Unsupervised Ultra-Fine Entity Typing with Distributionally Induced Word Senses

Özge Sevgili[1][0000−0002−3078−3478], Steffen Remus[1][0000−0003−4303−8781], Abhik Jana[2][0000−0002−4485−0002], Alexander Panchenko[3][0000−0001−6097−6118], and Chris Biemann[1][0000−0002−8449−9624]

[1] Language Technology Group, Universität Hamburg, Informatikum, Vogt-Kölln-Straße 30, 22527 Hamburg, Germany
oezge.sevgili.ergueven@studium.uni-hamburg.de
{steffen.remus,chris.biemann}@uni-hamburg.de
[2] School of Electrical Sciences, Indian Institute of Technology Bhubaneswar, PIN - 752050, Odisha, India
abhikjana@iitbbs.ac.in
[3] Center for Artificial Intelligence Technologies, Skolkovo Institute of Science and Technology, Bolshoy Boulevard 30, bld. 1, 121205, Moscow, Russia
A.Panchenko@skoltech.ru

Abstract. The lack of annotated data is one of the challenging issues in an ultra-fine entity typing, which is the task to assign semantic types for a given entity mention. Hence, automatic type generation is receiving increased interest, typically to be used as distant supervision data. In this study, we investigate an unsupervised way based on distributionally induced word senses. The types or labels are obtained by selecting the appropriate sense cluster for a mention. Experimental results on an ultra-fine entity typing task demonstrate that combining our predictions with the predictions of an existing neural model leads to a slight improvement over the ultra-fine types for mentions that are not pronouns.

Keywords: unsupervised ultra-fine entity typing · entity typing with induced word senses.

1 Introduction

Ultra-fine entity typing (UFET) is the task of assigning semantic types to an entity mention in context [6]. There exist numerous diverse types, e.g., consider the sentence - "Olympic National Park came into the national park system in 1938 and has been a favorite destination for naturalists and tourists ever since." - the types for "Olympic National Park" are ***geographical_area, national_park, space, region, location, landmark, park, place***. Ultra-fine types can be helpful for natural language understanding tasks, for example, Sui et al. [36] leverage ultra-fine entity types from entity descriptions in a zero-shot entity linking task. Yet, those large type sets lead to difficulties in annotating mentions for humans [7].

This causes a challenge of the scarcity of annotated data. There are more than 10K labels, in this task. Therefore, methods to create automatic annotations have been proposed. We explore the leverage of distributionally induced word senses, since we believe the induced word senses can help to understand and disambiguate the given mention.

For this challenge, some studies propose to generate labels to be used as distant supervision with different strategies [6, 7]. For example, Dai et al. [7] modify sentences by inserting [MASK] tokens with hypernym extraction patterns and obtain predictions from the pre-trained language model (PLM). In a similar vein, Ding et al. [9] and Li et al. [18] also leverage PLMs in different scenarios with different goals, i.e., to apply prompt learning or re-formulate the task, respectively. Qian et al. [32] and Liu et al. [20] generate labels under the setting without access to a knowledge base rather based on a large amount of data, and the underlying techniques of their solutions are quite similar to our investigation. However, their evaluations are on a fine-grained entity typing (FET) task. We rather focus on UFET task with richer type set described as free-form phrases. There are also some works for zero-shot [39, 23, 9], and unsupervised [12, 13] way of solutions, mostly using either a labeled data or knowledge base, e.g. Wikipedia information. In our study, we do not make use of any such knowledge bases or labeled data.

In this work, to produce ultra-fine types, we leverage the API of the JoBimText framework [35, 4], which provides sense clusters with hypernym labels (i.e., IS-As) for a queried term in an unsupervised and knowledge-free way based on a distributional thesaurus. The appropriate sense for a particular mention is selected based on the cosine similarity between vectorial representations of contextual information and each sense cluster information of the mention. The hypernym labels for the selected sense are our final prediction. Our goal in this work is to explore the potential of this approach in UFET task. Thus, the contribution of this paper is the investigation of the labels from the JoBimText in the UFET task in an unsupervised way. We experiment a combination of the neural approach predictions by Choi et al. [6] with JoBimText based predictions to explore their complementarity. We utilize predictions from Choi et al. [6] here, since they set the baselines while releasing the UFET dataset. With this combination, we observe a slight improvement of the F1 score for ultra-fine types for explicit mentions.

2 Related Work

Ultra-Fine Entity Typing There are several lines of research on UFET [6] and FET [19, 10]. FET contains smaller set of labels, e.g., 112 types in Ling and Weld [19], and labels are in an ontology, e.g., *location/city*. UFET is more diverse and finer grained, containing more than 10K labels as free-form noun phrases without an ontology. Some studies investigate hierarchies/dependencies or correlations in the types in different ways [15, 40, 37, 21, 24, 22], inter alia. While most attention is on English typing, several work on other languages [17].

