Ultra-fine Entity Typing with Indirect Supervision from Natural Language Inference

Bangzheng Li[◇†*], Wenpeng Yin[‡], and Muhao Chen[◇]

[◇]University of Southern California, USA
[†]University of Illinois at Urbana-Champaign, USA
[‡]Temple University, USA
vincentleebang@gmail.com; wenpeng.yin@temple.edu;
muhaoche@usc.edu

Introduction

Existing systems formulate the task as a multi-way classification problem and train directly or distantly supervised classifiers. This causes two issues:

- the classifiers do not capture the type semantics because types are often converted into indices;
- systems developed in this way are limited to predicting within a pre-defined type set, and often fall short of generalizing to types that are rarely seen or unseen in training

Introduction

Our method: LITE

a new approach that formulates entity typing as a natural language inference (NLI) problem, making use of

- the indirect supervision from NLI to infer type information meaningfully represented as textual hypotheses and alleviate the data scarcity issue
- a learning-to-rank objective to avoid the pre-defining of a type set.

Experiments show:

- state-of-the-art performance on the UFET task
- strong generalizability (works well on new data containing unseen types)

Method

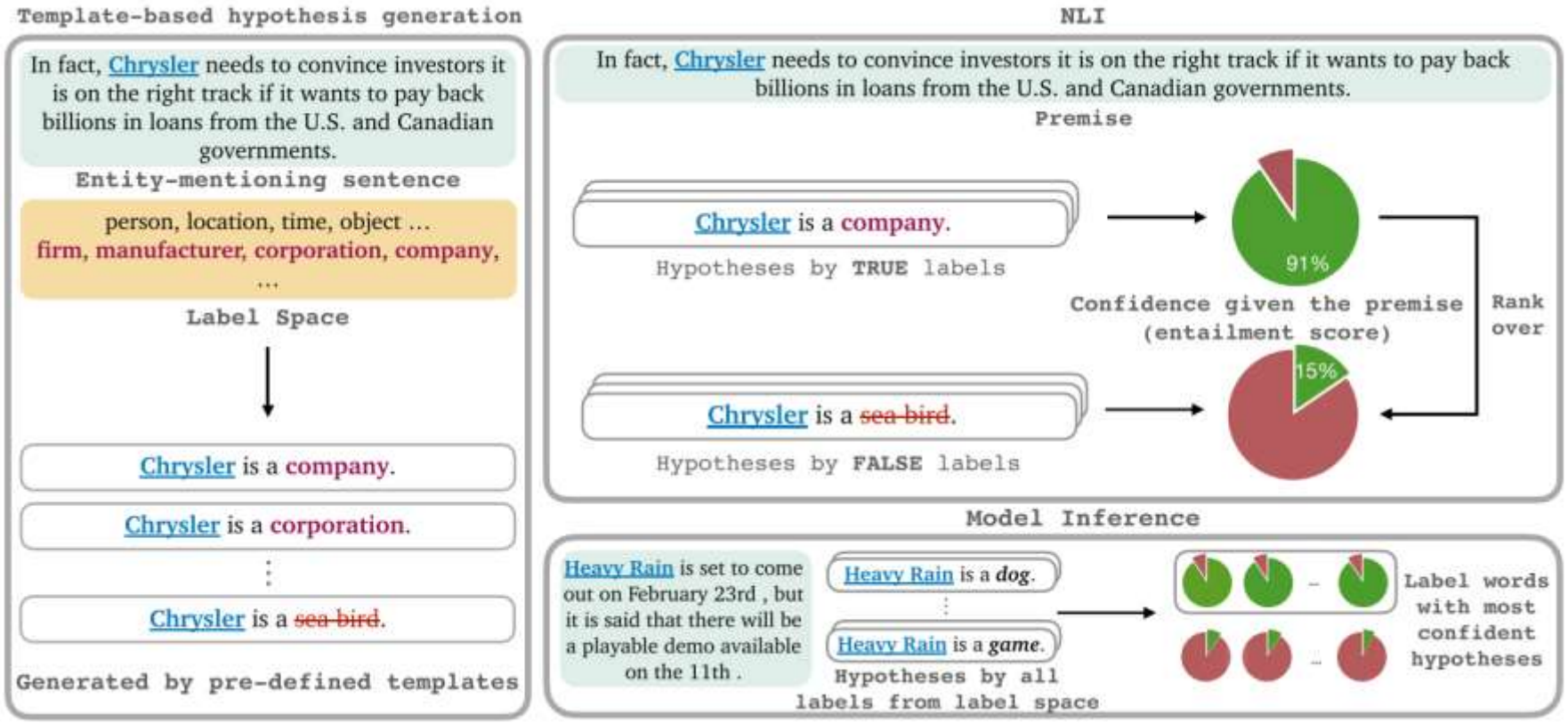
- Problem Definition

The input of an entity typing task is a sentence s and an entity mention of interest e $\in$ s. This task aims at typing e with one or more type labels from the label space L.

For instance, in "*Jay is currently working on his Spring 09 collection, which is being sponsored by the YKK Group*.", the entity "*Jay*" should be labeled as person, designer, or creator instead of organization or location.

Method

- Framework



Method

- Type Description Generation

Taxonomic statement. The first template directly connects the entity mention and the type label with an ‘‘is-a’’ statement, namely, ‘‘[ENTITY] is a [LABEL]’’

Contextual explanation. The second template generates a declarative sentence that adds a context-related connective. The generated type description is in the form of ‘‘In this context, [ENTITY] is referring to [LABEL]’’.

