Detection, Disambiguation, Re-ranking: Autoregressive Entity Linking as a Multi-Task Problem

Anonymous ACL submission

Abstract

We propose an autoregressive entity linking model, that is trained with two auxiliary tasks, and learns to re-rank generated samples at inference time. Our proposed novelties address two weaknesses in the literature. First, as recent improvements in entity linking suggest learning mention detection explicitly could increase performance, we train mention detection as an auxiliary task. Second, previous work suggests that re-ranking could help correct prediction errors. We add a new, auxiliary task, match prediction, to learn re-ranking. Without the use of a knowledge base or candidate sets, our model sets a new state of the art in two benchmark datasets of entity linking: COMETA in the biomedical domain, and AIDA-CoNLL in the news domain. We show through ablation studies that each of the two auxiliary tasks increases performance, and that re-ranking is an important factor to the increase. Finally, our low-resource experimental results suggest that performance on the main task benefits from the knowledge learned by the auxiliary tasks, and not just from the additional training data.

1 Introduction

Entity linking (Zhang et al., 2010; Han et al., 2011) is the task of linking mentions of entities in a text document to concepts in a knowledge base. It is a basic building block used in many NLP applications, such as question answering (Pouran Ben Veyseh, 2016; Yu et al., 2017; Dubey et al., 2018; Shah et al., 2019), word sense disambiguation (Raganato et al., 2017; Uslu et al., 2018), text classification (Basile et al., 2015; Scharpf et al., 2021), and social media analysis (Liu et al., 2013; Yamada et al., 2015; Waitelonis and Sack, 2016).

The task of entity linking (EL) can be decomposed into two subtasks: Mention Detection (MD) and Entity Disambiguation (ED). Many statistical and LSTM-based methods propose to cast EL as a two-step problem, and optimize for both MD and ED (Guo et al., 2013; Luo et al., 2015; Cornolti et al., 2016; Ganea and Hofmann, 2017).

Figure 1: Example of an Entity Linking (EL) source text and generated outputs. Entity mentions to be recognized and disambiguated are denoted in **blue** in the source text. In the outputs, **red** denotes errors, **green** denotes correct answers, **yellow** denotes close matches.

Recent entity linking methods based on language models propose to cast entity linking as a single, end-to-end trained task (Broscheit, 2019; Poerner et al., 2020; El Vaigh et al., 2020). An example is autoregressive entity linking (Petroni et al., 2021; De Cao et al., 2021b), which formulates entity linking as a language generation problem, where mention detection is learned implicitly. In contrast, a more recent, non-autoregressive approach (De Cao et al., 2021a) shows that learning mention detection explicitly can increase performance.

Methods based on word embedding models (Basaldella et al., 2020) propose to learn entity disambiguation by mapping embedding spaces. Their high accuracy at 10 results show that re-ranking could increase entity linking performance.

Contributions. In this paper, we propose an autoregressive entity linking method, that is trained jointly with two auxiliary tasks, and learns to re-

rank generated samples at inference time. Our proposed novelties address two weaknesses in the literature. First, autoregressive entity linking learns mention detection implicitly, but recent methods show learning MD explicitly could increase performance (De Cao et al., 2021a). We propose to add MD as an auxiliary task, that explicitly teaches the model to learn where entity mentions are within the input and target sentences. Second, previous work suggests that re-ranking could correct prediction errors (Basaldella et al., 2020). We propose to train a second, new auxiliary task, called *Match Prediction*. This task teaches the model to re-rank generated samples at inference time. We define match prediction as a classification task where the goal is to identify whether entities in a first sentence were correctly disambiguated in the second sentence. We train this second task with samples generated by the model at each training epoch. At inference time, we then rank the generated samples using our match prediction scores.

Our multi-task learning model outperforms the state of the art in two benchmark datasets of entity linking across two domains: COMETA (Basaldella et al., 2020) from the biomedical and social media domain, and AIDA-CoNLL (Hoffart et al., 2011) from the news domain. We show through four ablation study experiments that each auxiliary task provides improvements on the main task. Then, we show that using our model's match prediction module to re-rank generated samples at inference time plays an important role in increasing performance. Finally, we devise three experiments where we train auxiliary tasks with a smaller dataset. Results suggest that our model's performance is not only due to more training datapoints, but also due to our auxiliary task definition.

2 Related Work

Entity Linking (EL). Entity Linking is often (Hoffart et al., 2011; Steinmetz and Sack, 2013; Piccinno and Ferragina, 2014; De Cao et al., 2021a) trained as two tasks: Mention Detection (MD) and Entity Disambiguation (ED). Mention detection is the task of detecting entity mention spans, such that an entity mention m is represented by start and end positions. A mention m refers to a concept in a given knowledge base. Entity disambiguation is the task of finding the right knowledge base concept for an entity mention, thereby *disambiguating* its meaning.

Early EL methods (Hoffart et al., 2011; Steinmetz and Sack, 2013; Daiber et al., 2013) rely on probabilistic approaches. Hoffart et al. (2011) propose a probabilistic framework for MD and ED, based on textual similarity and corpus occurrence. They test their framework using the entity candidate sets available in the AIDA-CoNLL dataset.

