CSCI 699 Spring 2019
Machine Learning for Knowledge Extraction & Reasoning
Homework 1*

Chi Zhang

USC ID: 6099-4134-05

Department of Computer Science

University of Southern California

February 18, 2019

1 Code Repository

`https://github.com/vermouth1992/CSCI699ml4know/tree/master/hw1`

2 Conditional Random Field

We train conditional random field model using external library sklearn-crfsuite [3] by following the tutorial `https://eli5.readthedocs.io/en/latest/tutorials/sklearn_crfsuite.html`. The main goal in this part is to explore the effect of various features and use them for the RNN-based model.

2.1 Feature Selection

We follow [4] and the tutorial and select the following set of features:

1. The word itself, last three characters and last two characters.
2. Whether the word is uppercase.
3. Whether the word is title. The word is title if the string is a titlecased string and there is at least one character, for example uppercase characters may only follow uncased characters and lowercase characters only cased ones. We use Python builtin function istitle().
4. Whether the word is digit.
5. Whether the word is float.
6. Whether the word contains hyphen.

*Instructor: Xiang Ren

Table 1: Performance of various features using CRF

Features Set	Validation F1	Testa F1
1	68.05	63.26
1 - 2	68.02	63.01
1 - 3	73.30	68.05
1 - 4	73.33	68.21
1 - 5	73.44	67.87
1 - 6	73.37	67.60
1 - 7	75.47	69.52
1 - 8	79.64	76.12

7. The Part-of-Speech tag of the word.

8. The context of the word. We follow [4] by using a window size of 3 that is shown to work best.

We use 25% of the training sentences as validation sentences. We measure the F1 score on validation sentences and testa sentences by adding the feature one by one. All the models are trained using LBFGS optimization with 100 iterations.

2.2 Result

We show the validation F1-score and testa data F1-score for models using various features in Table 1. We summarize the most important features as follows:

- By adding "istitle" feature, the performance boosts by 5%.
- By adding "POStag" feature, the performance boosts by 2%.
- By adding "context" feature, the performance boosts around 6%.

To analyze which label benefits from those features, we show label-specific performance of important features in Figure 1. B-FAC and B-LAW benefits a lot from istitle feature. This is because most B-FAC and B-LAW starts with capitalized characters such as Article-II and Education Improvement. The boost of Postag comes from B-Person, B-ORG, B-NORP and B-LOC. This is because most of these labeled word is NNP. The context feature is crucial important for I-tag because I-tag always comes after B-tag.

3 RNN-based Model

3.1 Overview

We implement RNN-based model using Pytorch version 1.0.0. We use GloVe [2] to initialize the embedding. After the embedding layer, we try BiLSTM vs. CNN layer followed by a fully-connected layer with softmax activation.

Label	Precision	Recall	F1	Support
B-EVENT	0.812	0.454	0.582	86
B-FAC	0.513	0.247	0.333	77
B-GPE	0.816	0.842	0.829	1630
B-LANGUAGE	0.842	0.533	0.653	30
B-LAW	0.571	0.286	0.381	28
B-LOC	0.667	0.424	0.518	165
B-NORP	0.787	0.821	0.804	683
B-ORG	0.783	0.563	0.655	1705
B-PERSON	0.845	0.647	0.733	1713
B-PRODUCT	0.826	0.297	0.437	64
B-WORK_OF_ART	0.609	0.156	0.248	90
I-EVENT	0.726	0.418	0.531	184
I-FAC	0.500	0.402	0.446	117
I-GPE	0.723	0.721	0.722	376
I-LANGUAGE			0.000	0
I-LAW	0.423	0.167	0.239	66
I-LOC	0.680	0.497	0.574	167
I-NORP	0.549	0.538	0.544	52
I-ORG	0.700	0.616	0.655	2365
I-PERSON	0.854	0.736	0.791	1155
I-PRODUCT	0.750	0.500	0.600	48
I-WORK_OF_ART	0.400	0.111	0.174	270
O	0.980	0.992	0.986	151747

(a) Feature set 1 - 2 (No istitle)

Label	Precision	Recall	F1	Support
B-EVENT	0.695	0.477	0.566	86
B-FAC	0.615	0.312	0.414	77
B-GPE	0.829	0.854	0.841	1630
B-LANGUAGE	0.857	0.600	0.706	30
B-LAW	0.700	0.500	0.583	28
B-LOC	0.644	0.515	0.572	165
B-NORP	0.807	0.848	0.827	683
B-ORG	0.788	0.706	0.745	1705
B-PERSON	0.821	0.733	0.775	1713
B-PRODUCT	0.833	0.391	0.532	64
B-WORK_OF_ART	0.408	0.222	0.288	90
I-EVENT	0.615	0.522	0.565	184
I-FAC	0.506	0.376	0.431	117
I-GPE	0.744	0.726	0.735	376
I-LANGUAGE			0.000	0
I-LAW	0.458	0.333	0.386	66
I-LOC	0.692	0.593	0.639	167
I-NORP	0.643	0.519	0.575	52
I-ORG	0.747	0.816	0.780	2365
I-PERSON	0.774	0.817	0.795	1155
I-PRODUCT	0.718	0.583	0.644	48
I-WORK_OF_ART	0.301	0.233	0.263	270
O	0.989	0.991	0.990	151747

