How Can Cross-lingual Knowledge Contribute Better to Fine-Grained Entity Typing?

Hailong Jin[1,2], **Tiansi Dong**[3], **Lei Hou**[1,2]*, **Juanzi Li**[1,2]
Hui Chen[4], **Zelin Dai**[4], **Qu Yincen**[4]
[1]Department of Computer Science and Technology, BNRist
[2]KIRC, Institute for Artificial Intelligence, Tsinghua University, Beijing 100084, China
[3]B-IT, University of Bonn, Germany
[4]Alibaba Group, Hangzhou, China
{jinhl, houlei}@mail.tsinghua.edu.cn

Abstract

Cross-lingual Entity Typing (CLET) aims at improving the quality of entity type prediction by transferring semantic knowledge learned from rich-resourced languages to low-resourced languages. In this paper, by utilizing multilingual transfer learning via the mixture-of-experts approach, our model dynamically capture the relationship between target language and each source language, and effectively generalize to predict types of unseen entities in new languages. Extensive experiments on multilingual datasets show that our method significantly outperforms multiple baselines and can robustly handle negative transfer. We questioned the relationship between language similarity and the performance of CLET. With a series of experiments, we refute the commonsense that the more source the better, and propose the *Similarity Hypothesis for CLET*.

1 Introduction

Fine-grained Entity Typing (FET) aims at labeling entity mentions in a particular context with one or more specific types organized in a type hierarchy. For example, *Donald Trump* is classified as having the path of following types: *President*, *Politician*, *Person*. *President* is a subclass of *Politician* that in turn is a subclass of *Person*. FET provides accurate type information, and is therefore quite useful for various downstream NLP tasks, such as entity linking (Onoe and Durrett, 2020; Chen et al., 2020a; Zhu et al., 2020), relation extraction (Vashishth et al., 2018; Kuang et al., 2020), text generation (Dong et al., 2021; Elsahar et al., 2018), and so on.

Supervised learning approaches to FET need huge amount of labeled training data (Ren et al., 2016; Shi et al., 2020; Chen et al., 2020b), and can be applied for a few rich-resourced languages, e.g., English, which have enough qualified labeled data.

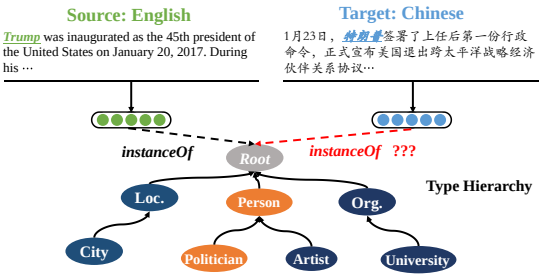

Figure 1: Example of Cross-lingual Entity Typing. We use knowledge from source language (English) to help with entity typing task in target language (Chinese).

For the vast majority of low-resourced languages, we have insufficient training data, or even do not have labeled data at all. However, languages are not independent, instead, some are more similar than others, and form a family tree. For example, Portuguese is similar to Spanish; Dutch can even be thought of as half way between German and English. This motivates us to utilize the knowledge from rich-resourced languages (source) to help to predict missing types in a low-resourced language (target), which is called the Cross-lingual Entity Typing (CLET). Previous research showed that transferring knowledge from multiple source languages could improve the performance of entity typing (Chen et al., 2019b). Recent research proposed a unified CLET model, trained with four rich source languages (English, Finnish, German, and Spanish), is able to accept over 100 different languages (Selvaraj et al., 2021). Behind such unified models is the assumption that the more rich-resourced languages a model has, the better the performance will be. This leads to the search of the best unified model for all low-resourced languages.

Here, we raise the question: How will the similarity between the source and the target languages affect the performance? To this end, we carefully select six languages as follows: German, English, and Dutch in the west Germanic family, Russian

* Corresponding author

Findings of the Association for Computational Linguistics: ACL 2022, pages 3071 - 3081

in the Slavic family, and Spanish in the Romance family. This five languages are in the European family[1]. We select Chinese in the Sino-Tibetan family, which is totally different from the other five languages, as illustrated in Figure 2. The more similar two languages are, the higher is their lowest common ancestor located in the language tree.

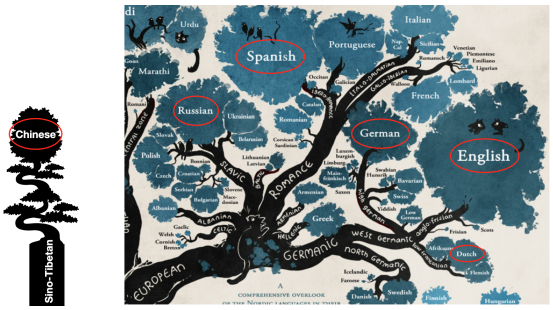

Figure 2: Six selected languages are marked with red circles.

Following the unsupervised multilingual transfer-learning setup, we use labeled data from source languages and unlabeled data from the target language, leverage multilingual BERT as feature encoder to produce language-independent features (Devlin et al., 2019), and use mixture-of-experts (MoE) approach (Jacobs et al., 1991; Shazeer et al., 2017) to capture the correlations between the target language and each source language. For each target example, the predicted posterior is a weighted combination of all the experts' predictions. Experts' weights reflect the proximity of the example to each source language. To further improve transfer quality, we apply a language discriminator to extract more language-invariant features from both source and target languages via adversarial learning. Extensive experiments show that our proposed method significantly outperforms multiple state-of-the-art monolingual methods.

