USTC-NELSLIP at SemEval-2022 Task 11: Gazetteer-Adapted Integration Network for Multilingual Complex Named Entity Recognition

Beiduo Chen [1] Jun-Yu Ma [1] Jiajun Qi [1] Wu Guo [1] Zhen-Hua Ling [1] Quan Liu [2]

[1]NERC-SLIP, University of Science and Technology of China [2]State Key Laboratory of Cognitive Intelligence, iFLYTEK Research

Introduction

- **Task** multilingual complex named entity recognition (NER) across 11 languages
- **Challenge** recognizing semantically ambiguous and complex entities in low-context settings
- **Solution** the gazetteer-adapted integration network (GAIN). This method first **adapts** the representations of gazetteer networks to those of language models by minimizing the KL divergence between them. After adaptation, these two networks are then **integrated** for backend supervised NER training. The GAIN method is applied to several state-of-the-art Transformer-based NER models with a **gazetteer** built from Wikidata.
- **Results** the final predictions are derived from an **ensemble** model. Our system ranked **1st** on three tracks (Chinese, Code-mixed and Bangla) and **2nd** on the other ten tracks.

Data Preparation & Basic Systems

- **Taxonomy** 6 labels (PER, LOC, CORP, GRP, PROD, CW) in the BIO scheme.
- **Data Preparation** an entity replacement strategy is adopted on the official training set using our own gazetteer to construct a double data-augmented set. In order to improve the performance of our models on low-context instances, a set of augmented data with pseudo labels are generated from the MS-MARCO QnA corpus (V2.1) and the ORCAS dataset.
- **Basic Systems** the XLM-RoBERTa large is mainly used as the pre-trained language model with an appended dense layer. Three mainstream NER backend classifiers are adopted: Softmax, CRF are classic sequential labeling methods that predict the tag of each token, and Span is a segment-based method that predicts the start and the end of an entity separately.

System Architecture

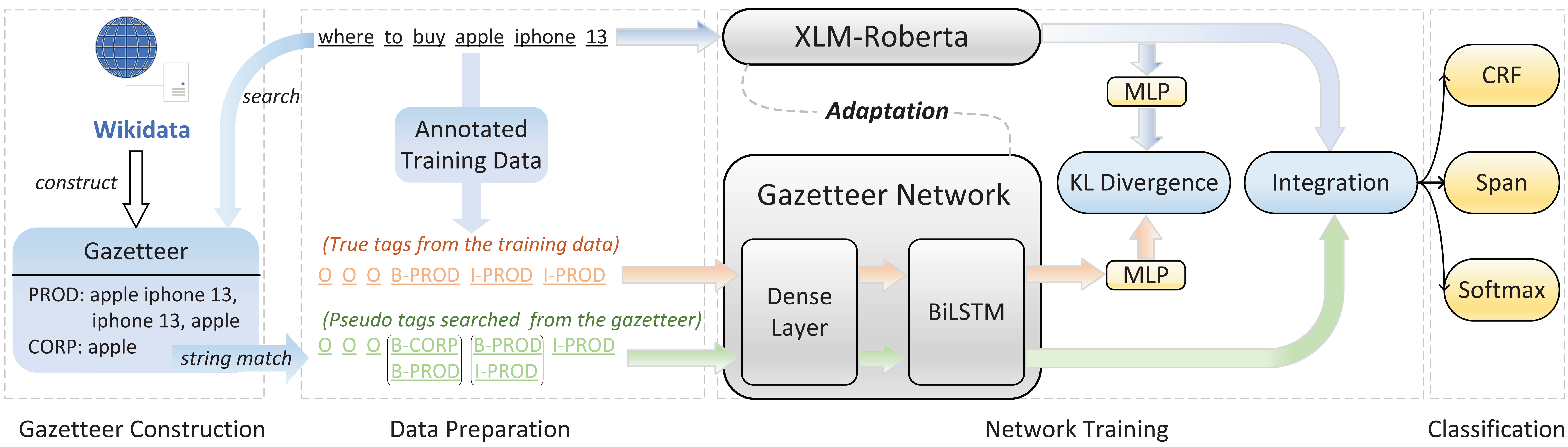

Figure 1. The overall structure of the proposed system. A gazetteer is built from Wikipedia and then integrated into the XLM-Roberta for complex named entity recognition.

Gazetteer Construction & Application

Words	O	B-CORP	I-CORP	B-PROD	I-PROD
where	1	0	0	0	0
to	1	0	0	0	0
buy	1	0	0	0	0
apple	0	1	0	1	0
iphone	0	0	0	1	1
13	0	0	0	0	1

Table 1. Example of the one-hot representation generated from a search tree built by our gazetteer. The rest 8 labels are zero.

Firstly every entity of the training set is searched in Wikidata. Then all the entity types returned are mapped to the NER taxonomy with 6 labels. Next, all Wikidata entities stored in these entity types can be added to the 6 labels gazetteer separately. Finally, a multilingual gazetteer is obtained that contains entities from 70K to 1M for each language. The gazetteer approximately has an average coverage rate of 57 percent.

Denote one sentence as $\mathbf{w} = (w_1, w_2, ...w_N)$. By feeding $\mathbf{w}$ into the language model such as the XLM-RoBERTa large, a semantic representation $\mathbf{e} \in \mathbb{R}^{N \times D}$ is obtained, where D is the hidden size. At the same time, the one-hot vector (As shown in Table 1) generated from the search tree is fed into a gazetteer network consisting of a dense layer and a BiLSTM. To match the hidden size of the language model, the output embedding $\mathbf{g}$ has the same size with $\mathbf{e}$.

