Sierra-Múnera, Alejandro; Westphal, Jan; Krestel, Ralf

Efficient Ultrafine Typing of Named Entities

Suggested Citation: Sierra-Múnera, Alejandro; Westphal, Jan; Krestel, Ralf (2023) : Efficient Ultrafine Typing of Named Entities, In: Proceedings of the Joint Conference on Digital Libraries (JCDL) 2023, ISBN 979-8-3503-9931-8, ACM, New York, pp. 205-214, https://doi.org/10.1109/JCDL57899.2023.00038

Efficient Ultrafine Typing of Named Entities

Alejandro Sierra-Múnera
Hasso Plattner Institute
University of Potsdam
Potsdam, Germany
alejandro.sierra@hpi.de

Jan Westphal
Hasso Plattner Institute
University of Potsdam
Potsdam, Germany
jan.westphal@student.hpi.uni-potsdam.de

Ralf Krestel
ZBW - Leibniz Information Centre
for Economics & Kiel University
Kiel, Germany
r.krestel@zbw.eu

***Abstract*—Ultrafine named entity typing (UFET) refers to the assignment of predefined labels to entity mentions in a given context. In contrast to traditional named entity typing, the number of potential labels is in the thousands and one mention can have more than one assigned type. Previous approaches either depend on large training datasets, or require inefficient encoding of all input-type combinations. Therefore, there is a need for investigating the efficiency during training and prediction of entity typing models in the ultrafine-grained setting, considering its distinctively bigger search space, compared to the coarse- and fine-grained tasks. To efficiently solve UFET, we propose DECENT, a lightweight model that encodes, using a pretrained language model, the input sentences separately from the type labels. Additionally, we make use of negative oversampling to speed up the training while improving the generalization of unseen types. Using an openly available UFET dataset, we evaluated the classification and runtime performance of DECENT and observed that training and prediction runtime is orders of magnitude faster than the current state-of-the-art approaches, while maintaining a competitive classification performance.**

***Index Terms*—ultrafine entity typing, entity typing, named entity recognition**

I. INTRODUCTION

One of the essential tasks of information extraction is the recognition and classification of named entities in text. These are critical tasks for digital libraries, especially for the extraction of valuable information from large amounts of text and for the retrieval of relevant documents and knowledge. Additionally, recognizing entity mentions in text is a key step in the construction of knowledge graphs.

The task can be broken down into two subtasks: named entity recognition (NER), where the boundaries of entity mentions are predicted, and named entity typing (NET), where the entity mention is classified into one or more pre-defined entity types. In the literature, named entity recognition (NER) is sometimes defined as one single task of recognition and classification. In the scope of this paper, we separate recognition from classification (typing) and focus on the latter.

Traditional NET focuses on a small set of coarse-grained entity types. For instance, CoNLL-03 [1] uses 4 general types, namely *person*, *location*, *organization* and *miscellaneous*. Finer-grained entity types have been proposed in different works [2]–[5], further defining hierarchies and increasing the number of entity types from a small set to hundreds (e.g. *person* could be further divided into *artist*, *politician*, *scholar*, *sportsmen*, etc.). However, with an increased number of types, additional challenges emerge. First, many entities and entity types are rarely (or even never) seen during training, resulting in the *long tail* problem [6]. Second, the assumption of mutual exclusion between entity types breaks, which means the task transforms from a single-label into a multi-label classification problem. Recent studies also introduced the task of ultrafine named entity typing, in which the number of entity types surpasses a thousand or even ten thousand to achieve a more precise semantic coverage than the fine-grained NET task [7]–[10].

Three examples taken from the UFET dataset [8] are depicted in Table I to illustrate the ultrafine-grained typing task. The first one assigns five different types to the entity *Ivan Lendl*. The task combines coarse-grain types like *person*, fine-grain like *player* and ultrafine-grain like *tennis_player*. The second example illustrates another additional challenge present in the task, which is the presence of mentions in the form of pronouns. This differs slightly from the traditional NET task, in which only direct mentions to the entities like the first example are considered. Finding the right entity given the pronoun and classifying accordingly is a particular challenging for UFET models. Lastly, in the third example, the nominal mention *the ball* differs from the named entities by containing simpler noun phrases with entity types like *football* that strongly depend on the context of the mention.

To this end, we propose DECENT, a model which is efficient in terms of the amount of data needed for training and the runtime needed for training and prediction, while maintaining competitive classification performance for the UFET task. We share the code for the model and the experiments online.[1]

II. RELATED WORK

For coarse-grained NER and NET, one single model is responsible for both finding the entity boundaries and classifying among the set of entity types. Notable models for this task are the ones proposed in [11], [12], [13], and [14].

a) Fine-grained Entity Typing.: [3] were the first to introduce a general fine-grained entity type ontology by separating a set of coarse-grained entity types into 147 fine-grained entity types. However, there was already an earlier approach using fine-grained named entity typing focused on person-related types [2]. [15] laid additional groundwork for the field

[1] https://github.com/HPI-Information-Systems/DECENT

TABLE I
EXEMPLES OF ENTITY MENTIONS AND THEIR ULTRAFINE-GRAINED ENTITY TYPES AS FOUND IN UFET

Sentence	Labels
Jose Luis Clerc was the defending champion but lost in the semifinals to **Ivan Lendl**.	person, athlete, player, tennis_player, winner
Pacino , 65 , will also direct the tragi - comedy 'Salomaybe?' while taking on the Herod role **he** has played on stage in both New York and Los Angeles, according to Daily Variety.	person, actor, artist, director, administrator, conductor, creator, performer, entertainer
Just a minute later Rafael Van der Vaart brought Real level , although replays suggest he controlled **the ball** with his hand, and then Xabi Alonso headed them into the lead .	object, ball, equipment, football

of fine-grained NET and emphasized the advantages of having a fine-grained type structure. [4] published one of the first large benchmark datasets for fine-grained NET called FIGER, which covers 112 different entity types. To structure the ontologies, researchers usually provide additional information in the form of hierarchies. For example, *actor* is a subtype of *person* or *location* is the supertype of *country*, *city*, etc. Many studies utilize external knowledge such as knowledge bases to improve fine-grained type inference, but [5] suggest restricting the acceptable labels to those that can be inferred from the local context only.