In contrast to other cross-lingual FET researches, our work explores how the similarity between source and target languages would affect the FET performance. Our experiment results surprisingly refute the commonly accepted assumption that the more and the richer the source languages are, the better performance it will be. Our results suggests the importance of the similarity between source and target languages. The more similar the source and the target are and the richer the source is, the better performance it will be. Adding a rich but dissimilar source may reduce the performance. This observation refutes the existence of the best unified model for all target languages. The best cross-lingual source languages shall be rich and selected among the cluster of the most similar languages to the target language.

The rest of this paper is organized as follows. Section 2 formally defines the problem of cross-lingual fine-grained entity typing. Section 3 describes our approach. Section 4 reports two groups of experiments, one to evaluate our method, the other to explore the relation between language similarity and the performance of type prediction. Section 5 reviews some related works. Section 6 concludes our work.

2 Problem Formulation

We use $\mathbb{S} = \{\mathcal{S}_i\}_{i=1}^{N}$ as the set of source languages, in which N is the number of source languages, $\mathcal{T}$ as the target language. Types are organized into a tree-structured hierarchy $\mathbf{Y}$, shared by all languages.

Based on the assumption that each mention can only have one type-path depending on the context, we represent each type-path uniquely by the terminal type (which might not be a leaf node). For example, type-path `root-person-athlete` can be represented as just `athlete`, while `root-person` can be unambiguously represented as the non-leaf `person`.

For each source language $\mathcal{S}_i \in \mathbb{S}$, we have a set of training data $\mathcal{S}_i = \{(x_t, y_t)\}_{t=1}^{|\mathcal{S}_i|}$. $x_t = (m_t, c_t)$ contains two parts, $m_t = \langle w_l, \ldots, w_r \rangle$ is an entity mention and $c_t = \langle w_1, \ldots, w_L \rangle$ is its context, both m_t and c_t are word sequences, where L is the context length and $1 < l \leq r < L$. y_t is the most specific type of m_t, corresponding to a unique type-path in $\mathbf{Y}$. For target language $\mathcal{T}$, we create a set of unlabeled data $\mathcal{T} = \{x_t\}_{t=1}^{|\mathcal{T}|}$. We formulate cross-lingual fine-grained entity typing (CLET) problem as follows:

Definition 1 *Given training data from source languages $\mathbb{S} = \{\mathcal{S}_i\}_{i=1}^{N}$, and unlabeled data from target language $\mathcal{T}$, we aim at learning a model $P(y|x)$ using the source training data and generalizing well to the target language. Given $x \in \mathcal{T}$, our task is to predict its most specific type $\hat{y}$ depending on the learned model $P(y|x)$.*

Notations The superscript and the subscript of an example denote the language from which it is

[1] https://thelanguagenerds.com/2019/feast-your-eyes-on-magnificent-linguistic-family-tree/

sampled, and its index, respectively. For instance, $x_j^{\mathcal{S}_i} = (m_j^{\mathcal{S}_i}, c_j^{\mathcal{S}_i})$ is the j^{th} example in $\mathcal{S}_i$. Sometimes we omit superscripts for brevity.

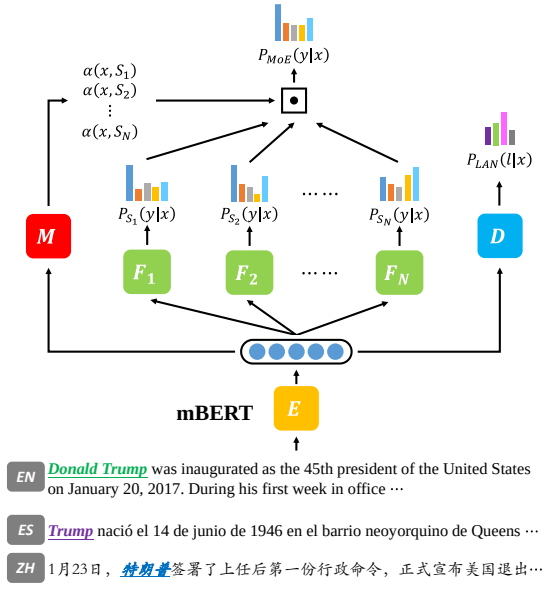

Figure 3: Framework of our proposed model. x is an example from any language. E is the shared encoder across all languages; $F_{\mathcal{S}_i}$ is the classifier on the i^{th} source language, the final prediction $P_{MoE}(y|x)$ is a weighted combination of all the classifiers' predictions; M is the metric learning component, which takes the encoding of x and source languages $\{\mathcal{S}_i\}_{i=1}^N$ as input and computes weight α; $\mathcal{D}$ is the language discriminator which is learned during adversarial training.

3 Methodology

3.1 Overview of Our Approach

We model the multiple source languages as a mixture of experts, and learn metric α to weight the experts for different target examples (Jacobs et al., 1991; Shazeer et al., 2017). Our model consists of four key components, as shown in Figure 3, namely the shared feature extractor E, a set of language-specific classifier $\{F_{\mathcal{S}_i}\}_{i=1}^N$, metric function M and language discriminator $\mathcal{D}$. Our model is a multi-task learning architecture, with a shared encoder of all languages, and language-specific classifier $F_{\mathcal{S}_i}$ for each language S_i. Each input is firstly encoded with E, and then fed to each classifier to obtain the language-specific predictions. The final predictions are then weighted based on the metric M.