Label substitution. Directly replaces the [ENTITY] in the original sentence with [LABEL]. Therefore, the NLI model will treat the modified sentence with a ‘‘type mention’’ as the hypothesis of the original sentence with the entity mention.

Method

- Type Description Generation

Templates	Type Descriptions	Premise-Hypothesis Pairs for NLI
Taxonomic Statement	**Jay** is a **producer**.	Premise: ‘‘**Jay** is currently working on his Spring 09 collection, . . . ’’ Hypothesis: ‘‘**Jay** is a **producer**.’’
Contextual Explanation	In this context, **career at a company** is referring to **duration**.	Premise: ‘‘No one expects a **career at a company** any more, . . . ’’ Hypothesis: ‘‘In this context, **career at a company** is referring to **duration**.’’
Label Substitution	**Musician** knows how to make a hip-hop record sound good.	Premise: ‘‘**He** knows how to make a hip-hop record sound good.’’ Hypothesis: ‘‘**Musician** knows how to make a hip-hop record sound good.’’

Method

- Modeling Label Dependency(?)

If there are ancestor types, we not only generate descriptions for each of the ancestor types, but also conduct learning among these type descriptions. The descendant type description would act as the premise and the ancestor type description would act as the hypothesis.

“London” (/location/city) → “London is a city” & “London is a location”

The more fine-grained type description “London is a city” can act as the premise of the more coarse-grained description “London is a location”, so as to help capture the dependency between two labels “city” and “location”. Such paired type descriptions are added to training and will be captured by the dependency loss $\mathcal{L}_d$

Method

- Learning Objective

s: sentence e: entity L: lable space

P : all true type labels of e in s $p \in P$

H(p): generated type description

entailment score: $\varepsilon(s, H(p)) \in [0, 1]$ for premise s and hypothesis H(p)

false label $p \notin \in L \setminus P$, negative sampling randomly selected

margin ranking loss:

$$\mathcal{L}_t = [\varepsilon(s, H(p')) - \varepsilon(s, H(p)) + \gamma]_+$$

[x]+ denotes the positive part of the input x (i.e., max(x, 0)) and γ is a non-negative constant.