More recently, neural methods propose to train end-to-end EL models. Francis-Landau et al. (2016) propose a convolutional neural EL model to take into account windows of context.

Kolitsas et al. (2018) propose a neural model for joint mention detection and entity disambiguation. They use a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) to encode spans of entities. They then embed candidate entities and train layers to score the likelihood of a match.

Sil et al. (2018) introduce an LSTM-based model that uses multilingual embeddings for zero-shot transfer from English-language knowledge bases.

EL as Language Modeling. Language modeling approaches have enabled new, end-to-end definitions of the entity linking task. These new settings enable to bypass the two-step MD-then-ED setting for entity linking, and propose to cast entity linking as a single task.

Broscheit (2019) propose to reformulate end-to-end EL problem as a token-wise classification over the entire set of the vocabulary. Their model is based on BERT (Devlin et al., 2019). The training combines mention detection, candidate generation, and entity disambiguation. If an entity is not detected, then the prediction is *O*. If an entity is detected, the classification head has to classify it as the corresponding particular entity within the vocabulary.

De Cao et al. (2021b) propose an autoregressive setting for EL. They use BART (Lewis et al., 2020) and cast entity linking as a language generation task. In this setting, the input is the source sentence with the entity mention. The goal is to generate an annotated version of the input sentence, such that the entity mention is highlighted and mapped to a knowledge base concept. Brackets and parentheses are used to annotate the entity mention and concept: "*I took the [flu shot] (influenza vaccine).*". They then introduce a constrained beam search to force the model to annotate. De Cao et al. (2021c) is a multilingual extension of this work.

EL as Embedding Space Mapping. Language models like BERT, as well as embedding models

like FastText (Bojanowski et al., 2017), enable to retrieve context-aware representations of entities and knowledge base concepts.

Basaldella et al. (2020) propose to map the embeddings of entity mentions to the embeddings of knowledge base concepts. For this purpose, they use the embeddings of FastText, as well as BioBERT (Lee et al., 2020), a BERT-based model trained on the PMC dataset. They find that the right mapping is more often found among the ten closest concept embeddings (accuracy at 10) rather than being the closest concept embedding (accuracy at 1). Their results suggest that generated sample re-ranking could improve entity linking systems.

Basaldella et al. (2020) also introduce the COMETA dataset: an entity linking benchmark based on social media user utterances on medical topics, and linked to the SNOMED-CT biomedical knowledge base (Donnelly et al., 2006). The dataset has four splits, based on whether the dev/test set entities are seen during training (stratified) or not (zeroshot), and on whether the entity mapping is context-specific (specific) or not (general). Liu et al. (2021a) propose a self-alignment pre-training scheme for entity embeddings, and show that it benefits the context-free splits (stratified general and zeroshot general). Liu et al. (2021b) propose MirrorBERT: a data-augmented approach for masked language models. Lai et al. (2021) and Kong et al. (2021) propose convolution-based and graph-based methods, respectively, for embedding mapping between entities and knowledge base concepts.

All of the above methods use knowledge base concepts. In our biomedical entity linking setting, we choose the harder zeroshot specific split. We propose to use the language modeling task setting instead of the embedding mapping method. We therefore bypass the need to embed each and every knowledge base concept, whereas only a small portion (<10%) of the SNOMED-CT knowledge base concepts are used in the COMETA dataset.

3 Multi-Task Learning for Autoregressive Entity Linking

We propose an autoregressive entity linking model, that is trained along with two auxiliary tasks, and uses re-ranking at inference time.

In this section, we first describe the main entity linking task. Then, we define the two auxiliary tasks: Mention Detection and a new task, called *Match Prediction*. Third, we train our multi-task learning architecture with a weighted objective. Finally, we propose to use the match prediction module for re-ranking during inference. An overview of our architecture is in Figure 2.

3.1 Autoregressive Entity Linking

We train autoregressive entity linking as a language generation task. We follow the setting of the encoder-decoder model of De Cao et al. (2021b). They train their model to generate the input sentence containing both the entity mention *and* the target entity, annotated with parentheses and brackets. For simplicity, we omit these annotations from the examples in the figures.

For entity linking (EL), we optimize the following negative log-likelihood loss:

$$\mathcal{L}_{\text{EL}} = -\sum_{i=1}^{N} \log P(y_i | y_1, ..., y_{i-1}, \mathbf{x}) \quad (1)$$

where $\mathbf{x}$ is the input sentence, and $\mathbf{y}$ is the output sentence of length N.

3.2 Entity Mention Detection

The first auxiliary task is mention detection (MD). The goal of this task is to teach the model to distinguish tokens that are part of entities from tokens that are not part of any entity. As a result, this task is in essence a token-wise binary classification task. This setting is similar to semantic role labeling (Carreras and Màrquez, 2005) or named entity recognition. Broscheit (2019) propose a similar task definition, but combine entity detection with entity disambiguation. Their task definition is a classification task over the entire knowledge base vocabulary, rather than our binary setting.