3.2 Feature Extractor E

We use multilingual BERT (mBERT) as feature extractor (Devlin et al., 2019), since it follows the same model architecture and training procedure as BERT and produces an effective cross-lingual word representation. Different from BERT, mBERT is pre-trained on concatenated Wikipedia data in 104 languages. Formally, given an example $x_i = (m_i, c_i)$ in any language, we utilize mBERT encoder to get its representation $E(x_i)$. Given a mention $m_i = \langle w_l, \ldots, w_r \rangle$ with its context $c_i = \langle w_1, \ldots, w_L \rangle$, we simply feed the sequence ([CLS], c_i, [SEP], m_i, [SEP]) to mBERT encoder and use the output of [CLS] token as the representation of the mention with its context.

3.3 Expert Classifier F

Each source $\mathcal{S}_i$ has a language-specific classifier (expert) $F_{\mathcal{S}_i}$. With the representation $E(x_j^{\mathcal{S}_i})$ of an example in $\mathcal{S}_i$, we employ a softmax classifier parameterized by $\theta_f^{\mathcal{S}_i} = [W_f^{\mathcal{S}_i}, b_f^{\mathcal{S}_i}]$ to get the language-specific prediction (i.e. posterior).

$$P_{\mathcal{S}_i}(y|x_j^{\mathcal{S}_i}) = \text{Softmax}(W_f^{\mathcal{S}_i} E(x_j^{\mathcal{S}_i}) + b_f^{\mathcal{S}_i}) \quad (1)$$

$$\hat{y} = \arg\max_y P_{\mathcal{S}_i}(y|x_j^{\mathcal{S}_i}) \quad (2)$$

where $W_f^{\mathcal{S}_i} \in \mathbb{R}^{K \times d_z}$ can be treated as the type embeddings, $b_f^{\mathcal{S}_i} \in \mathbb{R}^K$ is the type bias, K is the number of types. The predicted type $\hat{y}$ is the type with maximum posterior probability. Since $F_{\mathcal{S}_i}$ is trained on labeled data from $\mathcal{S}_i$, so it will pay more attention on **language-specific** feature in $\mathcal{S}_i$.

3.4 Mixture of Experts

Given an entity x from the target language, we model its posterior distribution as a mixture of posteriors produced by experts trained on different source language data:

$$P_{MoE}(y|x) = \sum_{i=1}^{N} \alpha(x, \mathcal{S}_i) P_{\mathcal{S}_i}(y|x) \quad (3)$$

$P_{\mathcal{S}_i}(\cdot)$ is the posterior distribution produced by the i^{th} source classifier $F_{\mathcal{S}_i}$ (i.e., the i^{th} expert). $\alpha(\cdot)$ is calculated by metric function M, it measures the similarities between the target language example x and each source languages $\{\mathcal{S}_i\}_{i=1}^N$.

We utilize *point-to-set* distance as metric function (Guo et al., 2018) to define the distance between entity x and a source $\mathcal{S}_i$ is defined as follow.

$$d(x, \mathcal{S}_i) = ((E(x) - \mu^{\mathcal{S}_i})^\top \mathbf{M}_{\mathcal{S}_i} (E(x) - \mu^{\mathcal{S}_i}))^{\frac{1}{2}} \quad (4)$$

where $\mu^{\mathcal{S}_i}$ is the mean encoding of $\mathcal{S}_i$. Each source $\mathcal{S}_i$ has a parameter matrix $\mathbf{M}_{\mathcal{S}_i}$, which is used to measure the similarity between an entity and this source. Based on the distance metric, confidence score is defined as $e(x, \mathcal{S}_i) = -d(x, \mathcal{S}_i)$. The final metric values $\alpha(x, \mathcal{S}_i)$ are then obtained by normalizing these scores:

$$\alpha(x, \mathcal{S}_i) = \text{Softmax}(e(x, \mathcal{S}_i)) \quad (5)$$

This metric approach hypothesizes that both input entity and source language distribution are important in weight assignment.

3.5 Language Discriminator $\mathcal{D}$

To further improve the quality, we adopt a language adversarial training module to minimize the divergence between source and target languages. In other words, feature extractor E should capture more **language-invariant** information. Different from $\{F_{\mathcal{S}_i}\}_{i=1}^{N}$, $\mathcal{D}$, as a language classifier, can be trained on unlabeled data in both source and target languages. Given an entity x, it takes $E(x)$ as input and predicts the likelihood of the language label of x. $\mathcal{D}$ is defined as a softmax classifier parameterized by $\theta_d = [W_d, b_d]$, where $W_d \in \mathbb{R}^{(N+1)\times d_z}$ and $b_d \in \mathbb{R}^{N+1}$.

$$P_{LAN}(l|x) = \text{Softmax}(W_d E(x) + b_d) \quad (6)$$

3.6 Model Training

Our model's parameters include $\theta_f^{\mathcal{S}_i}$, θ_d and $\mathbf{M}_{\mathcal{S}_i}$. We utilize language-adversarial training method to optimize parameters in language discriminator $\mathcal{D}$ and other components, separately. During the training process, E aims at confusing $\mathcal{D}$, so that $\mathcal{D}$ cannot predict the language in which a sample is written. The hypothesis is that if $\mathcal{D}$ cannot recognize the language of the input, the extracted features will contain more language-invariant information. We propose to use meta-training approach to learn the parameters in experts ($\theta_f^{\mathcal{S}_i}$) and metric function ($\mathbf{M}_{\mathcal{S}_i}$) simultaneously. With each iteration through the training data, we update parameters in the mBERT encoder as well as parameters in our model. The training part is described in more detail in Appendix (Alg.1).

4 Experiments

A series of experiments are conducted to evaluate our CLET method and to examine how the language similarity affects the performance of CLET. Our source code is available[2] for reference.