Method

- Learning Objective

p_{an}: ancestor type

p_{de}: descendant type

$$\mathcal{L}_d = [\varepsilon(H(p_{de}), H(p'_{an})) - \varepsilon(H(p_{de}), H(p_{an})) + \gamma]_+$$

The eventual learning objective is to optimize the joint loss:

$$\mathcal{L} = \frac{1}{|S|} \sum_{s \in S} \frac{1}{|P_s|} \sum_{p \in P_s} \mathcal{L}_t + \lambda \mathcal{L}_d$$

Experiments

- Ultra-Fine Entity Typing

Model		P	R	F1
UFET-biLSTM (Choi et al., 2018)		48.1	23.2	31.3
LabelGCN (Xiong et al., 2019)		50.3	29.2	36.9
LDET (Onoe and Durrett, 2019)		51.5	33.0	40.1
Box4Types (Onoe et al., 2021)		52.8	38.8	44.8
LRN (Liu et al., 2021)		**54.5**	38.9	45.4
MLMET (Dai et al., 2021)		53.6	45.3	49.1
LITE	NLI	1.5	7.1	2.5
	L	48.7	45.8	47.2
	D+L	27.5	**56.4**	37.0
	NLI+D+L	45.4	49.9	47.4
	NLI+L	52.4	48.9	**50.6**
	–w/o label dependency	53.3	46.6	49.7

Experiments

- Fine-grained Entity Typing

Model		OntoNotes		FIGER	
		macro-F1	micro-F1	macro-F1	micro-F1
Hierarchy-Typing (Chen et al., 2020b)		73.0	68.1	83.0	79.8
Box4Types (Onoe and Durrett, 2020)		77.3	70.9	79.4	75.0
DSAM (Hu et al., 2020)		83.1	78.2	83.3	81.5
SEPREM (Xu et al., 2021)		–	–	86.1	82.1
MLMET (Dai et al., 2021)		85.4	80.4	–	–
LITE	pre-trained on NLI+UFET	**86.6**	**81.4**	80.1	74.7
	NLI+task-specific training	86.4	80.9	**86.7**	**83.3**

Experiments

- Analysis

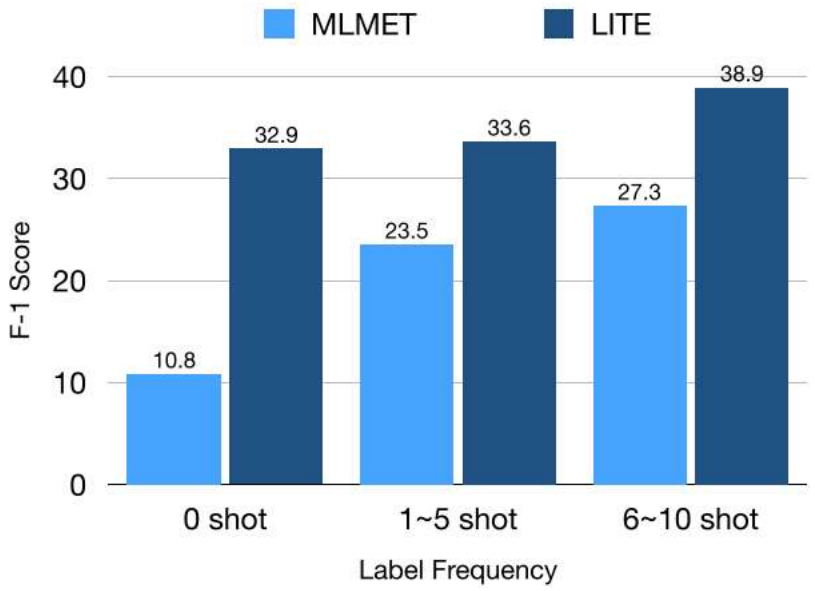
different type description template

Templates	LITE$_{NLI+L}$			LITE$_{NLI+D+L}$			LITE$_{D+L}$		
	P	R	F1	P	R	F1	P	R	F1
Taxonomic Statement	52.4	48.9	**50.6**	45.4	49.9	**47.4**	27.5	56.4	**37.0**
Contextual Explanation	50.8	49.2	50.2	45.3	48.5	46.8	26.9	55.4	36.2
Label Substitution	47.4	49.3	48.3	42.5	50.7	46.2	24.8	59.3	35.0

Experiments

- Analysis

few and zero-shot prediction



Experiments

- Analysis

time efficiency

In general, LITE has much less training cost, of around 40 hours, than the previous strongest (data-augmentation-based) model MLMET, which requires over 180 hours, on the UFET task.

