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]National Engineering Research Center for Speech and Language Information Processing, University of Science and Technology of China

[2]State Key Laboratory of Cognitive Intelligence, iFLYTEK Research

Outline

- **Introduction**
- Data Preparation & Basic Systems
- Gazetteer Construction & Application
- Gazetteer-Adapted Integration Network (GAIN)
- Experiments & Results

Task Description

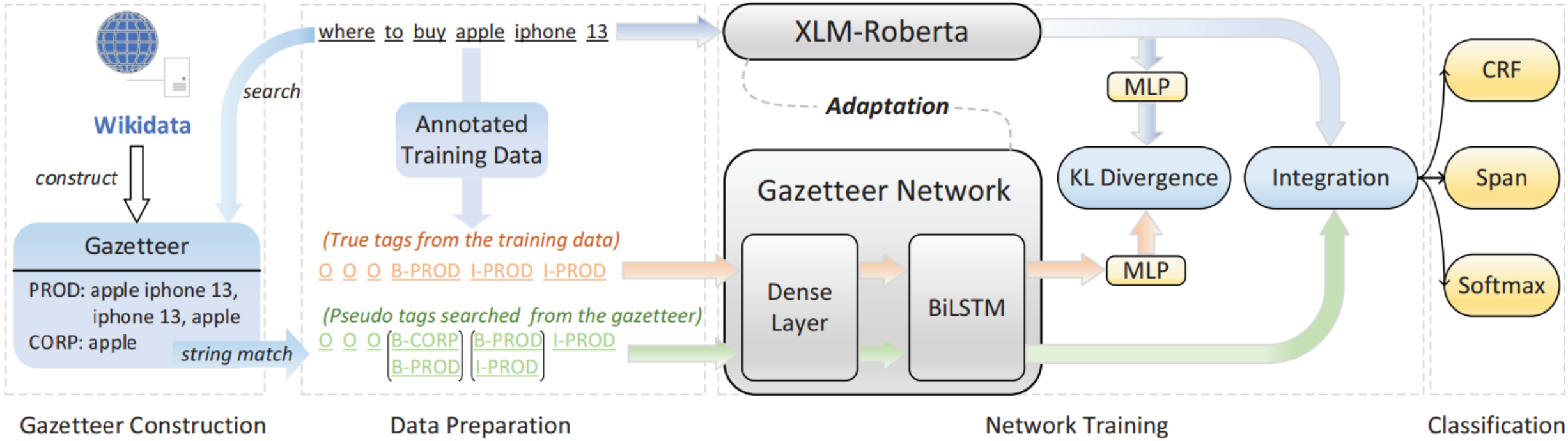
- Complex named entity recognition across 11 languages
- Focusing on recognizing semantically ambiguous and complex entities in short and low-context settings
- Example
  - Ambigous entity: “*On the beach*” (Creative Works)
  - Polysemy: “*Apple*” (Production, Corporation)
  - Emerging: “*Mission Impossible*” (Creative Works)

Recent Solutions

- Integrate external knowledge or gazetteers into networks
- Gazetteer
  - a kind of entity knowledge base
  - store entities by different labels
  - widely used in named entity recognition
- Ordinary solutions
  - get one-hot representations of a sentence by a seach tree constructed from the gazetteer
  - integrate the one-hot embedding with the semantic representation from the language model for classifying
  - concatenation / weighted summation

Our system

- To integrate the gazetteer with the language model better



Outline

- Introduction
- **Data Preparation & Basic Systems**
- Gazetteer Construction & Application
- Gazetteer-Adapted Integration Network (GAIN)
- Experiments & Results

Data Description

- Official data
  - 13 tracks (English, Spanish, Dutch, Russian, Turkish, Korean, Farsi, German, Chinese, Hindi, Bangla, Code-Mixed, Multilingual)
  - 15300 training samples / 800 validation samples
  - 6 labels (PER, LOC, CORP, GRP, PROD, CW) in the BIO scheme
- External challenge
  - short and low-context settings: need more entity information
  - add questions and short search queries in the test phase

Data Augment

- “data wiki”

  an entity replacement strategy is adopted using our own gazetteer built from Wikidata to construct a double data-augmented set.
- “data query”

  a set of augmented data with pseudo labels are generated from the MS-MARCO QnA corpus (V2.1) and the ORCAS dataset.
- “data code-mixed”

  for every sentence in ``data-wiki'' and ``data-query'', the entities inside are randomly replaced with their translations recorded by Wikidata. In this way, a set of annotated code-mixed data are built.

