EnCore: Fine-Grained Entity Typing by Pre-Training Entity Encoders on Coreference Chains

Frank Mtumbuka and **Steven Schockaert**
Cardiff University, UK
{MtumbukaF,SchockaertS1}@cardiff.ac.uk

Abstract

Entity typing is the task of assigning semantic types to the entities that are mentioned in a text. In the case of fine-grained entity typing (FET), a large set of candidate type labels is considered. Since obtaining sufficient amounts of manual annotations is then prohibitively expensive, FET models are typically trained using distant supervision. In this paper, we propose to improve on this process by pre-training an entity encoder such that embeddings of coreferring entities are more similar to each other than to the embeddings of other entities. The main problem with this strategy, which helps to explain why it has not previously been considered, is that predicted coreference links are often too noisy. We show that this problem can be addressed by using a simple trick: we only consider coreference links that are predicted by two different off-the-shelf systems. With this prudent use of coreference links, our pre-training strategy allows us to improve the state-of-the-art in benchmarks on fine-grained entity typing, as well as traditional entity extraction.

1 Introduction

Entity typing is a fundamental task in Natural Language Processing (NLP), with important applications to entity linking (Onoe and Durrett, 2020) and relation extraction (Peng et al., 2020; Zhong and Chen, 2021), among others. In recent years, the main focus has been on fine-grained entity typing (Ling and Weld, 2012; Gillick et al., 2014), where around 100 different entity types are considered, or even ultra-fine entity typing (Choi et al., 2018), where around 10000 types are considered. A key challenge then consists in compiling enough training data. This is particularly problematic because the distribution of entity types is highly skewed, with many types occurring only rarely in text. The main strategy thus far has been to create automatically labelled training sets. For instance, Ling and Weld (2012) relied on the fact that entity mentions in Wikipedia are linked to the article of the corresponding entity, which is in turn linked to Freebase (Bollacker et al., 2008). Entity mentions in Wikipedia can thus be linked to their Freebase types without any manual effort. However, these distantly supervised training sets are still highly skewed. As a result, models trained on such datasets may concentrate more on learning to recognise the most prevalent entity types than on deriving meaningful entity representations (i.e. embeddings which accurately capture semantic types of entities).

For this reason, we propose to first train a general-purpose entity encoder, which maps entity mentions to meaningful embeddings, independent of a particular label set. We can then train an entity type classifier in the usual way, using the embeddings from our encoder as input. Our approach relies on a supervision signal that has thus far remained largely unexplored for entity typing: coreference chains. In particular, we train an entity encoder with contrastive learning to represent coreferring entity mentions close to each other in the embedding space. While conceptually straightforward, this training signal forces the entity encoder to identify subtle cues in the context of an entity mention, to characterise the entity at a level which is sufficiently fine-grained to distinguish it from other entities. Our strategy only need access to an off-the-shelf coreference resolution system. This means that we can train the entity encoder on different genres of text and generate as much training data as is needed.

Figure 1 illustrates the three main steps of our approach. In the first step, an off-the-shelf coreference resolution system is applied to a large collection of stories. Second, we use contrastive learning to train an entity encoder, which maps mentions from the same coreference chain to similar vectors, while mentions from different chains are mapped to dissimilar vectors. In the third step, to learn a fine-grained entity typing model, we simply train a

Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 1768–1781

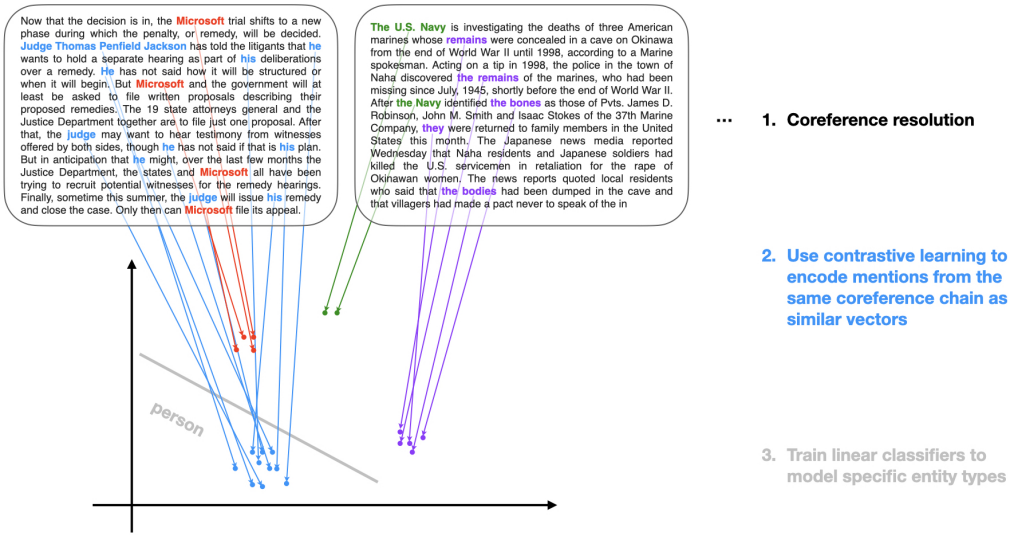

Figure 1: Illustration of our proposed strategy. In the first step, an off-the-shelf coreference resolution method is used to identify coreference chains in stories. In the second step, we use contrastive learning to train an encoder which maps mentions from the same coreference chain to similar vectors. In the third step, we use standard training data to learn a linear classifier for each considered entity type.

linear classifier in the resulting embedding space for each considered entity type.

An important challenge in implementing the proposed strategy is that coreference resolution systems are still far from perfect. Whenever two mentions are erroneously assumed to be referring to the same entity, the entity encoder is trained on a noisy signal, which has a detrimental impact on the overall performance of the method. In our experiments, we found that the success of our strategy indeed strongly depends on the quality of the coreference resolution system that is used. In fact, our best results are obtained by using two different systems, and only keeping coreference links that are predicted by both. When adopting this strategy, our model outperforms the current state-of-the-art in three entity typing benchmarks.

