Entity Linking via Explicit Mention-Mention Coreference Modeling

Anonymous ACL submission

Abstract

Learning representations of entity mentions is a core component of modern entity linking systems for both candidate generation and making linking predictions. In this paper, we present and empirically analyze a novel training approach for learning mention and entity representations that is based on building minimum spanning arborescences (i.e., directed spanning trees) over mentions and entities across documents to explicitly model mention coreference relationships. We demonstrate the efficacy of our approach by showing significant improvements in both candidate generation recall and linking accuracy on the Zero-Shot Entity Linking dataset and MedMentions, the largest publicly available biomedical dataset. In addition, we show that our improvements in candidate generation yield higher quality re-ranking models downstream, setting a new SOTA result in linking accuracy on MedMentions. We further demonstrate that our improved mention representations are effective for the discovery of new entities via cross-document coreference.

1 Introduction

Natural language corpora, such as biomedical research papers (Leaman and Lu, 2016), news articles (Milne and Witten, 2008; Hoffart et al., 2011), and, more generally, web page text (Gabrilovich et al., 2013; Lazic et al., 2015a), often contain ambiguous mentions of entities. Resolving this ambiguity requires mentions to either be linked to a knowledge base (KB) of entities or discovered as a new KB concept if no suitable entry exists. Grounded entity mentions can be beneficial for tasks such as question-answering (Das et al., 2019), semantic search (Leaman and Lu, 2016), recommendation ranking (Noia et al., 2016), and KB construction (Ling et al., 2015). The task is made particularly challenging in zero-shot settings, where not every entity has labeled training data (Lin et al., 2017; Logeswaran et al., 2019). In such settings, a common approach is to make use of entity descriptions, types, and aliases to form entity representations, which are then used for making predictions.

Learned vector representations of entity mentions are an integral part of modern linking systems (Gillick et al., 2019; Wu et al., 2020, inter alia). These representations are used for (a) retrieving a short-list of entity candidates for a mention for use with a re-ranker (Wu et al., 2020), (b) making linking predictions directly (Zhang et al., 2021; Liu et al., 2020; Sung et al., 2020), and (c) performing coreference by clustering mentions to form entities (Logan IV et al., 2020).

In this work, we present a new objective and training procedure for learning mention and entity representations that explicitly models mention coreference relationships. Our proposed method uses a supervised clustering training objective based on forming a directed minimum spanning tree, or *arborescence*, over mentions and entities. We hypothesize that such coreference links provide a useful inductive bias because the two tasks are inherently related (Angell et al., 2021; FitzGerald et al., 2021). We thoroughly analyze the performance of the proposed training procedure in each of the aforementioned use cases on MedMentions (Mohan and Li, 2019) and ZeShEL (Logeswaran et al., 2019), two challenging datasets that require zero-shot generalization at inference.

Retrieving Candidates. We illustrate that our approach yields mention and entity representations useful for candidate retrieval. We show improvements over baselines that use similarly parameterized models, achieving gains of at least 7.94 and 0.93 points in *recall*@64 over two standard dual-encoder training procedures on MedMentions and ZeShEL, respectively. We also consider the linking capacity of our learned embeddings without re-ranking and find that their performance (i.e *recall*@1) indeed improves upon our baselines. Our best performing models show gains of 13.61 & 15.46 points in linking accuracy on MedMentions and 12.06 & 1.52 points on ZeShEL.

Linking Predictions. We further consider the improvement in downstream training of full cross-attention re-ranker models using higher quality candidates generated by our approach. We show consistent gains in linking accuracy on MedMentions, setting a new state-of-the-art with a 1.63 point gain over the previous best model. We also note that our proposed approach shows mixed results on ZeShEL, with one variant outperforming all compared models by at least 1.19 points, while the other two underperform the baselines. We analyze this behavior in a later section and discuss the characteristics of the data distribution sufficient to make our approach effective.

Cross-Document Coreference. Finally, we illustrate that the learned representations can be used to perform coreference of mentions *across* documents. This indicates that they could be used to discover entities in settings where there is limited or no existing knowledge base of entities.

2 Arborescence-based Training for Mention & Entity Representations

In this section, we describe our proposed approach for constructing training objectives for dual-encoders that model mention coreference relationships.

2.1 Problem Definition

Each document d of a corpus $\mathcal{D}$ contains a set of entity mention spans $\mathcal{M}^d = \{m_1^d, m_2^d, \ldots, m_N^d\}$. All mentions in the corpus are given by $\mathcal{M} = \bigcup_{d \in \mathcal{D}} \mathcal{M}^d$. Following (Logeswaran et al., 2019; Angell et al., 2021), we assume that these mentions are pre-identified spans of text.

Entity Linking Formally, we define the task of entity linking as follows: given a knowledge base of entities $\mathcal{E}$ and a set of mentions $\mathcal{M}$, predict an entity $e_i^d \in \mathcal{E}$ for each mention m_i^d. We use $e^{\star}{}_i^d$ to refer to the ground truth entity label for m_i^d.

Zero-Shot Linking The zero-shot task refers to the setting where there are entities in the knowledge base that do not have any labeled training data. Linking decisions must instead rely on provided information for entities, such as a descriptions, aliases, and/or entity types.

Coreference We also consider a setting in which the KB of entities is not known in advance and entities must be *discovered*. For this task, we map every entity mention m_i^d to a cluster and assign a coreference label $c_i^d \in \mathcal{C}$ that is independent of the entity labels in the KB.

2.2 Coreference-based Similarity

In order to jointly train both the mention and entity encoders, we define a similarity measure and an analogous procedure for sampling positive training examples that intersperses the selection of coreferent mentions and gold entities based on a single-linkage structure formed by the representations generated by the model snapshot. We construct k-nearest neighbor graphs over coreferent mention and entity clusters, followed by the application of a pruning algorithm to generate arborescence (directed MST) structures rooted at entity nodes. In this way, the resultant edges represent pairs of positive examples used for training.