	EN	ES	DE	ZH	NL	RU
#train	74,543	19,764	23,709	13,711	16,528	24,918
#dev	35,275	9,334	11,276	6,446	7,521	12,527
#test	50,265	13,181	15,868	9,294	10,736	16,371

Table 1: Dataset Statistics. **EN**: English, **DE**: German, **ES**: Spanish, **ZH**: Chinese, **NL**: Dutch, **RU**: Russian.

4.1 Model Evaluation

4.1.1 Experiment Setting

Dataset We construct our dataset based on the MVET dataset constructed from Wikipedia and Freebase (Yaghoobzadeh and Schütze, 2018). Each entity in MVET has an name in English, names in other languages, Freebase ID, and FIGER types. MVET contains 102 FIGER types (Ling and Weld, 2012), which forms a 3-level type hierarchy. For each entity, we utilize hyper-link in Wikipedia to find a sentence containing this entity mention. We collect data for six languages: English, German, Spanish, Chinese, Dutch, and Russian. Table 1 shows the statistics.

Metrics To evaluate the performance of our proposed method, we use Accuracy (**Strict-F1**), Micro-averaged F1 (**Mi-F1**) and Macro-averaged F1 (**Ma-F1**), which have been used in many FET systems (Ling and Weld, 2012; Ren et al., 2016; Xu and Barbosa, 2018; Xin et al., 2018).

Baselines We compare our model with five state-of-the-art monolingual methods and two our models as follows: (1) **AttNER** is an attentive neural model that utilizes a fixed attention mechanism to focus on relevant expressions in context (Shimaoka et al., 2017); (2) **NFETC** utilizes a variant of cross-entropy loss function and hierarchical loss normalization to handle *out-of-context* noise and *overly-specific* noise (Xu and Barbosa, 2018); (3) **LTR** utilises a hybrid classification method beyond binary relevance to exploit type inter-dependency with latent type representation (Lin and Ji, 2019); (4) **MLL2R** uses multi-level learning to rank approach that embraces type hierarchy during both training and prediction (Chen et al., 2020b); (5) **VAT** alleviates dataset shift problem in FET by combining the proposed masked VAT with denoising methods (Shi et al., 2020). **Our**$_{no\_adv}$ is a variant of our model **Our**, which removes language

[2] https://github.com/SIGKDD/CLET

discriminator $\mathcal{D}$. For each baseline, we apply the same feature extractor, given a mention m_i with its context c_i, we feed the sequence ([CLS], c_i, [SEP], m_i, [SEP]) to mBERT encoder and use the output of [CLS] token as the representation of the mention with its context.

Parameter Settings We implement our approach with PyTorch 1.2.0. In all experiments, Adam is used for optimizers, with learning rate 0.0002 for Chinese and 0.001 for European languages, and weight decay 10^{-8} for all languages. Batch size is 32 for Chinese experiment and 64 for European languages. We use the cased multilingual BERT-BASE with 12 Transformer blocks, 768 hidden units, 12 self-attention heads, GELU activations, a dropout rate of 0.1 and learned positional embeddings. WordPiece embeddings are employed to split a word into subwords, which are then directly fed into the model without any other preprocessing. Hyper-parameters are empirically selected and utilized in all experiments as follows: $\lambda = 0.2$ and $\gamma = 0.005$ for Chinese as the target language, $\lambda = 0.2$ and $\gamma = 0.01$ for Russian as the target language, $\lambda = 0.25$ and $\gamma = 0.005$ for Dutch as the target language.

4.1.2 Overall Comparison Results

We take English, German and Spanish as source languages ($N = 3$), and one of the three remaining as target language. Table 2 show that our model consistently outperforms the state-of-the-art monolingual entity typing methods on three target languages. This shows that our model has the strong ability to transfer knowledge to new languages. Our model outperforms the best baseline with 4.0% and 3.9% in Mi-F1 and Ma-F1 on *Chinese* dataset, with 6.4% and 5.2% in Mi-F1 and Ma-F1 on *Dutch* dataset, with 6.0% and 6.2% in Mi-F1 and Ma-F1 on *Russian* dataset, respectively.

4.1.3 Analysis

Compared with monolingual methods, our method has two advantages. First, it can explicitly capture the relationship between a target entity and different source languages via a mixture-of-experts approach. In testing, metric module will calculate the similarity between the target entity and each source language. If the test entity is more similar to $\mathcal{S}_i$ training examples, the trained metric function M will predict a higher α for the expert $F_{\mathcal{S}_i}$. Second, we utilize language discriminator to further improve transfer quality between different languages. We fine-tune all the parameters from mBERT as well as parameters in our model jointly. Our cross-language approach can be seen as an effective way to augment training data for entity typing using different languages of data available.

Our full model outperform its variant (which removes language discriminator $\mathcal{D}$) in all target languages consistently. This shows that language adversarial training really improve transfer quality on new language, because language adversarial training can be viewed as a kind of pre-training in target language.

4.1.4 Unseen Entities

We aim at testing whether our model is able to predict types for new entities.

Data We remove entities which appear during training (in any source language), we call this entity-level zero-shot learning. We take English, German and Spanish as source languages, and one of the three remaining as target language.

Results and Analysis Table 4 shows that the performance slightly decreased. This shows that our model has a degree of memory ability, in part because our model can extract and learn language-specific and language-invariant features for entities. These features appear in any source language training data, so in testing they help to make accurate judgements.

4.1.5 Type Size and Performance

The aim is to evaluate whether type size could effect the performance of type prediction.