For the challenge of the scarcity of annotated data, Li et al. [18] re-formulate the task as natural language inference (NLI) and leverage indirect supervision from NLI. Some works attempt to create more labeled data automatically and use it as distant supervision. A typical way is to obtain types from the knowledge base after linking the mention to the entity [6]. Choi et al. [6] propose to utilize head words of mentions as a distant supervision. Dai et al. [7] insert [MASK] token with a few tokens (e.g., "such as") to create an artificial Hearst pattern [11] close to a mention to retrieve the predictions from BERT masked language model [8]. Qian et al. [32] attempt to tackle FET without a knowledge base, in which they generate automatically labeled data from a large-scale unlabeled corpus and then propose a training method using this data. Since the automatically labeled data might contain noise, several studies provide solutions to denoise it [25, 26]. Some others deal with also zero-shot scenarios [39, 23, 38, 18, 9] inter alia. Among them, works Zhou et al. [39] and Ding et al. [9] rely on unlabeled data. Ding et al. [9] apply prompt-learning to generate type labels using PLMs, and for zero-shot set-up, they further propose a self-supervised method relying on contrastive learning. Zhou et al. [39] utilize knowledge base information. Huang et al. [12, 13] provide unsupervised solution, again using knowledge base information.

Among all, our study is more relevant to Qian et al. [32], in terms of applying a Hearst pattern to large data and applying the clustering without accessing the knowledge base. In a similar vein, Liu et al. [20] propose an NLP system that supports unsupervised FET by applying a Hearst pattern and clustering. However, both evaluate on FET task, while our focus is on UFET, which contains more fine-grained types. In Dai et al. [7], the labels are generated automatically from PLMs, and Ding et al. [9] provide also a zero-shot solution. In comparison, our study investigates particularly the usage of the JoBimText API on this task, which is a simpler scenario than their models. In comparison with Zhou et al. [39], Huang et al. [12, 13], we do not use the knowledge base information.

JoBimText Applications Several works utilize information provided by JoBimText in different tasks [14, 1, 30], inter alia. Among them, the most similar studies might be unsupervised knowledge-free word sense disambiguation by Panchenko et al. [27, 29], in which a word in a context is disambiguated using the induced senses of JoBimText. Inspired by them, we conduct a similar approach to the UFET task.

3 Method

JoBimText Framework In our work, we generate labels relying on the JoBimText framework [4] as an end-user of the API[4] [35] provided by this framework. The underlying technology of this framework involves a holing operation as the first step, which performs the split of a term (Jo) and a contextual feature (Bim) based on structural observations of text (e.g., dependency parsing). Next,

[4] http://ltmaggie.informatik.uni-hamburg.de/jobimviz/#

by pruning these terms and features (based, e.g., on some significance scores, like LMI [16]) and by aggregating terms based on their overlapping contextual features, a distributional thesaurus (DT), a graph of terms, is constructed. Furthermore, they cluster an ego/neighboring graph (i.e., a sub-graph containing similar terms to a particular term) of a DT entry (i.e., term) using the Chinese Whispers algorithm [2] to get a sense information of an entry in terms of its similar entries. Each induced sense is labeled based on the information of IS-A relationship between terms with their frequencies, collected by applying IS-A (hypernym) patterns [11] on a text collection. The API [35] allows to access the information of the JoBimText framework (for more information, see [3, 35, 4, 34]).

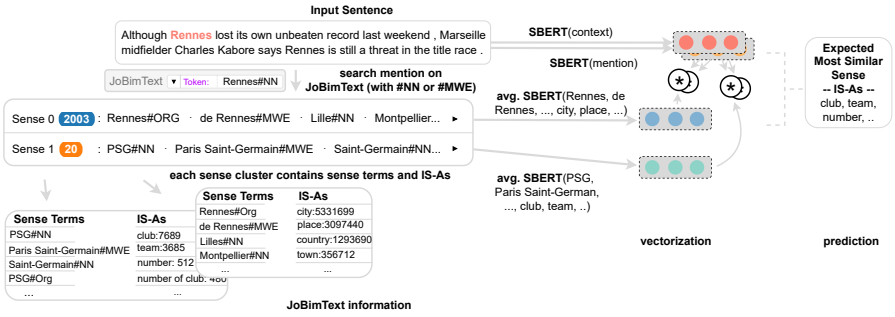

Fig. 1. A sample prediction process: search for a mention on JoBimText (mimicked from API) to get sense clusters containing terms and labels (IS-As), vectorize clusters by averaging SBERT vectors of each term (and label), compute cosine similarities between context (and mention) vector and clusters, and obtain IS-As of the most similar cluster as a final prediction.

Method In our setup, we query a mention into the JoBimText API and obtain sense clusters of this mention. The most appropriate sense cluster is selected based on the vectorial similarity between the context that the mention appeared in, and a sense cluster. To compute this similarity, each sense cluster is vectorized by using the sense terms (and labels) with Sentence BERT (SBERT) [33] (note that some clusters may not have hypernyms/terms, so we skip them). And context (and mention) is also vectorized using SBERT. The hypernym labels of the most similar sense are the final type predictions, as exemplified in the Figure 1.