During the inference step, it takes about 35 seconds per new sentence for our model to do inference with a fixed type vocabulary of over 10,000 different labels while a common multi-way classifier merely requires around 0.2 seconds.

LITE is much more efficient on dynamic type vocabulary. It requires almost no re-calculation when new, un-mappable labels are added to an existing type set but multi-way classifiers need re-training with an extended classifier every time (e.g., over 180 hours by the previous SOTA).

Conclusion

- Analysis

We propose a new model, LITE, that leverages indirect supervision from NLI to type entities in texts. Through template-based type hypothesis generation, LITE formulates the entity typing task as a language inference task and meanwhile the semantically rich hypothesis remedies the data scarcity problem in the UFET benchmark.

Our experimental results illustrate that LITE promisingly offers SOTA on UFET, OntoNotes, and FIGER, and yields strong performance on zero-shot and few-shot types. LITE pretrained on UFET also yields strong transferability by outperforming SOTA baselines when directly make predictions on OntoNotes and FIGER.

PromptNER: prompting for fewshot named entity recognition

Anonymous authors
Paper under double-blind review

Abstract

In this paper, we introduce PromptNER, an algorithm for few-shot and cross-domain NER. To adapt to a new NER task, PromptNER requires a set of **entity definitions**, and a set of **few-shot examples**, along with **explanatory text** justifying the applicability of each entity tag.
Given a sentence, PromptNER prompts an LLM to produce a list of potential entities along with correspondi**n**g explanations justifying their compatibility with the provided entity type definitions.
PromptNER achieves state-of-the-art performance on both few-shot NER and Cross Domain NER.

Input

Defn: An entity is a person(person), university(university), scientist(scientist), organisation(organisation), country(country), location(location), scientific discipline(discipline), enzyme(enzyme), protein(protein), chemical compound(chemicalcompound), chemical element(chemicalelement), event(event), astronomical object(astronomicalobject), academic journal(academicjournal), award(award), or theory(theory). If an entity does not fit the types above it is (misc)

Example 1: He attended the U.S. Air Force Institute of Technology for a year , earning a bachelor 's degree in aeromechanics , and received his test pilot training at Edwards Air Force Base in California before his assignment as a test pilot at Wright-Patterson Air Force Base in Ohio .

Answer:
1. U.S. Air Force Institute of Technology | True | as he attended this institute is likely a university (university)
2. bachelor 's degree | False | as it is not a university, award or any other entity type
3. aeromechanics | True | as it is a scientific discipline (discipline)
4. Edwards Air Force Base | True | as an Air Force Base is an organised unit (organisation)
5. California | True | as in this case California refers to the state of California itself (location)
6. Wright-Patterson Air Force Base | True | as an Air Force Base is an organisation (organisation)
7. Ohio | True | as it is a state (location)

Method

- Conditional Generation Language Model

we leverage the power of pretrained LLMs which have been trained on conditional generation tasks.

Using general Seq-to-Seq models opens up modeling possibilities, allowing us to pursue strategies based on chain-of-thought-like reasoning.

Method

- Modular Definitions

In our approach to few-shot, NER, each problem is defined not only by a small set of exemplars (the few-shot examples) but also by a per-domain definition. Here, the modular definition consists of a natural language description of what does (and does not) constitute an entity. This can be useful in instances where the typical natural language connotation of the word ‘entity’ may include concepts the specific NER task would want to exclude.