2 Related Work

Entity Typing The standard approach to entity typing is to use a fine-tuned Language Model (LM) of the BERT family (Devlin et al., 2019) to obtain embeddings of entity mentions (Zhong and Chen, 2021; Ye et al., 2022) and then train a standard classifier on top of these embeddings. Some alternative strategies have also been explored. For instance, Li et al. (2022a) cast the problem of entity typing as a natural language inference (NLI) problem. The main drawback of the NLI approach is that it requires individual testing for every entity type, making it highly inefficient for fine-grained entity typing. Large Language Models (LLMs) are similarly impractical to use in most application settings. Even when disregarding efficiency concerns, the impact of LLMs on the task of entity typing has thus far been limited (Han et al., 2023). The most successful approaches use a form of multi-task fine-tuning to adapt LLMs to information extraction tasks, but they still fail to consistently outperform BERT (Wang et al., 2023).

Fine-grained Entity Typing Most work on fine-grained entity typing uses distant supervision of some kind. As already mentioned in the introduction, one strategy is to rely on Wikipedia links in combination with an external knowledge base (Ling and Weld, 2012). A common problem with distantly supervised datasets is that they can be noisy: the fact that an entity has a particular type does not necessarily imply that this information is expressed in a given sentence mentioning that entity. To address this issue, several authors have proposed strategies for denoising distantly supervised datasets for entity typing (Ren et al., 2016; Onoe and Durrett, 2019; Pan et al., 2022). Given that two sentences referring to the same entity may emphasise different elements, a similar problem can also arise in our case. For example, we might have a sentence referring to Ben Affleck as an actor and another referring to him as a director. As the embedding of an entity mention should capture the semantic type that is represented in the relevant sentence context, using such sentence pairs will

confuse the model. We may anticipate that such instances will be rare, however, as we only take into account co-referring entity mentions that originate from the same story. Another possible source of noise comes from mistakes that are made by the coreference resolution system. This effect will be analysed in Section 4.

Pre-training Entity Encoders Previous work has already explored a number of pre-training strategies for learning entity representations. First, methods such as SpanBERT (Joshi et al., 2020) focus on learning better representations of text spans. Within this class of methods, strategies that rely on InfoNCE have also been considered (Wang et al., 2020). While our method also uses InfoNCE, the training signal is fundamentally different: the aforementioned methods focus on learning span representations, using tasks such as reconstructing the correct order of tokens in shuffled text spans. Such models have not proven superior to the standard BERT model for entity typing. In our experiments, we also found that modelling text spans is not essential for entity typing, as our best configuration simply uses the embedding of the head token of an entity span (see Section 4.2). Another line of work, which includes models such as ERNIE (Zhang et al., 2019), KnowBERT (Peters et al., 2019), LUKE (Yamada et al., 2020), KEPLER (Wang et al., 2021c) and K-Adapter (Wang et al., 2021a), improve LMs by modelling entities as separate tokens and leveraging information from knowledge graphs. The main focus of these models is to improve the amount of factual knowledge that is captured, rather than on learning the representations of (possibly) previously unseen entities.

Our approach also has some similarities with the matching-the-blanks model for relation extraction (Baldini Soares et al., 2019). The idea of this model is to learn a label-independent relation encoder, similar to how we are learning a label-independent entity encoder. In their case, the supervision signal comes from the idea that sentences mentioning the same pair of entities are likely to express the same relationship, hence the relation embeddings obtained from such sentences should be similar. Building on this approach, a number of authors have recently used InfoNCE to encode similar ideas (Han et al., 2021; Wan et al., 2022; Wang et al., 2022). Varkel and Globerson (2020) use a contrastive loss to pre-train a mention encoder for coreference resolution based on two heuristics: (i) if the same name appears multiple times in a document, the corresponding embeddings should be similar and (ii) the mention encoder should be able to reconstruct masked pronouns. The usefulness of contrastive learning for pre-training BERT encoders has also been observed more generally, for instance for learning sentence, phrase and word embeddings (Gao et al., 2021; Liu et al., 2021a,b; Wang et al., 2021b; Li et al., 2022b).

Leveraging Coreference Chains To the best of our knowledge, the idea of pre-training an entity encoder based on coreference chains has not yet been considered. However, a number of authors have proposed multi-task learning frameworks in which coreference resolution and entity typing are jointly learned, along with other tasks such as relation and event extraction (Luan et al., 2018; Wadden et al., 2019). Surprisingly, perhaps, such approaches have failed to outperform simpler entity typing (and relation extraction) models (Zhong and Chen, 2021).

3 Our Approach

In Section 3.1, we first discuss the basic entity typing model that we rely on in this paper. Section 3.2 subsequently describes our proposed pre-training strategy based on coreference chains.

3.1 Entity Typing

Let us assume that we are given a sentence in which some entity mentions are highlighted, e.g.:

> *[Alice] was unsure what was wrong with [the patient in front of her].*

Our aim is to assign (possibly fine-grained) semantic types to these entity mentions. For instance, using the FIGER (Ling and Weld, 2012) taxonomy, the first mention should be assigned the types *Person* and *Doctor*, while the second mention should be assigned *Person*. To make such predictions, a given entity mention e in sentence s is first mapped to an embedding $\mathsf{Enc}(s, e) \in \mathbb{R}^n$ using an encoder. For the experiments in our paper, this encoder takes the form of a language model from the BERT family (Devlin et al., 2019). Specifically, we use the final-layer embedding of the head word of the given entity span as the representation of the mentioned entity. For instance, for the second mention in the aforementioned example, *the patient in front of her*, we use the embedding of the head word, *patient*, as the representation of the entity span. This is motivated by the fact that the head word is most likely to

reflect the semantic type of the entity (Choi et al., 2018). We find the head word using the SpaCy dependency parser[1].

We pre-train the entity encoder Enc based on coreference chains, as will be explained in Section 3.2. For each entity type t, we learn a vector $\mathbf{a}_t \in \mathbb{R}^n$ and bias term $b_t \in \mathbb{R}$. The probability that the mention m should be assigned the type t is then estimated as:

$$P(t|s,e) = \sigma(\mathbf{a}_t \cdot \mathsf{Enc}(s,e) + b_t) \quad (1)$$

with σ the sigmoid function. This entity type classifier is trained using binary cross-entropy on a standard labelled training set. The encoder Enc is optionally also fine-tuned during this step. When using the classifier for entity typing, we assign all labels whose predicted probability is above 0.5.