We query a mention with NN (noun) or MWE (multi-word expression) tag, depending on whether a mention contains a single token or multiple tokens. Other tags, e.g., ADJ, are not considered since the mentions in the UFET dataset

[6] are pronouns, nominal expressions, and named entity mentions[5]. JoBimText might not provide information for all mentions, like long phrases, e.g., "the building , a violation of the Clean Air Act". Thus, we try to shorten the mentions in several ways: extraction of a head word (root word) of the mention or extraction of n-grams located close to the beginning/end of the mention due to an observation of some mentions (e.g., "shipments for the month", "the social and economic development"). Note that Choi et al. [6] also utilize a head word of the mention, however, they directly take it as a weak label, while our goal is to shorten the mention to search later on the JoBimText. Additionally, singularization is applied for the mention as an additional configuration based on the observation that the singular version of some mentions is in the JoBimText, while the plural is not, e.g., "shareholders". JoBimText might still not provide any information for the short mentions, for which we assign a *person* label as it is the most used label in the development set. The coverage for our reported results, in Table 2, is 87.74.

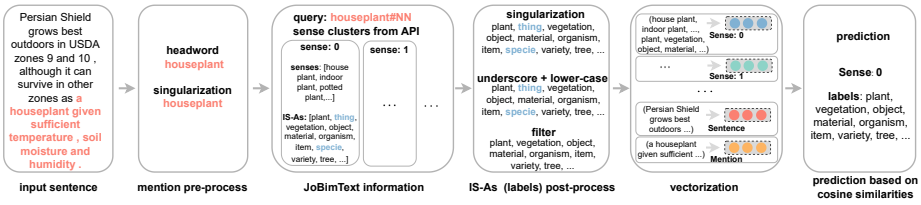

Fig. 2. A sample overview of the process is illustrated. In this sample, head word is used to shorten mention, and post-process is shown for labels of sense: 0, and IS-A labels are included for vector representation. The prediction is based on the cosine similarities of sentence and sense vectors, and mention and sense vectors.

In a similar vein, we apply some post-processing steps due to some mismatches between predictions (i.e., IS-As) and the type vocabulary of Choi et al. [6] (e.g., predictions may involve *people*, while the vocabulary contains *person*, not *people*). We first follow Dai et al. [7] for post-processing: the labels are singularized and filtered if they are not in the type vocabulary. In addition to them, we add underscores for the multi-token labels, e.g., *tennis_player*, and make them lower-case if not. We remove a label *thing* among our predicted labels since we consider it as a noisy label due to its high frequency. A sample overview containing pre-/post-process steps is shown in Figure 2.

[5] For named entity mentions, some other tags, like e.g. Org, Loc, would be helpful, however, the goal of entity typing itself to produce such labels, and so, we use only NN/MWE tags.

4 Experiments

Dataset The experiments are performed on an English UFET dataset provided by Choi et al. [6]. The type vocabulary contains 10331 labels. The dataset consists of a training set, development set, and test set, each with 1998 samples.

Baselines

- **first cluster or random cluster** There is an order of sense clusters in JoBimText, based on the score of related terms. We either always choose the first cluster with terms and labels, or choose any sense randomly. The same pre- and post-process steps are applied as the configuration, which will be explained in 4.1.
- **Choi et al. (2018) [6]** generate representations through the pre-trained word embeddings, bi-directional LSTM, CNN, and train the model with a multitask objective.
- **Dai et al. (2021) [7]** generate labels through the BERT masked language model and leverage the generated labels in the training entity typing model.
- **Li et al. (2022) [18]** treat each sentence as a premise and generate a hypothesis through the candidate type to formulate the task as NLI. Here, the learning objective is learning-to-rank.

4.1 Implementation Details

JoBimText provides many DTs including different language supports from various corpora. In this study, we use the DT constructed from the DepCC corpus [28]. It is built from the web-scale data from the Common Crawl[6], which provides access to large amounts of data. As a Sentence BERT model, we utilize "all-mpnet-base-v2", since it is the best performing one (on average performance) among current models[7].

We consider several parameters or features and select amongst them based on a simple manual search, in our implementation, as shown in Table 1. In Table 1, the first row represents the methodology, where we shorten the mention before searching in the JoBimText API, where n-grams are extracted using NLTK [5][8] and head words are extracted with the stanza library/toolkit[9] [31]. For some mentions (three mentions in the development set and eight mentions in the test set), there can be more than one head word, for which we use the first head word by default. The second item in the table is applying singularization to mentions

[6] The JBT DepCC model uses a 2016 snapshot of the Common Crawl (https://commoncrawl.org/).

[7] https://www.sbert.net/docs/pretrained_models.html

[8] Some mentions start with "a, an, the" (or upper case of them), for which we take the tokens after the first one, if we experiment with the beginning of mention. The punctuations (like, e.g. .,"–) as well as the tokens "-LRB-" and "-RRB-" that we think they would be used for brackets are removed.

[9] https://github.com/stanfordnlp/stanza

Table 1. Parameters or features, their possible values, and the selection based on our simple manual search on the development set.