b) Ultrafine Entity Typing.: Even more detailed entity type sets have been studied, for which models have to classify the entities among extensive collections of types. The standard benchmark for the ultrafine named entity typing task, which usually spans thousands of possible entity types, is the UFET dataset by [8]. Their ontology consists of 9 coarse, general types (e.g. person, organization, etc.), 121 fine-grained types (e.g. artist, athlete, etc.), and 10,201 ultra-fine types (e.g. martial artist, street artist, etc.) with no predefined hierarchy. They provide 5,994 human-annotated examples evenly split into train, validation, and test set. In these examples, only 2,519 entity types from the full set were annotated. In addition, there is distantly supervised data in the form of 5.2M samples, automatically labeled through entity linking and 20M samples labeled by assigning types using to the head words of nominal entity mentions. Further, [8] designed UFET-BILSTM that serves as a baseline for future methods. In this approach, each input representation is a concatenation of the context representation and the mention representation. They use word and character embeddings, and use the distantly supervised data to compute entity type embeddings, which are then used to infer type probabilities for a particular sentence and mention. Other models use UFET-BILSTM as the base architecture and improve specific aspects like label correlations [16] and mention representations [17]. A different approach, BOX4TYPES [18], utilize box embeddings [19] to achieve named entity typing by projecting the textual representation of the input into a high-dimensional hyper-rectangular box space.

[9] train the multi-label classifier MLMET on a combination of the human-annotated and the distantly supervise data from [8], with additionally generated data from using masked language modeling [20]. Their approach requires large amounts of training data to cover all entity types. They employ all the distantly supervised portion of the UFET dataset. In addition, the training of the model has 3 different stages: first pretraining with the weakly supervised data, then fine-tuning with the human annotated data, and finally a self-training stage.

[10] use the indirect supervision from natural language inference (NLI) for their model LITE to type named entities. They convert named entity typing into an NLI task by treating the input with the entity mention as the premise and creating the hypothesis from a candidate label using a template in the form "<Entity> is a <Type>". They employ RoBERTa-large [21] that has already been fine-tuned on the MNLI dataset [22]. They further fine-tune the model on the manually annotated part of the UFET dataset with a learning-to-rank objective by sampling an incorrect label for each correct type. LITE predicts the entailment score for every type label in the UFET dataset ontology given an entity-mentioning input to produce a prediction of entity types, which can be inefficient in terms of runtime at prediction time.

In contrast to the aforementioned models, our approach does not require huge amounts of labeled data, making it more data efficient and reducing training and prediction runtime significantly. In addition, our model avoids the costly full encoding of each mention-type pair during prediction and does not rely on natural language templates or task descriptions.

III. DECENT

Our proposed model (**D**ecoupled **E**ncoding and **C**ross-Attention for **E**fficient **N**amed Entity **T**yping) combines a lightweight design with a special learning procedure to enable fast training and prediction.

Similar to LITE [10], we use a pretrained language model (PLM) for contextualized embeddings and the semantics of type labels, making it possible to classify entity types that were not present in the training data. With this, the model does not require examples for each of the ultrafine entity types, avoiding the need of an extensive (distantly) supervised dataset. However, we do not encode sentence and entity types as NLI pairs. We argue that a lighter model on independently encoded sentences and entity types is more efficient and opens the possibility of using more negative samples.

Given an input $x = (s, m)$ with the context $s = (s_1, \ldots, s_k)$ and the marked entity mention $m = (m_1, \ldots, m_l)$ our objective is to predict all applicable entity types $T \subseteq \mathcal{T}$ from the ontology $\mathcal{T}$. For example:

- "The movie 'Gran Torino' stars **Clint Eastwood**"

In this sentence with the marked entity *Clint Eastwood*, we want to assign types such as *person*, *actor*, *entertainer*, etc.

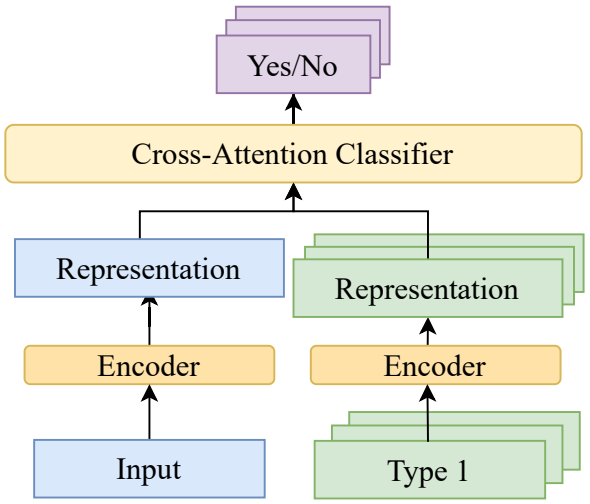

Fig. 1. DECENT's general architecture for ultrafine named entity typing.

To independently encode the context s and the different entity types, our model $\mathcal{M}$ consists of two parts, with a design inspired by FASTMATCH [23]. An overview of our architecture is shown in Figure 1.

First, the encoder $\mathcal{E}$ encodes the input x and the type label $t \in \mathcal{T}$ separately, resulting in the corresponding embedded representations x^h and y^h.

$$x^h = \mathcal{E}(x)$$
$$y^h = \mathcal{E}(t)$$

This way, the embeddings of the type labels can also be precomputed and stored, as they do not depend on the input x. Similar to LRN [24], we add special tokens $[M/]$ and $[/M]$ enclosing the entity mention m in the input sentence x. The special tokens are used later to distinguish the entity from the rest of the sentence.