Data We measure our model's performance on different types. They are grouped into two groups: Head Type Group and Tail Type Group. Head Type Group has 24 types, each has at least 300 entities; Tail Type Group has 15 types, each has at most 20 entities.

Results and Analysis Macro-averaged F1 metrics are reported in Table 3. Note that we use a different evaluation metric to calculate the F1 score for a type (Ren et al., 2016). Experiments results show that our model outperform other baselines and works for both type groups. Generally, the performance on Head Type Group is better than Tail Type Group. Our model consistently outperforms the other methods on Tail Type Group. This shows that our model can deal with rare types. As types in Head Type Group are more coarse-grained and

Methods	EN+DE+ES→ ZH			EN+DE+ES→ NL			EN+DE+ES→ RU		
	Strict	Ma-F1	Mi-F1	Strict	Ma-F1	Mi-F1	Strict	Ma-F1	Mi-F1
AttNER	0.551	0.702	0.722	0.557	0.707	0.724	0.546	0.689	0.706
NFETC	0.582	0.739	0.753	0.572	0.722	0.742	0.561	0.711	0.729
MLL2R	0.575	0.721	0.740	0.581	0.736	0.752	0.572	0.724	0.745
VAT	0.586	0.744	0.761	0.587	0.750	0.765	0.577	0.729	0.749
LTR	0.594	0.753	0.772	0.585	0.744	0.761	0.580	0.733	0.757
Our$_{no\_adv}$	0.626	0.775	0.791	0.636	0.794	0.822	0.623	0.788	0.812
Our	**0.636**	**0.792**	**0.812**	**0.640**	**0.802**	**0.829**	**0.628**	**0.795**	**0.817**

Table 2: Overall performance on three target languages.

Methods	Head Type Group					Tail Type Group				
	Org.	Person	Loc.	Work	Avg.	Durg	Law	Algorithm	TV channel	Avg.
AttNER	0.554	0.527	0.552	0.518	0.524	0.216	0.273	0.316	0.335	0.327
NFETC	0.574	0.539	0.565	0.537	0.536	0.292	0.289	0.344	0.352	0.365
MLL2R	0.568	0.548	0.587	0.562	0.542	0.303	0.318	0.326	0.361	0.377
VAT	0.586	0.623	0.607	0.583	0.547	0.288	0.329	0.357	0.348	0.369
LTR	0.613	0.605	0.612	0.592	0.554	0.316	**0.342**	0.337	0.350	0.392
Our$_{no\_adv}$	0.677	0.647	0.625	**0.615**	0.561	0.317	0.329	0.382	0.373	**0.414**
Our	**0.693**	**0.652**	**0.631**	0.613	**0.565**	**0.325**	0.331	**0.386**	**0.388**	0.407

Table 3: Performance on different types.

Target	Our			Our$_{no_adv}$		
	Strict	Ma-F1	Mi-F1	Strict	Ma-F1	Mi-F1
ZH	0.627	0.778	0.794	0.621	0.768	0.782
NL	0.635	0.792	0.818	0.627	0.781	0.807
RU	0.622	0.785	0.809	0.615	0.773	0.793

Table 4: Typing performance on unseen entities.

have more training data than types in Tail Type Group, our model performs better in predicting types in Head Type Group.

4.2 How does Language Similarity Affect Cross-lingual Type Prediction?

The idea of using rich source languages to predict entity types in low resource language may lead to following two hypotheses: (1) the richer the source is, the better predicting performance it will be; (2) the more sources, the better. Following experiments and the experiment results in Table 2 refute the two hypotheses and show that the similarity between source and target plays an important role.

4.2.1 Dataset

Languages are grouped into three level of similarity: (1) the similar level has three languages: English, German, and Dutch. All are in the west Germanic language family; (2) the less similar level consists of five languages in three language categories: Spanish in the Romance language family, Russian in the Slavic language family, and three languages from the similar level in the Germanic language family; (3) the dissimilar level consists of six languages in two language families: Chinese in the Sino-Tibetan language family, and the five European languages in the less similar level.

4.2.2 The Similar Group

Language similarity English, German, and Dutch are west Germanic languages. They are similar. Spanish is Romance language, and is less similar to English, German, and Dutch.

Experiments & results We conducted six experiments: (1) three experiments by selecting any one from {EN, DE, ES} as source; (2) three experiments select any two from {EN, DE, ES} as sources. Experiment results in Table 5 show that: (1) Comparing with using English or German as single source language, using both of them achieves the best performance. Dutch is one of the closest relatives of both German and English and is colloquially said to be "roughly in between" them. For Dutch, some linguistic features are similar with English, some features are more similar with Ger-