3.2 Pre-training the Entity Encoder

To pre-train the entity encoder Enc, we start from a collection of stories (e.g. news stories). Using off-the-shelf coreference resolution systems, we identify mentions within each story that are likely to refer to the same entity. Let us write (s,e) to denote an entity mention e appearing in sentence s. Then we consider the following self-supervision signal: if (s_1,e_1) and (s_2,e_2) are co-referring mentions, then the contextualised representations of e_1 and e_2 should be close to each other in the embedding space. In particular, we use a contrastive loss to encode that the representations of the tokens appearing in e_1 and e_2 should be more similar to each other than to the tokens appearing in the mentions of other entities.

Each mini-batch is constructed from a small set of stories $\{S_1, ..., S_k\}$. Let us write X_i for the set of entity mentions (s,e) in story S_i that belong to some coreference chain. To alleviate the impact of noisy coreference links, we adopt two strategies:

- We only include coreference links that are predicted by two separate coreference resolution systems. This reduces the number of spurious links that are considered.
- As negative examples, we only consider entity mentions from different stories. This prevents us from using entity mentions that refer to the same entity, but were missed by the coreference resolution system.

[1] https://spacy.io/api/dependencyparser

Let us write T_i for the set of tokens of the mentions in X_i. For a given token t, we write $\mathsf{Enc}(t)$ for its contextualised representation. We write $T = T_1 \cup ... \cup T_k$ and $T_{-i} = T \setminus T_i$. For a given token t, we write C_t for the set of tokens that are part of the same coreference chain. The encoder is trained using InfoNCE (van den Oord et al., 2018):

$$\sum_{i=1}^{k} \sum_{t \in T_i} \sum_{t' \in C_t} \log \frac{exp\left(\frac{\cos(\mathsf{Enc}(t),\mathsf{Enc}(t'))}{\tau}\right)}{\sum_{t''} exp\left(\frac{\cos(\mathsf{Enc}(t),\mathsf{Enc}(t''))}{\tau}\right)} \quad (2)$$

where t'' in the denominator ranges over $T_{-i} \cup \{t\}$. The token pairs in the numerator correspond to positive examples, i.e. tokens whose embeddings should be similar, while the denominator ranges over both positive and negative examples. The temperature $\tau > 0$ is a hyper-parameter, which controls how hard the separation between positive and negative examples should be.

Given a mention (s,e), the model can often infer the semantic type of the entity based on the mention span itself. To encourage the model to learn to identify cues in the sentence context, we sometimes mask the entity during training, following existing work on relation extraction (Baldini Soares et al., 2019; Peng et al., 2020). Specifically, for each input $(s,e) \in X$, with 15% probability we replace the head of the entity span by the [MASK] token. Note that, unlike previous work, we only mask the head word of the phrase.

Finally, following Baldini Soares et al. (2019), we also use the Masked Language Modelling objective during training, to prevent catastrophic forgetting. Our overall loss thus becomes:

$$\mathcal{L} = \mathcal{L}_{\text{entity}} + \mathcal{L}_{\text{MLM}}$$

where $\mathcal{L}_{\text{entity}}$ is the loss function defined in (2) and $\mathcal{L}_{\text{MLM}}$ is the masked language modelling objective from BERT (Devlin et al., 2019).

4 Experimental Analysis

In this section, we evaluate the performance of our proposed strategy on (fine-grained) entity typing.[2]

Experimental Setup In all our experiments, we initialise the entity encoder with a pre-trained language model. We consider `bert-base-uncased`[3],

[2] Our implementation and pre-trained models are available at https://github.com/fmtumbuka/EACL_EnCore

[3] https://huggingface.co/docs/transformers/model_doc/bert

Dataset	# Types	Train	Dev.	Test
ACE 2005	7	26.5K	6.4K	5.5K
OntoNotes	89	3.4M	8K	2K
FIGER	113	2M	1K	0.5K

Table 1: Overview of the considered benchmarks, showing the number of entity types, and the number of entity mentions in the training, development and test sets.

`albert-xxlarge-v1`[4] and `roberta-large`[5] for this purpose, as these are commonly used for entity typing. We use the Gigaword corpus[6] as the collection of stories. This corpus consists of around 4 million news stories from four different sources. We use two state-of-the-art coreference resolution systems: the **Explosion AI** system Coreferee v1.3.1[7] and the **AllenNLP** coreference model[8]. As explained in Section 3.2, we only keep coreference links that are predicted by both of these systems. Once the encoder has been pre-trained, we train an entity type classifier on the standard training set for each benchmark. We report results for two different variants of this process: one where the entity encoder is fine-tuned while training the entity type classifiers and one where the encoder is frozen. We will refer to these variants as *EnCore* and *EnCore-frozen*, respectively. We train all of the models for 25 epochs with the AdamW optimizer (Loshchilov and Hutter, 2019) and save the checkpoint with the best result on the validation set. The temperature τ in the contrastive loss was set to 0.05.

Benchmarks Our central hypothesis is that the proposed pre-training task makes it possible to learn finer-grained entity representations. As such, we focus on fine-grained entity typing as our main evaluation task. We use the OntoNotes (Gillick et al., 2014) and FIGER (Ling and Weld, 2012) benchmarks. OntoNotes is based on the news stories from the OntoNotes 5.0 corpus[9]. We use the entity annotations that were introduced by Gillick et al. (2014), considering a total of 89 different entity types (i.e. 88 types + *other*). They also introduced a distantly supervised training set, consisting of 133K automatically labelled news stories. FIGER considers a total of 113 types (i.e. 112 types + *other*). The test set consists of sentences from a student newspaper from the University of Washington, two local newspapers, and two specialised magazines (on photography and veterinary). Along with this test set, they also provided automatically labelled Wikipedia articles for training. For fine-grained entity typing, we report the results in terms of macro and micro-averaged F1, following the convention for these benchmarks.

