CSCI 699: Assignment #1
Named Entity Recognition

Last updated: September 13, 2018

Overview

In the first assignment, you will gain hand-on experience in Named Entity Recognition (NER) as a sequence labeling task. Two commonly used methods will be explored in this assignment. Please download the data and starter code from dropbox.

You should complete this assignment individually and submit the assignment to TA (Hongtao Lin) by email (`lin498@usc.edu`) before **9/26 11:59 PM PST**. 10 points will be deducted for late submissions.

Introduction to NER

For a given a word in a context, we want to predict whether it represents one of four categories:

- Person (PER): e.g. "Martha Stewart", "Obama", "Tim Wagner", etc. Pronouns like "he" or "she" are not considered named entities.
- Organization (ORG): e.g. "American Airlines", "Google", "Department of Defense".
- Location (LOC): e.g. "Germany", "Panama Canal", "Brussels", but not unnamed locations like "the bar" or "the farm".
- Miscellaneous (MISC): e.g. "Japanese", "USD", "1,000", "Englishmen".

We formulate this as a 5-class classification problem, using the four above classes and a null-class (O) for words that do not represent a named entity (most words fall into this category). For an entity that spans multiple words ("Department of Defense"), each word is separately tagged, and every contiguous sequence of non-null tags is considered to be an entity.

Dataset

We use CONLL 2003 dataset [1] in our assignment. The dataset consists of three parts: `train`, `testa` and `testb`. We intentionally remove the labels from `testb` for final evaluation. For development purpose, we suggested splitting `train` set into `train-dev` and `dev` for hyperparameter tuning.

Warning: Please be aware that CONLL 2003 dataset is copyrighted, so make sure to delete these data after the assignment. Do not distribute it!

The data format is described in CONLL website[1]. Note that the original labels apply IOB (Inside-outside-beginning) tagging scheme. Feel free to modify the tagging scheme (BIOS, BIOES etc) and explore how it affects system performance.

[1] `https://www.clips.uantwerpen.be/conll2003/ner/`

Evaluation

We use **entity-level** F_1 score as final evaluation. More specifically, suppose we have a sentence ($x(t)$) with the named entities tagged above each token ($y(t)$) as well as hypothetical predictions produced by a system ($\hat{y}(t)$):

$y(t)$	**I-MISC**	**B-MISC**	**I-MISC**	**O**	**I-PER**	**I-PER**	**O**	**O**	**O**
$\hat{y}(t)$	I-MISC	O	I-MISC	O	I-PER	I-PER	O	O	I-MISC
$x(t)$	Australian	Davis	Cup	captain	John	Newcombe	signalled	his	resignation.

In this example, all named entities are labeled with I-TYPE, except for consecutive entities ("Australian" and "Davis Cup"), in which the first token in the following entity is marked with B-TYPE. Note that this tagging scheme is conceptually different from what we learned in lecture. You might want to change the tagging scheme in some ways.

The entity-level precision, recall and F_1 is calculated as follows:

- Precision is the fraction of predicted entity name spans that line up exactly with spans in the gold standard evaluation data. In our example, "Cup" would be marked incorrectly because it does not cover the whole entity, i.e. "Davis Cup", and we would get a precision score of $\frac{2}{4}$.
- Recall is similarly the number of names in the gold standard that appear at exactly the same location in the predictions. Here, we would get a recall score of $\frac{2}{3}$.
- F_1 is the harmonic mean of the two. and would be $\frac{2}{7}$ in our example.

Deliverable

The deliverable for this assignment consists of three parts. Each part will be evaluated and taken into consideration for the final score:

1. **Code repository (30 points)**. The code to replicate your results.

   We will evaluate your code based on the completeness (all necessary code and data should be included) and the quality (good coding style, enough comments and well-organized structure).

2. **Prediction results (30 points)**. For each of the two models, you are required to hand in a prediction result for `testb` set in a required format. The file should be exactly the same as `testb`, with one additional column indicating predicted tag for each token.

   You will be evaluated based on **entity-level** F_1 score in test set. Basically, there is a positive correlation between F_1 score and points you get.

3. **Report (40 points)**. In this assignment, you are encouraged to explore and analyze different techniques in NER. The report is a place to write down your investigation process and interesting findings. Besides, you are required to include the following information in your report: (1) Open source libraries used. (2) References.

   We will evaluate your report based on the amount of work, the depth of analysis and clarity of writing.

1 Conditional Random Field

Conditional random field (CRF) is a type of discriminative undirected probabilistic graphical model suitable for sequence labeling task [1, 2]. [3] and [4] analyze the design challenges and features in CRF-based NER model. You can find other papers and try out their implementations and features. Any open source libraries can be used for this task.

Deliverable. In the report, please describe the libraries you used and the features you explored. Optionally, you can analyze how each feature affects model performance.

Deliverable. You are required to hand in a prediction result `crf_results.testb` in required format.

2 RNN-based Model

RNNs are a natural model for dealing with sequence labeling task. You are asked to implement one of RNN-based models (vanilla RNN, GRU, LSTM, Bi-LSTM, etc.) with either **Tensorflow** or **PyTorch**. You can also use pretrained word embeddings (GloVE [5]) to initialize your neural network.

You are more than welcome to explore more advanced systems in deep learning. Here are some ideas that may be helpful (not guaranteed):

- Combine RNN layer with CRF for global optimization [6].
- Add additional features (character-level word embedding etc) to each RNN step.
- Change tagging scheme.

Deliverable. In the report, please describe the libraries and versions you use, the model architecture. Optionally, you can describe your modifications and explorations on RNN-based models.

Deliverable. You are required to hand in a prediction result `rnn_results.textb` in required format.

References

[1] Erik F Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147. Association for Computational Linguistics, 2003.

[2] Vijay Krishnan and Christopher D Manning. An effective two-stage model for exploiting non-local dependencies in named entity recognition. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 1121–1128. Association for Computational Linguistics, 2006.

[3] Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155. Association for Computational Linguistics, 2009.

[4] Maksim Tkachenko and Andrey Simanovsky. Named entity recognition: Exploring features. In *KONVENS*, pages 118–127, 2012.

[5] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[6] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016.