$\mathbb{S}+\mathcal{T}$	Baseline			Our$_{no_adv}$			Our		
	Strict	Ma-F1	Mi-F1	Strict	Ma-F1	Mi-F1	Strict	Ma-F1	Mi-F1
EN→NL	0.561	0.709	0.726	0.627	0.776	0.793	*0.634*	*0.789*	*0.814*
DE→NL	0.580	0.735	0.751	0.631	0.786	0.809	*0.638*	*0.799*	*0.826*
ES→NL	0.556	0.702	0.719	*0.619*	*0.766*	*0.782*	0.615	0.758	0.773
EN+DE→NL	0.592	0.758	0.774	0.642	0.805	0.833	**0.652**	**0.819**	**0.843**
DE+ES→NL	0.572	0.723	0.743	0.620	0.768	0.783	*0.632*	*0.788*	*0.811*
EN+ES→NL	0.564	0.715	0.733	0.623	0.771	0.787	*0.628*	*0.780*	*0.797*
EN+DE+ES→NL	0.587	0.750	0.765	0.636	0.794	0.822	*0.640*	*0.802*	*0.829*
EN→RU	0.586	0.742	0.766	0.614	0.773	0.792	*0.619*	*0.781*	*0.801*
DE→RU	0.564	0.713	0.727	*0.601*	*0.759*	*0.775*	0.597	0.752	0.770
ES→RU	0.584	0.739	0.762	0.626	0.793	0.814	**0.633**	**0.802**	**0.827**
EN+DE→RU	0.583	0.736	0.760	0.618	0.779	0.799	*0.622*	*0.784*	*0.808*
DE+ES→RU	0.576	0.727	0.744	0.615	0.775	0.795	*0.624*	*0.789*	*0.811*
EN+ES→RU	0.591	0.751	0.773	0.620	0.782	0.803	*0.627*	*0.795*	*0.815*
EN+DE+ES→RU	0.580	0.733	0.757	0.623	0.788	0.812	*0.628*	*0.795*	*0.817*
EN→ZH	0.573	0.719	0.738	0.615	0.754	0.771	*0.621*	*0.768*	*0.782*
DE→ZH	0.559	0.696	0.713	0.604	0.742	0.760	*0.611*	*0.750*	*0.768*
ES→ZH	0.577	0.726	0.744	0.610	0.748	0.763	*0.617*	*0.760*	*0.775*
EN+DE→ZH	0.590	0.748	0.766	0.623	0.771	0.786	*0.631*	*0.783*	*0.805*
DE+ES→ZH	0.583	0.737	0.756	0.619	0.764	0.779	*0.626*	*0.775*	*0.792*
EN+ES→ZH	0.580	0.731	0.750	*0.632*	*0.785*	*0.807*	0.628	0.778	0.797
EN+DE+ES→ZH	0.594	0.753	0.772	0.626	0.775	0.791	**0.636**	**0.792**	**0.812**

Table 5: Typing performance on target language with different source language combinations.

man, so take both of them into account is the best way; (2) Adding less similar language source, here, adding Spanish to English and German, decreases the performance from **0.652** (shown in Table 3) to **0.640** (shown in Table 2). This shows that we do not need to use all of the training data in different languages, sometimes it may mislead the model's judgment.

4.2.3 The Less Similar Group

Language similarity English and German are Germanic languages, Spanish is a Romance language. As a Slavic language, Russian is less similar to Spanish, and much less similar to English and German.

Experiments & results We conducted six experiments: (1) three experiments by selecting any one from {EN, DE, ES} as source; (2) three experiments by selecting any two from {EN, DE, ES} as source. Experiment results in Table 5 show that: (1) Using Spanish as the single source reaches the best performance **0.633**; (2) If Spanish appears in the source language set, the performance is better; (3) Adding less similar language source (adding EN and DE to SP) may decrease the performance. The performance **0.628** of using all three languages is shown in Table 2; (4) Adding similar language source (adding German to English, or vice versa) improves the performance.

4.2.4 The Dissimilar Group

Language similarity English, German, and Spanish as source in the European language family, and Chinese as the target in the Sino-Tibetan language family, and is significantly dissimilar from the European languages.

Experiments & results We conducted six experiments: (1) three experiments by selecting any one from {EN, DE, ES} as source; (2) three experiments by selecting any two from {EN, DE, ES} as source. Experiment results in Table 5 show adding dissimilar source consistently increases the performance. The best performance **0.636** is achieved by using all three source languages, shown in Table 2. English is relatively more important than German and Spanish, to predict Chinese entity types. Besides the fact that English has more training samples than German and Spanish, English has

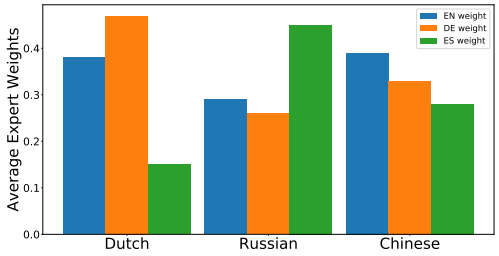

Figure 4: Average expert weights aggregated on language level.

the poorest inflection system among all of Indo-European and Ural-Altaic languages, and shares some similarity with Chinese, in the sense that the word-order plays the dominant role in conveying meanings (Bates et al., 1984; Li et al., 1993).

4.2.5 From Expert Weights to Peep How Language Similarity Affects Type Prediction

We take the setting of using three source languages, and compute the instance-level expert weights for each entity, then average across all entities in the validation set, resulting a final language-level average expert weight for each source language. Fig. 4 shows the average expert weights for each target language, and further strengthens our claim that the more similar between the source language and the target, the larger weight the source language expert will be. In particular, (1) for Dutch, German Expert has the largest weight, and English and German Experts have much larger weights than Spanish Expert; (2) for Russian, Spanish Expert has much larger weight than German and English Experts, and English Expert has lightly larger weight than German Expert; (3) for Chinese, English Expert has the largest weight that is slightly larger than German Expert that is slightly larger than Spanish.