In this task, we train the model to predict where the tokens of the entities are in the input sentence and in the target (annotated) sentence. Therefore, this auxiliary task has to output two sequences of entity indicators: "*E*" for entity mention or concept tokens, and "*O*" for all other tokens. To train our model to generate sequences for the input and target sentences, we augment our existing dataset. We create two datasets of the same size: the first has sequences of entity indicators for the input sentences, and the second has sequences of entity indicators for the target sentences.

As shown at the left of Figure 2, we use two different tagging heads for mention detection: one for

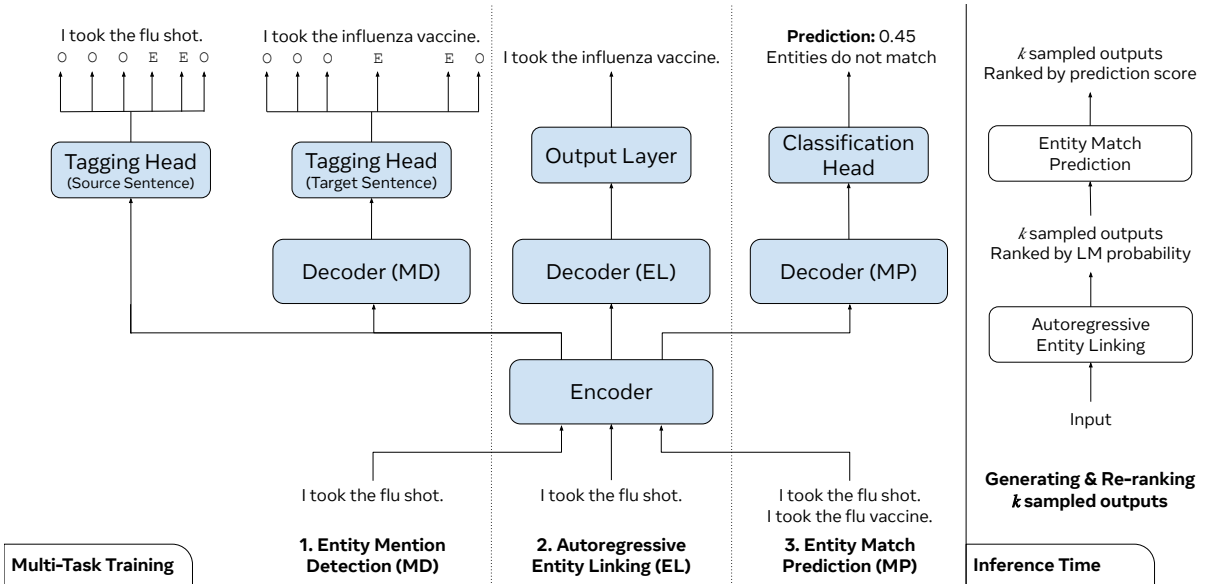


Figure 2: Architecture of our proposed multi-task autoregressive entity linking model. Each task is trained using a shared encoder and a task-specific decoder and output layer. The auxiliary mention detection task uses datasets derived from one entity linking dataset, whereas the match prediction task uses sampled outputs. At inference time, we use the match prediction module to re-rank generated samples.

the input sentence, and one for the output sentence. We use two tagging heads as the model learns different mappings from two different kinds of input. For the input sentence, we feed the encoder embeddings to the first tagging head. We cast this as a classification problem. For mention detection on the output sentence, we use a separate decoder, and feed this decoder's embeddings to the second tagging head. We cast this task as a generation task. For both tasks, we optimize a cross entropy (CE) loss. In summary, we optimize the following loss function for mention detection (MD):

$$\begin{aligned}\mathcal{L}_{\text{MD}} =& \text{CE}\left(Enc(\mathbf{x}), Ent(\mathbf{x})\right) \\ &+ \text{CE}\left(Dec(Enc(\mathbf{x})), Ent(\mathbf{y})\right)\end{aligned} \tag{2}$$

where $Enc(\cdot)$ is the encoder representation, $Dec(\cdot)$ is the decoder representation, and $Ent(\cdot)$ indicates the corresponding sequence of entity indicators.

3.3 Entity Match Prediction

In their biomedical entity linking experiments using word embedding space mapping, Basaldella et al. (2020) find that accuracy at 10 is often more than double the accuracy at 1. They then suggest that re-ranking could significantly improve performance. We build on this observation to introduce the second auxiliary task: entity match prediction (MP). The goal of this task is to teach the model to re-rank generated samples based on the input sentence, with the aim to help narrow the gap with the accuracy at 10 scores.

The input to this task is composed of two sentences: the first one is the input sentence, and the second is a sentence where entity mentions are replaced by entities that may or may not be the matching target entities. We train the model to predict whether the entities match (score of 1) or not (score of 0) between both sentences.

At regular intervals during training, we generate k samples for each input sentence using beam search on the autoregressive entity linking part of the trained model. We then form k sentence pairs. The corresponding ground truth label for a given sentence pair indicates whether the entities match or not. This data generation setting exposes the model to its own successes and failures in the main entity linking task.