Graph-based Dissimilarity Let G be a graph with nodes $V = \mathcal{M} \cup \mathcal{E}$ and directed edges $E \subset V \times V$. Each edge (x, y) of the graph has an associated weight $w_{x,y}$. We define a dissimilarity function f between two nodes $u, v \in V$ to be the weight of the minimax path between the nodes, i.e.

$$f(u,v) = \begin{cases} \min\limits_{p \in u \rightsquigarrow v} \max\limits_{(x,y) \in p} w_{x,y}, & \text{if connected}(u,v) \\ \infty & \text{otherwise} \end{cases} \tag{1}$$

where $\text{connected}(u, v)$ is true if there exists a directed path from node u to v in G, and $u \rightsquigarrow v$ is the set of all paths between u and v. In words, the dissimilarity between u and v is the minimum of the largest edge weights in all paths between the two nodes, and this is often referred to as the "bottleneck edge". This measure has the property of emitting low dissimilarities between nodes even when the direct edge weight $w_{u,v}$ is high by connecting them through a chain of low-weight edges, providing an inductive bias well-suited for coreference, i.e. not all pairs of points in a cluster are nearby (Figure 1). This inductive bias is not achieved if we sum edge weights and simply find the minimum path.

Edge Weights With this definition of dissimilarity, we now define how edge weights are calculated. We use two models: a mention-pair affinity model, $\phi : \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{R}$, and a mention-entity affinity model, $\psi : \mathcal{E} \times \mathcal{M} \rightarrow \mathbb{R}$. An edge between two mentions m_i and m_j has weight:

$$w_{m_i, m_j} = -\phi(m_i, m_j), \tag{2}$$

and the weight of the edge from entity e to m_i is:

$$w_{e, m_i} = -\psi(e, m_i) \tag{3}$$

Each of $\phi(\cdot, \cdot)$ and $\psi(\cdot, \cdot)$ are independently parameterized by dual-encoder transformer models

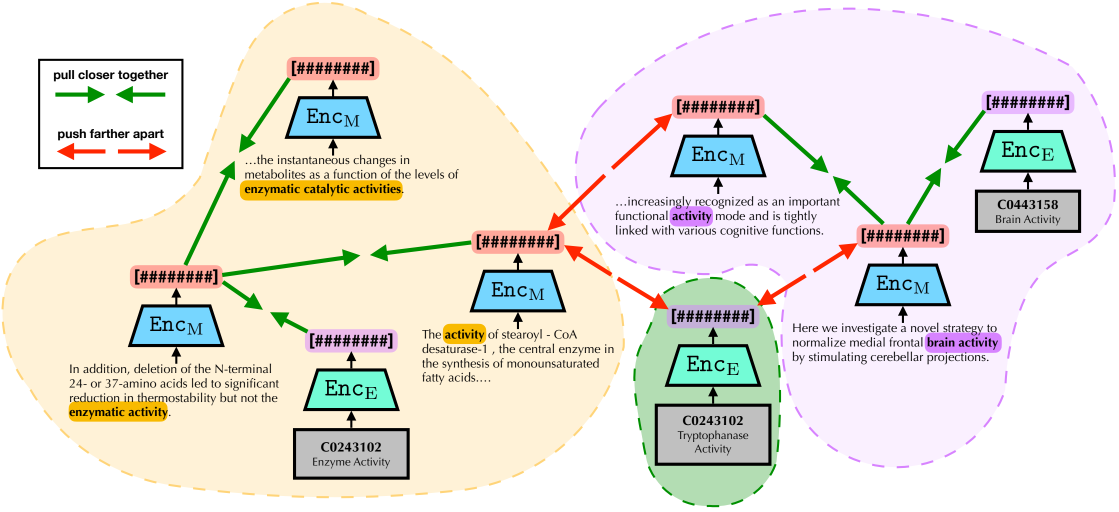
pull closer together
push farther apart
[########]
Enc_M
…the instantaneous changes in metabolites as a function of the levels of enzymatic catalytic activities.
[########]
Enc_M
In addition, deletion of the N-terminal 24- or 37-amino acids led to significant reduction in thermostability but not the enzymatic activity.
[########]
Enc_E
C0243102
Enzyme Activity
[########]
Enc_M
The activity of stearoyl - CoA desaturase-1 , the central enzyme in the synthesis of monounsaturated fatty acids.…
[########]
Enc_M
…increasingly recognized as an important functional activity mode and is tightly linked with various cognitive functions.
[########]
Enc_E
C0243102
Tryptophanase Activity
[########]
Enc_M
Here we investigate a novel strategy to normalize medial frontal brain activity by stimulating cerebellar projections.
[########]
Enc_E
C0443158
Brain Activity

Figure 1: **Arborescence-based Training Objective for Mention & Entity Representations**. Shown above is an illustrative example our proposed training objective for a dual-encoder ($\texttt{Enc}_{\mathrm{M}}$, $\texttt{Enc}_{\mathrm{E}}$) on real mentions and entities from the MedMentions data set. Mentions are highlighted in context and entities are represented using grey boxes with the name and unique identifier for the entity in UMLS. First, each mention and entity is encoded into a dense vector representation using the respective transformer encoder. Pairs of mentions and mention-entity pairs are then selected based on our arborescence-based procedure. The embeddings of these pairs are encouraged to be pulled closer together if both endpoints are contained in the pruned arborescence structure (represented by the shaded regions), or encouraged to be pushed farther apart if the endpoints are sampled as hard negatives.

(Gillick et al., 2019; Humeau et al., 2019), one for mentions ($\texttt{Enc}_{\mathrm{M}}$), and one for entities ($\texttt{Enc}_{\mathrm{E}}$). The affinity models are simply the inner products of the associated encoded representations:

$$\begin{aligned} \phi(m_i, m_j) &= \texttt{Enc}_{\mathrm{M}}(m_i)^T \texttt{Enc}_{\mathrm{M}}(m_j) \\ \psi(e, m_i) &= \texttt{Enc}_{\mathrm{E}}(e)^T \texttt{Enc}_{\mathrm{M}}(m_i) \end{aligned} \tag{4}$$

For the mention encoder, $\texttt{Enc}_{\mathrm{M}}$, the transformer input is the surrounding mention context with the mention span marked by special tokens `[START]` and `[END]`:

$$\texttt{[CLS]}\, c_{\text{left}}\, \texttt{[START]}\, m_i\, \texttt{[END]}\, c_{\text{right}}\, \texttt{[SEP]}$$

where c_{left} and c_{right} are the left and right contexts of the mention m_i in the document. For the entity encoder, $\texttt{Enc}_{\mathrm{E}}$, the transformer takes as input the title and description of the entity:

$$\texttt{[CLS]}\, e_{\text{title}}\, \texttt{[TITLE]}\, e_{\text{desc}}\, \texttt{[SEP]}$$

In this input, e_{desc} is the token sequence corresponding to the description of the entity, which could include natural language text related to the entity, such as a "wiki" entry, a list of entity aliases, or any other available features useful in forming an entity representation.

