Project 2: Named Entity Recognition

Introduction:

Named Entity Recognition

Named-entity recognition (NER) (also known as entity identification, entity chunking and entity extraction) is a subtask of information extraction that seeks to locate and classify named entity mentions in unstructured text into pre-defined categories such as the person names, organizations, locations, medical codes, time expressions, quantities, monetary values, percentages, etc.

NER systems have been created that use linguistic grammar-based techniques as well as statistical models such as machine learning. Hand-crafted grammar-based systems typically obtain better precision, but at the cost of lower recall and months of work by experienced computational linguists .

Use Cases of NER Models can be: Automatically Summarizing Resumes, Optimizing Search Engine Algorithms, Powering Recommender Systems, Simplifying Customer Support, etc.

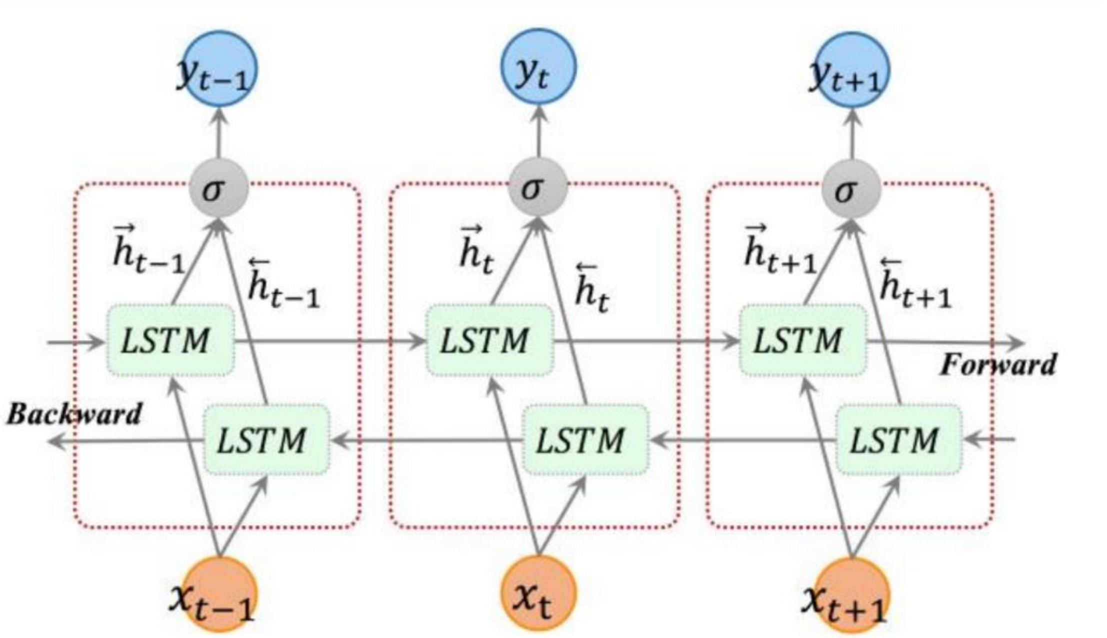
Named Entity Recognition and Classification (NERC) is a process of recognizing information units like names, including person, organization and location names, and numeric expressions including time, date, money and percent expressions from unstructured text. The goal is to develop practical and domain-independent techniques in order to detect named entities with high accuracy automatically.

Deep learning is a set of algorithms and techniques inspired by how the human brain works. Text classification has benefited from the recent resurgence of deep learning architectures due to their potential to reach high accuracy with less need of engineered features. The two main deep learning architectures used in text classification are Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN).

On the one hand, deep learning algorithms require much more training data than traditional machine learning algorithms, i.e. at least millions of tagged examples. On the other hand, traditional machine learning algorithms such as SVM and NB reach a certain threshold where adding more training data doesn’t improve their accuracy. In contrast, deep learning classifiers continue to get better the more data you feed them with.

Bidirectional LSTMs

Long short-term memory (LSTM) cells are the building block of recurrent neural networks (RNNs). While plain LSTM cells in a feedforward neural network process text just like humans do (from left to right), BLSTMs also consider the opposite direction. This allows the model to uncover more patterns as the amount of input information is increased.



Methods:

Dataset: Conll

Since 1999, CoNLL has included a shared task in which training and test data is provided by the organizers which allows participating systems to be evaluated and compared in a systematic way. Descriptions of the participating systems and an evaluation of their performances are presented both at the conference and in the proceedings.

