.33
WNED-WIKI	92.18	92.31	+0.13	9/45	20.00	29/45	64.44
SLINKS-TAIL	98.78	98.89	+0.11	1/2	50.00	1/2	50.00
SLINKS-SH.	76.88	77.21	+0.33	3/31	9.68	20/31	64.52
SLINKS-TOP	77.32	77.32	+0.00	0/52	0.00	49/52	94.23
Average	84.62	84.75	+0.13		10.87		79.43

Table 22: Effect of replacing LLM “None of the above” (NOTA) predictions with the Dual Encoder (DE) top-1 prediction as a fallback. We report accuracy when NOTA is kept as the final prediction (w/ NOTA), accuracy when NOTA predictions are replaced by the DE top-1 candidate (w/ Fallback), and the resulting accuracy difference (Δ). For all NOTA cases, we additionally report how often the DE top-1 prediction is indeed correct and, on the other hand, how often the gold entity is absent from the top-10 candidate set, both as counts and percentages.

Max. possible LLM Acc. / Overall Acc. Upper bounds on LLM and overall accuracy assuming the LLM would always choose correctly if the gold label is among the candidates.

For spans that are not forwarded to the LLM (*certain* cases, i.e., where the dual encoder prediction is accepted directly), we report the number of **Correct** and **Incorrect** predictions and compute **DE Acc.%_{Certain Cases}**.

We observe a substantial gap between the dual encoder’s accuracy on *certain* (accepted) and *uncertain* (later refined) spans. While the model performs consistently well on accepted spans, accuracy on uncertain spans is markedly lower, with particularly large differences on datasets such as CWEB and TWEEKI. This confirms that LLM refinement is particularly beneficial for low-certainty

spans.

Examining the LLM’s interventions, most changes are beneficial: the number of *Incorrect* > *Correct* predictions consistently exceeds *Correct* > *Incorrect*, indicating that the LLM generally improves disambiguation quality. However, many incorrect predictions remain uncorrected (*Incorrect* > *Incorrect*). This is partly because the gold label is missing from the candidate set (see *Gold Not in Top 10*). However, even when the gold label is present, errors still occur.

Overall, these results demonstrate the effectiveness of LLM-based refinement while also highlighting remaining potential for improvement, as current performance remains several percentage points below the theoretical maximum accuracy.

Mention Context	Gold	DE Top- k and LLM Prediction
“...Walters said of O’Donnell , who is on vacation this week ...” (MSNBC)	Rosie_O’Donnell	1. Lawrence_O’Donnell (-20.89) 2. Rosie_O’Donnell (-20.96) 3. The_Rosie_O’Donnell_Show (-21.45) ... LLM: Rosie_O’Donnell SELECTED : LLM corrected DE mistake. <i>Iter. VerbalizED</i> : Also correctly predicted label.
“Upgrade makes aging Mars rovers smarter ...so that they can roll around the Red Planet with less help from humans.” (MSNBC)	Mars	1. Red_giant (-23.70) 2. Great_Red_Spot (-23.78) 3. Red_dwarf (-24.57) ... 10. Mars (-25.16) LLM: Mars SELECTED : LLM corrected DE mistake. <i>Iter. VerbalizED</i> : Incorrect label Red_Planet_Mars (movie).
“ Grexit already? So Greece is going to leave EU too?” (REDDIT-COMMENTS)	Greek_withdrawal_from_the_eurozone	1. Brexit (-21.93) 2. Greek_withdrawal_from_the_eurozone (-22.24) 3. Brexit_withdrawal_agreement (-23.61) ... LLM: Greek_withdrawal_from_the_eurozone SELECTED : LLM corrected DE mistake. <i>Iter. VerbalizED</i> : Incorrectly kept Brexit.
“Alright. ESPN not in HD has sound. HD doesnt. Boo.” (TWEET)	High-definition_television	1. ESPN_Goal_Line_&_Bases_Loaded (-26.88) 2. High-definition_television (-27.18) 3. High-dynamic-range_television (-27.54) ... LLM: High-definition_television SELECTED : LLM corrected DE mistake. <i>Iter. VerbalizED</i> : Incorrect label HD_Radio.

Table 23: Qualitative examples of successful and erroneous LLM refinements. We show the dual encoder (DE) top- k candidates (truncated for readability) and the subsequent LLM predictions. We compare the final predictions of **SELECTED** and the iterative **VerbalizED** variant (Rücker and Akbik, 2025), with green indicating a correct and red indicating an incorrect final prediction. (Table continues on next page.)