2.3 Training Procedure

We now define our approach for training the affinity models, $\phi(\cdot, \cdot)$ and $\psi(\cdot, \cdot)$, and their associated encoders, $\texttt{Enc}_{\mathrm{M}}$ and $\texttt{Enc}_{\mathrm{E}}$. Our objective is to optimize the dissimilarity function $f(\cdot, \cdot)$ such that the procedure infers a set of clusters that each contain exactly one entity, and every mention is assigned to the cluster containing its ground truth entity. We optimize $f(\cdot, \cdot)$ using mini-batch gradient descent by sequentially building batches of mentions $B \subset \mathcal{M}$ over the training data, where each $m_i \in B$ has its gold entity defined by $e_i^\star$. We then build a graph G_B with nodes consisting of each $m_i \in B$, each mention coreferent to $m_i \in B$, and the set of gold entities for each $m_i \in B$. For every m_i, we build a set of directed edges defined by

$$\begin{aligned} E_{m_i} = &\left\{ (e_i^\star, m_\ell) \,\middle|\, m_\ell \in \mathcal{M}_{e_i^\star} \right\} \\ &\cup \left\{ (m_\ell, m_p) \,\middle|\, m_\ell, m_p \in \mathcal{M}_{e_i^\star} \right\} \end{aligned} \tag{5}$$

The complete set of edges in graph G_B for a mini-batch B is then given by $E(G_B) = \bigcup_{m_i \in B} E_{m_i}$. Observe that the resultant edges ensure that each connected component contains exactly one entity (namely, the gold entity for the mentions in that connected component).

Forming Clusters for Positive Sampling. The graph G_B is input to a constrained clustering procedure that partitions a graph G into disjoint clusters $\mathcal{C} = \{C_1, \ldots, C_M\}$ such that each cluster contains at most one entity. There are three constraints that every $C \in \mathcal{C}$ must satisfy:

$$\begin{aligned}
&\text{(i)}\ \ |C \cap \mathcal{E}| \leq 1,\\
&\text{(ii)}\ \ \forall u, v \in C,\ \text{connected}(u, v) \implies f(u, v) \leq \lambda,\\
&\text{(iii)}\ \ \forall u, v \in C,\ \text{connected}(u, v) \vee \text{connected}(v, u)
\end{aligned}$$

where λ is a hyperparameter representing the dissimilarity threshold over which edges between nodes are dropped. We set $\lambda = \infty$ during training. These constraints ensure that (i) there is at most one entity in each cluster, (ii) if u is reachable from v then every edge in the path from v to u has a weight $\leq \lambda$, and (iii) each node in the cluster has a path connecting itself with every other node in the cluster. We solve this constrained clustering problem, i.e., partition graph G, using a process similar to Angell et al. (2021).

Specifically, we first remove all edges in graph G with weight greater than threshold λ. We then evaluate each edge $(u, v) \in E$ in descending order of dissimilarity and check if its presence violates any of the three constraints defined above, removing the edge from E if it does. If not, we evaluate whether there is an entity in the connected component of node u, i.e. $|C_u \cap \mathcal{E}| = 1$. If $|C_u \cap \mathcal{E}| = 1$, we temporarily drop edge (u, v) and check whether v can still be reached by an entity node. If reachable, we permanently drop (u, v), maintaining the validity of constraint (i) as well as our minimax dissimilarity function $f(\cdot, \cdot)$. If an entity cannot reach v, we retain edge (u, v), preserving the connectivity of the cluster, and iterate further. Our predicted clusters are the resultant connected components in the partitioned graph G.

Using this clustering procedure on G_B, we construct a partitioned target graph $G_B^\star = \{E_{m_i}^\star \mid m_i \in B\}$. We use $E_{m_i}^\star$ to optimize the parametric encoder models. Note that each mention node in a target edge set $E_{m_i}^\star$ has only one incoming edge originating from either an entity or a mention, and the selection of $E_{m_i}^\star$ was done in a way to minimize $f(\cdot, \cdot)$ between mentions and entities with coreferent labels on the subgraph of the mini-batch.

For every cluster with an entity node, the edge structure is a directed analogue of the minimum spanning tree, where there exists a directed path from the entity node to every other node in the cluster. This structure is often referred to as the *minimum spanning arborescence*, thus lending its name to our method, i.e. ARBORESCENCE-based linking.

Negative Sampling. Akin to the graph embedding objectives used by Nickel and Kiela (2018) and others, we construct our objective by sampling hard negative edges. For each mention $m_i \in B$, the set of negative edges $\mathtt{N}(m_i)$ is the $k/2$ lowest-weight incoming edges from $\mathcal{E} \setminus \{e_i^\star\}$ and the $k/2$ lowest-weight incoming edges from $\mathcal{M} \setminus \mathcal{M}_{e_i^\star}$, where k is a specified hyperparameter.

Loss Function. We define $\Gamma(m_i) = \{u \mid (u, m_i) \in E_{m_i}^*\} \cup \{u \mid (u, m_i) \in \mathtt{N}(m_i)\}$ to be the set of all neighbors with an outgoing edge to m_i in the training graph. Let $\mathbb{I}_{u,m_i}$ be the indicator variable such that $\mathbb{I}_{u,m_i} = 1$ if $(u, m_i) \in E_{m_i}^*$ and $\mathbb{I}_{u,m_i} = 0$ otherwise. Our loss function with respect to each mention $m_i \in B$ is then defined as follows:

$$\begin{aligned}
\mathcal{L}(m_i) = \sum_{u \in \Gamma(m_i)} \Big(&\mathbb{I}_{u,m_i} \log(\sigma_u(w_{u,m_i})) \qquad (6)\\
&+ (1 - \mathbb{I}_{u,m_i}) \log(1 - \sigma_u(w_{u,m_i}))\Big),
\end{aligned}$$

where $\sigma(\cdot)$ is the softmax function over all edges in $\Gamma(m_i) \times \{m_i\}$. The loss for the entire batch B is the mean of losses over all mentions in B. Optimizing this loss function requires simultaneously increasing the likelihood of the positive edges and decreasing the likelihood of the negative edges. This objective and training routine are inspired by the supervised single-linkage clustering proposed by Yadav et al. (2019), but differs in the choice of loss function and selection of negative examples. We also experimented with the standard cross-entropy loss, but found its performance subpar.