It may be the case that no generated sample contains entities that match the input sentence, and therefore that all labels for a pair are 0. In this case, the model would not be shown what an example of matching entities looks like. To mitigate this issue, we decide to add one additional sentence pair, where the second sentence is the target sentence used in the autoregressive entity linking training. We add this additional sentence pair to all datapoints for consistency.

We train entity match prediction using a mean squared error loss:

$$\mathcal{L}_{\text{MP}} = \left(P^{\text{MP}}(\hat{\mathbf{y}}|\mathbf{x}) - 1\right)^2 + \sum_{i=1}^{k} \left(P^{\text{MP}}(\mathbf{y}_i^s|\mathbf{x}) - \hat{y}_i^{\text{MP}}\right)^2 \quad (3)$$

where $\hat{\mathbf{y}}$ is the target sentence, $\mathbf{y}_i^s$ is the i-th generated sample, $P^{\text{MP}}(\cdot|\cdot)$ is the probability that the entities in the left-hand sequence match the ones in the right-hand sequence, and $\hat{y}_i^{\text{MP}}$ is the ground truth label for entity match prediction for the i-th generated sample.

De Cao et al. (2021a) propose to rank candidate concepts from a predefined set after the detecting entity mentions. In our case, we do not learn to rank predefined sets of candidates, nor do we rank concepts. Instead, we generate sentences using beam search, and propose to learn to re-rank them.

3.4 Multi-Task Learning

We propose to optimize simultaneously for all three tasks using a single loss function. We set one weight for each auxiliary task. We discuss the task weight hyperparameter tuning in §4.3.

Given the losses defined in equations 1, 2, and 3, our loss function for multi-task learning is as follows:

$$\mathcal{L}_{\text{MTL}} = \mathcal{L}_{\text{EL}} + \lambda_{\text{MD}}\mathcal{L}_{\text{MD}} + \lambda_{\text{MP}}\mathcal{L}_{\text{MP}} \quad (4)$$

where λ_{MD} and λ_{MP} are the auxiliary task weights for mention detection and match prediction, respectively.

As shown in Figure 2, we use three separate decoders for training: one for each task. We use two separate tagging heads for mention detection. For the match prediction task, we feed the last decoder output to the classification head. This follows the training scheme of BART (Lewis et al., 2020) for sentence classification tasks.

Our model architecture is inspired by MT-DNN (Liu et al., 2019), a multi-task model that obtained state-of-the-art results across many NLP tasks involving sentence representation. In the MT-DNN architecture, the encoder is shared across tasks, and prediction heads are task-specific. Nonetheless, other multi-task architectures remain compatible with our auxiliary tasks and re-ranking, which are the novelties we focus on in this work.

	AIDA-CoNLL		COMETA
Split	**Documents**	**Mentions**	**Mentions**
Train	942	18,540	13,714
Dev	216	4,791	2,018
Test	230	4,485	4,283

Table 1: Statistics of Entity Linking benchmark datasets.

3.5 Inference-time Re-ranking

In order to bridge some of the gap between accuracy at 1 and accuracy at 10 (Basaldella et al., 2020), we propose to use the entity match prediction module to re-rank generated samples. The right side of Figure 2 illustrates the process.

At inference time, we first generate k samples ranked by their language modeling probability. We then use the separate entity match prediction (MP) decoder to predict an entity match probability. To do so, we input the source sentence and a generated sample to the MP decoder. We use the resulting MP probabilities to re-rank the k generated samples. We select the sample with the highest MP probability to compute the evaluation metrics.

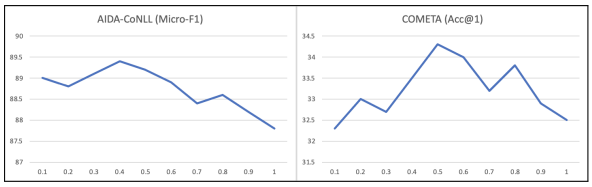
4 Experiments

4.1 Datasets and Setup

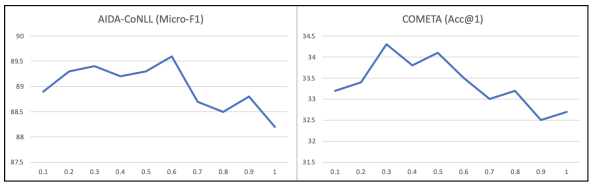
We use two benchmark datasets for English-language entity linking. We use the standard data splits for both datasets, as detailed in Table 1.

AIDA-CoNLL (Hoffart et al., 2011) is a dataset consisting of annotated news articles from the Reuters Corpus (Lewis et al., 2004). The knowledge base concepts come from the titles of the English-language Wikipedia. Each news article contains multiple entity mentions. Articles are sometimes too long for the maximum sequence length of our model. We follow De Cao et al. (2021a) and cut the articles into separate chunks. We use the Micro-F1 metric for evaluation. We only evaluate mentions present in the knowledge base, following the *In-KB* setting (Röder et al., 2018), in line with previous work (De Cao et al., 2021b,a). This dataset contains candidates for each entity mention. Our method does not use these entity candidates, although several baselines do (Kolitsas et al., 2018; Martins et al., 2019; De Cao et al., 2021a).