	Parameter/Feature	Possible values	Selection
1	mentions shorten options	- head word - n-gram beginning tokens (for n=1 - 2 - 3) - n-gram end tokens (for n=1 - 2 - 3)	head word
2	apply singularization to mention	true - false	true
3	cluster types	200,200 - 200,50 - 50,50	50,50
4	number of terms for representation	10 - 20 - 30	10
5	number of labels for representation	false - 10 - 20 - 30	10
6	weighting average	false - rank - cos. sim.	false
7	include mention similarity	true - false	true
8	include mention	true - false	false
9	number of predictions	5 - 10 - 15 - 20	10

before the search in API, for which we use inflect library[10], however, it might result in some mistakes for some cases as discussed in Section 5. To avoid the case that the term ends with "s" (e.g. "access"), we double-check its morphological property using the stanza toolkit whether it is singular or plural[11]. The third row in Table 1 is for cluster types available in the API to determine the number of some entries[12] for the Chinese Whispers algorithm. Cluster representations are created using the sense terms, and the fourth item in the table is to determine how many terms to include. Similarly, the cluster labels are optionally included while creating representations, as shown in the 5th row with the number options. While averaging, weighting is possible with weights either from the similarity between a mention and a considered term/label or from the order present in the JoBimText with the formula, 1/its order (meaning if it is ranked first, weight becomes 1, second: 0.5, third: 0.33, fourth: 0.25..), as in the sixth entry in the table. While computing similarities for the final decision, similarities are between a context and a sense, or also including the similarities between the mention and a sense. Item eight is to determine that the context contains either only left- and right-context words or also mention words, inserted in between. The last entry is for the number of predicted labels.

We try to find the best parameters and features in the development set with a simple manual search. Among the experiments, the configuration with the best F1 score, which we can reach so far, consists of the parameters and features, as shown in the last column of Table 1.

4.2 Evaluation

In Table 2, we report P (precision), R (recall), and F1 by following recent works [7, 18, 24]. The scores are computed with the evaluation script provided by Choi

[10] https://pypi.org/project/inflect/

[11] Here, we cross-check only the last token of the mention and we dismiss the cases of the singularized word that is located in different place. For example, "princess of Brunswick-Wolfenbuttel" is singularized as "princes of brunswick-wolfenbuttell".

[12] http://ltmaggie.informatik.uni-hamburg.de/jobimtext/documentation/sense-clustering/

Table 2. Unsupervised ultra-fine entity typing performance on UFET test set. without pronouns: the results are for the mentions that are not pronouns (1210 samples, in the test set), 5 preds.: the results contain the first 5 predictions from Choi et al. [6] and our first 5 predictions, Ours-PRP: pronoun mentions are searched with PRP tag.

Model	Total			Coarse			Fine			Ultra-Fine		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
first cluster	17.4	18.1	17.7	34.1	42.9	38.0	13.0	18.0	15.1	6.5	9.6	7.7
random cluster (avg. of 5 runs)	16.8	16.2	16.5	43.4	43.3	43.4	12.5	14.5	13.4	5.5	7.5	6.3
Li et al. (2022) [18]	53.3	**56.4**	**50.6**	-	-	-	-	-	-	-	-	-
Dai et al. (2021) [7]	**53.6**	45.3	49.1	-	-	-	-	-	-	-	-	-
Choi et al. (2018) [6]	47.1	24.2	32.0	60.3	63.4	61.8	41.2	38.7	39.9	42.2	9.4	15.4
Ours	20.1	19.3	19.7	42.3	41.9	42.1	16.1	20.0	17.8	8.9	11.0	9.8
Ours-PRP	25.6	22.0	23.7	58.7	53.3	55.9	23.2	20.0	21.5	9.5	12.0	10.6
Choi et al. (2018) [6] + Ours	23.2	33.0	27.3	49.0	74.4	59.0	24.4	46.2	31.9	13.9	17.7	15.5
Choi et al. (2018) [6] + Ours (5 preds.)	27.3	30.3	28.7	51.9	72.5	60.5	30.6	43.8	36.0	16.7	14.4	15.5
Choi et al. (2018) [6] + Ours-PRP	25.2	33.4	28.7	54.4	74.5	62.9	29.9	46.2	36.3	14.5	18.4	16.2
Choi et al. (2018) [6] + Ours-PRP (5 preds.)	29.4	30.4	29.9	56.7	72.8	63.8	34.3	44.0	38.5	17.4	14.7	15.9
without pronouns												
Choi et al. (2018) [6]	46.7	19.6	27.7	50.3	50.8	50.5	44.1	36.0	39.6	**50.2**	7.8	13.4
Ours	18.8	25.4	21.6	46.0	47.4	46.7	23.1	34.7	27.8	12.2	17.7	14.5
Choi et al. (2018) [6] + Ours	21.1	33.4	25.9	43.0	68.6	52.8	25.2	48.7	33.2	13.8	**21.1**	**16.7**
Choi et al. (2018) [6] + Ours (5 preds.)	26.4	29.6	27.9	46.4	65.6	54.3	31.8	44.9	37.2	17.7	16.8	**17.3**