3 Experiments

We are interested in investigating the following empirical research questions:

- Does our proposed approach improve the recall of candidate generators?
- Do improvements in candidate generation at training lead to improvements in downstream re-ranker models?
- Does our approach result in better learned mention embeddings that can be used for coreference / discovering entities when a KB does not exist?

Experiment Details Our experiments are run on top of BLINK (Wu et al., 2020), a PyTorch (Paszke et al., 2019) implementation of dual- and cross-encoder architectures for entity linking, with

		MedMentions							ZeShEL						
Training Method	Recall@	1	2	4	8	16	32	64	1	2	4	8	16	32	64
In-Batch Negatives		58.70	69.01	75.87	80.03	83.14	85.54	87.73	39.27	53.02	62.98	70.32	75.97	80.27	84.04
k-NN Negatives		56.85	65.96	71.68	76.50	80.31	83.51	86.11	49.81	60.59	68.24	74.11	78.07	81.53	84.77
TF-IDF ‡		50.8	63.8	73.4	79.2	82.3	84.6	85.3	-	-	-	-	-	-	-
In-Batch Negatives ‡‡		-	-	-	-	-	-	-	-	-	-	-	-	-	82.06
Arborescence †		**72.31**	**80.88**	86.09	**89.86**	**92.36**	**94.31**	**95.67**	50.31	61.04	68.34	74.26	78.40	82.02	85.11
1-NN Arborescence †		71.99	80.78	**86.10**	89.61	91.92	93.75	95.23	**51.33**	62.00	69.03	74.67	**78.86**	81.97	85.13
1-Rand Arborescence †		71.27	80.17	85.44	89.09	91.65	93.34	94.88	50.86	**62.09**	**69.36**	**75.05**	78.78	**82.50**	**85.70**

Table 1: **Dual-Encoder Retriever Results: Recall@k** (†Proposed methods; ‡Angell et al. (2021); ‡‡Wu et al. (2020))

		MedMentions	ZeShEL
$\lvert\mathcal{M}\rvert$	Train	120K	49K
	Dev	40K	10K
	Test	40K	10K
$\lvert\mathcal{E}\rvert$	Train	19K	26K
	Dev	9K	7K
	Test	8K	7K
$\lvert\mathcal{E} \setminus \mathcal{E}_{\text{Train}}\rvert$	Dev	4K	7K
	Test	4K	7K

Table 2: **Dataset Statistics**. $|\mathcal{M}|$ is the number of mentions. $|\mathcal{E}|$ is the number of unique entities in the labeled partition (not the total KB size). $|\mathcal{E} \setminus \mathcal{E}_{\text{Train}}|$ is the number of *zero-shot* entities. The total KB size of MedMentions and ZeShEL is 2.3M and 492K, respectively.

model fine-tuning performed over only BERT-base, since gains from pre-trained LM size are unrelated to our approach. For more details, see Appendix §A.1.

3.1 Datasets

We run experiments on two entity linking datasets that both require generalization to unseen entities at test time. Each document in the datasets contains a set of entity mention spans, which are pre-defined using common mention detection heuristics. KB entities are composed of two metadata attributes – an entity *title* and *description*, which are natural language sequences of text. ZeShEL, additionally, contains a fine-grained *type* specification, which is needed due to the diverse disjoint domains contained in the dataset. The statistics for both datasets are reported in Table 2.

MedMentions (Mohan and Li, 2019) is a collection of titles and abstractions of bio-medical research papers. The KB that is used for this dataset is the 2017AA full-version of UMLS. The validation and test sets contain both entities that are present in the training set as well as entities that are zero-shot (never seen at training time). We use the author-recommended ST21pv subset.

ZeShEL (Logeswaran et al., 2019) is a collection of crowd-sourced wikis, which are divided into train, validation, and test splits such that no Fandom topic overlaps across the sets. In this way, all entities that appear at validation and test time are not seen during training.

3.2 Dual-Encoder Retrieval

In order to evaluate the benefit of explicitly modeling coreference relationships, we construct three variants of our proposed dual-encoder training objective, which jointly trains both the mention-mention similarity function $\phi(\cdot, \cdot)$ and the mention-entity similarity function $\psi(\cdot, \cdot)$. We compare to baselines that only explicitly train $\psi(\cdot, \cdot)$, and rely on the structure of $\phi(\cdot, \cdot)$ sharing representations with $\psi(\cdot, \cdot)$ to provide meaningful mention-mention similarities. Our proposed objectives are identical to each other except in how the positive training pairs are constructed, while our baselines differ in the selection of negatives.

Arborescence In this training variant, for each mention query, we first construct a fully-connected graph of the ground truth coreferent mention cluster along with the gold entity. We then apply the pruning procedure described in the previous section to compute an arborescence rooted at the entity node. From the resultant graph, each pair of a mention and its incoming-edge node (which can either be a coreferent mention or the gold entity) is then treated as a positive example for training. Following previous work by Gillick et al. (2019), we use hard negative mining with $k = 10$ negatives composed of equal number of mention and entities.

1-NN Arborescence Instead of constructing a fully-connected k-NN graph over the entire gold cluster, in this variant we approximate the arborescence structure by pruning a restricted graph of

Re-ranker	Candidate Retriever	Accuracy Overall	 Seen	 Unseen	Oracle Self	 Union
k-NN Negatives	Dual (In-Batch Negatives)	73.31	77.58	58.47	**80.78**	47.96
k-NN Negatives	Dual (k-NN Negatives)	70.76	77.05	48.85	79.90	21.12
MST & k-NN (Angell et al., 2021)	TF-IDF (Angell et al., 2021)	74.1	77.3	**62.9**	-	-
k-NN Negatives	Dual (Arborescence) †	**75.73**	**79.97**	60.99	76.09	**75.64**
k-NN Negatives	Dual (1-NN Arborescence) †	74.73	78.91	60.19	75.48	74.71
k-NN Negatives	Dual (1-Rand Arborescence) †	74.89	79.39	59.22	75.75	74.95

Table 3: **MedMentions: Cross-Encoder Linking Results**: We report the re-ranker accuracy trained using the candidates generated by each retriever variant. (†Proposed methods)

only the gold entity, the query mention, and the most similar within-cluster mention neighbor of the query. We keep all other details of the training procedure identical to the first variant.

1-Rand Arborescence A third training objective we explore modifies the initial k-NN graph construction by restricting the nodes to the gold entity, the query mention, and a *random* within-cluster mention neighbor of the query, instead of the nearest-neighbor.