Basic Systems

- Language Model
  - the XLM-RoBERTa large is mainly used as the pre-trained language model with an appended dense layer
  - several state-of-the-art pre-trained monolingual models are also used
- Backend Classifiers
  - Softmax and CRF are classic sequential labeling methods that predict the tag of each token
  - Span is a segment-based method that predicts the start and the end of an entity separately

Outline

- Introduction
- Data Preparation & Basic Systems
- **Gazetteer Construction & Application**
- Gazetteer-Adapted Integration Network (GAIN)
- Experiments & Results

Gazetteer Construction

- Wikidata
  - a free and open knowledge base
  - for example, “apple” can be annotated as a kind of fruit or a well-known high-tech corporation in America. Thus, the word “apple” is given both PROD and CORP labels.
- Construction Procedure
  - every entity of the training set is searched in Wikidata
  - all the entity types returned are mapped to the NER taxonomy
  - all Wikidata entities stored in these entity types can be added to the 6 labels gazetteer separately
  - a multilingual gazetteer is obtained that contains entities from 70K to 1M for each language. The average coverate rate is 57%.

Gazetteer 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

Take the sentence "where to buy apple iphone 13" for example. By string matching with the gazetteer, "apple iphone 13", "iphone 13" and "apple" are found in the PROD gazetteer, while "apple" is also found in the CORP gazetteer. Then a 13-dimension one-hot vector will be generated for every word as shown in the table.

Denote one sentence as **w** = (*w1, w2, …wN*). By feeding **w** into the language model such as the XLM-RoBERTa large, a semantic representation **e** is obtained. At the same time, the one-hot vector 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 **g** has the same size with **e** .

Outline

- Introduction
- Data Preparation & Basic Systems
- Gazetteer Construction & Application
- **Gazetteer-Adapted Integration Network (GAIN)**
- Experiments & Results

The Proposed GAIN Method

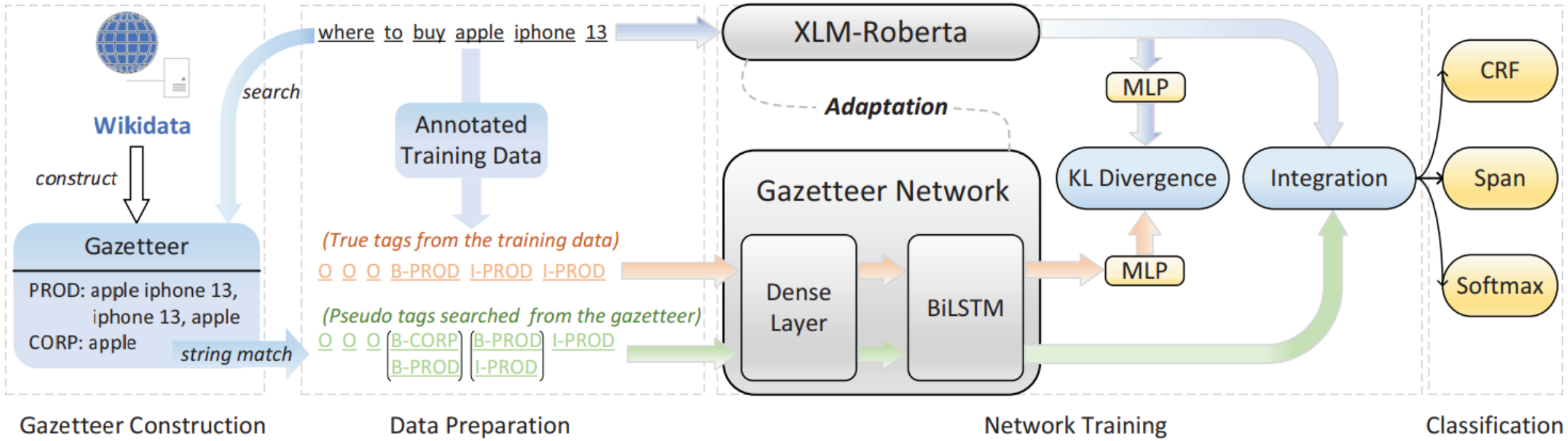
- For a sentence **w** of the training set, denote **g***r* and **g** are the gazetteer representation of true tags T and searched pseudo tags respectively. **e** is the semantic representation generated from the language model. *{***g***r,* **e***}* are projected to *{***g***rt,* **e***t}* by two separate linear layers.

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

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

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

The Architechture of the GAIN



Outline

- Introduction
- Data Preparation & Basic Systems
- Gazetteer Construction & Application
- Gazetteer-Adapted Integration Network (GAIN)
- **Experiments & Results**

Official Results

Track	Team Num.	F1	Rank
English (EN)	30	0.8547	2
Spanish (ES)	18	0.8544	2
Dutch (NL)	15	0.8767	2
Russian (RU)	14	0.8382	2
Turkish (TR)	15	0.8552	2
Korean (KO)	17	0.8636	2
Farsi (FA)	15	0.8705	2
German (DE)	16	0.8905	2
Chinese (ZH)	21	0.8169	1
Hindi (HI)	17	0.8464	2
Bangla (BN)	18	0.8424	1
Multilingual	26	0.853	2
Code-mixed	26	0.929	1

Main 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**

Coverate Rate Trials

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

Thank You !!

Code at GitHub