Comparing Multiple Entities at Once for Efficient and Effective Entity Linking

Anonymous ACL submission

Abstract

Popular approach in entity linking is a two-step reranking process where bi-encoder first retrieves top-K candidates and more powerful cross-encoder reranks them. While showing performance boosts beyond entity linking across multiple domains, such as open-domain question answering and dialogue selection, this two-step architecture suffers from cross-encoder's slow runtime and lack of scalability making it impractical in real-time service. To overcome this challenge, we propose comparing multiple entities (CME) framework which compares multiple pre-computed entity embeddings jointly with the mention embedidng over small layers of bi-directional transformer layers. CME shows that it is efficient, 29.3x faster than cross-encoder, and effective *reranker*, 10% improvement over bi-encoder. We additionally show the effect of CME as *second-stage retriever* which surpasses the performance of state-of-the-art retrievers on various K. This in turn shows that use of CME can make cross-encoder reranking system faster (1.6x-2x) while maintaining the performance. Lastly, we conduct extensive comparison of CME to other reranker and retrieval models including newly proposed methods in the ablation study.

1 Introduction

The predominant approach for entity linking (EL) consists of two stages: retrieval and reranking. Typically, a bi-encoder model is used to efficiently retrieve K relevant entities as candidates and then an expressive cross-encoder model effectively reranks the retrieved candidates, where K is significantly smaller than the total number of entities (N), i.e., $K \ll N$ (Wu et al., 2020; Zhang and Stratos, 2021). This framework emerges from the efficiency of bi-encoder in searching over a large space, and the superior performance of a cross-encoder(Nogueira and Cho, 2019), at an expense of higher computational cost.

Cross-encoder captures fine-grained interactions between mention and entity texts by jointly encoding both pieces of text into a single embedding using bi-directional tranformer and eventually produces a single compatibility score. However, this deep interaction between tokens is computationally expensive. In contrast, bi-encoder independently encodes mentions and entities, then evaluates the proximity of an entity to a mention via dot product. This allows for the fast retrieval of the most appropriate entitities, with the help of maximum inner-product search (Malkov and Yashunin, 2018; Johnson et al., 2019; Guo et al., 2020), for a mention by storing large entity encodings in databases.

The typical reranking system with bi- and cross-encoder faces two primary challenges. First, the cross-encoder lacks the scalability to manage a large number of K candidates, rendering the reranking process vulnerable to error propagation. Specifically, this occurs when the bi-encoder fails to capture a true entity within a limited pool of retrieved candidates, leading to the inevitable failure of the whole reranking system. The lack of scalability of cross encoder stems from its requirement to access the raw texts of both mention and entity. Second, when considering serving times of applications that implement entity linking, the runtime of cross-encoder makes it impractical as its speed is two ordrers of magnitude slower than bi-encoder (Humeau et al., 2019).

In this work, we propose a second-stage reranker CME (Comparing Multiple Entities) which compares mention embeding and pre-computed multiple entity embeddings at once through a few layers of bi-directional transformer layers. The idea is to build a model that is scalable under incresing size of K. To do so, we chose the strategy of comparing multiple entities all together and actively utilizing the precomputed entity embeddings as bi-encoder does.

In order to investigate effectiveness of CME, we

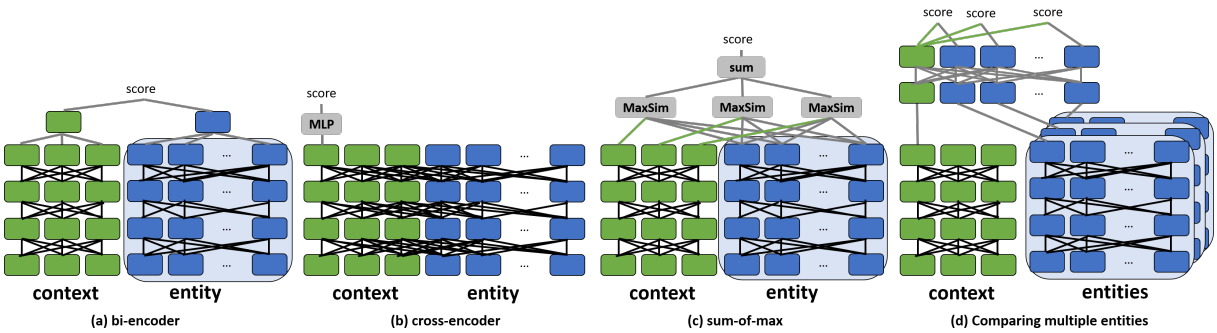

Figure 1: Overview of Models in the Entity Linking Framework: This figure illustrates the existing approaches (a, b, and c) alongside the proposed 'Comparing Multiple Entities' model. Areas shaded in blue indicate embeddings that are available for offline indexing.