We also experiment on standard entity typing, using the ACE 2005 corpus[10], which covers the following text genres: broadcast conversation, broadcast news, newsgroups, telephone conversations and weblogs. It differentiates between 7 entity types. For this benchmark, the entity spans are not provided. We thus need to identify entity mentions in addition to predicting the corresponding types. We treat the problem of identifying entity span as a sequence labelling problem. We follow the strategy from Hohenecker et al. (2020), but start from our pre-trained entity encoder rather than a standard LM. We summarise this strategy in Appendix A. We use the standard training/development/test splits that were introduced by Li and Ji (2014). Following standard practice, we report the results in terms of micro-averaged F1. We take individual sentences as input. Existing work on this benchmark jointly evaluates span detection and entity typing, i.e. a prediction is only correct if both the span and the predicted type are correct. We will refer to this as the *strict* evaluation setting, following Bekoulis et al. (2018). We also consider the *lenient* setting from, where a prediction is scored as correct as soon as the type is correct and the predicted span *overlaps* with the gold span.

Table 1 summarises the main characteristics of the considered datasets.

Baselines We report results for a number of simplified variants of our main model. First, we consider a variant which uses the same strategy for training the entity type classifier as our full model, but without pre-training the entity encoder on the Gigagword corpus. This variant is referred to as the *base model*. Second, we investigate a setup in which the entity encoder is pre-trained on Gigaword, but only using the masked language modelling (MLM) objective. This setting, which we refer to as *MLM-only*, allows us to analyse to what extent improvements over the base model are due

[4] https://huggingface.co/docs/transformers/model_doc/albert

[5] https://huggingface.co/docs/transformers/model_doc/roberta

[6] https://catalog.ldc.upenn.edu/LDC2003T05

[7] https://github.com/explosion/coreferee

[8] https://demo.allennlp.org/coreference-resolution

[9] https://catalog.ldc.upenn.edu/LDC2013T19

[10] https://catalog.ldc.upenn.edu/LDC2006T06

to the continued training of the language model.

For reference, we also compare our models with the published results of state-of-the-art models. For fine-grained entity typing, we consider the following baselines: **DSAM** (Hu et al., 2021) is an LSTM-based model, which we include as a competitive baseline; **Box4Types** (Onoe et al., 2021) uses hyperboxes to represent mentions and types, to take advantage of the hierarchical structure of the label space; **PICOT** (Zuo et al., 2022) uses a contrastive learning strategy based on the given type hierarchy; **Relational Inductive Bias (RIB)** (Li et al., 2021) uses a graph neural network to model correlations between the different labels. Entity mentions are encoded using a transformer layer on top of pre-trained ELMo (Peters et al., 2018) embeddings; **LITE** (Li et al., 2022a) assigns entity types by fine-tuning a pre-trained Natural Language Inference model; **SEPREM** (Xu et al., 2021) improves on the standard RoBERTa model by exploiting syntax during both pre-training and fine-tuning, and then using a standard entity typing model on top of their pre-trained model; **MLMET** (Dai et al., 2021) extends the standard distantly supervised training data, using the BERT masked language model for generating weak labels; **DenoiseFET** (Pan et al., 2022) uses a denoising strategy to improve the quality of the standard distantly supervised training set, and furthermore exploits prior knowledge about the labels, which is extracted from the parameters of the decoder of the pre-trained BERT model; **PKL** (Li et al., 2023) improves on DenoiseFET by incorporating pre-trained label embeddings.

For ACE 2005, we consider the following baselines: **DyGIE++** (Wadden et al., 2019) uses multi-task learning to jointly train their system for coreference resolution, entity typing, relation extraction and event extraction; **TableSeq** (Wang and Lu, 2020) jointly trains a sequence encoder for entity extraction and a table encoder for relation extraction; **UniRe** (Wang et al., 2021d) also uses a table based representation, which is shared for entity and relation extraction; **PURE** (Zhong and Chen, 2021) uses BERT-based models to get contextualised representations of mention spans, which are fed through a feedforward network to predict entity types; **PL-Marker** (Ye et al., 2022) builds on PURE by introducing a novel span representation.

4.1 Results

Table 2 summarises the results for fine-grained entity typing. As can be seen, EnCore outperforms the base and *MLM-only* models by a large margin, which clearly shows the effectiveness of the proposed pre-training task. Remarkably, EnCore-frozen performs only slightly worse. The best results are obtained with `roberta-large`. Our model furthermore outperforms the baselines on both OntoNotes and FIGER, except that RIB achieves a slightly higher micro-averaged F1 on FIGER. It should be noted that several of the baselines introduce techniques that are orthogonal to our contribution in this paper, e.g. denoising the distantly supervised training sets (DenoiseFET), incorporating prior knowledge about the type labels (PKL) and exploiting label correlations (RIB), which would likely bring further benefits when combined with our pre-training strategy.

Table 3 summarises the results for standard entity typing (ACE 2005). We can again see that EnCore consistently outperforms the MLM-baseline, which in turn consistently outperforms the base model. Comparing the different encoders, the best results for our full model are obtained with `albert-xxlarge-v1`, which is consistent with what was found in previous work (Zhong and Chen, 2021; Ye et al., 2022). Finally, we can see that our full model outperforms all baselines.

4.2 Analysis

We now analyse the performance of our method in more detail. For this analysis, we will focus on ACE 2005 under the lenient setting and OntoNotes. Throughout this section, unless mentioned otherwise, we use `bert-base-uncased` for the encoder.

Encoding Entity Spans We represent entities using the embedding of the head word. In Table 4 we compare this approach with the following alternatives:

MASK We replace the entity mention by a single MASK token and use the final-layer encoding of this token as the embedding of the entity.

Prompt Given a mention (s, e), we append the phrase "The type of e is [MASK]." The final-layer encoding of the MASK-token is then used as the mention embedding.

Masked triple This strategy is similar to *Prompt* but instead of appending a sentence, we append the phrase "<e, hasType, [MASK]>".