COMETA (Basaldella et al., 2020) is a dataset of biomedical entity mentions from social media (Reddit) utterances. In this dataset, each user-written



(a) Choosing the optimal λ_{MD}, setting $\lambda_{\text{MP}} = 0.3$.



(b) Choosing the optimal λ_{MP}, given the optimal λ_{MD}.

Figure 3: Task weight tuning on the dev set for Mention Detection (MD) and Match Prediction (MP). We first optimize for λ_{MD} (a), and then λ_{MP} (b).

MD	MP	Rk	AIDA-CoNLL Micro-F1	COMETA Acc@1
Ablation of Auxiliary Tasks and Re-ranking				
✗	✗	✗	86.4	31.2
Ablation of Auxiliary Tasks				
✓	✗	✗	87.5	31.9
✗	✓	✓	88.8	34.1
Ablation of Re-ranking				
✓	✓	✗	87.8	32.8
MD, MP and Re-ranking (Ours)				
✓	✓	✓	89.6	34.3

Table 2: Results of the ablation studies on the dev sets. We perform ablation studies on Mention Detection (**MD**), Match Prediction (**MP**), and the re-ranking of generated samples (**Rk**).

utterance contains exactly one entity mention. The metric used to evaluate this dataset is accuracy at 1 (Acc@1). We measure Acc@1 by checking whether the correct knowledge base concept is present in the top generated sample. We use the zeroshot specific split, where the entity mention and disambiguation pairs in the test set are not seen during training, and the entity linking is context-specific.

4.2 Training Details

We use BART Large (Lewis et al., 2020) as our base model. We use three decoders, all initialized from the same checkpoint decoder. We train for 100 epochs on AIDA-CoNLL, and for 10 epochs on COMETA. We use the same model checkpoint as De Cao et al. (2021b), which is trained on an English Wikipedia dataset for entity linking. We generate $k = 10$ samples for the Match Prediction training and validation, as well as for inference.

4.3 Task Weight Tuning

For each dataset, we optimize the auxiliary task weights λ_{MD} for mention detection, and λ_{MP} for match prediction. We select these hyperparameters based on the highest performance in Micro-F1 (AIDA-CoNLL) or accuracy at 1 (COMETA) on the dev set.

We trial all values from 0.1 to 1.0 with 0.1 increments, for both task weights. We start by optimizing λ_{MD} given $\lambda_{\text{MP}} = 0.3$, and then optimize λ_{MP} given the optimal λ_{MD} weights. The results are in Figure 3. The graphs show that performance on the main entity linking task can vary visibly when the weights of the auxiliary tasks change. Moreover, the optimal task weights are different for every dataset and domain: we find that the optimal auxiliary task weights are $\lambda_{\text{MD}} = 0.4$ and $\lambda_{\text{MP}} = 0.6$ for AIDA-CoNLL, and $\lambda_{\text{MD}} = 0.5$ and $\lambda_{\text{MP}} = 0.3$ for COMETA. We use these task weights for the next experiments.

4.4 Ablation Studies

We perform two types of ablation studies to analyze the added value of our novelties. First, we evaluate how do the two auxiliary tasks and the re-ranking impact entity linking performance. Second, we implement a low-resource scenario for the auxiliary tasks, as we ask whether the main task benefits more from the knowledge learned the auxiliary tasks, or from the additional training data.

Auxiliary Tasks and Re-ranking. Our main novelties are multi-task learning with mention detection and match prediction, and the re-ranking of generated samples at inference time. The auxiliary tasks aim to explicitly teach the model how to detect mentions of entities, and how to predict whether entities were correctly disambiguated given an input sentence and a generated sample.

We perform ablation studies to gauge the added value of each task and re-ranking. We perform four additional experiments, keeping the same number of model parameters. First, we perform an ablation of both auxiliary tasks and the re-ranking, by setting $\lambda_{\text{MD}} = 0.0$ and $\lambda_{\text{MP}} = 0.0$, and not changing the order of the generated samples. Second, we remove the match prediction training objective ($\lambda_{\text{MP}} = 0.0$), and therefore also remove the re-ranking, but we keep the optimally weighted mention detection objective. Third, we remove

the mention detection training objective by setting $\lambda_{\text{MD}} = 0.0$, but we keep the optimally weighted mention prediction objective, along with the re-ranking. Finally, we keep both optimally weighted auxiliary tasks, but remove the inference-time re-ranking of generated samples.

We show the results of all ablation experiments on the dev sets in Table 2. We notice that the lowest scores are obtained when both auxiliary tasks and re-ranking are ablated. This shows the added value of all of our main novelties on the main entity linking task. In addition, each auxiliary task individually increases performance, as shown on the second and third row of results. The auxiliary match prediction task along with re-ranking provide a larger performance increase than the auxiliary mention detection task alone. This could be due to the fact that the match prediction task gets a larger number of samples to train on. Finally, the difference in performance between our model and the re-ranking ablation study shows that re-ranking of generated samples is an important contribution to the final performance. This result backs the suggestion of Basaldella et al. (2020) that re-ranking can bridge some of the gap between accuracy at 1 and accuracy at 10.