Then we apply a lightweight classifier $\mathcal{C}$ on top of the contextualized embeddings produced by $\mathcal{E}$. The internal classifier's architecture is shown in Figure 2.

The classifier incorporates a cross-attention [25], [26] mechanism $\mathcal{C}_A$ that performs token-to-token interaction between the embeddings of the input and the type label. In contrast to the original scaled dot-product attention, we use V as both keys and values (see Equation 1). In our case, Q and V refer to sequences of token embeddings, each having the size of encoder dimension $d_\mathcal{E}$.

$$A(Q,V) = softmax(\frac{QV^T}{\sqrt{d}})V \in \mathbb{R}^{n \times d} \tag{1}$$

We apply the attention mechanism to the embeddings x^h and y^h, treating x^h as the queries, y^h as values, and vice versa. The vector representations x^c and y^c encode the interactions between the sentence and the type label.

$$x^c = \mathcal{CA}(x^h, y^h)$$
$$y^c = \mathcal{CA}(y^h, x^h)$$

Similar to FASTMATCH, we apply the attention mechanism again to aggregate the local interactions between tokens into a global representation. To achieve this, we compute the attention for the representation token $[M/]$ in the input sequence:

$$x^s_{[M/]} = A(x^c_{[M/]}, x^c)$$

As stated by [23], the intuition is that $x^c_{[M/]}$ is the representation of the cross-attention interaction between the sequences. Therefore, the token-wise interactions consistent with the representation should receive more weight. For the representation of the type label after cross-attention, we compute the average of the type label embeddings y^c.

Lastly, we feed x^r, the concatenation of all the input and label representations from before and after cross-attention, into a multi-layer perceptron $\mathcal{C}_F$ with one hidden layer of size $|\mathcal{H}|$, that outputs an annotation probability $p \in [0,1]$ for the type $t \in \mathcal{T}$ (Equation 3).

$$x^r = x^h_{[M/]} \oplus x^s_{[M/]} \oplus avg(y^c) \oplus avg(y^h) \tag{2}$$

$$\begin{aligned} p &= \mathrm{P}(t \in \mathcal{T}|x) \\ &= \mathcal{C}_F(x^r) \\ &= \sigma(W_2 \times \mathrm{ReLU}(W_1 \times x^r + b_1) + b_2) \in (0,1) \end{aligned} \tag{3}$$

a) Training Procedure.: The classification head computes the probability p for one specific entity type y, thus, similar to LITE, non annotated types are assumed to be negative types. Let T be all the annotated types from the label space $\mathcal{T}$ for the input $x = (s,m)$. Therefore, we define $\overline{T} = \mathcal{T} \setminus T$ as the entity types from the ontology that *have not been assigned* to x. For simplicity, we refer to T and $\overline{T}$ as the positive and negative types, respectively. LITE uses one negative type for each positive type during training, but because the classification head in our model is lighter, we opt for oversampling from the negative types.

For each input x, we sample one positive type from T and r_{neg} negative types from $\overline{T}$ and train with the respective class labels. This is particularly important for large ontologies such as the one of the UFET dataset [8], for which the majority of entity types have not been manually annotated. As our model outputs a classification probability $p \in (0,1)$ for each entity type, we use binary cross entropy loss as our loss function.

b) Prediction Procedure.: At prediction time, for an input (s,m), s is encoded, and the cross attention head computes p for each of the entity types $t \in \mathcal{T}$. Note that all the entity types can be pre-encoded and the classification head is significantly lighter compared to the PLM encoder. If p is above a certain threshold γ, we choose t as one of the predictions for (s,m).