Baselines We compare to two baselines following previous work: (1) training $\psi(\cdot,\cdot)$ with random negatives (In-batch Negatives) where each gold entity for a mention in a training batch is treated as a negative example for all other mentions in the batch, and (2) training $\psi(\cdot,\cdot)$ with hard negatives (k-NN Negatives) similar to the negative mining in our proposed methods albeit with only mention-*entity* positive selection.

Results In Table 1, we report the test set *recall*@64 for each dual-encoder model, where the prediction is evaluated as a hit if the gold entity is retrieved in the top-64 candidates of the model. On each dataset, we additionally include the performance of candidate generators used by previous works that we compare to.

We find that models trained with explicit coreference relationships outperform those that incorporate this relationship only indirectly. For *recall*@64, our proposed methods improve over the baseline models by at least 7.94 percentage points on MedMentions and 0.93 points on ZeShEL. Even at linking, or *recall*@1, our proposed methods show similar improvements, with gains of 13.61 and 1.52 points over the next best baseline models. We perform a more comprehensive analysis of the dual-encoder linking performance and describe our inference approach and results in Appendix §A.2 and §A.3.

We posit that much of the observed gains in recall using our proposed methods result from higher quality mention embeddings generated due to a wide array of surface forms available to mention queries at training. Since each training example evaluates not only the gold entity but also its coreferent mentions, this leads to better generalization of representations. We evaluate this improvement in representations in the clustering/coreference setting in Section 3.5.

We also provide representative examples of predictions comparing the candidates generated by our proposed Arborescence to the retriever from Angell et al. (2021) in Appendix Table 7.

3.3 Cross-Encoder Re-ranking

To answer our second research question, we compare 5 cross-attention models, which are trained using entity candidates generated by the dual-encoder variants discussed in the previous experiment. Training and inference batches are constructed by concatenating each mention with an entity candidate separated by a `[SEP]` token. Similar to Wu et al. (2020), we use the top-64 retrieved entities as hard negatives during training and as linking candidates during inference.

Results We report the cross-encoder linking accuracy for MedMentions in Table 3. We additionally report the breakdown of accuracy on subsets of test mentions for which the ground truth entities were not evaluated ("unseen") during training, illustrating the zero-shot capability of the models. We also include the current state-of-the-art results by Angell et al. (2021), which uses an n-gram based model for candidate generation and two cross-encoder models, one each for mention-mention and mention-entity scoring, as the re-ranker. We observe that each cross-encoder trained with candidates generated by an arborescence-based model outperforms the baselines, including the current

			Oracle			
			Self		Union	
Candidate Retriever	**Acc.**	**Macro**	**Acc.**	**Macro**	**Acc.**	**Macro**
Dual (In-Batch Negatives)	61.27	60.93	**64.96**	67.81	62.91	66.13
Dual (k-NN Negatives)	61.02	60.47	63.66	66.91	61.30	64.72
Dual (In-Batch Negatives) (Wu et al., 2020)	-	61.34	-	-	-	-
Dual (Arborescence) †	60.72	60.36	62.64	65.90	61.04	64.39
Dual (1-NN Arborescence) †	60.47	60.48	63.20	66.70	61.03	64.77
Dual (1-Rand Arborescence) †	**62.35**	**62.53**	64.94	**67.90**	**63.33**	**66.51**

Table 4: **ZeShEL: Cross-Encoder Linking Results: Unnormalized Accuracy**. We report the accuracy of the re-ranker trained using the candidates generated by each retriever. (†Proposed methods)

SOTA by at least 0.63 points, and the best performing model – Arborescence – achieves 1.63 point gains. We note, however, that Angell et al. (2021) does better on unseen entities by 1.91 points compared to Arborescence, which might be a result of the within-document nature of their TF-IDF candidate retriever.

Table 4 contains linking results for ZeShEL, where each reported model varies only in the method used for retrieving the entity candidates, while the cross-encoder re-ranker training method is held constant (k-NN Negatives with $k = 64$). Since ZeShEL is completely zero-shot, we do not include a seen-unseen analysis. We follow Wu et al. (2020) and report the unnormalized accuracy, which is calculated as the percentage of successes out of the total number of query mentions in the test set, and the macro-averaged unnormalized accuracy, which is a simple average of the unnormalized accuracies over the different "worlds", or domains, in the test set. We find that the best performing model is 1-Rand Arborescence, with a 1.19 point difference in macro-averaged accuracy over the next best model (Wu et al., 2020).

We also note that, unlike on MedMentions, not all of our proposed models have higher accuracy than the mention-entity baselines. Since a key motivation for the proposed arborescence-based methods is to explicitly model coreference relationships during training, we expect performance gains to be strongly correlated with the number of coreference links present within the dataset. We analyze the two datasets in terms of the number of mentions for each KB entity. This can be thought of as how large each cluster of coreferent mentions is. We report a histogram distribution in Figure 2. We find that the clusters in ZeShEL are typically very small (at most 3), whereas in MedMentions, each cluster has many more mentions with maximum sizes of 1256, 434, and 447 across the train, validation, and test sets.

3.4 Oracle Inference

In this setting, we isolate the re-ranking capability of the cross-encoder from the quality of the candidates retrieved at *inference*. This setting also removes the upper-bound on re-ranker accuracy by artificially injecting the ground-truth entity in the top-64 candidates retrieved at inference for each mention where retrieval failed. An additional setting we explore holds this oracle candidate set constant across each variant of the cross-encoder by taking a union over all dual-encoder candidate sets, and then proceeding to inject the ground-truth. This construction provides a way to purge the factor of candidate retrieval quality at inference, which otherwise conflates the comparison of re-ranking performance. We refer to these oracle settings as Self and Union, respectively.

Results As seen in column **Oracle** of Table 3, the baseline models show higher linking accuracy than our proposed methods when the gold entity is guaranteed to be present in the original candidate set. However, the performance of the baseline models drops significantly ($\geq$ 32 points) when evaluated with the Union candidate set, while the arborescence-based models show a ± 0.9 point variation. We believe this discrepancy clearly highlights the poor quality of candidates retrieved by the baseline models compared to our proposed methods. This also explains the inflation in accuracy of the baselines on the Self set due to the trivial discrimination task presented to the cross-encoders. We further point to linking performance on the Union set, which provides the more challenging task of differentiating between higher quality candidates that are similar and argue that the large performance difference ($\geq$ 26.75 points) is strongly

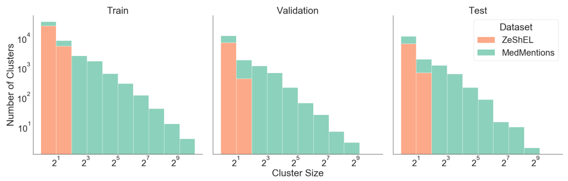

Figure 2: **Cluster Distribution**. We count the number of mentions in each coref cluster. Clusters in ZeShEL are typically very small, compared to the MedMentions clusters which have considerably more mentions.

indicative of the greater linking capacity of our proposed methods.

