Named Entity Recognition for short text

Saad Khan, 3010290

NER

- Named Entity recognition(NER) is a subtask of information extraction that seeks to locate and classify named entities in text into pre-defined categories such as the names of persons, organizations, locations, monetary values etc.

Named Entity recognition – Problem

- Input

  Plain text – T

- Output

  The spans of T that constitute proper names and **classification** of **entity types**.

Example

Input: Vancouver is a coastal seaport city on the mainland of British Columbia. The city's mayor is Gregor Robertson.

Location

Output: ***Vancouver*** is a coastal seaport city on the mainland of ***British Columbia***. The city's mayor is ***Gregor Robertson***.

Location **Person**

Bryan Perozzi Stony Brook University Polyglot-NER: Massive Multilingual Named Entity Recognition

Named Entity Recognition

Existing NER pipelines

- NLTK.
- Stanford.
- OpenNLP.
- SpaCy.
- Stanford Named Entity Recognizer.

What I used

- MIT Information Extraction
  - Based on dlib - a high-performance machine-learning library. MITIE makes use of several state-of-the-art techniques including the use of distributional word embeddings and Structural Support Vector Machines.
- MTA
  - A Multi-task Approach for Named Entity Recognition on Social Media Data. The system uses a Multi-task Neural Network as a feature extractor.

MTA

- Multi-task learning & combination of POS tags and gazetteers representation to feed the network.
- Took advantage of multi-task setting by adding NE segmentation (secondary) along with NE categorization (primary).
- Methodology
  - Feature Representation
    - Character Representation
    - Word Representation
    - Lexical Representation
  - Model Description
    - Character Level BLSTM
    - Word Level BLSTM
    - Lexicon Network
    - Multi-task network

MTA - Continued

- Character Representation
  - Use an orthographic encoder to encapsulate capitalization, punctuation, word shape, and other orthographic features.
  - Handles non-ASCII characters as well.
  - Example : “3rd Workshop !” becomes “ncc Ccccccc p” as numbers are mapped to ‘n’, letters to ‘c’ (or ‘C’ if capitalized), and punctuation marks to ‘p’. NonASCII characters are mapped to ‘x.
- Word Representation
  - Two different representations
    - First : Pre-trained embeddings with 400 million tweets.
    - Second : uses POS tags generated by CMU Twitter POS tagger.

MTA - Continued

- Character Level CNN
  - CNN level architecture to learn word shapes and some orthographic features at the character level representation.
  - Characters embedded into $R^{d \times l}$ dimensional space – d (dimension of the features per character), l (maximum length of characters per word).
  - Apply 2-stacked convolutional layers – *global average pooling*
  - Finally, the result is passed to a fully-connected layer – *Activation Function : Rectifier Linear Unit (ReLU)*.
  - Resulting vector – Input for rest of network.