Impact of additional training data. In this subsection, we ask whether the main task benefits more from the knowledge learned by the auxiliary tasks, or from the large sizes of the auxiliary task datasets. The mention detection task has two datapoints for every entity linking datapoint, while the match prediction task has $k + 1 = 11$ datapoints for every entity linking datapoint. Therefore, in a given training epoch, there are more datapoints to train on for the auxiliary tasks in comparison with the main task.

We devise three experiments to gauge whether a lower amount of training datapoints for auxiliary tasks impacts the main task results. We propose a low-resource regimen of training for auxiliary tasks, such that we bring the ratio of training datapoints down to 1:1 between the auxiliary tasks and the main task. We train on one out of every two MD datapoints, and on one of out every 11 MP datapoints. In other words, we skip 50% of the training data of the MD task, and 91% of the training data of the MP task. We spread out the input such that, at each training step, the model sees one EL input sentence, one MD input sentence, and one MP input sentence pair. In each epoch, we skip the same datapoints so that the model only sees a reduced number of training datapoints.

% of Train Set		AIDA-CoNLL	COMETA
MD	**MP**	**Micro-F1**	**Acc@1**
Ablation of Auxiliary Tasks and Re-ranking			
0%	0%	86.4	31.2
Low-Resource Experiments			
50%	9%	88.5	34.0
50%	0%	89.3	33.4
0%	9%	88.5	33.8
No Low-Resource (Ours)			
100%	100%	89.6	34.3

Table 3: Results on the dev sets of the low-resource experiments. We reduce the training datasets of the auxiliary mention detection **MD** and match prediction **MP** tasks to gauge whether the main task continues to benefit from multi-task learning. We add the first and last row of results as reference points for comparison.

In the first experiment, we train for both auxiliary tasks on a train set ratio of 1:1 with the main task. In the second and third experiments, we apply the low-resource setting only to the mention detection task, and only to the match prediction task, respectively. In all three experiments, we keep the same selection of skipped datapoints for each task, and we keep re-ranking.

We show the results of the low-resource experiments in Table 3. For reference, we add the results from our model and the model without auxiliary task nor re-ranking from Table 2. The results show that globally, there is a slight decrease in performance when the training set is smaller, compared to our model. However, the low-resource experiments show a significant increase in performance compared to the ablation experiment of the first row. This shows that our proposed method's edge does not only come from the additional training data, but also from our formulation of the auxiliary tasks, and the re-ranking of generated samples.

4.5 Results and Discussion

AIDA-CoNLL. The test results for the AIDA-CoNLL dataset are on Table 4. Our model establishes a new state of the art for this task.

Note that our model is autoregressive and, compared to the state-of-the-art autoregressive model on AIDA-CoNLL De Cao et al. (2021b), our method shows a 2.0-point improvement in Micro-F1 score. This increase shows that our model is able to correct some errors with the re-ranking at

Method	Micro-F1
Hoffart et al. (2011)	72.8
Steinmetz and Sack (2013)	42.3
Daiber et al. (2013)	57.8
Moro et al. (2014)	48.5
Piccinno and Ferragina (2014)	73.0
Kolitsas et al. (2018)	82.4
Peters et al. (2019)	73.7
Broscheit (2019)	79.3
Martins et al. (2019)	81.9
van Hulst et al. (2020)	80.5
Févry et al. (2020)	76.7
Kannan Ravi et al. (2021)	83.1
De Cao et al. (2021a)	85.5
Autoregressive Entity Linking Models	
De Cao et al. (2021b)	83.7
Our model	**85.7**

Table 4: Results on the AIDA-CoNLL test set.

inference time, and that our multi-task setting benefits the main entity linking task.

Our model scores a Micro-F1 0.2 higher than the model of De Cao et al. (2021a). However, De Cao et al. (2021a) use a predefined candidate set of concepts, whereas the autoregressive models – including our own – do not. This shows that our model is able to bypass the knowledge base, and that our method leverages language modeling to gain knowledge of the news domain.

COMETA. There are no pre-defined sets of candidate concepts in the COMETA dataset. In this task, there is a knowledge base of biomedical concepts from which the model can choose. Similarly to our AIDA-CoNLL setting, our model does not use the knowledge base.

We consider three baselines for our biomedical entity linking benchmark. The first baseline is the embedding mapping method of Basaldella et al. (2020). They use BioBERT and a max-margin loss with negative target embeddings. The second baseline is the BERT- and classification-based method of Broscheit (2019). We train this baseline by classifying tokens into the concepts present in the COMETA dataset, as opposed to the entire vocabulary of $350K$ knowledge base concepts. This is for computational purposes, as a $350K$-way classification would be difficult to train. The third baseline is the autoregressive, single-task model of De Cao et al. (2021b). We train this baseline as a reference point for our model.

Method	Acc@1
Basaldella et al. (2020)	27.0
Broscheit (2019)	24.5
Autoregressive Entity Linking Models	
De Cao et al. (2021b)	30.9
Our model	**32.4**

Table 5: Results on the COMETA test set.