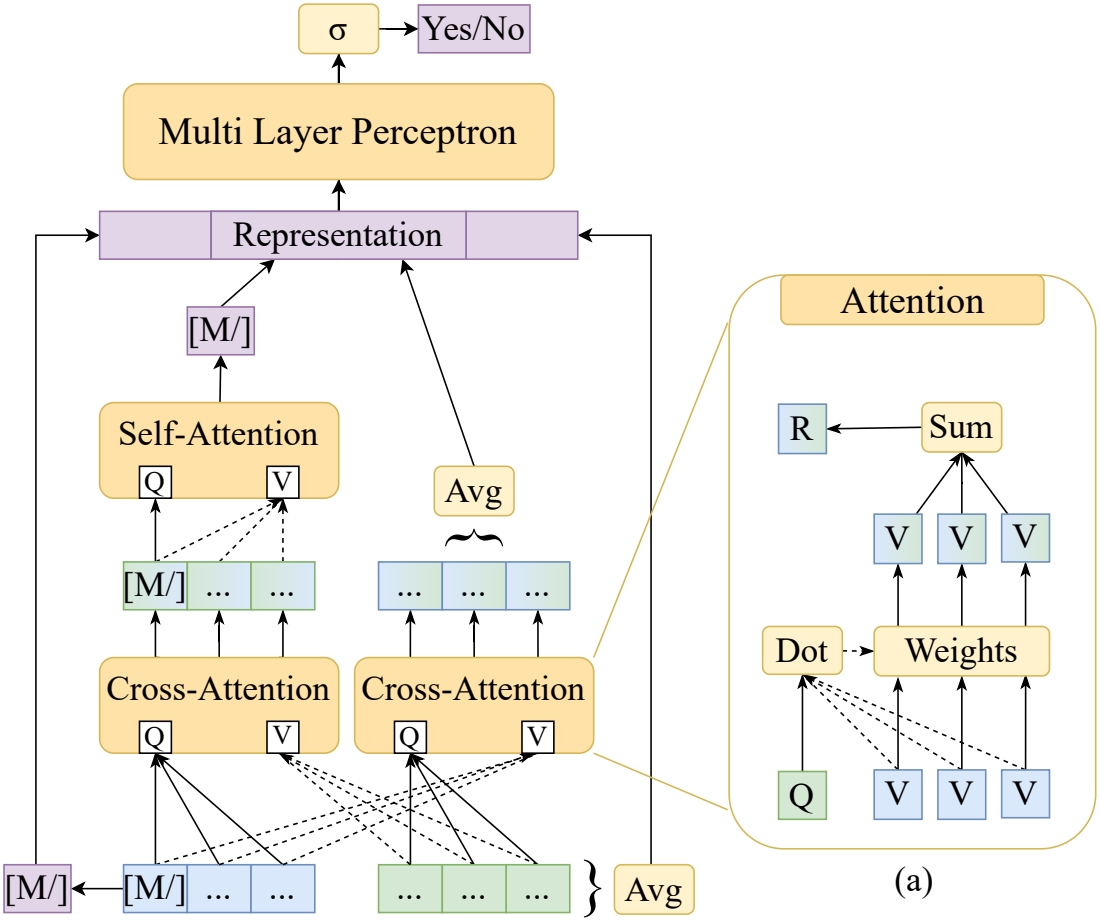

Fig. 2. The cross attention classification head as it processes the embeddings of input and type label. (a) shows the details of the cross-attention mechanism.

IV. Experiments

To evaluate the validity of our approach, we conduct experiments on the UFET benchmark from [8] and compare the runtime and classification performance of our model DECENT with UFET-BILSTM [8], MLMET [9] and LITE [10], which correspond to the original proposed model with UFET, and the two latest state-of-the-art models.

For our model, we use RoBERTa-large [21] as the encoder, which was also used by LITE. We use the implementation from Huggingface's Transformers library [27][2]. We conduct a hyperparameter search for the following parameters using the validation set and optimize for macro-F1 score. The following values are used for the final evaluation:

- Learning rate (encoder): 5×10^{-6} j
- Learning rate (head): 5×10^{-4}
- Dropout probability (head): 0.5
- Negative oversampling rate r_{neg}: 31
- Prediction threshold γ : 0.9

[2]https://huggingface.co/roberta-large

TABLE II
GENERAL PREDICTION PERFORMANCE ON THE TEST SPLIT OF UFET. ALL THE REPORTED RESULTS ARE MEASURED IN MACRO-AVERAGED SCORES

Model	P	R	F1
UFET-BILSTM	48.1	23.2	31.3
MLMET	**53.6**	45.3	49.1
LITE_{H}	48.7	45.8	47.2
$\text{LITE}_{\text{w/o lab dep.}}$	53.3	46.6	49.7
LITE	52.4	**48.9**	**50.6**
DECENT	50.8	48.7	49.7

† marks scores calculated with the publicly available model checkpoint.
Bold means best model and underline means second best

A. Classification Performance

Firstly, we want to evaluate the quality of the predictions of DECENT in comparison to the baselines. In case of LITE we also evaluate different configurations: LITE_{H} is only trained using the human-annotated data and $\text{LITE}_{\text{w/o lab dep.}}$, which is pretrained on the MLNI benchmark [22] but does not rely on additional information about the label space in the form of label dependencies.

TABLE III
PREDICTION PERFORMANCE FOR DIFFERENT TYPE GRANULARITIES IN UFET

Model	Coarse (9)			Fine (121)			Ultrafine (10,201)		
	P	R	F1	P	R	F1	P	R	F1
UFET-BILSTM	60.3	61.6	61.0	40.4	38.4	39.4	42.8	8.8	14.6
MLMET†	69.3	**84.9**	76.3	47.0	**65.5**	54.7	**47.5**	31.9	38.1
LITE†	**72.3**	82.8	77.2	**57.7**	59.2	**58.4**	41.7	**39.5**	**40.5**
DECENT	71.6	84.8	**77.6**	55.7	60.2	57.9	42.7	36.5	39.4

† marks scores calculated with the publicly available model checkpoint.
Bold means best model and underline means second best.

TABLE IV
COMPARISON OF PREDICTION PERFORMANCE FOR DIFFERENT TYPES OF ENTITY MENTIONS IN UFET

Model	Named Entity			Pronoun			Nominal		
	P	R	F1	P	R	F1	P	R	F1
MLMET†	**57.6**	55.6	**56.6**	54.8	50.8	52.7	**49.1**	39.4	43.7
LITE†	54.0	**58.8**	56.3	54.6	**53.7**	54.2	46.9	**45.0**	**45.9**
DECENT	54.9	57.0	56.0	**57.0**	**53.7**	**55.3**	44.4	42.2	43.3

† marks scores calculated with the publicly available model checkpoint.

For comparison, we use the macro-averaged precision (P), recall (R) and F1 scores. The F1 score is computed as the harmonic mean of macro-averaged P and R. For the baseline methods, we compare against their reported results.

The prediction results are shown in Table II. Here we see that DECENT performs better than the original UFET dataset baseline UFET-BILSTM and has comparable results to both, MLMET and LITE. Interestingly, our approach performs better than LITE when it is solely trained on the human-annotated data (LITE_{H}) and performs similar to the model pretrained on the natural language inference dataset MNLI ($\text{LITE}_{\text{NLI+H}}$). Only additional information through label dependencies allow for better prediction performance compared to our approach. We attribute this to the negative oversampling, which allows DECENT to learn not only from the explicit annotations, but from a bigger portion of the ontology.

We perform a more in-depth analysis on the predictions of the models separating the entity types into their levels of granularity as defined originally in the UFET dataset. The corresponding metrics are presented in Table III. Here we notice that DECENT performs consistently better than MLMET regarding the macro-F1 score, although our approach was never trained with positive examples for the majority of entity types. DECENT also performs similarly to LITE, even though LITE has a more sophisticated encoding procedure and an additional loss objective in the form of label dependency, which uses inferred hierarchical relations among entity types.