Special tokens: full span We add the special tokens <*m*> and </*m*> around the entire entity

Model	LM	OntoNotes		FIGER	
		macro	micro	macro	micro
DSAM	LSTM	83.1	78.2	83.3	81.5
Box4Types	BL	77.3	70.9	79.4	75.0
PICOT	BL	78.7	72.1	84.7	79.6
RIB	ELMo	84.5	79.2	87.7	**84.4**
LITE	RL	86.4	80.9	86.7	83.3
SEPREM	RL	-	-	86.1	82.1
MLMET	BBc	85.4	80.4	-	-
DenoiseFET	BB	87.2	81.4	86.2	82.8
DenoiseFET	RL	87.6	81.8	86.7	83.0
PKL	BB	87.7	81.9	86.8	82.9
PKL	RL	87.9	82.3	87.1	83.1
Base model	BB	76.9	72.9	78.6	76.1
	ALB	77.9	74.8	80.2	77.4
	RL	82.8	80.1	82.3	79.5
MLM-only	BB	81.6	78.7	80.2	77.9
	ALB	82.7	79.8	81.5	79.6
	RL	85.4	81.4	85.8	82.1
EnCore-frozen	BB	87.3	80.6	87.1	82.2
	ALB	87.9	81.9	87.7	82.9
	RL	88.3	82.7	87.8	83.6
EnCore	BB	87.6	81.9	87.3	82.9
	ALB	88.7	82.9	87.9	83.8
	RL	**88.9**	**83.4**	**88.4**	84.1

Table 2: Results for fine-grained entity typing, in terms of macro-F1 and micro-F1 (%). BB stands for bert-base-uncased, BBc stands for bert-base-cased, BL stands for bert-large-uncased, ALB stands for `albert-xxlarge` and RL stands for `roberta-large` DenoiseFET results are taken from (Li et al., 2023); all other baseline results are taken from the original papers.

	Strict			Lenient		
	BB	ALB	RL	BB	ALB	RL
DyGIE++$^{\diamond}$	88.6	-	-	-	-	-
UniRe$^{\diamond}$	88.8	90.2	-	-	-	-
PURE$^{\diamond}$	90.1	90.9	-	-	-	-
PL-Marker$^{\diamond}$	89.8	91.1	-	-	-	-
PURE	88.7	89.7	-	-	-	-
TableSeq	-	89.4	88.9	-	-	-
Base model	86.8	87.1	86.9	90.3	90.8	90.6
MLM-only	87.1	87.8	87.5	90.7	91.2	90.9
EnCore-frozen	89.9	90.5	90.1	91.8	92.3	92.0
EnCore	**90.8**	**91.9**	**91.0**	**92.4**	**93.1**	**92.6**

Table 3: Results for entity typing on ACE 2005, in terms of micro-F1 (%). BB stands for bert-base-uncased, ALB stands for `albert-xxlarge` and RL stands for `roberta-large`. Configurations with $^{\diamond}$ rely on cross-sentence context and are thus not directly comparable with our method.

Strategy	OntoNotes	
	macro	micro
MASK	70.7	66.8
Prompt	72.1	68.7
Masked triple	72.8	69.4
Special tokens: full span	75.2	70.8
Special tokens: head	76.1	71.3
Head word	76.9	72.9

Table 4: Comparison of different strategies for encoding entity spans (using `bert-base-uncased`).

span. We take the final-layer encoding of the <*m*> token as the embedding of the entity.

Special tokens: head In this variant, we add the special tokens <*m*> and </*m*> around the head word of the entity span.

Head word This is the method adopted in our main experiments. In this case, we simply use the embedding of the head word of the entity mention, without using special tokens.

In all cases, we use the entity typing model that was described in 3.1. Note that we do not consider ACE 2005 for this analysis, as the entity spans have to be predicted by the model for this dataset, which means that aforementioned alternatives cannot be used. For this analysis, we train the entity encoder on the training data of the considered benchmark, without using our coreference based pre-training strategy. The results in Table 4 show that using the embedding of the head word clearly outperforms the considered alternatives. Another interesting observation is that encapsulating the head of the entity mention performs slightly better than encapsulating the entire entity span, whereas it is the latter variant that is normally used in the literature. It is also notable, and somewhat surprising, that *Masked triple* outperforms *Prompt*.

Pre-training Strategies In Table 5 we compare four strategies for pre-training the entity encoder based on coreference chains. In particular, we analyse the effect of two aspects:

- When training our model, the negative examples for the contrastive loss (Section 3.2) are always selected from other stories. Here we analyse the impact of choosing these negative examples from the same story instead.

- During training, in 15% of the cases, we mask the head of the entity span. Here we consider two other possibilities: (i) never masking the entity span and (ii) masking the entire span.

Neg. samples	Masking	ACE05	OntoNotes	
		micro	macro	micro
Same story	None	83.9	82.1	74.9
Same story	Entire span	84.7	82.9	75.3
Different stories	Entire span	88.8	86.2	78.9
Different stories	Head	91.8	87.3	80.6

Table 5: Comparison of different strategies for pre-training the entity encoder (using `bert-base-uncased`).

Coreference Systems	ACE05	OntoNotes	
	micro	macro	micro
Explosion AI	86.4	83.4	79.4
AllenNLP	90.7	86.8	80.1
Explosion AI + AllenNLP	91.8	87.3	80.6

Table 6: Comparison of different coreference resolution strategies (using `bert-base-uncased`).

Choosing the negative examples from the same story has a number of implications. First, it may mean that false negatives are included (i.e. coreference links that were missed by the system). Second, it means that the overall number of negative examples becomes smaller, since they have to come from a single story. However, these downsides may be offset by the fact that negative examples from the same story may be harder to discriminate from the positive examples, since the story context is the same, and using harder negatives is typically beneficial for contrastive learning. For this analysis we use EnCore-frozen. As can be seen in Table 5, choosing negative examples from the same story overall has a clearly detrimental impact. We also find that masking is important, where masking only the head of the entity span leads to the best results. This masking strategy has not yet been used in the literature, to the best of our knowledge.

Coreference Resolution In Table 6 we analyse the importance of using only high-quality coreference links. In particular, we compare three configurations: (i) using all links predicted by the Explosion AI system; (ii) using all links predicted by the AllenNLP system; and (iii) using only the links that are predicted by both systems. For this analysis, we use EnCore-frozen. As can be seen, the AllenNLP system overall performs better than the Explosion AI system. However, the best results are obtained by combining both systems.

Model	One Label		Two Labels		Three labels	
	macro	micro	macro	micro	macro	micro
MLM-only	79.8	75.6	53.0	50.9	39.1	38.4
EnCore	82.7	78.7	59.8	58.5	44.6	43.6

Table 7: Comparison of the MLM-only and EnCore models (using `roberta-large`) on partitions of the OntoNotes test set.