4.2.6 Similarity Hypothesis for Cross-Lingual Entity Typing

As a summary, we propose the *Similarity Hypothesis for CLET* as follows: *The more similar the source and the target are, the better the performance will be; A large set of source languages with a high deviation of similarity performs worse than one of its subsets whose members are more similar to the target than other sources.*

5 Related Work

Fine-grained Entity Typing. FET research targets at utilising sentence-level context for making predictions (Ling and Weld, 2012) and (Gillick et al., 2014), in which they created the commonly used *FIGER* and *OntoNotes* datasets. (Shimaoka et al., 2017) proposed an attentive LSTM network model to encode an entity context, and proposed an attention mechanism to allow the model to focus on relevant expressions in a context. (Xu and Barbosa, 2018) studied two kinds of noises, namely, *out-of-context* noise and *overly-specific* noise in training data. (Wu et al., 2019) leveraged a novel cost function to jointly model the correlation among hierarchical types and label noises. (Xiong et al., 2019) presented an effective method to impose label-relational inductive bias on fine-grained entity typing models. (Onoe and Durrett, 2019) investigated the problem of denoising distant training data for entity typing tasks. (Chen et al., 2019a) regularized distantly supervised models with Compact Latent Space Clustering (CLSC) to effectively utilize noisy data. (Lin and Ji, 2019; Shi et al., 2020) employed contextualized word representations to further boosts the performance.

Cross-lingual task in NLP. To tackle the low-resourced problem, many cross-lingual transfer learning models have been proposed. Most of the research focuses on bilingual transfer case. (Xu and Yang, 2017) introduced a framework for distillation of discriminative knowledge across languages, focusing on the domain/distribution mismatch issues in cross-lingual text classification problem. (Chen et al., 2018) utilized an adversarial deep averaging network to extract language-invariant features for cross-lingual sentiment classification. (Wu et al., 2020) proposed a teacher-student learning method, where NER models in the source languages are used as teachers to train a student model on unlabeled data in the target language. Recently, some researches focus on the multi-source scenario, also known as multilingual transfer learning (MLTL). (Chen et al., 2019b) used both language-invariant and -specific features to improve the performance on the target language. (Karamanolakis et al., 2020) presented a cross-lingual text classification method, which extracts and transfers a small number of task-specific seed words, and creates a teacher that provides weak supervision to train a more powerful student in the target language.

6 Conclusions and Discussions

We address the problem of Cross-Lingual Entity Typing (CLET) in an unsupervised setting, and propose a mixture-of-experts (MoE) approach to dynamically capture the relation between the target language and each source language, which enables to acquire more knowledge from source languages. Experiments on multi-lingual datasets show that this approach outperforms various baselines and can effectively predict types of unseen entities in new languages. The presented work is the first to investigate how language similarity affects the performance of CLET, and propose the *Similarity Hypothesis*. This will be helpful for the empirical selection of source languages, and raises new questions, such as how we can quantitatively compute and compare similarities among languages.

7 Acknowledgements

This work is supported by the National Key Research and Development Program of China (2020AAA0106501), the grants from the Institute for Guo Qiang, Tsinghua University (2019GQB0003) and Beijing Academy of Artificial Intelligence, and Alibaba Inc.

References

E. Bates, B. MacWhinney, C. Caselli, A. Devescovi, F. Tatale, and V. Venza. 1984. A Cross-linguistic Study of the Development of Sentence Interpretation Strategies. *Child Development*, 55:341–354.

Bo Chen, Xiaotao Gu, Yufeng Hu, Siliang Tang, Guoping Hu, Yueting Zhuang, and Xiang Ren. 2019a. Improving distantly-supervised entity typing with compact latent space clustering. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2862–2872.

Shuang Chen, Jinpeng Wang, Feng Jiang, and Chin-Yew Lin. 2020a. Improving entity linking by modeling latent entity type information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7529–7537.

Tongfei Chen, Yunmo Chen, and Benjamin Van Durme. 2020b. Hierarchical entity typing via multi-level learning to rank. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8465–8475.

Xilun Chen, Ahmed Hassan, Hany Hassan, Wei Wang, and Claire Cardie. 2019b. Multi-source cross-lingual model transfer: Learning what to share. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3098–3112.

Xilun Chen, Yu Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Weinberger. 2018. Adversarial deep averaging networks for cross-lingual sentiment classification. *Transactions of the Association for Computational Linguistics*, 6:557–570.

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

Xiangyu Dong, Wenhao Yu, Chenguang Zhu, and Meng Jiang. 2021. Injecting entity types into entity-guided text generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 734–741.

Hady Elsahar, Christophe Gravier, and Frederique Laforest. 2018. Zero-shot question generation from knowledge graphs for unseen predicates and entity types. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 218–228.

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

Jiang Guo, Darsh Shah, and Regina Barzilay. 2018. Multi-source domain adaptation with mixture of experts. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4694–4703.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87.

Giannis Karamanolakis, Daniel Hsu, and Luis Gravano. 2020. Cross-lingual text classification with minimal resources by transferring a sparse teacher. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 3604–3622.

Jun Kuang, Yixin Cao, Jianbing Zheng, Xiangnan He, Ming Gao, and Aoying Zhou. 2020. Improving neural relation extraction with implicit mutual relations. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pages 1021–1032. IEEE.

P. Li, E. Bates, and B. MacWhinney. 1993. Processing a Language without Inflections: A Reaction Time Study of Sentence Interpretation in Chinese. *Journal of Memory and Language*, 32:169–192.

Ying Lin and Heng Ji. 2019. An attentive fine-grained entity typing model with latent type representation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6198–6203.

Xiao Ling and Daniel S Weld. 2012. Fine-grained entity recognition. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Yasumasa Onoe and Greg Durrett. 2019. Learning to denoise distantly-labeled data for entity typing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2407–2417.

Yasumasa Onoe and Greg Durrett. 2020. Fine-grained entity typing for domain independent entity linking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8576–8583.