et al. [6][13]. We report the results of our method on the test set by Choi et al. [6] using the features/parameters as explained in the previous section. We also report the results of the first/random cluster baseline (with the same possible parameters applied without the ones to choose the best cluster). The improvement over first and random cluster baselines suggests that our method is able to disambiguate the induced sense at some level. Additionally, the first cluster baseline scores are pretty good, so we can say that the first sense among the induced senses is prominent in the dataset. We perform an alternative experiment by searching pronoun mentions with PRP tag rather than NN/MWE and we can see some improvement there[14]. Most of the recent works are supervised (e.g., [18, 7], in Table 2), and thus we cannot directly compare them with our results. For this reason, we combine our predictions with the predictions from the Choi et al. [6] model and check if additional predictions from our approach improve the scores[15]. They release their best model and the prediction file from this model[16], and for this experiment, we use only this prediction file.

Our solution cannot produce good hypernyms for pronouns. Therefore, we also compare the predictions for explicit mentions only, excluding pronouns[17].

[13] https://github.com/uwnlp/open_type/blob/master/scorer.py

[14] Note that the coverage is changed to 80.73 in this experiment.

[15] We directly concatenate the predictions, and then we keep unique labels (technically we make the concatenation set).

[16] http://nlp.cs.washington.edu/entity_type – Pretrained model/outputs (best_model/test.json)

[17] We consider pronouns: "i, me, myself, we, us, ourselves, he, him, himself, she, her, herself, it, itself, they, them, them-

Additionally, we collect our first 5 predictions, for each mention, and the first 5 predictions from the model by Choi et al. [6]. Based on the combination results

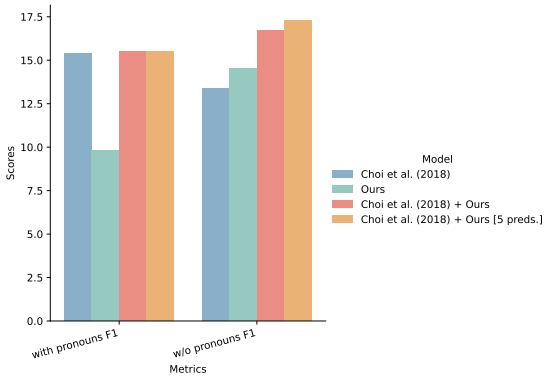

Fig. 3. Results on ultra-fine granularity are shown.

shown in Table 2, the ultra-fine F1 scores are improved when the predictions of explicit mentions are combined in both cases all predictions and 5 predictions, which suggests our labels are complementary to the predictions by Choi et al. [6], in this set-up, as also shown in Figure 3. Overall, we experiment with all granularities but only ultra-fine worked well for the tested dataset, which is a potential limitation of the approach.

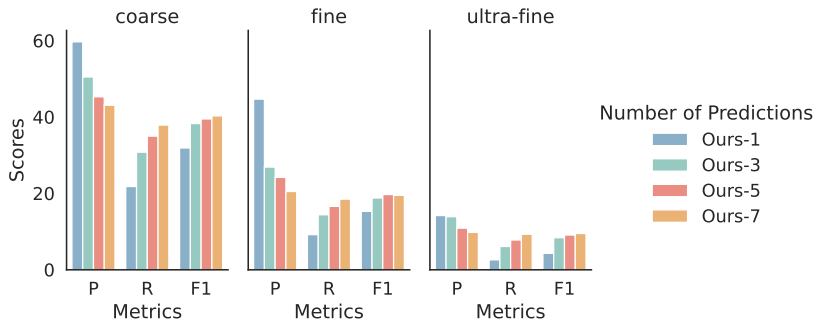

Fig. 4. Our results with different number of predictions on test set are shown. Ours-{1,3,5,7}: the results contain the first {1,3,5,7} prediction(s).

selves, you, yourself" (and upper case of them), with references – https://github.com/HKUST-KnowComp/MLMET/blob/main/prep.py#L9 and https://en.wikipedia.org/wiki/English_pronouns#Full_list

Table 3. A sample per category: Mentions are marked as red, the predictions, and gold labels are examplified for each category. 1: Context-dependent or pronoun mentions, 2: JoBimText does not contain the referred sense information, 3: The labels are not the matching perfectly, 4: Some preprocessing issues, 5: The JoBimText labels are not that relevant, 6: Wrong sense selection.