In Table 4, we report both the micro accuracy and macro-averaged accuracy for the two oracle sets. We observe that 1-RAND ARBO performs the best on the UNION set, but is marginally outperformed by IN-BATCH on micro accuracy on the SELF set by 0.02 points. In contrast to the fluctuation on MedMentions, the relative stability of results on the oracle candidate sets indicates that the candidates generated by each model have similar quality.

3.5 Mention Coreference

We evaluate the quality of the learned mention representations for cross-document coreference. Entity labels of each mention are its ground truth cluster assignment. To form clusters, we build mention-only arborescences using the clustering procedure described in Section 2.3, tuning the threshold value, λ, based on the validation data. In Table 5, we report the Adjusted Rand Index (ARI) clustering scores using each of the representation learning objectives using dual-encoders. For both ZeShEL and MedMentions, we report ARI on all the test mentions (denoted ALL). For MedMentions, we report two additional settings: (1) ARI when clustering mentions with ground truth entity not seen at training (denoted UNSEEN ONLY) and (2) clustering on all mentions but evaluating only on the set in (1) (denoted ALL/UNSEEN). Representations learned with the ARBORESCENCE objective performs best on each setting, aligning with the inductive bias.

4 Related Work

Entity Linking Entity linking has been widely studied (Milne and Witten, 2008; Cucerzan, 2007; Lazic et al., 2015b; Gupta et al., 2017; Raiman and Raiman, 2018; Kolitsas et al., 2018; Cao et al., 2021, inter alia). Dutta and Weikum (2015) combine clustering-based cross-document coreference decisions and linking around sparse bag-of-word representations not well suited for the embedding-based representations used in this work. Hoffart et al. (2011); Cheng and Roth (2013); Ganea and Hofmann (2017); Le and Titov (2018) use global objectives instead of independent predictions, measuring the compatibility of entity links. Zhang and Stratos (2021) use noise contrastive estimation to mine hard negatives for the linking task.

Setting	MedMentions ALL	 ALL/UNSEEN	 UNSEEN ONLY	ZeShEL ALL
IN-BATCH NEGATIVES	0.37	0.71	0.71	0.31
K-NN NEGATIVES	0.26	0.73	0.80	0.29
ARBORESCENCE	**0.51**	**0.83**	**0.85**	**0.34**
1-NN ARBORESCENCE	0.47	0.75	0.83	**0.34**
1-RAND ARBORESCENCE	0.35	0.63	0.81	0.32

Table 5: **Coreference Results**. We report the Adjusted Rand Index achieved by clustering (§2.3) the embeddings produced by each model. We evaluate on three settings: ALL (clustering & evaluating on all test set mentions), ALL/UNSEEN (clustering all mentions, evaluating on mentions with ground truth entity not seen in train), UNSEEN ONLY (clustering & evaluating on mentions with ground truth entity not in train).

Cross-document Coreference Models have been developed for the cross-document coreference setting where no entity KB is known in advance (Bagga and Baldwin, 1998; Gooi and Allan, 2004; Singh et al., 2011; Barhom et al., 2019; Cattan et al., 2020; Caciularu et al., 2021; Ravenscroft et al., 2021; Logan IV et al., inter alia).

Alternatives to Cross-Encoders Our work demonstrates how clustering-based training and prediction improves dual-encoder based models for linking and discovery. If prediction efficiency, and not training efficiency, was the only concern, one could use model distillation (Hinton et al., 2015; Izacard and Grave, 2021, inter alia). We could also consider models such as poly-encoders as an alternative to dual-encoders (Humeau et al., 2020).

5 Conclusion

We presented a novel approach for learning mention and entity representations for use in entity linking candidate generation and prediction, as well as in the discovery of new entities. Our approach uses an objective that explicitly incorporates mention-to-mention coreference relationships. We demonstrated its empirical effectiveness through analysis on two datasets, MedMentions and the Zero-Shot Entity Linking dataset. As future work, we hope to further analyze these objectives with the lens of efficiency, distillation, and domain transfer.

6 Ethical Considerations

The base models, which we fine-tuned, and evaluation datasets are all publicly available. We will also make our code and models publicly available. There are several ways in which entity linking/entity resolution models could be biased and there is the potential for those biases to have harmful downstream consequences. There is a large body of work studying the biases of language models (such as those used for fine-tuning here) and coreference models. Most notably in understanding when error rates in coreference differ across certain populations (e.g., genders, races, or any entity-type more broadly). If entity linking and discovery systems are used to build / populate knowledge-bases, those systems may propagate these biased predictions. This could be particularly problematic if one used such a biased knowledge-base with this realization. For instance, if entity mentions are author names on citation data and the entities are scientific authors, statistics like h-index or citation count could be biased if the algorithms used to disambiguate the author names are biased. Lastly, we note entity linking and discovery are related to surveillance and tracking in computer vision, which bear a substantial weight of ethical considerations.

References

Rico Angell, Nicholas Monath, Sunil Mohan, Nishant Yadav, and Andrew McCallum. 2021. Clustering-based inference for biomedical entity linking. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 79–85, Montreal, Quebec, Canada. Association for Computational Linguistics.

Shany Barhom, Vered Shwartz, Alon Eirew, Michael Bugert, Nils Reimers, and Ido Dagan. 2019. Revisiting joint modeling of cross-document entity and event coreference resolution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4179–4189, Florence, Italy. Association for Computational Linguistics.

Avi Caciularu, Arman Cohan, Iz Beltagy, Matthew E Peters, Arie Cattan, and Ido Dagan. 2021. Cross-document language modeling. *arXiv preprint arXiv:2101.00406*.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *International Conference on Learning Representations*.

Arie Cattan, Alon Eirew, Gabriel Stanovsky, Mandar Joshi, and Ido Dagan. 2020. Streamlining cross-document coreference resolution: Evaluation and modeling. *arXiv preprint arXiv:2009.11032*.