Following the analysis performed by MLMET [9] we also analyze the performance with different types of entity mentions present in UFET. Here the mentions are classified as named entities (e.g. USA) when the words start with a capital letter, pronouns (e.g. they) defined by a pre-defined set and nominal (e.g. the president) for the rest of the mentions. The results for each category can be seen in Table IV. They illustrate the strengths and weaknesses of the three models. DECENT performs equally well for named entities, but slightly worse for nominal mentions compared to LITE. This is expected as nominal mentions are convoluted, and it can be difficult for the PLM to encode the headword properly. However, as LITE encodes the input together with the label, it should be easier for the model to focus on the critical parts in a long nominal mention. As we can see, our approach performs noticeably better for pronoun mentions, which is a surprise considering the full encoding of LITE.

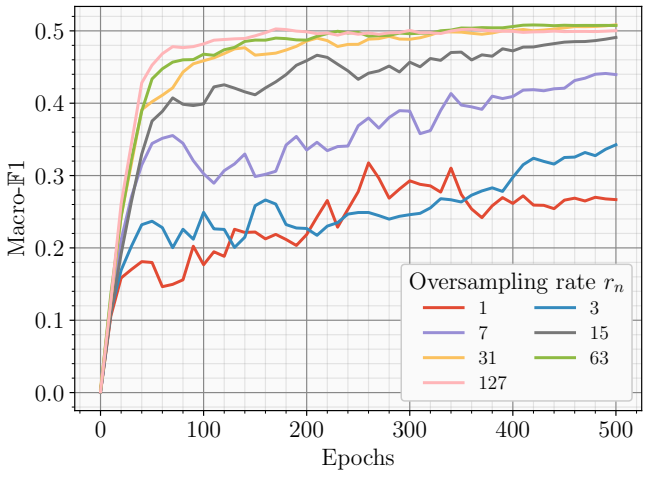

Fig. 3. Macro-F1 score on the validation set, throughout the training process, using different oversampling rates. The lines were smoothed using a moving average of window 3 for better readability

a) Oversampling Rate.: Being able to sample multiple negative types for each of the positive examples is one of the success factors of DECENT. More precisely, for each positive type, we sample r_{neg} negative types where r_{neg} refers to the negative oversampling rate. To analyze the impact of r_{neg}, we conduct experiments with the following negative oversampling rates: $\{1, 3, 7, 15, 31, 63, 127\}$. We show the development of the macro-F1 score on the validation split during training in Figure 3. Equally sampling of positive and negative samples ($r_{neg} = 1$) does not produce desirable results. The sampling rates $\{31, 63, 127\}$ seem to work evenly well and only differ when the model starts converging. Naturally, the higher the oversampling rate, the earlier the convergence starts, as the effective batch size is much larger. On the other hand, if the oversampling rate is too high, there is an increased risk of overfitting, which we can observe in a slight drop in prediction performance for $r_{neg} = 127$ at the end of the training. This assumption is further supported by the increasing validation loss for this rate towards the end of training. Therefore, choosing the negative oversampling rate as low as possible while maintaining prediction performance is desirable.

b) Zero-Shot and Few-Shot Performance.: One of the benefits of using the label semantics for entity typing is the ability to run predictions with unseen types. For instance, in the UFET dataset, the entity type *castle* is not present in the training split, however a model which learned to predict types such as *building*, can use the relatedness between the two labels to infer predictions for the first. Among MLMET, LITE, and DECENT, only LITE and DECENT can perform such generalization. MLMET, on the other hand, cannot directly predict new entity types without retraining; thus it must rely on weak supervision to observe examples from all the types during training. Therefore, we additionally evaluate the ability of the models for predicting entity types not present or rarely present in the human-annotated portion of the UFET dataset. The results are shown in Table V.

The evaluated entity types are split in three groups, depending on their frequency in the training set; zero, one to five, and six to ten shots. We compare micro-averaged F1 scores in these cases. The reason for not using macro-averaged F1 is that there might be sentences without entity types falling into these 3 groups, hence macro-scores might aggregate over ill-defined sample-wise scores.

DECENT performs better than MLMET on all three groups, although our model had only access, during training, to a limited number of training examples ($2K$ vs. $29M$). This also underlines the superiority of semantic typing, which we exploit by using label encodings. However, our approach falls behind compared to LITE. This suggests that pair-wise encoding with multiple self-attention layers has a strong generalization potential compared to our lightweight classification head.

c) Fine-grained Typing.: Our model is designed to predict types from an ultrafine-grained ontology; however, we compare the performance of DECENT under the fine-grained setting. For that, we use the commonly used OntoNotes [5] and FIGER [4] datasets. In addition, for LITE and DECENT we tested two different strategies, transferring the model trained on UFET to the fine-grained setup without fine-tuning, and training directly on the new datasets.

TABLE V
ZERO-SHOT AND FEW-SHOT PERFORMANCE IN MICRO-AVERAGED SCORES. $F1_0$ REFERS TO THE PERFORMANCE ON ENTITY TYPES NOT PRESENT IN THE HUMAN-ANNOTATED PORTION OF UFET. $F1_{m-n}$ REFERS TO ENTITY TYPES ANNOTATED BETWEEN *m* AND *n* TIMES.

Model	$F1_0$	$F1_{1\text{-}5}$	$F1_{6\text{-}10}$
MLMET	16.8	28.8	33.6
LITE	**23.8**	**32.1**	**36.4**
DECENT	17.0	31.2	35.6

TABLE VI
PREDICTION PERFORMANCE FOR THE FINE-GRAINED NAMED ENTITY TYPING BENCHMARKS ONTONOTES [5] AND FIGER [4].