Gazetteer-Adapted Integration Network

The GAIN method is proposed as a two-stage training strategy. For a sentence $\mathbf{w}$ of the training set, denote $\mathbf{g}_r \in \mathbb{R}^{N \times D}$ and $\mathbf{g}$ are the gazetteer representation of true tags T and searched pseudo tags respectively. $\{\mathbf{g}_r, \mathbf{e}\}$ are projected to $\{\mathbf{g}_r^t, \mathbf{e}^t\} \in \mathbb{R}^{N \times 13}$ by two separate linear layers, where the semantic meaning is transferred to the tags meaning as a kind of logits distributions. In the first stage, the adaptation between the two networks is conducted:

$$L_1(\mathbf{w}) = \mathrm{KL}(sg(\mathbf{g}_r^t)||\mathbf{e}^t) + \mathrm{KL}(sg(\mathbf{e}^t)||\mathbf{g}_r^t) \tag{1}$$

where $\mathrm{KL}(\cdot)$ is the KL divergence calculation and $sg(\cdot)$ is used to stop back-propagating gradients. In the second stage, the supervised training is implemented with the gazetteer:

$$L_2(\mathbf{w}) = \mathrm{Classifier}(f(\mathbf{g}, \mathbf{e}), \mathrm{T}) \tag{2}$$

where $f(\cdot)$ denotes ordinary integration methods like concatenation or weighted summation. $\mathrm{Classifier}(\cdot)$ represents one of the three mainstream backend classifiers. During the whole second-stage training, a multitask learning goal is conducted shown as:

$$L_3(\mathbf{w}) = \alpha L_1(\mathbf{w}) + L_2(\mathbf{w}) \tag{3}$$

where α is a hyperparameter that is manually set for different fusion and backend methods.

Experiments

Strategy	Classifier	BN	DE	EN	ES	FA	HI	KO	NL	RU	TR	ZH	MIX
A	CRF	0.771	0.886	0.846	0.834	0.78	0.771	0.813	0.878	0.802	0.835	0.866	0.654
	Softmax	0.763	0.879	0.849	0.836	0.783	0.767	0.811	0.871	0.792	0.835	0.862	0.652
	Span	0.793	0.896	0.853	0.845	0.806	0.802	0.831	0.879	0.809	0.839	0.884	0.696
B	CRF	0.816	0.906	0.865	0.857	0.821	0.8	0.853	0.888	0.817	0.865	0.908	0.788
	Softmax	0.799	0.901	0.865	0.859	0.824	0.796	0.851	0.879	0.815	0.864	0.901	0.786
	Span	0.811	0.917	0.871	0.857	0.818	0.825	0.864	0.887	0.82	0.858	0.906	0.792
C	CRF	0.841	0.943	0.891	0.87	0.835	0.831	0.871	0.902	0.829	0.884	0.913	0.833
	Softmax	0.829	0.931	0.888	0.872	0.839	0.822	0.868	0.897	0.831	0.882	0.909	0.835
	Span	0.832	0.935	0.892	0.874	0.836	0.837	0.879	0.901	0.836	0.872	0.912	0.823
weighted token-vote		**0.864**	**0.955**	**0.922**	**0.892**	**0.855**	**0.853**	**0.899**	**0.916**	**0.843**	**0.903**	**0.922**	**0.865**

Table 2. All macro-F1 scores on the validation set in the main experiments.

Coverage Rate	BN	DE	EN	ES	FA	HI	KO	NL	RU	TR	ZH	MIX	avg
0	0.784	0.897	0.856	0.847	0.8	0.775	0.839	0.892	0.806	0.855	0.863	0.662	0.823
30%	0.791	0.898	0.861	0.845	0.804	0.799	0.84	0.893	0.814	0.856	0.872	0.694	0.831
50%	0.858	0.901	0.867	0.844	0.807	0.871	0.866	0.897	0.814	0.861	0.903	0.709	0.85
70%	0.891	0.907	0.868	0.854	0.811	0.899	0.894	0.901	0.82	0.869	0.904	0.732	0.863
100%	0.974	0.973	0.942	0.92	0.903	0.978	0.938	0.94	0.91	0.91	0.964	0.914	0.934

Table 3. F1 scores of the gradient coverage rate trial. "Coverage Rate" means the number of entities in official data also found in our gazetteer / the number of entities in official data. "avg" denotes the average result of all tracks.

To explore the effectiveness of the proposed GAIN method, a large number of trials are conducted on the official data. All scores under the concatenation integration setting on the validation set are listed in Table 2. Strategy A denotes basic NER systems, B denotes ordinary integration method with the gazetteer, and C denotes the GAIN method. "weighted token-vote" represents the ensemble of all our models including those with the weighted summation integration method not listed, and achieves the best performance on the validation set. The results demonstrate that the gazetteer plays a pivotal role in processing complex entities.

Our gazetteer can only reach approximately 57% coverage rate over the entities in the official data. Intuitively, the higher the coverage rate over the entities reaches, the better the performance can achieve. We carry out a toy experiment to explore this conjecture. As shown in Table 3, not as expected, scores on many tracks don't improve in step with the increase of the coverage rate when it does not reach 100%. This situation mostly happens in languages that already have good scores, like DE and EN. To explore the reason why the gazetteers under 100% coverage rate don't work, we check the weight λ of weighted summation fusion method. It's surprising to find the λ is nearly zero when the coverage rate is not 100%, which indicates that our models almost don't use the gazetteer information.

Further fine-grained tests find that only when the coverage rate exceeds the basic accuracy of the model, the λ starts to have a non-zero value. An empirical conclusion can be drawn that the gazetteer network and the language model almost process information separately, and the final integration module simply selects the better one for classifying. Thus, this study starts to figure out an explicit way to adapt the two networks, and the GAIN method is designed.