also investigated other strategies that can utilize precomputed entity embeddings as well as revisiting previous methods. We find that it is nontrival to perform significantly better than bi-encoder in the context of zero-shot entity linking. On the other hand, the proposed CME framework shows 11% performance improvement at a marginal extra latency, 0.22x of that bi-encoder and 0.03x of cross-encoder's latency (Table 1). In addition, we show that CME can act as a second-stage retriever that can provide higher recall than existing state-of-the-art retrievers (Table 3). We further show that this can reduce overall latency of cross-encdoer reranking system by 0.5-6x by reducing the number of K candidates that cross-encoder examines while performing similarly (Table 4).

The main contribution of the paper is as follows:

- We propose a novel CME framework that is both effective in performance and effcient by utilizing pre-computed entity embeddings (§3). Additionally, we show that CME can benefit from domain transfers of sentence encoder while standard bi-encoder did not (§4.5).
- We provide an in-depth analysis of various ways of utilizing pre-computed entity embeddings and show non-trivial to utilize pre-computed entity embeddings (§4.5).
- We show that CME can act as a second-stage retriever which can increase the recall of the retriever at a marginal cost (0.2x runtime of bi-encoder) and thus improve the cross-encoder reranking performance as well.
- By presenting that CME can act as both reranker and enhanced retriever, we provide flexible options to users depending on their need. For higher accuracy, one can utilize CME as second-stage retriever, while for scenarios with a restricted time budget, CME can be utilized as an efficient reranker.

2 Background and Related Work

2.1 Task Description

Entity linking Entity linking (EL), also known as named entity disambiguation, refers to the process of associating named entity mentions with unique entities in a knowledge base, such as, Wikipedia (Sevgili et al., 2022). Formally, given a mention m in a context, an entity linking system predicts a unique entity e_i from an entity set $\mathcal{E} = \{e_i\}_{i=1,\ldots,N}$ that matches the identity of mention m. It is assumed that there exists an entity $e_i \in \mathcal{E}$ where each mention can be correctly mapped. The size of the entity set, N, is typically large. The entity linking system is required to both *efficiently* search for relevant entities in the large entity set and *effectively* identify the correct entities.

Zero-shot entity linking We focus on a general and challenging setting of the task, *zero-shot entity linking (zeshel)* (Logeswaran et al., 2019). In zeshel, training and testing sets are from separate domains that are characterized by non-overlapping entity sets $\mathcal{E}_{\text{train}}$ and $\mathcal{E}_{\text{test}}$ where $\mathcal{E}_{\text{train}} \cap \mathcal{E}_{\text{test}} = \emptyset$. The entities are defined by short textual descriptions. Zeshel systems thus need to comprehend the semantics of *unseen* entity descriptions in order to match entities with the contextualized mention representations.

2.2 Existing Methods

Retrieve and rerank Entity linking systems commonly comprises two stages: efficient retrieval and effective reranking. Given a mention m in a context sentence $ctxt$, a fast retriever will score

