CSCI 699: Assignment #1
Named Entity Recognition

Last updated: January 29, 2019

1 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 Blackboard.

You should complete this assignment individually and submit the assignment in Blackboard before **2/18 2:00 PM PST**. Please refer to Sec. 3 for detailed instructions on submission.

1.1 Task

In the task of Named Entity Recognition, we aim to detect the existence of meaningful entities in a sentence. The detected entities should fall into a predefined set of categories. Specifically, in our assignment, we cares about the following 11 categories of entities as documented below:

PERSON	People, including fictional
NORP	Nationalities or religious or political groups
FACILITY	Buildings, airports, highways, bridges, etc.
ORGANIZATION	Companies, agencies, institutions, etc.
GPE	Countries, cities, states
LOCATION	Non-GPE locations, mountain ranges, bodies of water
PRODUCT	Vehicles, weapons, foods, etc. (Not services)
EVENT	Named hurricanes, battles, wars, sports events, etc.
WORK OF ART	Titles of books, songs, etc.
LAW	Named documents made into laws

We formulate this as a sequence labeling problem, with each token being classified into one of the 12 classes – 11 predefined classes and a null-class (O) for tokens that do not belong to 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.

1.2 Dataset

We use Ontonotes Release 5.0 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 Ontonotes Release 5.0 dataset is copyrighted, so make sure to delete these data after the assignment. Do not distribute it!

1.3 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 ($\hat{y}(t)$):

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

In this example, we apply BIO tagging scheme – All entities starts with a tag of I-TYPE, except for the first token of the entity (labeled as B-TYPE).

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.

2 Models

2.1 Conditional Random Field

Conditional random field (CRF) is a type of discriminative undirected probabilistic graphical model suitable for sequence labeling task [2, 3]. [4] and [5] 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.

2.2 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 [6]) 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:

- Combine RNN layer with CRF for global optimization [7].
- Add CNN layers on top of LSTM [8].
- Add additional features (character-level word embedding etc) to each RNN step.
- Change tagging scheme from BIO to BIOS, BIOSE.
- Try contextualized sentence embedding from ELMo [9] or BERT [10].

3 Submission Guideline

3.1 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. Several things to note about code repository:

   (a) Please document the instructions on how to reproduce the results in `README.md`.

   (b) You should include the external resources that are necessary to reproduce the result. If it's too large (e.g. pretrained word vector), you can provide specific instructions on how to access it in `README.md`.

   (c) The model binary file and checkpoints shall be removed from code.

   You can choose to include the compressed codebase in Blackboard or create a github repository and include TA (github account: Hunter-Lin) in it.

   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 result (30 points)**. Two prediction results for `testb` set (named as `crf_output.txt` and `rnn_output.txt`).

   Each line in your result file should be in similar format with `testb` data, with each line being a single output label for corresponding token. For example, the first few lines of your result can be:

   ```
   B-PERSON
   O
   B-ORG
   ...
   ```

   You will be evaluated based on F_1 score in `testb` set. Basically, there is a positive correlation between F_1 score and points you get. (Tip: check the of lines in your prediction output so that it matches the original data.)

3. **Report (40 points)**. A report named as `report.pdf`.

   In this assignment, you are highly 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.

The above deliverable should be included into a single zip file named as `FirstName_LastName_HW1.zip`.

3.2 Late Policy

According to our syllabus, you are given **4 late days** for the assignments and project proposal/survey (no late days for the final project report), to be used in integer amounts and distributed as you see fit. Additional late days will each result in a deduction of 10 points (out of 100) for the corresponding assignment.

References

[1] Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*, 2013.

[2] 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.

[3] 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.

[4] 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.

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

[6] 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.

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

[8] Jason PC Chiu and Eric Nichols. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370, 2016.

[9] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.