Xiao Cheng and Dan Roth. 2013. Relational inference for wikification. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1787–1796, Seattle, Washington, USA. 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.

Rajarshi Das, Ameya Godbole, Dilip Kavarthapu, Zhiyu Gong, Abhishek Singhal, Mo Yu, Xiaoxiao Guo, Tian Gao, Hamed Zamani, Manzil Zaheer, et al. 2019. Multi-step entity-centric information retrieval for multi-hop question answering. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 113–118.

Sourav Dutta and Gerhard Weikum. 2015. C3EL: A joint model for cross-document co-reference resolution and entity linking. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 846–856, Lisbon, Portugal. Association for Computational Linguistics.

Nicholas FitzGerald, Jan A Botha, Daniel Gillick, Daniel M Bikel, Tom Kwiatkowski, and Andrew McCallum. 2021. Moleman: Mention-only linking of entities with a mention annotation network. *arXiv preprint arXiv:2106.07352*.

Evgeniy Gabrilovich, Michael Ringgaard, and Amarnag Subramanya. 2013. Facc1: Freebase annotation of clueweb corpora, version 1 (release date 2013-06-26, format version 1, correction level 0).

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.

Dan Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning dense representations for entity retrieval. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*.

Chung Heong Gooi and James Allan. 2004. Cross-document coreference on a large scale corpus. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL*

2004, pages 9–16, Boston, Massachusetts, USA. Association for Computational Linguistics.

Nitish Gupta, Sameer Singh, and Dan Roth. 2017. Entity linking via joint encoding of types, descriptions, and context. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2681–2690, Copenhagen, Denmark. Association for Computational Linguistics.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

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

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2019. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *International Conference on Learning Representations*.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *International Conference on Learning Representations*.

Gautier Izacard and Edouard Grave. 2021. Distilling knowledge from reader to retriever for question answering. In *International Conference on Learning Representations*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*.

Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. 2018. End-to-end neural entity linking. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 519–529, Brussels, Belgium. Association for Computational Linguistics.

Nevena Lazic, Amarnag Subramanya, Michael Ringgaard, and Fernando Pereira. 2015a. Plato: A selective context model for entity resolution. *Transactions of the Association for Computational Linguistics*, 3:503–515.

Nevena Lazic, Amarnag Subramanya, Michael Ringgaard, and Fernando Pereira. 2015b. Plato: A selective context model for entity resolution. *Transactions of the Association for Computational Linguistics*, 3:503–515.

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.

Robert Leaman and Zhiyong Lu. 2016. Taggerone: joint named entity recognition and normalization with semi-markov models. *Bioinformatics*, 32(18):2839–2846.

Ying Lin, Chin-Yew Lin, and Heng Ji. 2017. List-only entity linking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 536–541, Vancouver, Canada. Association for Computational Linguistics.

Xiao Ling, Sameer Singh, and Daniel S. Weld. 2015. Design challenges for entity linking. *Transactions of the Association for Computational Linguistics*, 3:315–328.

Fangyu Liu, Ehsan Shareghi, Zaiqiao Meng, Marco Basaldella, and Nigel Collier. 2020. Self-alignment pretraining for biomedical entity representations. *arXiv preprint arXiv:2010.11784*.

Robert L. Logan IV, Matt Gardner, and Sameer Singh. 2020. On importance sampling-based evaluation of latent language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2171–2176, Online. Association for Computational Linguistics.

Robert L Logan IV, Andrew McCallum, Sameer Singh, and Daniel Bikel. Benchmarking scalable methods for streaming cross document entity coreference.

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

David Milne and Ian H Witten. 2008. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 509–518.

Sunil Mohan and Donghui Li. 2019. Medmentions: A large biomedical corpus annotated with umls concepts. *arXiv preprint arXiv:1902.09476*.

Maximillian Nickel and Douwe Kiela. 2018. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *International Conference on Machine Learning*, pages 3779–3788. PMLR.

Tommaso Di Noia, Vito Claudio Ostuni, Paolo Tomeo, and Eugenio Di Sciascio. 2016. Sprank: Semantic path-based ranking for top-n recommendations using linked open data. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(1):1–34.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*.

Jonathan Raiman and Olivier Raiman. 2018. Deeptype: multilingual entity linking by neural type system evolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

James Ravenscroft, Arie Cattan, Amanda Clare, Ido Dagan, and Maria Liakata. 2021. Cd2cr: Co-reference resolution across documents and domains. *arXiv preprint arXiv:2101.12637*.

Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. 2011. Large-scale cross-document coreference using distributed inference and hierarchical models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 793–803, Portland, Oregon, USA. Association for Computational Linguistics.

Mujeen Sung, Hwisang Jeon, Jinhyuk Lee, and Jaewoo Kang. 2020. Biomedical entity representations with synonym marginalization. *arXiv preprint arXiv:2005.00239*.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Zero-shot entity linking with dense entity retrieval. In *EMNLP*.

Nishant Yadav, Ari Kobren, Nicholas Monath, and Andrew Mccallum. 2019. Supervised hierarchical clustering with exponential linkage. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6973–6983. PMLR.

Sheng Zhang, Hao Cheng, Shikhar Vashishth, Cliff Wong, Jinfeng Xiao, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Knowledge-rich self-supervised entity linking. *arXiv preprint arXiv:2112.07887*.

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

A Appendix

A.1 Experiment Details

Our experiments are run on top of BLINK (Wu et al., 2020), a PyTorch (Paszke et al., 2019) implementation of dual- and cross-encoder architectures for entity linking, with model fine-tuning performed over only BERT-base, since gains from pre-trained LM size are unrelated to our approach. Each training procedure is run on a single machine using 2 NVIDIA Quadro RTX 8000 GPUs. Our dual-encoder models for ZeShEL and MedMentions have 218M and 230M parameters, respectively. Each variant is optimized using mini-batch gradient descent using the Adam optimizer for 5 epochs using a mini-batch size of 128 to accumulate the gradients. Experiments with batch sizes < 128 performed poorly, possibly due to increased fluctuation of gradients, and sizes > 128 were computationally infeasible to run with our available compute resources. For ZeShEL, the dual-encoder models are trained using 192 warm-up steps and learning rates of 1e-5, 3e-5, and 3e-5 for In-batch, k-NN, and Arborescence-based models, respectively. For MedMentions, each model is trained using 464 warm-up steps and a learning rate of 3e-5. All cross-encoder models are trained with a mini-batch size of 2, learning rate of 2e-5, and an additional linear layer. Our MedMentions and ZeShEL cross-encoder models have 108M and 109M parameters, respectively. We use FAISS[1] (Johnson et al., 2017) for fast nearest-neighbor search during graph construction at both training and inference. For MedMentions, the execution time was 70 mins to embed and index 2M entities and 120K mentions, and 20 mins to perform exact nearest-neighbor search for the 120K mentions.