Mention Context	Gold	DE Top- k and LLM Prediction
“ Daniel Day Lewis ? I would cast him as the Guild Navigator just to see what he would do to prepare for the role.” (REDDIT-COMMENTS)	Daniel_Day-Lewis	1. Daniel_Day-Lewis (-21.90) 2. Jason_Lewis_(adventurer) (-25.24) 3. Noah_Lewis (-26.17) ... LLM: N/A (LLM not invoked) SELECTED : Keeping correct DE prediction. <i>Iter. VerbalizED</i> : Also not selected for refinement, kept correct prediction.
“International police seek to stymie illegal art traffic ” (AQUAINT)	Art_theft	1. Art_theft (-25.38) 2. Illegal_entry (-25.47) 3. Illegal_per_se (-25.89) 4. Smuggling (-25.94) ... LLM: Smuggling SELECTED : LLM worsened a correct DE prediction. <i>Iter. VerbalizED</i> : Kept the correct label.
“Boise vs oklahoma in the championship would be hilarious. #ncaa” (TWEEKI)	Oklahoma_Sooners_football	1. University_of_Oklahoma (-25.55) 2. Oklahoma_Sooners (-25.76) 3. 2008-09_Oklahoma_Sooners_men’s_basketball_team (-26.10) ... 10. 2018_Oklahoma_Sooners_football_team (-26.30) LLM: Oklahoma_Sooners SELECTED : Both DE and LLM incorrect; correct entity not in candidate set. <i>Iter. VerbalizED</i> : Also kept incorrect label.
“I kept reading Black Panther , but my head visualizing Pink Panther...” (REDDIT-COMMENTS)	Black_Panther_(film)	1. Black_panther (-20.50) 2. Black_Panther_Party (-21.89) 3. Black_Panther_(character) (-22.28) 4. Black_Panther_(film) (-23.45) ... LLM: N/A (LLM not invoked) SELECTED : Keeping DE error (the animal). <i>Iter. VerbalizED</i> : Did not re-predict and kept error.

Table 23: (Continued.) Qualitative examples of successful and erroneous LLM refinements.

Metric	AIDA	TWEEKI	REDDIT- POSTS	REDDIT- COMMENTS	CWEB	WNED- WIKI	SLINKS- TAIL	SLINKS- SHADOW	SLINKS- TOP	Average across datasets
# Spans	4485	856	703	637	11115	6765	899	904	904	100%
<i>Certain</i> Cases: Dual Encoder only – No LLM prediction										
# Spans _{DE} only	3560	661	589	521	8109	6076	883	602	579	79.31% of all spans are <i>certain</i> (DE Only) 90.45% thereof correct 9.55% thereof incorrect 90.45
Correct	3225	593	573	497	6891	5818	878	478	472	
Incorrect	335	68	16	24	1218	258	5	124	107	
DE Acc. % _{Certain} Cases	90.59	89.71	97.28	95.39	84.98	95.75	99.43	79.40	81.52	
<i>Uncertain</i> (Refined) Cases: Dual Encoder + LLM										
# Spans _{DE} + LLM	925	195	114	116	3006	689	16	302	325	20.69% of all spans are refined (DE + LLM) 30.68% thereof Correct > Correct 26.55% thereof Incorrect > Correct 9.60% thereof Correct > Incorrect 33.16% thereof Incorrect > Incorrect 40.28 57.25 23.30% thereof do not have gold in top 10 76.70% thereof have gold in top 10 127.27% thereof incorrect 76.70
Correct > Correct	309	60	38	29	575	248	5	98.5	112.5	
Incorrect > Correct	251.5	45.5	22	35	486	163.5	4	119.5	112	
Correct > Incorrect	62	8	15	17	245	91	3	13.5	10.5	
Incorrect > Incorrect	302	81.5	39	35	1700	186.5	4	70.5	90	
DE Acc. % _{Uncertain} Cases	40.11	34.87	46.49	39.66	27.28	49.20	50.00	37.09	37.85	
LLM Acc. % _{Uncertain} Cases	60.60	54.10	52.63	55.17	35.50	59.72	56.25	72.18	69.08	
Gold Not in Top 10	189	65	19	21	1229	103	5	36	72	
Gold in Top 10	736	130	95	95	1777	586	11	266	253	
Gold in Top 10 > LLM Incorrect	175.5	27.5	35	31	734	194	3	48	28.5	
Max. poss. LLM Acc. %	79.57	66.67	83.33	81.90	59.12	85.05	68.75	88.08	77.85	
Overall (<i>Certain</i> + <i>Refined</i> Cases)										
DE Acc. %	80.18	77.22	89.05	85.24	69.37	91.01	98.55	65.27	65.82	80.19
Overall Acc. %	84.40	81.60	90.04	88.07	71.54	92.08	98.67	76.99	77.04	84.49
Max. poss. Overall Acc. %	88.32	84.70	95.02	92.94	78.16	94.86	99.00	82.30	80.20	88.39

Table 24: Additional performance metrics for our final SELECTED using GPT-5 mini. We show counts and accuracy for both the *Certain* and the *Uncertain* (refined) cases, as well as the *Overall* results. Results may contain decimals as they were averaged from multiple runs. We also include a column which depicts average values across datasets for each metric in percent, relative to a sensible base metric. We obtain the averages across datasets by calculating the percentage of each individual dataset first and then averaging these percentages.