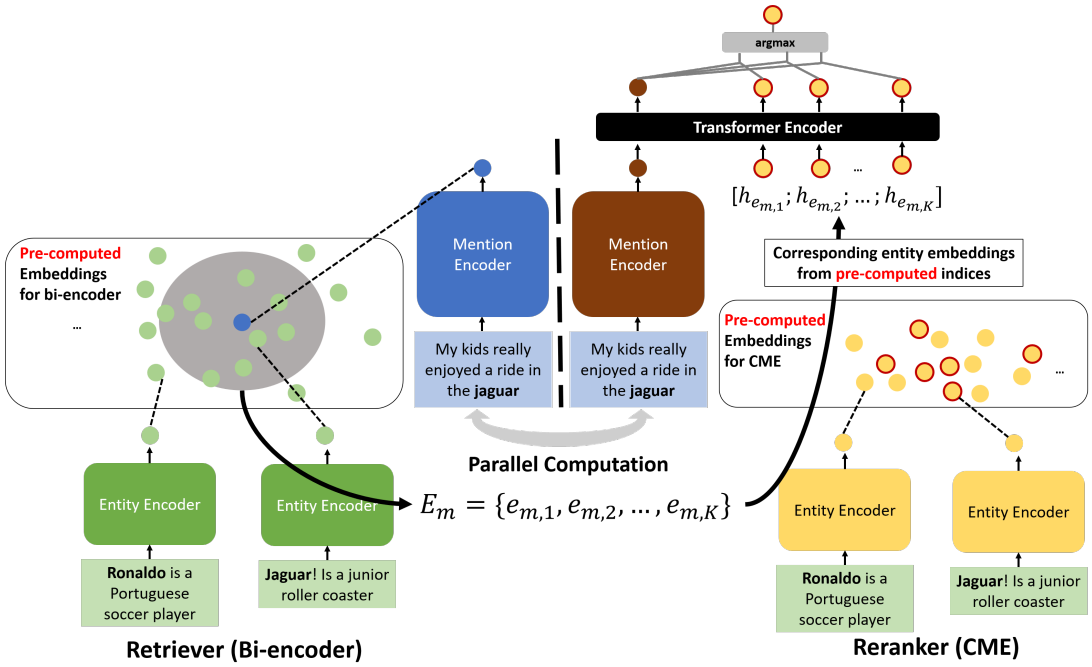

Figure 2: Overview the proposed CME system that compares multiple entities at once for efficient and effective entity linking.

the mention m with each entity $e \in \mathcal{E}$. Although the retriever is designed to be fast, its top-1 accuracy tends to be suboptimal. Practitioners therefore identify a candidate entity set

$$E_m = \{e_{m,1}, e_{m,2}, \ldots, e_{m,K}\} \subseteq \mathcal{E}, \quad (1)$$

whose elements are K most relevant entities in entity set $\mathcal{E}$ according to the retriever, to be reranked.

A reranker $s_\theta(m, ctxt, e_{m,j})$ is a model learned to assign fine-grained matching scores between a mention m and each candidate $e_{m,j}$ from the relatively small set of entities E_m predicted by the retriever. It is an expressive model that is generally slower but more accurate than the retriever. The entity with the highest score,

$$\hat{e}_m = \arg\max_{e_{m,j} \in E_m} s_\theta(m, ctxt, e_{m,j}), \quad (2)$$

is finally output as the entity where mention m should be linked.

Score functions Given a mention in a context and an entity description, their matching score can be computed by diverse functions. The retrieval step requires fast functions while the reranking step focuses more on accuracy. Logeswaran et al. (2019) first retrieve candidate entities for a mention using the bag-of-words *BM25* retrieval function and then use an expressive *cross-encoder* model to score the retrieved candidates. The cross-encoder is a transformer encoder model that inputs the mention context tokens and entity description tokens simultaneously to produce a matching score.

Gillick et al. (2019) concurrently uses a *bi-encoder* architecture to embed mention context tokens and entity description tokens separately and compute the dot point of the embeddings. The mention embedding only needs to be calculated once when compared to all the entities and vice versa. The bi-encoder allows fast *maximum inner-product search (MIPS)* but is less expressive than the cross-encoder (Figure 1(a-b)). BLINK (Wu et al., 2020) combines the advantages of both types of scores functions, first using a bi-encoder for retrieving relevant entities and then using a cross-encoder to rerank them.

Humeau et al. (2019) proposes a *poly-encoder* that learns global rather than token level self-attention features to be faster than the cross-encoder and more accurate than the bi-encoder. Zhang and Stratos (2021) proposes negative sampling strategies to improve the bi-encoder training and proposes a *sum-of-max* late fusion model as a reranker that is both faster and more accurate than the poly-encoder. We adopt their sampling strategies and explore a previously overlooked opportunity to achieve a reranker that is more accurate than the sum-of-max model while remaining much

more efficient than the cross-encoder. In particular, we proposes to process multiple candidate entities (CME) at once to rerank them.

Other entity linking systems We also compare our method with the following methods for completeness. Barba et al. (2022) revisits entity linking as a text extraction task. De Cao et al. (2020) is an autoregressive model which generates the title of the corresponding entity with constrained beam search. Our proposed CME is more accurate than both. Xu et al. (2023) proposes a reranker that consists of a cross-encoder reader module and a selection module that extracts candidate tokens based on embeddings from the reading module. While this method is more accurate than the cross-encoder, it is even slower than the cross-encoder, which already has almost 30× more extra latency than our CME.

3 Proposed Method

We propose an entity linking system consisting of a bi-encoder retriever and a CME reranker.