Cat.	Context	Search Mention	Predictions	True Labels
1	"They need to allow the international humanitarian organizations full and unobstructed access because they are obstructing access right now , " -Sollom- said .	Sollom	no information (assigned person)	person
2	Following a full-scale tour in support of its previous album , -Binaural- -LRB- 2000 -RRB- , Pearl Jam took a year-long break .	Binaural	company, format, feature, technology, brand, variety, mode, track, stuff, film	object, album
3	" People are getting deported for -even minor offenses- like not having an ID or a driver 's license , " said Cesar Espinosa of America for All , a group that helps immigrants in Houston .	offense	offense, **crime**, activity, charge, incident, matter, case, law, act, **violation**	**violation**, difficulty, wrongdoing, error, consequence, problem, trouble, event, **crime**
4	The devastating 21 September earthquake of 1999 left Taiwan 's landscape covered in scars , but researchers discovered that -the places least damaged by the quake- were areas of natural forest .	damaged	no information (assigned person)	city, area, location, town, region, space
5	Jack Byrne , chairman of Fireman 's Fund , said this disaster will test the catastrophe reinsurance market , causing -these rates- to soar .	rate	disease, illness, condition, information, cancer, side_effect, event, number, effect, rate	share, value, capital, price, stock
6	It was formerly located at Six Flags -New Orleans before it was relocated to Six Flags Fiesta Texas- and rethemed to Goliath .	Orleans	venue, stadium, attraction, **place**	town, city, placement, space, state, location, **place**, park

We also take the first 1, 3, 5, 7 prediction(s) as the final prediction(s) and see the decrease of precision and the increase of recall, when the number of predictions are increased, as shown in Figure 4.

5 Error Analysis and Limitations

Error Analysis We conduct an error analysis on 100 random samples in the test set shown in Table 3, from our predictions explained in Section 4.2. Based on this analysis, we classify errors into five categories: 1 - Context-dependent or pronoun mentions, where the induced senses of JoBimText are not that useful for pronoun mentions, and mentions, where the labels are expected to be generated context dependently, e.g., for a given name as shown in Table 3. 2 - JoBimText does not contain the referred sense information, although it can provide many different and fine-grained induced senses. For the sample mention "Binaural", most senses are related to Binaural beats or headsets[18]. 3 - The labels are not matching perfectly. 4 - Some pre-processing issues, which are discussed in detail, in limitations paragraph below. For the sample data in Table 3, if "place" is searched, more relevant information can be collected. 5 - The labels of JoBimText are not that relevant, though the correct sense (or one of the correct senses) is

[18] Note that there is one sense (sense 14), which contains person names from the group Pearl Jam as terms, however there is no IS-As, and thus we cannot take into account.

selected. 6 - Wrong sense selection, for example, for "Orleans", induced sense 0 is the right sense in JoBimText, however, the method selects another cluster. Note that some samples included in one group of a category can also be included in another category, e.g., some pronouns (category-1) do not have the referred sense information from JoBimText (category-2). Note also that some labels can be mixed for the sense.

Limitations One of the goals of ultra-fine entity typing is to generate context-dependent labels for a mention, e.g., the label for "Leonardo DiCaprio" could be *passenger* depending on its context [7]. JoBimText may not produce context-dependent labels, as discussed. In UFET, mention types can be nominals, named entities, and pronouns, and for pronouns, JoBimText is unable to produce good labels and clusters. There are some mistakes or limitations specifically due to the pre-processing steps. Since head word extraction returns only one token, named entities cannot be taken properly, e.g., for the mention "Los Angeles", the head word is "Los". Sometimes, the head word loses the main information, e.g. for the mention "Perhaps the biggest of those factors", the head word is "biggest", although "factors" might be a better token for this mention. There are also some limitations due to the singularization step. Sometimes it can singularize the name entities, for example for the mention "the Cleveland Browns" with "Browns" head word, after singularization the word becomes "Brown". If the plural word is not in the last token, the singularization might fail for compounds. As explained earlier, we double-check whether the token is plural using features of stanza, however, sometimes this check causes a mistake. For instance, "works" is labeled as a verb by stanza, and so it is not singularized. The induced senses might be so fine-grained for some terms, e.g., "plan" has 14 senses (with cluster 200,200).

In some cases, the labels might be mixed and produce noise information. Therefore, the predictions are far from being usable directly in real-world and the misuse of the predictions might result in wrong information, as also discussed in Ding et al. [9].

6 Conclusion and Future Work

In this study, we generated ultra-fine entity type labels using the JoBimText framework in an unsupervised way. We observed a slight improvement when we combine our predictions with the predictions from Choi et al. [6] for the mentions that are not pronouns, and this suggests that the labels produced through JoBimText contain helpful information. The improvement is due to the drop of the precision in favor of recall. That means, JoBimText has good lexical coverage with numerous labels, but they are also noisy.

There are several promising further directions, such as, we consider an unsupervised solution in this work, yet the produced labels can be used as a weak label of some supervised models as in some previous models. Our produced unsupervised labels might help when supervised labels are not sufficient. We try

to find good features and parameters based on a manual search, however, their combinatorial behavior is open for research, and further improvement over the search space by better tuning parameters is possible[19].

Acknowledgements The work was partially supported by a Deutscher Akademischer Austauschdienst (DAAD) doctoral stipend and the DFG funded JOIN-T project BI 1544/4.