Method

- Potential Entity Output Template

we create a template structure for the output of the LLM which allows it to emulate reasoning and justify why a phrase is predicted as an entity of a specific type. The exact structure is one where each line of the output mentions a distinct candidate entity, a decision on whether or not the candidate should be considered an entity and an explanation for why or why not along with what entity type it belongs to.

Experiments

- Standard Low Resource NER

Method	CoNLL
COPNERHuang et al. (2022b)	75.8 ± 2.7
EntLMMa et al. (2021)	51.32 ± 7.67
FactMixYang et al. (2022)	60.8
ProMLChen et al. (2022)	79.16 ± 4.49
UIELu et al. (2022)	67.09
CONTaiNERDas et al. (2022)	75.8 ± 2.7
PMRXu et al. (2022)	65.7 ± 4.5
PromptNER ~~T5XXL (Us)~~	45.66 ± 12.43
PromptNER GPT3.5 (Us)	78.62 ± 4.62
PromptNER GPT4 (Us)	$\mathbf{83.48 \pm 5.66}$

FewShot Learning ($0 < k < 5$) on CoNLL dataset. Results show micro-F1 averages and associated standard deviation over 3 runs when available. The results for all competing methods are taken from the tables reported in their respective publications and papers.

Experiments

- Cross Domain NER

Method	k	Politics	Literature	Music	AI	Sciences
FactMixYang et al. (2022)	100	44.66	28.89	23.75	32.09	34.13
LANERHu et al. (2022a)	100-200	74.06	71.11	78.78	65.79	71.83
CPNERChen et al. (2023)	100-200	76.35	72.17	80.28	**66.39**	**76.83**
EnTDAHu et al. (2022b)	100	72.98	68.04	76.55	62.31	72.55
PromptNER T5XXL (Us)	**2**	39.43	36.55	41.93	30.67	46.32
PromptNER GPT3.5 (Us)	**2**	71.74	64.15	77.78	59.35	64.83
PromptNER GPT4 (Us)	**2**	**78.61**	**74.44**	**84.26**	64.83	72.59

With CoNLL as source domain. k is the number of target domain datapoints used by each method. Results show micro-F1 scores. Despite using only 1%–2% of the data our method achieves state-of-the-art performance on three of the five datasets

Experiments

- Biomedical Domain NER

Method	GENIA
CONTaiNERDas et al. (2022)	44.77 ± 1.06
BCLMing et al. (2022)	46.06 ± 1.02
SpanProtoShen et al. (2021)	41.84 ± 2.66
PACL	49.58 ± 1.82
PromptNER T5XXL (Us)	25.13 ± 3.22
PromptNER GPT3.5 (Us)	$\mathbf{52.80 \pm 5.15}$
PromptNER GPT4 (Us)	$\mathbf{58.44 \pm 6.82}$

Few-shot Learning ($0 < k < 5$) on GENIA dataset. Results show micro-F1 averages and associated standard deviation over 3 runs when available

Experiments

- Recent NER Datasets

Methods	TweetNER	FaBNER
CONTaiNER	12.83 ± 6.3	9.42 ± 5.1
ProML	18.82 ± 4.2	19.61 ± 3.2
PromptNER GPT3.5	30.5 ± 5.6	17.84 ± 4.8
PromptNER GPT4	$\mathbf{43.5 \pm 5.3}$	$\mathbf{24.35 \pm 4.1}$

Performance of PromptNER on TweetNER and FaBNER. GPT3.5 outperforms both methods significantly on TweetNER, and is competitive with all other methods on FaBNER.