3.1 Model Architectures

Sentence Encoder After identifying the candidate entity set E_m, our model begins from obtaining embeddings for the context sentence tokens x_m, which contains the mention m, and for the entity description sentence tokens $\mathrm{x}_{e_{m,j}}$. Similar to bi-encoder, embeddings are derived for both the context sentence ($\mathbf{h}_m^{sent}$) encompassing the mention and the entity description sentence ($\mathbf{h}_{e_{m,j}}^{sent}$) using respective context and entity encoders, $\texttt{Enc}_{ctxt}$ and $\texttt{Enc}_{ent}$. The embeddings are calculated as follows:

$$\mathbf{h}_m^{sent} = \texttt{aggregator}\left(\texttt{Enc}_{ctxt}\left(\mathrm{x}_m\right)\right), \quad (3)$$

$$\mathbf{h}_{e_{m,j}}^{sent} = \texttt{aggregator}\left(\texttt{Enc}_{ent}\left(\mathrm{x}_{e_{m,j}}\right)\right) \quad (4)$$

Following the approach in Wu et al. (2020), the embeddings for both the context sentence and the entity description are limited to a maximum of 128 word-piece tokens. This includes the use of special tokens such as [CLS], [SEP], and custom tokens that denote the locations of mentions and entities[1]. The aggregator function extracts the [CLS] token embedding from the last layer of the transformer encoder.

[1]These include [mention_start], [mention_end], and [ENT] tokens

Comparing Multiple Entities (CME)

$$\left[\mathbf{h}_m^{CME}; \mathbf{h}_{e_{m,1}}^{CME}; \ldots; \mathbf{h}_{e_{m,K}}^{CME}\right] = \texttt{Enc}_{CME}(\left[\mathbf{h}_m^{sent}; \mathbf{h}_{e_{m,1}}^{sent}; \ldots; \mathbf{h}_{e_{m,K}}^{sent}\right]) \quad (5)$$

Analogous to the bi-encoder retrieval, the final prediction e_m^* is determined by computing the dot product score between the mention and entity embeddings:

$$e_m^* = \arg\max_{e_{m,j} \in E_m} \mathbf{h}_m^{CME} \cdot \left(\mathbf{h}_{e_{m,j}}^{CME}\right)^\top \quad (6)$$

3.2 Training

Optimization Given an entity set for training $\tilde{E}_m = \{\tilde{e}_{m,1}, \tilde{e}_{m,2}, \ldots, \tilde{e}_{m,K}\}$, score function s_θ that assigns a score to each entity in the set. The score function outputs a probability distribution over the entities, representing the likelihood of each entity being the correct one. The loss function for our model is a combination of multi-class cross-entropy, which is regularized by Kullback-Leibler (KL) divergence between the reranker's scores and the retriever's scores. The loss function is formulated as follows:

$$\mathcal{L}(m, E_m) = -\lambda_1 \sum_{i=1}^{K} y_i \log(p_i) + \lambda_2 \sum_{i=1}^{K} p_i \log\left(\frac{p_i}{q_i}\right) \quad (7)$$

where y_i represents the ground truth label for each entity $e_{m,i}$, p_i is the predicted probability for entity $e_{m,i}$ from the score function s_θ, q_i is the probability of the same entity from the retriever's distribution, and λ_1 and λ_2 are coefficients forming a convex combination of the two losses.

Negative Sampling In contrast to previous studies (Wu et al. (2020); Xu et al. (2023)), which train a reranker using a fixed set of top-k candidates from the retriever, our approach adopts a technique similar to hard negative sampling for training retriever (Zhang and Stratos (2021)). Some negative entities are sampled based on the retriever's scoring for mention-entity pairs:

$\forall j \in \{1, \ldots, K\} \setminus \{\text{gold index}\}$,

$$e_{m,j} \sim \frac{\exp(s_{\text{retriever}}(m, \tilde{e}_{m,j}))}{\sum_{k=1, k \neq \text{gold index}}^{K} \exp(s_{\text{retriever}}(m, \tilde{e}_{m,k}))} \quad (8)$$

To provide competitive and diverse negatives for the reranker, p% of the negatives are fixed as the top-k negatives, while the others are sampled following the score distribution.

Sentence Encoder Initialization The initial starting point of the sentence encoder can significantly impact performance. The transformer encoder may be initialized with BERT (Devlin et al. (2018)) or with other BERT-based fine-tuned models. These include models fine-tuned on the Zeshel dataset (Yadav et al. (2022)) or other datasets (Gao and Callan (2022) ; Wu et al. (2020)).

3.3 Inference

In contrast to cross-encoders where entity embeddings are not saved in advance (Logeswaran et al., 2019; Wu et al., 2020), our model can compute and index entity sentence embeddings as it is processed independently to context information (see Figure 2). During inference, it is only required to compute the context sentence embedding, obtain embeddings for each mention and entity via the CME module, and then identify the candidate with the highest scores through an inner-product calculation.