Model	OntoNotes	FIGER
MLMET	85.4	-
LITE (transfer)	**86.6**	80.1
LITE (trained)	86.4	**86.7**
DECENT (transfer)	83.5	75.0
DECENT (trained)	81.2	84.6

The results of the fine-grained evaluation are observed in Table VI, where we see that LITE remains the better performing model. DECENT performs worse with both strategies. We can see that our approach is also suited for fine-grained named entity typing, but lacks in performance. We believe that in the fine-grained scenario, the negative oversampling effect is limited. Therefore, we deduce that our approach is more fitting for the ultrafine-grained named entity typing scenario, as the advantage of our model design becomes more apparent with a much larger ontology.

d) Ablation Study.: To analyze the influence of different components of our approach, we consider the following variants regarding the model design. First, we evaluate the performance of our model without the cross-attention between input and label tokens ($\text{DECENT}_{\text{NO CA}}$). The cross-encoded representations of the input and label are replaced with randomly initialized vectors that are optimized during training to keep the number of parameters in the classification head the same. These vectors are the same for every input-label combination. Second, we consider one version of DECENT trained by sampling only one negative type for every positive type ($\text{DECENT}_{\text{NO OS}}$), effectively omitting the negative oversampling ($r_n = 1$). Third, we consider a model variant that samples negative types solely from types present in the training split ($\text{DECENT}_{\text{NO OOV}}$). This configuration examines whether there is merit in predicting the annotation probability for type labels only considered for negative examples during training. Fourth, we evaluate a variant of our approach using RoBERTa-base [21], instead of RoBERTa-large as PLM backbone ($\text{DECENT}_{\text{BASE}}$).

The results are shown in Table VIII. Here we can see that all variants perform significantly worse than the proposed model.

TABLE VII
EXEMPLARY PREDICTIONS OF OUR MODEL WITH AND WITHOUT THE CROSS-ATTENTION MECHANISM. **CORRECT PREDICTIONS** ARE MARKED IN BOLD.

Input	Labels	Prediction
In practice, this conviction means that if the journalist continues his critical line in his publications, **he** will probably end up behind bars", said Adil Soz.	person, adult, investigator, journalist, male, professional, reporter, writer, communicator	DECENT: **person**, **adult**, **journalist**, **male**, **reporter**, **writer**, criminal w/o cross-attention: **person**
The Thomas Viaduct, located over Levering Avenue at the entrance to the Patapsco Valley State Park, is **the oldest stone curved bridge** in the world.	object, bridge, structure, construction	DECENT: **object**, **bridge**, **structure**, **construction** w/o cross-attention: **object**, **structure**, location, place, building, facility, road

TABLE VIII
PREDICTION PERFORMANCE ACHIEVED BY DIFFERENT VARIANTS OF OUR APPROACH.

Model	P	R	F1
DECENT	50.8	**48.7**	**49.7**
$\text{DECENT}_{\text{NO CA}}$	46.3	42.6	44.4
$\text{DECENT}_{\text{NO OS}}$	**61.1**	17.9	27.7
$\text{DECENT}_{\text{NO OOV}}$	45.5	43.4	44.4
$\text{DECENT}_{\text{BASE}}$	47.8	48.3	48.1

Negative oversampling has the largest effect on model performance among the modifications we considered for the ablation study. Having more negative examples effectively improves recall, resulting in a better balance between precision and recall. The performance of the other two variants compared to the baseline models (Table II) show the importance of these components in the architecture of DECENT for obtaining competitive results. In contrast, using the base version of the PLM instead of the large version does not have a strong negative impact on the performance of the model.

Two examples of predictions with and without the cross-attention mechanism are depicted in Table VII. Both examples underline how the cross-attention mechanism helps to focus on important information in the context, thereby improving the prediction.

B. Qualitative Analysis

In order to illustrate the differences between the predictions of DECENT and LITE, we present in Table IX a selection of examples from the validation and test splits of UFET and the final types selected by both models. *E1* and *E2* are successful cases, favoring the prediction of DECENT in comparison with LITE. In the first case, our model predicts the same number of types as LITE, but with a higher precision. This is consistent with the results found in Table IV regarding performance on pronouns. The second one is a case of better recall, in which both models have perfect precision, but DECENT classifies the mention with more types, thus having better coverage of the ground truth.

E3, on the other hand, shows better recall from LITE, that includes all three types in the predictions, whereas our model only includes one. We speculate that the reason behind this is the label dependency loss included in LITE. In this case *organization*, *government* and *administration* are types which are likely to appear together, showing some degree of correlation.

C. Runtime Performance

Although the classification performance is important to understand the quality of the model predictions, our main contribution is in the efficiency during training and prediction. Our hypothesis is that the decoupled encoding of input and label, in combination with a dedicated cross-attention classification head, will reduce the time needed to predict which of the entity types correspond to a particular entity mention. For training, our model makes use of only the human annotated portion of the UFET dataset, which, along with the lightweight model, should result in lower training times.

In order to compare the models, we measure the runtimes on a single NVIDIA DGX A100 (40GB) GPU with 32 CPU cores and 32GB RAM. We use each model's maximum possible batch size for a fair comparison. We use the training and validation splits of the UFET dataset for measuring training and prediction times, respectively. To approximate the full training times we execute the training until the epoch runtime becomes stable, and then extrapolate this value according to the suggested number of epochs for each model. For MLMET we use the number of steps according to the published implementation[3], and extrapolate according to the number of samples used for each stage. For LITE, we measure epoch duration and extrapolate with the suggested number of epochs 2000, and finally, DECENT is trained for 500 epochs.

For prediction, we use the validation split of the UFET dataset and measure the runtime for each model. Prediction with MLMET is straightforward, due to its multi-label classifier nature. In the case of LITE, it is necessary to encode each input-type pair. Prediction with our approach involves encoding all inputs and then feeding them along with each pre-encoded type into the cross-attention classification head, then comparing each probability against the pre-defined threshold.