The test results of the COMETA dataset experiments are on Table 5. Our model is able to exceed over five percentage points the baselines that use the knowledge base concepts. This shows that our method can efficiently generalize without the need for a knowledge base, but only through learning about the biomedical domain. Note that we use the zeroshot specific split here, where the entity mention and disambiguation pairs in the test set are not seen during training. Moreover, our model exceeds the autoregressive single-task baseline by 1.5%. This increase shows that our multi-task setting and re-ranking can generalize, and increase performance under zeroshot settings.

5 Conclusions

We propose a multi-task learning and re-ranking approach to autoregressive entity linking. Our main two novelties address two weaknesses in the literature. First, whereas autoregressive entity linking can increase performance, mention detection is only learned implicitly. We propose to cast this problem as a language generation task while explicitly teaching the model how to detect entity mentions. Second, previous work suggests that a sizeable portion of errors could be corrected with re-ranking. We propose to use samples generated at training time to teach the model to re-rank outputs.

We devise four ablation study experiments, and show that our model benefits from both auxiliary tasks and re-ranking. In particular, we show that re-ranking plays a major role in increasing entity linking scores. Then, we propose three low-resource experiments for auxiliary tasks. The results show that our model's performance is not only due to additional training datapoints, but also due to how we defined our auxiliary tasks. Finally, our model establishes a new state of the art in both COMETA and AIDA-CoNLL. The increases in performance across both datasets show that our model can learn and leverage domain-specific knowledge, without using a candidate set or a knowledge base.

References

Marco Basaldella, Fangyu Liu, Ehsan Shareghi, and Nigel Collier. 2020. Cometa: A corpus for medical entity linking in the social media. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3122–3137.

Pierpaolo Basile, Valerio Basile, Malvina Nissim, and Nicole Novielli. 2015. Deep tweets: from entity linking to sentiment analysis. In *Proceedings of the Italian Computational Linguistics Conference (CLiC-it 2015)*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

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

Xavier Carreras and Lluís Màrquez. 2005. Introduction to the conll-2005 shared task: Semantic role labeling. In *Proceedings of the ninth conference on computational natural language learning (CoNLL-2005)*, pages 152–164.

Marco Cornolti, Paolo Ferragina, Massimiliano Ciaramita, Stefan Rüd, and Hinrich Schütze. 2016. A piggyback system for joint entity mention detection and linking in web queries. In *Proceedings of the 25th International Conference on World Wide Web*, pages 567–578.

Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N Mendes. 2013. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems*, pages 121–124.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021a. Highly parallel autoregressive entity linking with discriminative correction. *arXiv preprint arXiv:2109.03792*.

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

Nicola De Cao, Ledell Wu, Kashyap Popat, Mikel Artetxe, Naman Goyal, Mikhail Plekhanov, Luke Zettlemoyer, Nicola Cancedda, Sebastian Riedel, and Fabio Petroni. 2021c. Multilingual autoregressive entity linking. *arXiv preprint arXiv:2103.12528*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Kevin Donnelly et al. 2006. Snomed-ct: The advanced terminology and coding system for ehealth. *Studies in health technology and informatics*, 121:279.

Mohnish Dubey, Debayan Banerjee, Debanjan Chaudhuri, and Jens Lehmann. 2018. Earl: joint entity and relation linking for question answering over knowledge graphs. In *International Semantic Web Conference*, pages 108–126. Springer.

Cheikh Brahim El Vaigh, François Torregrossa, Robin Allesiardo, Guillaume Gravier, and Pascale Sébillot. 2020. A correlation-based entity embedding approach for robust entity linking. In *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 949–954. IEEE.

Thibault Févry, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, and Tom Kwiatkowski. 2020. Entities as experts: Sparse memory access with entity supervision. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4937–4951, Online. Association for Computational Linguistics.

Matthew Francis-Landau, Greg Durrett, and Dan Klein. 2016. Capturing semantic similarity for entity linking with convolutional neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1256–1261.

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.

Stephen Guo, Ming-Wei Chang, and Emre Kiciman. 2013. To link or not to link? a study on end-to-end tweet entity linking. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1020–1030.

Xianpei Han, Le Sun, and Jun Zhao. 2011. Collective entity linking in web text: a graph-based method. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 765–774.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

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.

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

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.

Luyang Kong, Christopher Winestock, and Parminder Bhatia. 2021. Zero-shot medical entity retrieval without annotation: Learning from rich knowledge graph semantics. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2401–2405, Online. Association for Computational Linguistics.

Tuan Lai, Heng Ji, and ChengXiang Zhai. 2021. Bert might be overkill: A tiny but effective biomedical entity linker based on residual convolutional neural networks. *arXiv preprint arXiv:2109.02237*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

David D Lewis, Yiming Yang, Tony Russell-Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Fangyu Liu, Ehsan Shareghi, Zaiqiao Meng, Marco Basaldella, and Nigel Collier. 2021a. Self-alignment pretraining for biomedical entity representations. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4228–4238.

Fangyu Liu, Ivan Vulić, Anna Korhonen, and Nigel Collier. 2021b. Fast, effective and self-supervised: Transforming masked languagemodels into universal lexical and sentence encoders. *arXiv preprint arXiv:2104.08027*.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496.