4 Experiments

4.1 Dataset

We use the zero-shot entity linking dataset (zeshel) created by Logeswaran et al. (2019) from Wikia[2]. The dataset consists of 16 domains, divided into 8 for training, 4 for validation and testing. It includes 49,275 labeled mentions in training and 10,000 unseen mentions each in validation and test sets. This dataset challenges linking entity mentions to unseen entities based on their descriptions, emphasizing zero-shot inference. The entity domain, also called "world", varying from 10K to 100K entities, is unique to each domain, testing the model's ability to generalize to new entities, with detailed statistics presented in Table 7 in appendix.

4.2 Training Details

All CME models are trained for 5 epochs using the AdamW optimizer. The learning rate is tuned over the set {1e-5, 2e-5, 5e-5}, the ratio of fixed negatives, denoted as p, is chosen from the set {0, 50}. The best model is selected based on its performance on the development set. The number of negatives is 63 and sampled from top-1024 candidates retrieved from bi-encoder. The effective batch size is 8, with batch size 2 and gradient accumulation steps 4. The training process takes $\tilde{4}$ hours on a single NVIDIA A100 GPU. The loss coefficients are set as $\lambda_1 = 0.2$ and $\lambda_2 = 0.8$. The number of layers and multi-head attention for CME module is two and four, respectively.

4.3 Evaluation Metric

In the evaluation of a retriever, we measure its effectiveness using recall@k, which represents the percentage of the gold entity found within the top-k retrieved entities. For evaluating a reranker, the primary metric is top-1 accuracy, since there is only one correct entity corresponding to each mention. The accuracy is categorized into unnormalized and normalized accuracy. Unnormalized accuracy is computed across all mention instances, while normalized accuracy is calculated for those mention sets that are successfully retrieved by the retriever. Our model was tested on each domain in validation and test sets. Performance metrics over each set were then calculated using either macro- or micro-averaging across different domains.

4.4 Results

4.4.1 CME as a reranker

We evaluated our model's reranking performance using the top-64 candidates generated by bi-encoder retrievers (Yadav et al. (2022)) and BM25 (Logeswaran et al. (2019)).

Baselines We conducted a comparative analysis of our model with several reranking methods. Cross-encoder (Logeswaran et al. (2019); Wu et al. (2020); Yadav et al. (2022)) is recognized as one of the most powerful baselines but is resource-intensive, as it jointly encodes context and entity tokens. The sum-of-max method (Zhang and Stratos (2021)), calculates the relevance score utilizing the entirety of mention and entity tokens (Khattab and Zaharia (2020)). Poly-encoder (Humeau et al. (2019)) is the model that utilizes attention mechanism between entity embedding and multiple context representation. Bi-encoder, a simpler approach, determines relevance scores by the dot product of individual mention and entity embeddings. ExtEND (Barba et al. (2022)) utilizes self-attention between mentions and entities, while making use of longformer Beltagy et al. (2020) to carry out self-attention between all tokens from each mention and

[2]Now known as Fandom: https://www.fandom.com

Table 1: Macro-Averaged accuracy (%), reranking latency, and index sizes from candidates from Bi-encoder (Yadav et al. (2022)) To elucidate the relative latency associated with the reranking process, we express the reranking latency as extra relative latency to the bi-encoder runtime (36.3 ms)under the same experiment setup. The best result is denoted in **bold** and the second-best result is underlined † is reported at Zhang and Stratos (2021), which is evaluated over candidate set with recall@64 91.93% for validation set and 83.48% for test set. *base* means BERT-base based model and *large* means BERT-large based model.

Method	Unnormalized acc. Valid	Unnormalized acc. Test	Normalized acc. Valid	Normalized acc. Test	Extra latency	Relative index size
Cross-encoder	**66.86**	**65.76**	**72.64**	**74.46**	+6.46x	-
Sum-of-max† (Zhang and Stratos (2021))	59.35	57.04	65.38	65.24	+0.02x	10.9x
Sum-of-max† (w/o indexing)	59.35	57.04	65.38	65.24	+4.70x	-
Poly-encoder 16† (Zhang and Stratos (2021))	55.90	54.87	61.56	62.65	+0.03x	1x
Poly-encoder 128† (Zhang and Stratos (2021))	55.98	55.17	61.67	62.95	+0.03x	1x
Bi-encoder	55.45	52.95	59.71	60.32	-	1x
Comparing Multiple Entities *(large)*	60.65	59.48	65.99	67.15	+0.28x	1.4x
Comparing Multiple Entities *(base)*	60.01	58.98	65.06	66.69	+0.22x	1.0x

Table 2: Test Normalized accuracy of CME model over retrieved candidates from BM25.