Xiang Ren, Wenqi He, Meng Qu, Lifu Huang, Heng Ji, and Jiawei Han. 2016. Afet: Automatic fine-grained entity typing by hierarchical partial-label embedding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1369–1378.

Nila Selvaraj, Yasumasa Onoe, and Greg Durrett. 2021. Cross-lingual fine-grained entity typing. *arXiv preprint arXiv:2110.07837*.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

Haochen Shi, Siliang Tang, Xiaotao Gu, Bo Chen, Zhigang Chen, Jian Shao, and Xiang Ren. 2020. Alleviate dataset shift problem in fine-grained entity typing with virtual adversarial training. In *IJCAI*, pages 3898–3904.

Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. 2017. Neural architectures for fine-grained entity type classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1271–1280.

Shikhar Vashishth, Rishabh Joshi, Sai Suman Prayaga, Chiranjib Bhattacharyya, and Partha Talukdar. 2018. Reside: Improving distantly-supervised neural relation extraction using side information. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1257–1266.

Junshuang Wu, Richong Zhang, Yongyi Mao, Hongyu Guo, and Jinpeng Huai. 2019. Modeling noisy hierarchical types in fine-grained entity typing: a content-based weighting approach. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 5264–5270. AAAI Press.

Qianhui Wu, Zijia Lin, Börje Karlsson, Jian-Guang Lou, and Biqing Huang. 2020. Single-/multi-source cross-lingual ner via teacher-student learning on unlabeled data in target language. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6505–6514.

Ji Xin, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2018. Improving neural fine-grained entity typing with knowledge attention. In *Thirty-second AAAI conference on artificial intelligence*.

Wenhan Xiong, Jiawei Wu, Deren Lei, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. 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)*, pages 773–784.

Peng Xu and Denilson Barbosa. 2018. Neural fine-grained entity type classification with hierarchy-aware loss. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 16–25.

Ruochen Xu and Yiming Yang. 2017. Cross-lingual distillation for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1415–1425.

Yadollah Yaghoobzadeh and Hinrich Schütze. 2018. Multi-multi-view learning: Multilingual and multi-representation entity typing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3060–3066.

Ming Zhu, Busra Celikkaya, Parminder Bhatia, and Chandan K Reddy. 2020. Latte: Latent type modeling for biomedical entity linking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9757–9764.

8 Appendix

Algorithm 1: model training

Input: Training data on multiple source languages $\mathbb{S} = \{\mathcal{S}_i\}_{i=1}^{N}$, test data on target language $\mathcal{T}$
Output: Cross-lingual Fine-grained Entity Typing model $P_{MoE}(y|x)$

repeat
  # $\mathcal{D}$ iteration, update parameters in language discriminator $\mathcal{D}$
  for $iter = 1$ *to* k **do**
    $\mathcal{L}_D \leftarrow 0$
    Sample N source mini-batches $\{x_t^{\mathcal{S}_1}\}_{t=1}^{m}, \cdots, \{x_t^{\mathcal{S}_N}\}_{t=1}^{m}$ from $\mathbb{S}$
    for $i = 1$ *to* N **do**
      Calculate cross-entropy loss of language label on source $\mathcal{S}_i$, and add to $\mathcal{L}_D$
    end
    Sample a target mini-batch $\{x_i^{\mathcal{T}}\}_{i=1}^{m}$ from $\mathcal{T}$
    Calculate cross-entropy loss of language label on target $\mathcal{T}$, and add to $\mathcal{L}_D$
    Update parameters in $\mathcal{D}$ using $\nabla\mathcal{L}_D$
  end
  # Main iteration, update parameters in encoder E, experts $\{F_{\mathcal{S}_i}\}_{i=1}^{N}$ and metric function M
  $\mathcal{L}_{moe}, \mathcal{L}_{sup}, \mathcal{L}_{adv} \leftarrow 0$
  Sample N source mini-batches $\{(x_t^{\mathcal{S}_1}, y_t^{\mathcal{S}_1})\}_{t=1}^{m}, \cdots, \{(x_t^{\mathcal{S}_N}, y_t^{\mathcal{S}_N})\}_{t=1}^{m}$ from $\mathbb{S}$
  for $i = 1$ *to* N **do**
    Set meta-target as $\mathcal{T}_{meta} = \mathcal{S}_i$, meta-sources as $\mathcal{S}_{meta} = \{\mathcal{S}_j\}_{j=1,j\neq i}^{N}$
    Calculate cross-entropy loss of type information on $\mathcal{T}_{meta}$, and add to $\mathcal{L}_{sup}$
    Calculate metric weight $\alpha(x, \mathcal{S}^{'})$ for each $x \in \mathcal{T}_{meta}$ and $\mathcal{S}^{'} \in \mathcal{S}_{meta}$
    Calculate MoE loss over $(\mathcal{S}_{meta}, \mathcal{T}_{meta})$ using α, and add to $\mathcal{L}_{moe}$
    Calculate cross-entropy loss of language label on $\mathcal{T}_{meta}$, and add to $\mathcal{L}_{adv}$
  end
  Sample a target mini-batch $\{x_i^{\mathcal{T}}\}_{i=1}^{m}$ from $\mathcal{T}$
  Calculate cross-entropy loss of language label on target $\mathcal{T}$, and add to L_{adv}
  $\mathcal{L} \leftarrow \lambda \cdot \mathcal{L}_{moe} + (1 - \lambda) \cdot \mathcal{L}_{sup} + \gamma \cdot \mathcal{L}_{adv}$
  Update parameters in E, $\{F_{\mathcal{S}_i}\}_{i=1}^{N}$ and M using $\nabla\mathcal{L}$
until *convergence*;