(b) Feature set 1 - 6 (No POStag)

Label	Precision	Recall	F1	Support
B-EVENT	0.661	0.500	0.570	86
B-FAC	0.611	0.286	0.389	77
B-GPE	0.835	0.863	0.849	1630
B-LANGUAGE	0.773	0.567	0.654	30
B-LAW	0.619	0.464	0.531	28
B-LOC	0.736	0.576	0.646	165
B-NORP	0.840	0.865	0.852	683
B-ORG	0.801	0.746	0.773	1705
B-PERSON	0.820	0.788	0.804	1713
B-PRODUCT	0.833	0.391	0.532	64
B-WORK_OF_ART	0.476	0.222	0.303	90
I-EVENT	0.569	0.538	0.553	184
I-FAC	0.543	0.376	0.444	117
I-GPE	0.749	0.731	0.740	376
I-LANGUAGE			0.000	0
I-LAW	0.500	0.303	0.377	66
I-LOC	0.682	0.605	0.641	167
I-NORP	0.634	0.500	0.559	52
I-ORG	0.767	0.835	0.800	2365
I-PERSON	0.789	0.870	0.828	1155
I-PRODUCT	0.757	0.583	0.659	48
I-WORK_OF_ART	0.393	0.252	0.307	270
O	0.991	0.993	0.992	151747

(c) Feature set 1 - 7 (No context)

Label	Precision	Recall	F1	Support
B-EVENT	0.754	0.500	0.601	86
B-FAC	0.522	0.312	0.390	77
B-GPE	0.851	0.882	0.867	1630
B-LANGUAGE	0.737	0.467	0.571	30
B-LAW	0.750	0.536	0.625	28
B-LOC	0.717	0.551	0.623	165
B-NORP	0.861	0.854	0.857	683
B-ORG	0.819	0.781	0.799	1705
B-PERSON	0.868	0.841	0.854	1713
B-PRODUCT	0.885	0.359	0.511	64
B-WORK_OF_ART	0.607	0.378	0.466	90
I-EVENT	0.627	0.484	0.546	184
I-FAC	0.533	0.410	0.464	117
I-GPE	0.784	0.782	0.783	376
I-LANGUAGE			0.000	0
I-LAW	0.733	0.333	0.458	66
I-LOC	0.695	0.587	0.636	167
I-NORP	0.750	0.462	0.571	52
I-ORG	0.808	0.859	0.833	2365
I-PERSON	0.861	0.897	0.878	1155
I-PRODUCT	0.839	0.542	0.658	48
I-WORK_OF_ART	0.670	0.459	0.545	270
O	0.992	0.994	0.993	151747

(d) Feature set 1 - 8 (All)

Figure 1: Label-specific performance of various features

3.2 Architecture Comparison

3.2.1 CNN vs BiLSTM

The number of layer is 1 for both architecture. The filter size of CNN is 3 and hidden size for both architecture is 64. The learning rate is set to 1e-3. The F1 score of CNN on testa data is 66.39% and the F1 score of BiLSTM on testa data is 70.77%. The approximate 4% performance boost indicates that name entity tags generally have long-term dependency instead of depending purely on local context words.

3.3 Additional Features

The word embedding can be viewed as features for the word itself. By using CNN or BiLSTM, we include features for context words. Like conditional random fielf, we add additional features including istitle, isdigit and isupper and part-of-speech tag to BiLSTM model and the F1 score on testa is 71.12%.

3.4 Number of Layers

We try 2 layer and 3 layer BiLSTM with 50% dropout. The F1 performance on testa dataset is 73.65% and 72.41%. The model becomes more and more overfitting by adding more layers.

3.5 Embedding Comparison

We try to use contextual embedding BERT [1] without case to finetune a transformer model based on tutorial `https://www.depends-on-the-definition.com/named-entity-recognition-with-bert/`. However, the final performance on testa data is only around 67% and it's hard to analyze what's going wrong.

3.6 Additional thoughts

GloVe and BERT is uncased embedding, which ignore the difference between Education Improvement and education improvement. However, as shown in Part 1, the character-level feature plays an important role in name entity recognition. Thus, we believe there would be huge performance boost by adding char-level features such as using CNN as extractor. Due to limited time, we leave this to future work.

References

[1] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[2] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *In EMNLP*, 2014.

[3] TeamHG-Memex. sklearn-crfsuite. https://eli5.readthedocs.io/en/latest/tutorials/.

[4] M. Tkatchenko and A. Simanovsky. Named entity recognition: Exploring features. In *KONVENS*, 2012.