Method	Forgotten Realms	Lego	Star Trek	Yugioh	**Macro Acc.**	**Micro Acc.**
Low Latency Methods						
ExtEnD (Barba et al. (2022))	79.62	65.20	73.21	60.01	69.51	68.57
GENRE (De Cao et al. (2020))	55.20	42.71	55.76	34.68	47.09	47.06
Comparing Multiple Entities (*large*; Ours)	84.30	70.43	75.08	62.54	73.08	71.99
Comparing Multiple Entities (*base*; Ours)	83.20	70.63	75.75	64.83	73.35	72.41
High Latency Methods						
Cross-encoder (Wu et al. (2020))	87.20	75.26	79.61	69.56	77.90	77.07
ReS (Xu et al. (2023))	88.10	78.44	81.69	75.84	81.02	80.40

candidate. GENRE (De Cao et al. (2020)) operates entity linking using a language model, generating entity titles corresponding to each mention. Finally, ReS (Xu et al. (2023)) uses a reading module and a selecting module, where the reading module functions as a cross-encoder and the selecting module predicts the entity on top of concatenated cross-encoder embeddings.

For candidates from bi-encoder we evaluated the CME against various baseline models, including the cross-encoder, sum-of-max, and bi-encoder. The bi-encoder checkpoint from Yadav et al. (2022) performed macro-averaged recall@64 92.04% on the validation set and 86.8% on the test set. Also, we've loaded cross-encoder from the same checkpoint. CME exhibited a significant improvement in unnormalized accuracy compared to the bi-encoder. This increase attributed to the integration of sentence encoder embeddings enhanced by a self-attention mechanism, leading to an accuracy increase of approximately 5-6 points.
Additionally, our model surpassed the performance of multi-vector models such as the sum-of-max and poly-encoder. In contrast to the sum-of-max, which either requires an index size 7-10 times larger or operates 16-21 times slower without indexing, CME demonstrated superior efficiency and effectiveness in reranking tasks while saving computation and memory resources. Although the poly-encoder showed lower latency compared to our model, its performance was relatively inferior, almost comparable with the bi-encoder.
When compared to the cross-encoder, CME was found to be less accurate, with a difference of about 6 points in unnormalized accuracy. However, it offered a considerable reduction in latency. This decrease in time complexity renders CME a more practical choice, especially considering it only necessitates 3-4% of the time required by the cross-encoder to run. The accuracy, additional latency, and index size are reported in Table 1.

For candidates from BM25 we also conduct reranking on candidates from BM25, which was released by Logeswaran et al. (2019). We reported accuracy on each world of test set in Table 2. In the comparison between CME and other models such as ExtEND and GENRE, it has been observed that CME exhibits superior performance. Besides this

Table 3: Recall@k on the Zeshel test set. The best result is denoted in **bold** and the second-best result is underlined. Numbers in parentheses (e.g., CME(128)) indicate the number of candidates from which CME selects, initially retrieved by the bi-encoder. * is re-implemented bi-encoder from Yadav et al. (2022) which is used for generating candidates for CME

Retriever	R@1	R@4	R@8	R@16	R@32	R@64
BM25	-	-	-	-	-	69.26
BLINK	-	-	-	-	-	82.06
Bi-encoder*	51.41	70.49	76.34	80.44	83.81	86.8
SOM	-	-	-	-	-	89.62
MuVER	43.49	68.78	75.87	77.65	85.86	89.52
Arbo-EL	50.31	68.34	74.26	78.40	82.02	85.11
MVD	52.51	73.43	79.74	84.35	88.17	**91.55**
CME (128)	**59.18**	**76.70**	**81.35**	85.25	87.74	89.52
CME (256)	59.13	76.60	81.06	**85.35**	88.13	90.40
CME (512)	59.01	76.48	81.00	85.07	**88.25**	90.75

advantage, our model comes with another advantage of a reduced computational burden. The efficiency of CME can be attributed to its utilization of a singular vector embedding strategy, which stands in contrast to ExtEND and GENRE, where there is a need to process the entire tokens for reranking.
In comparing CME with the Cross-encoder and the Read-and-Select (ReS), it is evident that although CME may not achieve the same level of performance as these models, it significantly benefits from lower computational requirements. The Cross-encoder and ReS models necessitate a more intensive computational process, as they concurrently encode each token from both the mention and the entity. In contrast, CME operates under a comparatively lighter computational burden.

4.4.2 CME as a Second-Stage Retriever

CME can be also used as a second-stage retriever, which can enhance recall over base retriever.