Performance on Fine and Coarse Labels In Table 7 we compare our full model with the *MLM-only* variant on different partitions of the OntoNotes test set. We specifically compare EnCore and *MLM-only* on those examples with one-level labels (5.3K); two-level labels (3.0K); and three-level labels (0.6K). Examples with one-level labels only require the model to determine the top-level entity type (e.g `/organisation`). Examples with two-level labels call for more precise finer-grained differentiations (e.g. `/organisation` and `/organisation/company`). Examples with three-level labels call for even more precision (e.g. `/organisation`, `/organisation/company` and `/organization/company/broadcast`). EnCore performs better than *MLM-only* in every scenario, as can be observed, with the difference being least pronounced in the case of single-level labels. This supports the idea that our pre-training technique is particularly useful for learning finer-grained entity types. A more detailed breakdown of the results, which is provided in the appendix, shows that EnCore consistenly outperforms *MLM-only* on all labels, both for OntoNotes and FIGER.

5 Conclusion

We have proposed a strategy which uses coreference chains to pre-train an entity encoder. Our strategy relies on the natural idea that coreferring entity mentions should be represented using similar vectors. Using a contrastive loss for implementing this intuition, we found that the resulting encoders are highly suitable for (fine-grained) entity typing. In our analysis, we found that restricting our strategy to high-quality coreference links was important for its success. We also found that focusing on the head of the entity span, rather than the span itself, was beneficial, both when it comes to representing the entity span and when it comes to masking entities during training (where only masking the head was found to be most helpful).

6 Limitations

Our model is pre-trained on individual sentences. This means that during testing, we cannot exploit cross-sentence context. Prior work has found such cross-sentence context to be helpful for benchmarks such as ACE2005, so it would be of interest to extend our model along these lines. Furthermore, we have not yet applied our model to ultra-fine entity typing, as this task requires us to cope with labels for which we have no, or only very few training examples. This would require combining our entity encoder with entity typing models that can exploit label embeddings, such as UNIST (Huang et al., 2022), which we have left as an avenue for future work.

Acknowledgements This research was supported by EPSRC grant EP/W003309/1 and undertaken using the supercomputing facilities at Cardiff University operated by Advanced Research Computing at Cardiff (ARCCA) on behalf of the Cardiff Supercomputing Facility and the HPC Wales and Supercomputing Wales (SCW) projects. We acknowledge the support of the latter, which is part-funded by the European Regional Development Fund (ERDF) via the Welsh Government.

References

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018. Adversarial training for multi-context joint entity and relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2830–2836, Brussels, Belgium. Association for Computational Linguistics.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018. Ultra-fine entity typing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 87–96, Melbourne, Australia. Association for Computational Linguistics.

Hongliang Dai, Yangqiu Song, and Haixun Wang. 2021. Ultra-fine entity typing with weak supervision from a masked language model. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1790–1799, Online. Association for Computational Linguistics.

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

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dan Gillick, Nevena Lazic, Kuzman Ganchev, Jesse Kirchner, and David Huynh. 2014. Context-dependent fine-grained entity type tagging. *CoRR*, abs/1412.1820.

Jiale Han, Bo Cheng, and Wei Lu. 2021. Exploring task difficulty for few-shot relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2605–2616, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ridong Han, Tao Peng, Chaohao Yang, Benyou Wang, Lu Liu, and Xiang Wan. 2023. Is information extraction solved by chatgpt? an analysis of performance, evaluation criteria, robustness and errors. *CoRR*, abs/2305.14450.

Patrick Hohenecker, Frank Mtumbuka, Vid Kocijan, and Thomas Lukasiewicz. 2020. Systematic comparison of neural architectures and training approaches for open information extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8554–8565, Online. Association for Computational Linguistics.

Yanfeng Hu, Xue Qiao, Luo Xing, and Chen Peng. 2021. Diversified semantic attention model for fine-grained entity typing. *IEEE Access*, 9:2251–2265.

James Y. Huang, Bangzheng Li, Jiashu Xu, and Muhao Chen. 2022. Unified semantic typing with meaningful label inference. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2642–2654, Seattle, United States. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Bangzheng Li, Wenpeng Yin, and Muhao Chen. 2022a. Ultra-fine entity typing with indirect supervision from natural language inference. *Transactions of the Association for Computational Linguistics*, 10:607–622.

Jiacheng Li, Jingbo Shang, and Julian McAuley. 2022b. UCTopic: Unsupervised contrastive learning for phrase representations and topic mining. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6159–6169, Dublin, Ireland. Association for Computational Linguistics.

Jinqing Li, Xiaojun Chen, Dakui Wang, and Yuwei Li. 2021. Enhancing label representations with relational inductive bias constraint for fine-grained entity typing. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 3843–3849. ijcai.org.

Na Li, Zied Bouraoui, and Steven Schockaert. 2023. Ultra-fine entity typing with prior knowledge about labels: A simple clustering based strategy. *CoRR*, abs/2305.12802.

Qi Li and Heng Ji. 2014. Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 402–412, Baltimore, Maryland. Association for Computational Linguistics.

Xiao Ling and Daniel S. Weld. 2012. Fine-grained entity recognition. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada*. AAAI Press.

Fangyu Liu, Ivan Vulić, Anna Korhonen, and Nigel Collier. 2021a. Fast, effective, and self-supervised: Transforming masked language models into universal lexical and sentence encoders. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1442–1459, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Qianchu Liu, Fangyu Liu, Nigel Collier, Anna Korhonen, and Ivan Vulić. 2021b. MirrorWiC: On eliciting word-in-context representations from pretrained language models. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 562–574, Online. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium. Association for Computational Linguistics.

Yasumasa Onoe, Michael Boratko, Andrew McCallum, and Greg Durrett. 2021. Modeling fine-grained entity types with box embeddings. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2051–2064, Online. Association for Computational Linguistics.

Yasumasa Onoe and Greg Durrett. 2019. Learning to denoise distantly-labeled data for entity typing. 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 2407–2417, Minneapolis, Minnesota. Association for Computational Linguistics.