The extrapolated runtimes for training and prediction on the validation set are shown in Table X. For training, we observe that the training of MLMET takes considerably longer

[3]https://github.com/HKUST-KnowComp/MLMET

TABLE IX
EXEMPLARY PREDICTIONS OF OUR MODEL, MLMET AND LITE. **TRUE POSITIVES** ARE MARKED IN BOLD.

	Input	Labels	Prediction
E1	In practice this conviction means that if the journalist continues his critical line in his publications, **he** will probably end up behind bars", said Adil Soz.	person, adult, investigator, journalist, male, professional, reporter, writer, communicator	DECENT: **person**, **adult**, **journalist**, **male**, **reporter**, **writer**, criminal MLMET: **person**, **journalist**, **male**, **reporter**, **writer**, criminal, man LITE: **person**, **adult**, **journalist**, **reporter**, **writer**, criminal, defendant, convict
E2	The Thomas Viaduct, located over Levering Avenue at the entrance to the Patapsco Valley State Park, is **the oldest stone curved bridge** in the world.	object, bridge, structure, construction	DECENT: **object**, **bridge**, **structure**, **construction** MLMET: **object**, **bridge**, **structure**, construction LITE: **object**, **bridge**, **structure**
E3	**It** urged Americans to maintain a low profile and be alert between Aug. 11 and Aug. 16.	organization, government, administration	DECENT: **organization** MLMET: **organization**, group, agency, report LITE: **organization**, **government**, **administration**, authority, document, message, report

TABLE X
TIME NEEDED FOR MODEL TRAINING AND PREDICTION

	MLMET	LITE	DECENT
Training Time	166h†	15h	4.5h
Prediction Time ‡	1.6s	3h	84s
Total⋆	167h	215h	6h

†33h with existing weak labels
‡for a single iteration over the validation set
⋆assuming validation is performed every 2% of training progress.

compared to the other model. That is because MLMET is trained on nearly 30M distantly supervised samples. This impacts both the pre-processing of those automatically generated labels, and the length of the different training iterations. Due to its multi-label classification nature, all the entity types need to go through the model during training. Additionally, there are multiple stages for training the model with different datasets. On the other hand, training LITE is faster than training MLMET. LITE can generalize from just a portion of entity types to the whole type ontology by making use of the label semantics, NLI, and label dependencies. Finally, DECENT is the fastest in training time, due to the simplicity of the model, and the label generalization benefits of using PLMs. Another advantage of DECENT is the negative oversampling, allowing the model to see more samples in less iterations.

In terms of prediction runtime, the disadvantages of LITE can be clearly noted, resulting from the pair-wise input-type encoding using a transformer architecture. MLMET has the best performance, due to the simpler encoding of the inputs compared to the others. Lastly, our model is significantly slower than MLMET, as a result of the more complex transformer encoder. But compared with LITE, we have a 131x runtime performance gain.

In general, if we consider both, training and prediction runtimes, DECENT is more efficient than the other evaluated approaches. Assuming a full training of the models using the UFET dataset, with regular validation steps, we see that our approach can be trained at least 25x faster than MLMET and LITE.

V. CONCLUSION

In this work, we presented DECENT, a model for the ultrafine named entity typing task, which is efficient in training and prediction runtime. In our model design, we decouple the encoding of input and the entity type label and thereby circumvent the encoding of every input-label combination with a pretrained language model. We defer the combination of input and type labels to a designated cross-attention classification head. This classification head allows us to query the input for a specific label without fully encoding them with a pretrained language model. Our study shows that this approach is 130 times faster than the state-of-the-art model LITE [10] while maintaining competitiveness regarding prediction performance.

The model design also allows us to perform semantic typing and classify entity types not seen during training. In addition, the architecture facilitates the oversampling of input types during training. We can sample more type labels from the ample label space for a given input and speed up the training process. Specifically, our model can be trained significantly faster than state-of-the-art approaches.

Although we present the benefits of DECENT, with the evaluation setup of the UFET dataset, there are some limitations of

the model worth mentioning. First, the hierarchical structure of the type ontology is not directly employed. Other models like LITE make use of such information. Also, our model depends on the representations computed with a transformer model, thus depending on its ability to encode a vocabulary. Models like BERT [20], define a fixed token vocabulary, potentially limiting the ability to represent all the words of a language. That might impact particularly the encoding of entity types in domain-specific settings with specialized names. As discussed in subsection IV-A, in scenarios where the amount of entity types is reduced and data and run-time efficiency are not critical, the benefits of our approach are less prominent.

For future work, we would like to understand under which circumstances models are confident enough to perform a zero-shot prediction and include this knowledge into a new approach. In this context, we will also experiment with the success of transferring approaches between different datasets and across different domains [28], [29]. Furthermore, we would like to investigate model biases during named entity typing and ways to mitigate them, extending the work of [30]. Lastly, we think contextualized entity types yield an exciting opportunity for future research, e.g., the entity type *director* can have multiple meanings. Combining the entity type with different definitions [31], [32] or providing an example of the type's usage should allow a pretrained language model to produce embeddings of higher quality.

ACKNOWLEDGMENT

This research was partially funded by the HPI Research School on Data Science and Engineering.

REFERENCES

[1] E. F. Tjong Kim Sang and F. De Meulder, "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition," in *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pp. 142–147, 2003.

[2] M. Fleischman and E. Hovy, "Fine grained classification of named entities," in *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002.

[3] C. Lee, Y.-G. Hwang, H.-J. Oh, S. Lim, J. Heo, C.-H. Lee, H.-J. Kim, J.-H. Wang, and M.-G. Jang, "Fine-grained named entity recognition using conditional random fields for question answering," in *Information Retrieval Technology* (H. T. Ng, M.-K. Leong, M.-Y. Kan, and D. Ji, eds.), AIRS 2006, (Berlin, Heidelberg), pp. 581–587, Springer Berlin Heidelberg, 2006.