Baselines Our baseline for comparison is the bi-encoder (Yadav et al. (2022)), upon which CME conduct the retrieval process. BM25 (Logeswaran et al. (2019)) is a variant of the traditional TF-IDF designed to measure the similarity between mentions and entity descriptions. BLINK is a standard bi-encoder fine-tuned with the Zeshel dataset, as presented by Wu et al. (2020). SOM (Zhang and Stratos (2021)) can also function as a retriever as well as reranker. ArboEL, which depends on graph structures to derive fine-grained entity representations (Agarwal et al. (2022)). Finally, we consider MVD as state-of-the-art, a multi-view entity retrieval system that necessitates the use of multiple embeddings for representing entities (Liu et al. (2023)).

Results In our two-stage retrieval framework, the process begins with the bi-encoder retrieving a larger candidate set from the entire entity set. Following this, CME takes over, narrowing down these candidates to 64 or fewer. This approach is essential for effective retrieval: from the result in Table 3. while the bi-encoder has a recall@64 of 87.95% on the test set, the subsequent use of CME in this two-stage process leads to a significant enhancement. Specifically, applying CME as a retriever results in a 2.8-point increase in recall@64, showing performance that is on par with MVD. Notably, CME outperforms other state-of-the-art (SOTA) models in recall@k for all k except 64. Results show that utilizing the model as a second-stage retriever, as opposed to merely a CME reranker, thereby providing a more refined selection of candidates for further reranking.
To assess the effectiveness of candidates generated by CME, we analyzed the cross-encoder's performance with varying numbers of candidates: 8, 16, and 64 from the bi-encoder, and 16 and 8 candidates from CME, derived from the initial 64 candidates from the bi-encoder. As demonstrated in Table 4, the cross-encoder showed optimal performance with 16 candidates from CME. This finding underscores CME's role in not only improving end-to-end accuracy by filtering out less relevant candidates but also in its ability to provide candidates that are more accurately identified by the cross-encoder. Moreover, this method significantly reduces time complexity, as the cross-encoder processes a smaller candidate set.

4.5 Ablation Study

Initialization of CME We conduct a systematic investigation of different initialization methods for our sentence encoder, as illustrated in Table 5. Initially, we employed the pre-trained BERT model (Devlin et al., 2018) as a baseline. To test domain adaptation capability of CME, we experimented with other pre-trained models, such as utilizing a bi-encoder fine-tuned for the Zeshel dataset (Yadav et al., 2022) and Cocondenser (Guo et al., 2020), a BERT-base-cased model further adapted for the MS-MARCO dataset. Additionally, for the BERT-large variant, we incorporated a bi-encoder fine-tuned on the Wikipedia entity linking dataset (Wu et al., 2020). The results, as shown in Table 5, indicate that the fine-tuning of pre-trained models

Table 4: Unnormalized accuracy of the Cross-encoder across various retriever configurations is presented. Instances where the Cross-encoder showed superior performance with candidates generated by CME compared to those from the Bi-encoder are underlined. The top-performing scenarios in each category are highlighted in **bold**. We measure extra latency to the bi-encoder runtime.

# of Candidates		Recall	Unnormalized accuracy					Extra
Bi-encoder	CME		Forgotten Realms	Lego	Star Trek	Yugioh	Macro	latency
64	-	87.91	80.83	**67.47**	64.18	50.56	65.76	234ms
16	-	81.52	80.17	66.14	63.69	49.64	64.91	100ms
64	16	84.96	**81.00**	67.15	64.70	**51.04**	**65.97**	146ms
8	-	77.71	78.92	65.14	62.76	48.64	63.87	67ms
64	8	81.25	80.58	66.06	**64.75**	50.50	65.48	113ms

on domain-specific datasets significantly improves their performance. Also, freezing sentence encoder did not yield high performance, which yields further fine-tuning is required for domain adaptation. This highlights the effective domain adaptation capabilities of language models when fine-tuned for new tasks, evidenced by the marked increase in unnormalized accuracy (U.Acc.) on both validation and test sets across different initialization methods. The Cocondenser model, in particular, demonstrated the highest gains in performance, underscoring the benefits of targeted fine-tuning for specific language understanding tasks.

Effect of CME Module We also examine the impact of the scale of BERT on the performance of our models, as detailed in Table 6. We initialized both the bi-encoder and the CME model with two variants: Cocondenser and BLINK-Wiki. Our findings reveal that simply training the bi-encoder, irrespective of the scale, did not lead to performance improvements. This outcome underscores the importance of the CME module in effectively adapting these models to the domain-specific tasks.