MTA - Continued

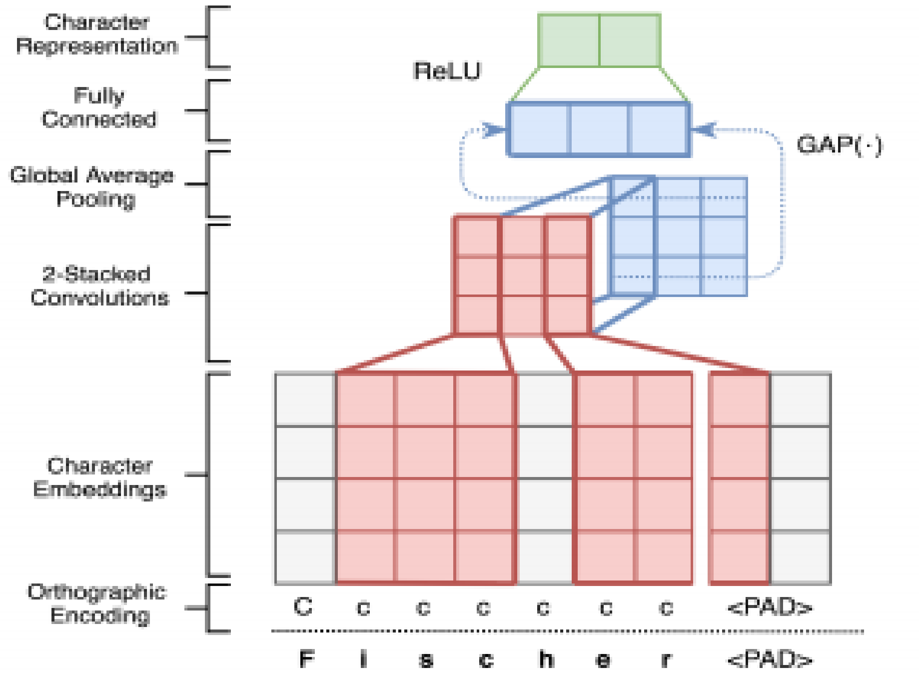

Figure 1: Orthographic character-based representation of a word (green) using a CNN with 2-stacked convolutional layers. The first layer takes the input from embeddings (red) while the second layer (blue) takes the input from the first convolutional layer. Global Average Pooling is applied after the second convolutional layer.

MTA - Continued

- Word Level BLSTM
  - Bi-directional LSTM to learn contextual information of sequence of words.
  - Word embeddings initialized with pre-trained twitter word embeddings – *word2vec.*
  - POS tags embeddings – randomly initialized using uniform distribution.
  - BLSTM layer extracts features from forward & backward (100 neurons per direction) direction & concatenates the results.

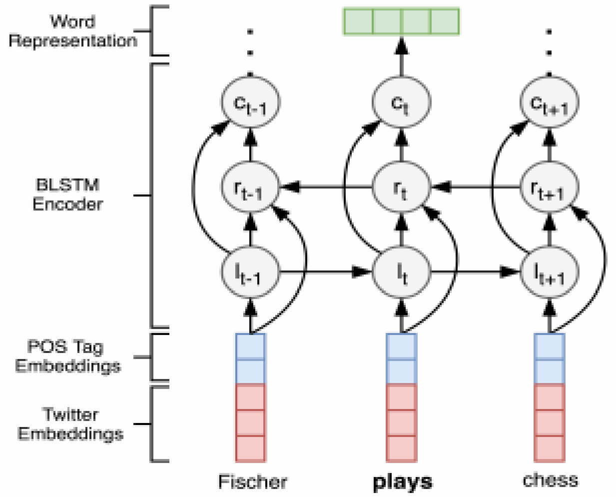

Figure 2: Word representation of POS-tag embeddings (blue) and Twitter word embeddings (red) using a BLSTM neural network.

MTA - Continued

- Lexicon network
  - Lexical representation vectors(input words) and feed them into a fully-connected layer.
  - *32* neurons on this layer and a *ReLU*.
- Multi-task network
  - 13-neuron layer – softmax activation function for NE categorization.
  - Single-neuron layer – sigmoid activation function for NE segmentation.
  - Add loss from both tasks & backpropagate.

W-NUT

- The WNUT workshop focuses on Natural Language Processing applied to noisy user-generated text, such as that found in social media, online reviews, crowdsourced data, web forums, clinical records and language learner essays.

NER

- Used WNUT17 dataset to train models (MTA & MITIE).
- Only three categories
  - Person
  - Organization
  - Location
- Used Precision, Recall and F1-score as measures to evaluate results for all the three categories individually.

Results - MTA

Entity	*Precision*	*Recall*	*F1-score*
Person	70.72	50.12	58.66
Organization	26.55	22.73	26.55
Location	*56.92*	*49.33*	*52.86*
Overall	*57.54*	*32.90*	*41.86*

Results - MITIE

Entity	*Precision*	*Recall*	*F1-score*
Person	58.46	35.43	44.12
Organization	100	1.52	2.99
Location	*66.67*	*20.00*	*30.77*

References

- Davis E. King. Dlib-ml: A Machine Learning Toolkit. Journal of Machine Learning Research 10, pp. 1755-1758, 2009.
- Paramveer Dhillon, Dean Foster and Lyle Ungar, Eigenwords: Spectral Word Embeddings, Journal of Machine Learning Research (JMLR), 16, 2015.
- T. Joachims, T. Finley, Chun-Nam Yu, Cutting-Plane Training of Structural SVMs, Machine Learning, 77(1):27-59, 2009.
- Aguilar, S. Maharjan, A.P. Lopez-Monroy and T. Solorio A Multi-task Approach for Named Entity Recognition in Social Media Data
- http://noisy-text.github.io/2017/
- https://www.slideshare.net/bperz/15-sdmpolyglot-ner