References

1. Anwar, S., Shelmanov, A., Panchenko, A., Biemann, C.: Generating lexical representations of frames using lexical substitution. In: Proceedings of the Probability and Meaning Conference (PaM 2020). pp. 95–103. Gothenburg (2020), https://aclanthology.org/2020.pam-1.13
2. Biemann, C.: Chinese whispers - an efficient graph clustering algorithm and its application to natural language processing problems. In: Proceedings of TextGraphs: the First Workshop on Graph Based Methods for Natural Language Processing. pp. 73–80. New York City (2006), https://aclanthology.org/W06-3812
3. Biemann, C., Coppola, B., Glass, M.R., Gliozzo, A., Hatem, M., Riedl, M.: JoBimText visualizer: A graph-based approach to contextualizing distributional similarity. In: Proceedings of TextGraphs-8 Graph-based Methods for Natural Language Processing. pp. 6–10. Seattle, WA, USA (2013), https://aclanthology.org/W13-5002
4. Biemann, C., Riedl, M.: Text: now in 2D! A framework for lexical expansion with contextual similarity. Journal of Language Modelling **1**(1), 55–95 (2013). https://doi.org/10.15398/jlm.v1i1.60
5. Bird, S., Klein, E., Loper, E.: Natural Language Processing with Python. O'Reilly Media Inc. (2009), https://www.nltk.org/book
6. Choi, E., Levy, O., Choi, Y., Zettlemoyer, L.: Ultra-fine entity typing. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 87–96. Melbourne, Australia (2018), https://www.aclweb.org/anthology/P18-1009
7. Dai, H., Song, Y., Wang, H.: 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). pp. 1790–1799. Online (2021). https://doi.org/10.18653/v1/2021.acl-long.141
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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). pp. 4171–4186. Minneapolis, MN, USA (2019). https://doi.org/10.18653/v1/N19-1423

[19] Our code can be found at: https://github.com/uhh-lt/unsupervised-ultra-fine-entity-typing

9. Ding, N., Chen, Y., Han, X., Xu, G., Wang, X., Xie, P., Zheng, H., Liu, Z., Li, J., Kim, H.G.: Prompt-learning for fine-grained entity typing. In: Findings of the Association for Computational Linguistics: EMNLP 2022. pp. 6888–6901. Abu Dhabi, United Arab Emirates (2022), https://aclanthology.org/2022.findings-emnlp.512
10. Gillick, D., Lazic, N., Ganchev, K., Kirchner, J., Huynh, D.: Context-dependent fine-grained entity type tagging (2016)
11. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics. p. 539–545 (1992), https://aclanthology.org/C92-2082
12. Huang, L., May, J., Pan, X., Ji, H.: Building a fine-grained entity typing system overnight for a new x (x = language, domain, genre) (2016)
13. Huang, L., May, J., Pan, X., Ji, H., Ren, X., Han, J., Zhao, L., Hendler, J.A.: Liberal entity extraction: Rapid construction of fine-grained entity typing systems. Big Data **5**(1), 19–31 (2017). https://doi.org/10.1089/big.2017.0012
14. Jana, A., Goyal, P.: Can network embedding of distributional thesaurus be combined with word vectors for better representation? In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). pp. 463–473. New Orleans, LO, USA (2018). https://doi.org/10.18653/v1/N18-1043
15. Jiang, C., Jiang, Y., Wu, W., Xie, P., Tu, K.: Modeling label correlations for ultra-fine entity typing with neural pairwise conditional random field. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. pp. 6836–6847. Abu Dhabi, United Arab Emirates (2022), https://aclanthology.org/2022.emnlp-main.459
16. Kilgarriff, A., Rychly, P., Smrz, P., Tugwell, D.: Itri-04-08 the sketch engine. Information Technology **105**(116), 105–116 (2004)
17. Lee, C., Dai, H., Song, Y., Li, X.: A Chinese corpus for fine-grained entity typing. In: Proceedings of the Twelfth Language Resources and Evaluation Conference. pp. 4451–4457. Marseille, France (2020), https://aclanthology.org/2020.lrec-1.548
18. Li, B., Yin, W., Chen, M.: Ultra-fine entity typing with indirect supervision from natural language inference. Transactions of the Association for Computational Linguistics **10**, 607–622 (2022). https://doi.org/10.1162/tacl_a_00479
19. Ling, X., Weld, D.: Fine-grained entity recognition. Proceedings of the AAAI Conference on Artificial Intelligence **26**(1), 94–100 (2021). https://doi.org/10.1609/aaai.v26i1.8122
20. Liu, L., Zhang, H., Jiang, H., Li, Y., Zhao, E., Xu, K., Song, L., Zheng, S., Zhou, B., Zhu, D., Feng, X., Chen, T., Yang, T., Yu, D., Zhang, F., Kang, Z., Shi, S.: TexSmart: A system for enhanced natural language understanding. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations. pp. 1–10. Online (2021). https://doi.org/10.18653/v1/2021.acl-demo.1
21. Liu, Q., Lin, H., Xiao, X., Han, X., Sun, L., Wu, H.: Fine-grained entity typing via label reasoning. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 4611–4622. Online and Punta Cana, Dominican Republic (2021). https://doi.org/10.18653/v1/2021.emnlp-main.378
22. López, F., Heinzerling, B., Strube, M.: Fine-grained entity typing in hyperbolic space. In: Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019). pp. 169–180. Florence, Italy (2019). https://doi.org/10.18653/v1/W19-4319