Yasumasa Onoe and Greg Durrett. 2020. Interpretable entity representations through large-scale typing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 612–624, Online. Association for Computational Linguistics.

Weiran Pan, Wei Wei, and Feida Zhu. 2022. Automatic noisy label correction for fine-grained entity typing. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 4317–4323. ijcai.org.

Hao Peng, Tianyu Gao, Xu Han, Yankai Lin, Peng Li, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2020. Learning from Context or Names? An Empirical Study on Neural Relation Extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3661–3672, Online. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

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.

Xiang Ren, Wenqi He, Meng Qu, Clare R. Voss, Heng Ji, and Jiawei Han. 2016. Label noise reduction in entity typing by heterogeneous partial-label embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1825–1834. ACM.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748.

Yuval Varkel and Amir Globerson. 2020. Pre-training mention representations in coreference models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8534–8540, Online. Association for Computational Linguistics.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span 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 5784–5789, Hong Kong, China. Association for Computational Linguistics.

Zhen Wan, Fei Cheng, Qianying Liu, Zhuoyuan Mao, Haiyue Song, and Sadao Kurohashi. 2022. Relation extraction with weighted contrastive pre-training on distant supervision. *CoRR*, abs/2205.08770.

Jue Wang and Wei Lu. 2020. Two are better than one: Joint entity and relation extraction with table-sequence encoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1706–1721, Online. Association for Computational Linguistics.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021a. K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1405–1418, Online. Association for Computational Linguistics.

Shufan Wang, Laure Thompson, and Mohit Iyyer. 2021b. Phrase-BERT: Improved phrase embeddings from BERT with an application to corpus exploration. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10837–10851, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shusen Wang, Bosen Zhang, Yajing Xu, Yanan Wu, and Bo Xiao. 2022. RCL: Relation contrastive learning for zero-shot relation extraction. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2456–2468, Seattle, United States. Association for Computational Linguistics.

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, Jihua Kang, Jingsheng Yang, Siyuan Li, and Chunsai Du. 2023. InstructUIE: Multi-task instruction tuning for unified information extraction. *CoRR*, abs/2304.08085.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021c. KEPLER: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194.

Yijun Wang, Changzhi Sun, Yuanbin Wu, Junchi Yan, Peng Gao, and Guotong Xie. 2020. Pre-training entity relation encoder with intra-span and inter-span information. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1692–1705, Online. Association for Computational Linguistics.

Yijun Wang, Changzhi Sun, Yuanbin Wu, Hao Zhou, Lei Li, and Junchi Yan. 2021d. UniRE: A unified label space for entity relation extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 220–231, Online. Association for Computational Linguistics.

Zenan Xu, Daya Guo, Duyu Tang, Qinliang Su, Linjun Shou, Ming Gong, Wanjun Zhong, Xiaojun Quan, Daxin Jiang, and Nan Duan. 2021. Syntax-enhanced pre-trained model. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5412–5422, Online. Association for Computational Linguistics.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online. Association for Computational Linguistics.

Deming Ye, Yankai Lin, Peng Li, and Maosong Sun. 2022. Packed levitated marker for entity and relation extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4904–4917, Dublin, Ireland. Association for Computational Linguistics.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 50–61, Online. Association for Computational Linguistics.

Xinyu Zuo, Haijin Liang, Ning Jing, Shuang Zeng, Zhou Fang, and Yu Luo. 2022. Type-enriched hierarchical contrastive strategy for fine-grained entity typing. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2405–2417, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

A Entity Span Detection

We treat the problem of entity span detection as a sequence labelling problem, following the strategy from Hohenecker et al. (2020). Specifically, each token in the input sentence is then labelled with an appropriate tag, which could either be one of the entity types from the considered dataset or a tag which denotes that the token does not belong to any entity span. To assign these tags, we again use the encoder that was pre-trained on coreference chains. However, rather than looking only at the head word of a given entity span, we now consider the embedding of every token in the sentence. Specifically, we train a linear classifier to predict the correct tag from the contextualised representation of each token, while optionally also fine-tuning the encoder. Since most tokens do not belong to any entity span, the training data will inevitably be highly imbalanced. For this reason, during training, we ignore the majority of tokens that are outside of any entity span. Specifically, following Hohenecker et al. (2020), we only consider such tokens when they are immediately preceding or succeeding an entity span.

B Additional Analysis

Prediction confidence In Table 8, we compare the confidence of the EnCore and MLM-only models for the gold label predictions. We observe that in the first example, EnCore more confidently predicts the label for *delegation* as */organization* than MLM-only, which places *delegation* in the more generic label class */other* with lower confidence. In the second and third case, we observe that EnCore is more certain to label the currency terms *dollars* and *RMB* with the second-level label */other/currency* than with the more general first level label */other*, whereas MLM-only assigns a very low confidence to */other/currency*. A similar pattern can also be observed in the last example. We have observed the same trend throughout the test set: EnCore consistently makes more confident predictions than MLM-only. This is especially evident for the second- and third-level labels.

Breakdown by Label A closer examination of the model outputs in Figure 2 reveals that EnCore consistently beats the MLM-only model across all entity types. The OntoNotes test set, for example, contains 1130 */person* gold labels. MLM-only predicts only 67.96% of these accurately, compared to 85.49% for EnCore. As an example of a label at the second level, there are 74 */person/artist* gold labels in the test set; the MLM-only model correctly predicts 21.62% of these, whereas EnCore correctly predicts 35.14%. At the third level, there are 58 */person/artist/author* gold labels. The MLM-only model predicts only 13.79% of them correctly, while EnCore predicts 25.86% correctly. These patterns are consistently seen over the whole label set. This is also true for the FIGER test set, as shown in Figure 3.