A.2 Dual-Encoder Inference Procedure

Building the Graph The structure of the graph G impacts the dissimilarity function by changing the paths between pairs of nodes in addition to changing which pairs of nodes are connected. We advocate for a simple, deterministic approach to construct this graph. For each mention m, construct E_m by (1) adding edges from m's k-nearest neighbor mentions in $\mathcal{M}$ to m, and (2) adding an edge from m's nearest entity to m:

$$E_m = \Bigg\{ (u, m) \,\Bigg|\, u \in \underset{m' \in \mathcal{M}}{\text{argmink}}\, w_{m',m} \vee u = \underset{e \in \mathcal{E}}{\text{argmin}}\, w_{e,m} \Bigg\} \quad (7)$$

The complete collection of edges E in G is given by $E(G) = \bigcup_{m \in \mathcal{M}} E_m$. There are other ways that one could conceivably pick the pairs of mentions to be connected in the graph. For example, one could use the minimum spanning tree over the mentions. This approach, however, has several drawbacks: (1)

[1] https://github.com/facebookresearch/faiss

		MedMentions				ZeShEL
Training	**Inference**	**Overall**	**Seen**	**Unseen**	**Inference**	**Overall**
In-Batch Negatives	Clustering (undirected)	59.11	61.88	49.45	Independent	39.27
k-NN Negatives	Independent	56.86	64.03	31.88	Independent	49.81
Arborescence †	Clustering (directed)	**72.19**	**77.48**	**53.79**	Independent	50.31
1-NN Arborescence†	Clustering (directed)	72.00	77.29	53.60	Clustering (directed)	**51.09**
1-Rand Arborescence†	Clustering (directed)	71.33	77.02	51.51	Clustering (directed)	50.85

Table 6: **Dual-Encoder Linking Results: Accuracy %** (†Procedures incorporating explicit mention-to-mention coreference relationships)

the directionality of nearest neighbor relationships is ignored leading to added noise in the graph, and (2) the resultant graph includes edges that clearly cross cluster boundaries due to this approach forcing all pairs of mentions to be connected.

Forming Clusters & Making Predictions To make linking decisions for each mention m_i^d, we assign the ID of the entity present in the mention's cluster as the linking label (or NIL if there is no entity in the cluster). Let $\mathcal{C}(m_i^d)$ be the predicted cluster of mention m_i^d, then:

$$e_i^d = \begin{cases} \mathcal{C}(m_i^d) \cap \mathcal{E}, & \text{if } |\mathcal{C}(m_i^d) \cap \mathcal{E}| = 1 \\ \text{NIL}, & \text{otherwise} \end{cases}. \tag{8}$$

Furthermore, the clusters we predict for in the entity discovery setting are exactly $\mathcal{C}$.

A.3 Experiment: Dual-Encoder Linking

Each model is evaluated using three inference procedures. Independent refers to predictions made using only mention-entity edges. This method was used by Wu et al. (2020) to generate candidates for a cross-encoder model trained on ZeShEL. Clustering (undirected) refers to a hierarchical agglomerative clustering (HAC) procedure, following previous work by Angell et al. (2021), which is akin to the procedure for positive sampling used for training our arborescence-based models, but with no edge directionality. Clustering (directed) adds directed edges to the previous method. For each model, we pick the best performing inference procedure on the dev set and report the test set performance.

We report the linking accuracy in Table 6 but leave out models from previous works since they do not report linking accuracy of their candidate generators. We specify the inference method used in each case, chosen based on the dev set accuracy of the models. Similar to our cross-encoder results in Table 3, we report the "seen" and "unseen" performance for MedMentions.

Mention	[...] Mutations of critical amino acids affected either *dsDNA* recombination or both ssDNA and dsDNA recombination indicating two separable functions , one of which is critical for dsDNA recombination and the second for recombination per se [...]
(Angell et al., 2021)	**DNA** (C0012854): (Chemical , DNA , Deoxyribonucleic Acid , substance : dna molecules ; dsDNA ; Deoxyribonucleic acid ; dna / desoxyribonucleic acid ; DNA / desoxyribonucleic acid ; DNA molecule ; DNA - Deoxyribonucleic acid [...]
Ours	**DNA , Double - Stranded** (C0311474): Chemical , substance : double stranded dna ; DNA , Double Stranded ; Double - Stranded DNA ; ds dna ; deoxyribonucleic acid double strand [...]
Mention	[...] mean dose , and maximum dose were significantly associated with parotid gland atrophy . Multivariate analysis indicated that only V5 was significantly associated with *atrophy*. Increasing V5 was a significant risk factor for parotid gland atrophy after carbon ion radiotherapy [...]
(Angell et al., 2021)	**Muscular Atrophy** (C0026846): Biologic Function , Muscular , diagnosis , disorder , finding , physical finding : atrophy ; muscle ; amyotrophy ; muscle atrophy was seen ; Wasting ; muscle ; Atrophies , Muscle ; Muscle thinning [...]
Ours	**Atrophy of parotid gland** (C0341045): (Biologic Function , disorder : atrophy ; parotid gland)
Mention	[...] This study aimed to determine the methylation phenotype in colorectal cancer for identification of predictive markers for chemotherapy *response*. We performed DNA methylation profiling on 43 non - recurrent and five recurrent colorectal cancer patients using the Illumina Infinium HumanMethylation450 Beadchip assay [...]
(Angell et al., 2021)	**Disease Response** (C1704632): Finding : Response ; response
Ours	**Response to treatment** (C0521982): Clinical Attribute , context - dependent category , finding , function , observable entity , situation : response to treatment ; response treatments ; Therapeutic response; successful treatment [...]

Table 7: **Improved Candidate Generation Yields Correct Entity Linking.** Above are examples of mentions where the candidate generation procedure from (Angell et al., 2021) fails to retrieve the correct entity, and thus, the cross-encoder is not able to correctly link the mention. Our dual-encoder is able to retrieve the correct entity in the candidate set of 64 entities, and then the cross-encoder is able to link each mention to the correct entity.