Furthermore, an intriguing observation was made when the bi-encoder was scaled up to BERT-large. Contrary to expectations, this upscaling resulted in diminished performance. This suggests that increasing the parameter count of BERT does not inherently guarantee enhanced performance. On the other hand, the incorporation of the CME module with both BERT-base and BLINK-Wiki significantly improved performance, as evidenced by the higher unnormalized accuracy (U.Acc.) on both validation and test sets. This indicates the crucial role of the CME module in leveraging the capabilities of BERT for effective domain adaptation, transcending the mere increase in model size.

Table 5: Comparison of unnormalized accuracy (U.Acc.) over different initialization methods.

Sentence Encoder	U.Acc. Valid	Test
BERT-base		
BERT (Devlin et al. (2018))	56.15	55.34
Bi-encoder (Yadav et al. (2022))	58.04	56.20
Bi-encoder (w/o BERT fine-tuning)	56.35	51.32
Cocondenser (Guo et al. (2020))	**60.01**	**58.98**
BERT-large		
BERT (Devlin et al. (2018))	58.61	58.07
BLINK-Wiki (Wu et al. (2020))	**60.65**	**59.48**

Table 6: Unnormalized accuracy (U.Acc.) of bi-encoder and CME for different sentence encoder

Model	Sentence Encoder	U.Acc. Valid	Test
Bi-encoder	Cocondenser	49.32	44.01
	BLINK-Wiki	49.12	46.10
CME	BERT-base	60.01	58.98
	BLINK-Wiki	60.65	59.48

5 Conclusion

Our study introduces the Comparing Multiple Entities (CME) model which can serve as both a retriever and a reranker for entity linking. By leveraging pre-computed entity embeddings, CME offers scalability under an increasing size of candidates. It acts as both an efficient reranker and an enhanced retriever, improving recall and reranking performance with minimal computational overhead. This dual capability provides flexibility based on user needs: higher accuracy through second-stage retrieval or time-efficient reranking. CME's efficiency in processing, with notable improvements in latency and computational burden, will make it a practical choice for real-world applications. The ablation studies further reveal the impact of different sentence encoder initializations on the model's performance. Overall, the findings underscore CME's capability to improve retrieval accuracy and efficiency in entity-linking tasks, offering valuable insights for future model optimizations.

References

Dhruv Agarwal, Rico Angell, Nicholas Monath, and Andrew McCallum. 2022. Entity linking via explicit mention-mention coreference modeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4644–4658.

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

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2020. Autoregressive entity retrieval. *arXiv preprint arXiv:2010.00904*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Luyu Gao and Jamie Callan. 2022. Unsupervised corpus aware language model pre-training for dense passage retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2843–2853.

Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning dense representations for entity retrieval. *arXiv preprint arXiv:1909.10506*.

Ruiqi Guo, Philip Sun, Erik Lindgren, Quan Geng, David Simcha, Felix Chern, and Sanjiv Kumar. 2020. Accelerating large-scale inference with anisotropic vector quantization. In *International Conference on Machine Learning*, pages 3887–3896. PMLR.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2019. Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring. *arXiv preprint arXiv:1905.01969*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Yi Liu, Yuan Tian, Jianxun Lian, Xinlong Wang, Yanan Cao, Fang Fang, Wen Zhang, Haizhen Huang, Denvy Deng, and Qi Zhang. 2023. Towards better entity linking with multi-view enhanced distillation. *arXiv preprint arXiv:2305.17371*.

Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. 2019. Zero-shot entity linking by reading entity descriptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3449–3460.

Yu A Malkov and Dmitry A Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):824–836.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

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

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.

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.

Nishant Yadav, Nicholas Monath, Rico Angell, Manzil Zaheer, and Andrew McCallum. 2022. Efficient nearest neighbor search for cross-encoder models using matrix factorization. *arXiv preprint arXiv:2210.12579*.

Wenzheng Zhang and Karl Stratos. 2021. Understanding hard negatives in noise contrastive estimation. *arXiv preprint arXiv:2104.06245*.

A Statistics of Zero-shot Entity Linking Dataset

Table 7: Staistics of Zero-shot Entity Linking (Zeshel) dataset.

Domain	**# of Mentions**	**# of Entities**
American Football	3898	31929
Doctor Who	8334	40281
Fallout	3286	16992
Final Fantasy	6041	14044
Military	13063	104520
Pro Wrestling	1392	10133
Star Wars	11824	87056
World of Warcraft	1437	27677
Training	**49275**	**332632**
Coronation Street	1464	17809
Mupptes	2028	21344
Ice Hockey	2233	28684
Elder Scrolls	4275	21712
Validation	**10000**	**89549**
Forgotten Realms	1200	15603
Lego	1199	10076
Star Trek	4227	34430
Yugioh	3374	10031
Test	**10000**	**70140**