Xiaohua Liu, Yitong Li, Haocheng Wu, Ming Zhou, Furu Wei, and Yi Lu. 2013. Entity linking for tweets. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1304–1311.

Gang Luo, Xiaojiang Huang, Chin-Yew Lin, and Zaiqing Nie. 2015. Joint entity recognition and disambiguation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 879–888.

Pedro Henrique Martins, Zita Marinho, and André FT Martins. 2019. Joint learning of named entity recognition and entity linking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 190–196.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity Linking meets Word Sense Disambiguation: a Unified Approach. *Transactions of the Association for Computational Linguistics*, 2:231–244.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54, Hong Kong, China. Association for Computational Linguistics.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, et al. 2021. Kilt: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544.

Francesco Piccinno and Paolo Ferragina. 2014. From tagme to wat: a new entity annotator. In *Proceedings of the first international workshop on Entity recognition & disambiguation*, pages 55–62.

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2020. E-bert: Efficient-yet-effective entity embeddings for bert. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 803–818.

Amir Pouran Ben Veyseh. 2016. Cross-lingual question answering using common semantic space. In *Proceedings of TextGraphs-10: the workshop on graph-based methods for natural language processing*, pages 15–19.

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 99–110.

Michael Röder, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. 2018. Gerbil–benchmarking named entity recognition and linking consistently. *Semantic Web*, 9(5):605–625.

Philipp Scharpf, Moritz Schubotz, and Bela Gipp. 2021. Towards explaining stem document classification using mathematical entity linking. *arXiv preprint arXiv:2109.00954*.

Sanket Shah, Anand Mishra, Naganand Yadati, and Partha Pratim Talukdar. 2019. Kvqa: Knowledge-aware visual question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8876–8884.

Avirup Sil, Gourab Kundu, Radu Florian, and Wael Hamza. 2018. Neural cross-lingual entity linking. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Nadine Steinmetz and Harald Sack. 2013. Semantic multimedia information retrieval based on contextual descriptions. In *Extended Semantic Web Conference*, pages 382–396. Springer.

Tolga Uslu, Alexander Mehler, Daniel Baumartz, and Wahed Hemati. 2018. fastsense: An efficient word sense disambiguation classifier. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Johannes M van Hulst, Faegheh Hasibi, Koen Dercksen, Krisztian Balog, and Arjen P de Vries. 2020. Rel: An entity linker standing on the shoulders of giants. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2197–2200.

Jörg Waitelonis and Harald Sack. 2016. Named entity linking in# tweets with kea. In *# Microposts*, pages 61–63.

Ikuya Yamada, Hideaki Takeda, and Yoshiyasu Takefuji. 2015. An end-to-end entity linking approach for tweets. In *5th Workshop on Making Sense of Microposts: Big Things Come in Small Packages,# Microposts 2015, at the 24th International Conference on the World Wide Web, WWW 2015*, pages 55–56. CEUR-WS.

Mo Yu, Wenpeng Yin, Kazi Saidul Hasan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2017. Improved neural relation detection for knowledge base question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 571–581.

Wei Zhang, Jian Su, Chew Lim Tan, and Wen Ting Wang. 2010. Entity linking leveraging automatically generated annotation. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1290–1298.

A Additional Training Details

Following De Cao et al. (2021b), we use constrained beam search to force the model to annotate mentions and concepts. We set the learning rate at $3 \cdot 10^{-5}$. We use an Adam optimizer where the betas are 0.9 and 0.999. The Adam epsilon is 10^{-8}, and the dropout is 0.1. The total number of parameters for our multi-task model is 915 million. We select the best model based on the lowest loss value on the dev set. We generate samples at every training epoch for COMETA, and at every 2 epochs for AIDA-CoNLL.

B Reproducibility Details

We will open-source the code and trained models along with the camera-ready version.

For training, we use 8 GPUs of 32GB each. The average runtime per training epoch is 3 minutes for COMETA, and 12 minutes for AIDA-CoNLL. All validation results for the reported test results of our best-performing models are in Table 3 of the main paper. We use Micro-F1 as metric for AIDA-CoNLL, and accuracy at 1 for COMETA. We describe the hyperparameter search for multi-tasking in §4.3.

We report the statistics about the dataset in Table 1. We did not exclude any data. For COMETA, one has to email Prof. Nigel Collier (`nhc30@cam.ac.uk`) to get the dataset. For AIDA-CoNLL, we use the pre-processed dataset available at this link: `https://mega.nz/folder/l4RhnIxL#_oYvidq2qyDIw1sT-KeMQA`.

For the low-resource experiments, we form the low-resource datasets as follows. At the very beginning of the experiment, we select which datapoints we will omit, and which ones we will train with. For our multi-task setting, we train at each step on one EL input sentence, two MD input sentences (technically one, for which we produce two outputs), and $k + 1 = 11$ MP input sentence pairs. In contrast, for our low-resource setting, we train at each step with one input for each task. For Match Prediction, we truncate the last few datapoints in the low-resource setting, as the sizes of the training sets in both datasets are not dividable by 11.