23. Obeidat, R., Fern, X., Shahbazi, H., Tadepalli, P.: 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). pp. 807–814. Minneapolis, MN, USA (2019). https://doi.org/10.18653/v1/N19-1087
24. Onoe, Y., Boratko, M., McCallum, A., Durrett, G.: 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). pp. 2051–2064. Online (2021). https://doi.org/10.18653/v1/2021.acl-long.160
25. Onoe, Y., Durrett, G.: 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). pp. 2407–2417. Minneapolis, MN, USA (2019). https://doi.org/10.18653/v1/N19-1250
26. Pan, W., Wei, W., Zhu, F.: Automatic noisy label correction for fine-grained entity typing. In: Raedt, L.D. (ed.) Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22. pp. 4317–4323 (2022). https://doi.org/10.24963/ijcai.2022/599
27. Panchenko, A., Marten, F., Ruppert, E., Faralli, S., Ustalov, D., Ponzetto, S.P., Biemann, C.: Unsupervised, knowledge-free, and interpretable word sense disambiguation. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 91–96. Copenhagen, Denmark (2017). https://doi.org/10.18653/v1/D17-2016
28. Panchenko, A., Ruppert, E., Faralli, S., Ponzetto, S.P., Biemann, C.: Building a web-scale dependency-parsed corpus from Common Crawl. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). pp. 1816–1823. Miyazaki, Japan (2018), https://aclanthology.org/L18-1286
29. Panchenko, A., Ruppert, E., Faralli, S., Ponzetto, S.P., Biemann, C.: Unsupervised does not mean uninterpretable: The case for word sense induction and disambiguation. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers. pp. 86–98. Valencia, Spain (2017), https://aclanthology.org/E17-1009
30. Pelevina, M., Arefiev, N., Biemann, C., Panchenko, A.: Making sense of word embeddings. In: Proceedings of the 1st Workshop on Representation Learning for NLP. pp. 174–183. Berlin, Germany (2016). https://doi.org/10.18653/v1/W16-1620
31. Qi, P., Zhang, Y., Zhang, Y., Bolton, J., Manning, C.D.: Stanza: A python natural language processing toolkit for many human languages. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations. pp. 101–108. Online (2020). https://doi.org/10.18653/v1/2020.acl-demos.14
32. Qian, J., Liu, Y., Liu, L., Li, Y., Jiang, H., Zhang, H., Shi, S.: Fine-grained entity typing without knowledge base. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 5309–5319. Online and Punta Cana, Dominican Republic (2021). https://doi.org/10.18653/v1/2021.emnlp-main.431
33. Reimers, N., Gurevych, I.: Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Nat-

ural Language Processing (EMNLP-IJCNLP). pp. 3982–3992. Hong Kong, Hong Kong (2019). https://doi.org/10.18653/v1/D19-1410
34. Riedl, M., Biemann, C.: Scaling to large^3 data: An efficient and effective method to compute distributional thesauri. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. pp. 884–890. Seattle, WA, USA (2013), https://aclanthology.org/D13-1089
35. Ruppert, E., Kaufmann, M., Riedl, M., Biemann, C.: JoBimViz: A web-based visualization for graph-based distributional semantic models. In: Proceedings of ACL-IJCNLP 2015 System Demonstrations. pp. 103–108. Beijing, China (2015). https://doi.org/10.3115/v1/P15-4018
36. Sui, X., Zhang, Y., Song, K., Zhou, B., Zhao, G., Wei, X., Yuan, X.: Improving zero-shot entity linking candidate generation with ultra-fine entity type information. In: Proceedings of the 29th International Conference on Computational Linguistics. pp. 2429–2437. Gyeongju, Republic of Korea (2022), https://aclanthology.org/2022.coling-1.214
37. Xiong, W., Wu, J., Lei, D., Yu, M., Chang, S., Guo, X., Wang, W.Y.: 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). pp. 773–784. Minneapolis, MN, USA (2019). https://doi.org/10.18653/v1/N19-1084
38. Zhang, T., Xia, C., Lu, C.T., Yu, P.: MZET: Memory augmented zero-shot fine-grained named entity typing. In: Proceedings of the 28th International Conference on Computational Linguistics. pp. 77–87. Barcelona, Spain (Online) (2020). https://doi.org/10.18653/v1/2020.coling-main.7
39. Zhou, B., Khashabi, D., Tsai, C.T., Roth, D.: Zero-shot open entity typing as type-compatible grounding. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 2065–2076. Brussels, Belgium (2018). https://doi.org/10.18653/v1/D18-1231
40. Zuo, X., Liang, H., Jing, N., Zeng, S., Fang, Z., Luo, Y.: Type-enriched hierarchical contrastive strategy for fine-grained entity typing. In: Proceedings of the 29th International Conference on Computational Linguistics. pp. 2405–2417. Gyeongju, Republic of Korea (2022), https://aclanthology.org/2022.coling-1.212