Ablations

- Pretrained Language Model

Model	Size	ConLL	Genia	Politics	Literature	Music	AI	Science	FewNERD
GPT4	**1.8T***	**83.48**	**58.44**	**78.61**	**74.44**	**84.26**	**64.83**	**72.59**	**72.63**
GPT3	175B	78.62	52.8	71.74	64.15	77.78	59.35	64.83	62.33
T5XXL	11B	45.66	19.34	39.43	36.55	41.93	30.67	46.32	23.2
T5XL	3B	24.12	10.5	18.45	18.62	25.79	10.58	26.39	8.35

Ablations

- Components of PromptNER

Def	FS	CoT	Cand	ConLL	Genia	Pol	Lit	Mus	AI	Sci	FewNERD	Avg Rank
✓	✓	✓	✓	**78.6**	**52.8**	**71.7**	**64.1**	**77.7**	**59.3**	**64.8**	**62.3**	1
✓	✓	✓	✗	71.6	38.5	61.3	46.3	60.2	34.2	46.8	57.3	3.5
✓	✓	✗	✓	75.1	49.2	70.4	54.9	70.6	53.6	60.5	42.4	2.1
✓	✗	✓	✓	68.1	23.2	20.3	21.3	24.5	40.7	40.6	34.6	5.6
✗	✓	✓	✓	63.3	46.2	57.7	49.6	50	29	50.8	34.8	4
✗	✓	✓	✗	54.8	37.2	49.8	37.3	54.7	27.8	21.7	18.8	5.6
✗	✓	✗	✗	49.7	39.3	42.5	40.3	48.6	24.5	35.9	16.1	6.1

Ablation over components of PromptNER on GPT3.5. Def: Definitions, FS: Few Shot Examples, CoT: Explanations required, Cand: Candidate entities in predicted list. Every component improves performance of the method in general, with the setting of all components vastly outperforming the traditional Few Shot Prompting and Chain-of-Thought Prompting methods

Comments

Soundness: 3 good
Presentation: 3 good
Contribution: 2 fair
Strengths: Since LLMs are quite popular these days, it is worth to see how LLMs can be used in the classic NLP tasks. This work explores the potential of LLMs in NER tasks and show they are useful in terms of cross-domain and low-resource scenarios.
Weaknesses: The biggest concern is that the method seems straightforward to me, and thus I think it lacks the core innovation in terms of the methodology. It is intuitive to inform the model of the definition, few-shots, and chain-of-thought to accomplish the task. I wish to see how these prompts are interacting with the final outputs so as to provide more insights on how the future work can learn from the prompt design or use the LLMs properly in classic NLP tasks. Also, I think it is more innovative to design specialized modules in the prompt engineering for NER tasks. Otherwise, the current work is similar to a strong baseline which is helpful to future work for sure, but may not be ready for a long research paper.

Comments

Soundness: 2 fair
Presentation: 2 fair
Contribution: 2 fair
Strengths: This paper demonstrated that with the GPT4 as the backbone, the proposed PROMPTNER showed good cross-domain NER identification ability. Several ablation experiments showed the effectiveness of each component.
Weaknesses: There are various studies on improving prompt strategies in this LLM area. Adding the entity definition on the prompt of LLM is not an innovative method. Based on the experiment, the most improvement of the proposed method comes from the powerful GPT 4 backend. Comparing the GPT4-driven model with models with much weaker LLM is not necessary and not fair (Table 2-4).

Comments

Soundness: 3 good
Presentation: 3 good
Contribution: 1 poor
Strengths: The paper introduces a very simple prompting template which can be easily integrated into relevant applications. The paper is clearly written and is easy to follow. it gives the detailed ablation study to help us understand the contribution of each components. The paper compares against a comprehensive list of previous work in its experiments.
Weaknesses: It seems to me that the major contribution of the SOTA performance comes from GPT4 more than some advanced prompting techniques of the paper where most of cases, the best performance is achieved only by GPT 4. T5 is way worse than the current SOTA. The prompting template is simply an application of CoT with few shot in-context learning. I am not convinced if whether this prompt template is very novel or has a significant originality of ideas and I was wondering whether other similar template can't achieve similar performance.
Questions: For the Cross Domain NER tasks, as Table 2 shows, only 2 examples are used in prompting and the F1 scores for AI and Sciences are not the highest. Why can't we add more examples (e.g. up to 200) to improve performance? Did we also consider fine-tune GPT to see if we can get the higher F1 scores?