	Sentence	Gold label	MLM-only	EnCore
(1)	At the beginning of 1993 , six cities such as Zhuhai , Foshan , etc. also organized a **delegation** to advertise in the US and Canada for students studying abroad.	/organization /other	0.26 0.54	0.60 0.15
(2)	Last year , its foreign exchange income was up to more than 2.1 billion US **dollars**, and in the first half of this year exports again had new growth.	/other /other/currency	0.63 0.04	0.97 0.98
(3)	In 1997 , this plant made over 4,400 tons of Mao - tai ; with sales income exceeding 500 million yuan RMB , and profit and taxes reaching 370 million **RMB** , both being the best levels in history.	/other /other/currency	0.31 0.02	0.94 0.96
(4)	In the near future , the Russian Tumen River Region Negotiation Conference will also be held in **Vladivostok**.	/location /location/city	0.25 0.07	0.98 0.73

Table 8: Comparison of the confidence of the *MLM-only* and *EnCore* models (with `roberta-large`) on sample cases from the OntoNotes test set. The words in **bold** in the input sentences are the entity spans' head word. The **MLM-only** and **EnCore** columns indicate the confidence of the *MLM-only* and *EnCore* models, respectively.

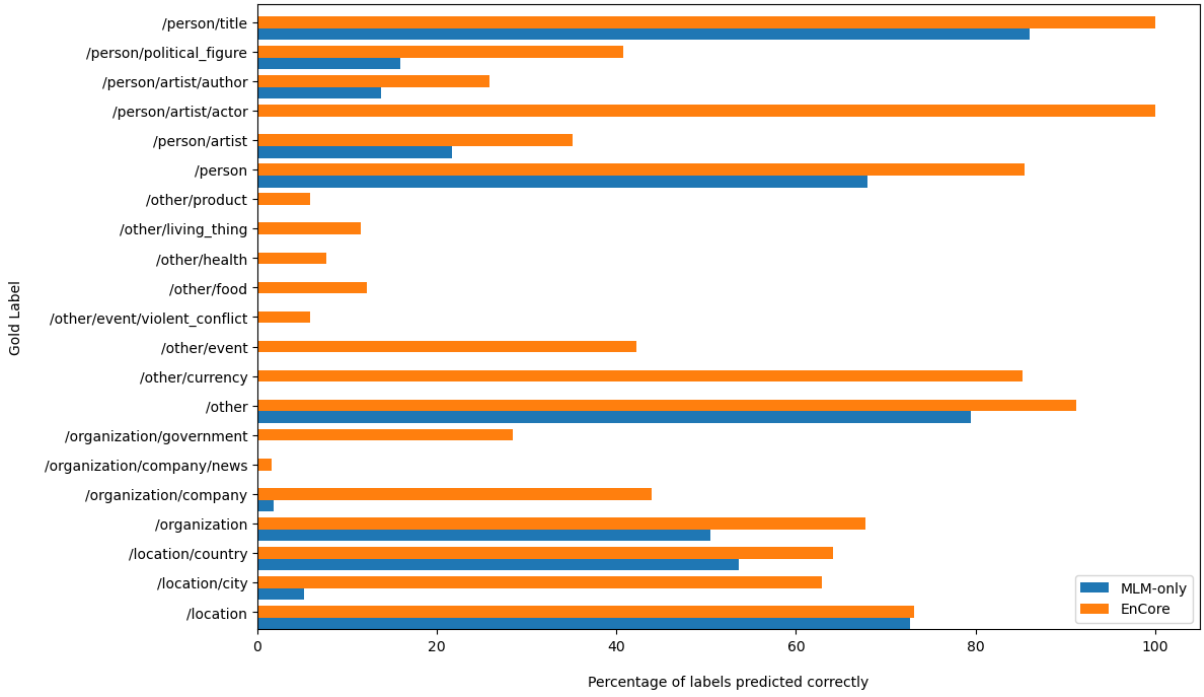

Figure 2: Comparison of the percentage of correct predictions per gold label by the MLM-only and EnCore models (with `roberta-large`) on the OntoNotes test set. The instances of a label that are accurately predicted are expressed as a percentage of the total number of occurrences of the corresponding gold label.

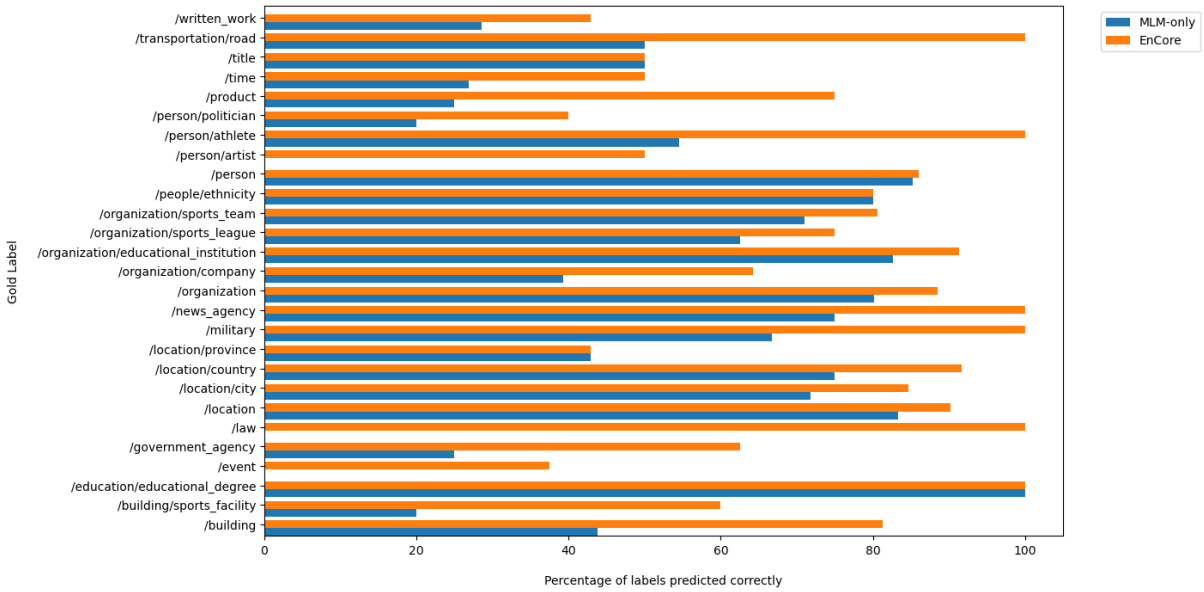

Figure 3: Comparison of the percentage of correct predictions per gold label by the MLM-only and EnCore models (with `roberta-large`) on the FIGER test set. The instances of a label that are accurately predicted are expressed as a percentage of the total number of occurrences of the corresponding gold label.