[4] X. Ling and D. S. Weld, "Fine-grained entity recognition," in *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, p. 94–100, AAAI Press, 2012.

[5] D. Gillick, N. Lazic, K. Ganchev, J. Kirchner, and D. Huynh, "Context-dependent fine-grained entity type tagging," *CoRR*, vol. abs/1412.1820, 2014.

[6] T. Lin, Mausam, and O. Etzioni, "No noun phrase left behind: Detecting and typing unlinkable entities," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, (Jeju Island, Korea), pp. 893–903, Association for Computational Linguistics, July 2012.

[7] L. Del Corro, A. Abujabal, R. Gemulla, and G. Weikum, "FINET: Context-aware fine-grained named entity typing," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, (Lisbon, Portugal), pp. 868–878, Association for Computational Linguistics, Sept. 2015.

[8] E. Choi, O. Levy, Y. Choi, and L. Zettlemoyer, "Ultra-fine entity typing," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Melbourne, Australia), pp. 87–96, Association for Computational Linguistics, July 2018.

[9] H. Dai, Y. Song, and H. Wang, "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)*, (Online), pp. 1790–1799, Association for Computational Linguistics, Aug. 2021.

[10] B. Li, W. Yin, and M. Chen, "Ultra-fine entity typing with indirect supervision from natural language inference," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 607–622, 2022.

[11] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (San Diego, California), pp. 260–270, Association for Computational Linguistics, June 2016.

[12] A. Akbik, D. Blythe, and R. Vollgraf, "Contextual string embeddings for sequence labeling," in *Proceedings of the 27th International Conference on Computational Linguistics*, (Santa Fe, New Mexico, USA), pp. 1638–1649, Association for Computational Linguistics, Aug. 2018.

[13] Y. Nie, Y. Tian, Y. Song, X. Ao, and X. Wan, "Improving named entity recognition with attentive ensemble of syntactic information," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, (Online), pp. 4231–4245, Association for Computational Linguistics, Nov. 2020.

[14] X. Wang, Y. Jiang, N. Bach, T. Wang, Z. Huang, F. Huang, and K. Tu, "Automated concatenation of embeddings for structured prediction," 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)*, (Online), pp. 2643–2660, Association for Computational Linguistics, Aug. 2021.

[15] S. Sekine, "Extended named entity ontology with attribute information," in *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, (Marrakech, Morocco), European Language Resources Association (ELRA), May 2008.

[16] W. Xiong, J. Wu, D. Lei, M. Yu, S. Chang, X. Guo, and W. Y. Wang, "Imposing label-relational inductive bias for extremely fine-grained 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)*, (Minneapolis, Minnesota), pp. 773–784, Association for Computational Linguistics, June 2019.

[17] Y. Onoe and G. Durrett, "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)*, (Minneapolis, Minnesota), pp. 2407–2417, Association for Computational Linguistics, June 2019.

[18] Y. Onoe, M. Boratko, A. McCallum, and G. Durrett, "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)*, (Online), pp. 2051–2064, Association for Computational Linguistics, Aug. 2021.

[19] L. Vilnis, X. Li, S. Murty, and A. McCallum, "Probabilistic embedding of knowledge graphs with box lattice measures," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Melbourne, Australia), pp. 263–272, Association for Computational Linguistics, July 2018.

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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)*, (Minneapolis, Minnesota), pp. 4171–4186, Association for Computational Linguistics, June 2019.

[21] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019.

[22] A. Williams, N. Nangia, and S. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," in *Proceedings of*

the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), (New Orleans, Louisiana), pp. 1112–1122, Association for Computational Linguistics, June 2018.

[23] S. Pang, J. Ma, Z. Yan, Y. Zhang, and J. Shen, "FASTMATCH: Accelerating the inference of BERT-based text matching," in *Proceedings of the 28th International Conference on Computational Linguistics*, (Barcelona, Spain (Online)), pp. 6459–6469, International Committee on Computational Linguistics, Dec. 2020.

[24] Q. Liu, H. Lin, X. Xiao, X. Han, L. Sun, and H. Wu, "Fine-grained entity typing via label reasoning," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, (Online and Punta Cana, Dominican Republic), pp. 4611–4622, Association for Computational Linguistics, Nov. 2021.

[25] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2015.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), p. 5998–6008, Curran Associates, Inc., 2017.

[27] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, (Online), pp. 38–45, Association for Computational Linguistics, Oct. 2020.

[28] C. Jia, X. Liang, and Y. Zhang, "Cross-domain NER using cross-domain language modeling," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (Florence, Italy), pp. 2464–2474, Association for Computational Linguistics, July 2019.

[29] Z. Liu, X. Yan, T. Yu, W. Dai, Z. Ji, S. Cahyawijaya, A. Madotto, and P. Fung, "Crossner: Evaluating cross-domain named entity recognition," *arXiv*, vol. abs/2012.04373, Dec. 2020.

[30] N. Xu, F. Wang, B. Li, M. Dong, and M. Chen, "Does your model classify entities reasonably? diagnosing and mitigating spurious correlations in entity typing," *arXiv*, vol. abs/2205.12640, May 2022.

[31] R. Obeidat, X. Fern, H. Shahbazi, and P. Tadepalli, "Description-based zero-shot fine-grained 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)*, (Minneapolis, Minnesota), pp. 807–814, Association for Computational Linguistics, June 2019.

[32] R. Aly, A. Vlachos, and R. McDonald, "Leveraging type descriptions for zero-shot named entity recognition and classification," 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)*, (Online), pp. 1516–1528, Association for Computational Linguistics, Aug. 2021.