Steps:

(1). Dealing with the raw data – corresponding to step 1

(2). Extracting the following features: the part-ofspeech tag, the lemma, all hypernyms, hyponyms, holonyms and meronyms – corresponding to step 2;

(3). Cleaning all the dataset – corresponding to step 3;

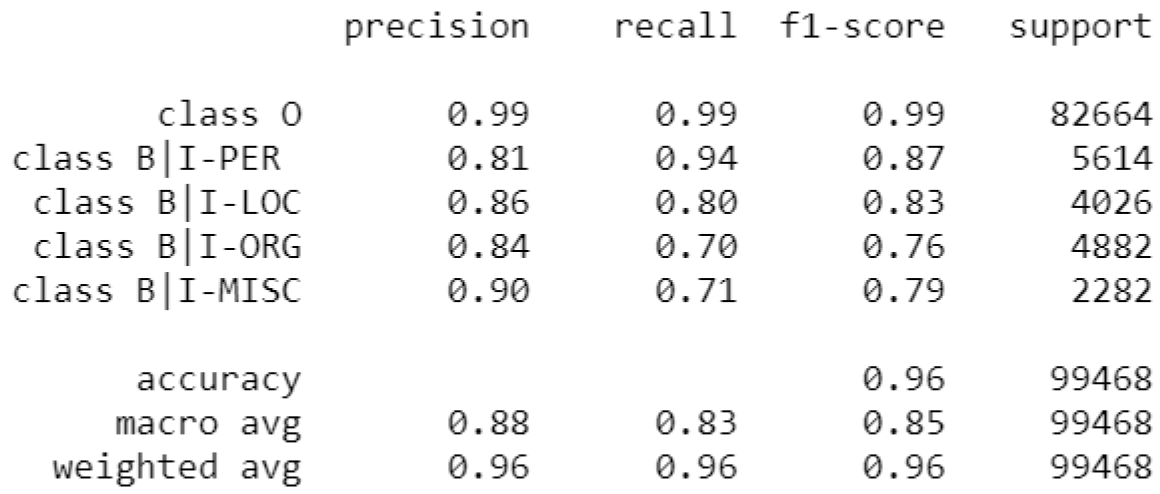
(4). Utilizing Machine Learning Logistic regression Algorithm to obtain the result – corresponding to step 4;

(5). Optimal: NER using deep learning

Result

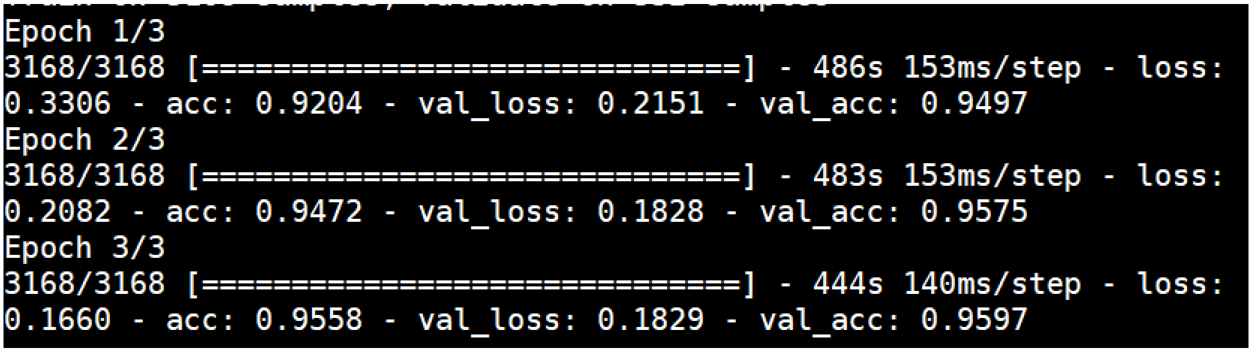
The result can be checked in two different files. One is the from “Step 4 ML”, and the other one is from ner_dl.py file.

Logistic regression

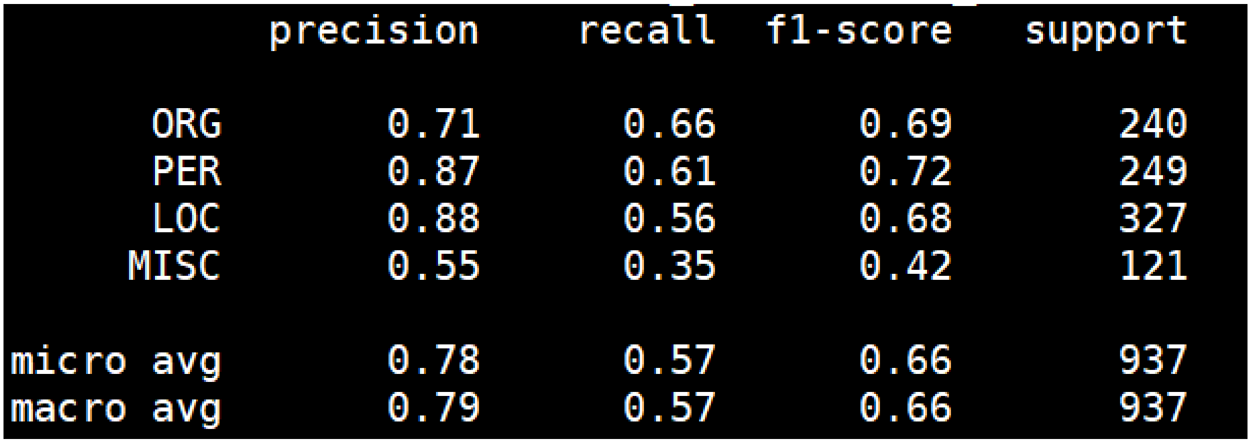


```
                precision    recall  f1-score   support

       class O       0.99      0.99      0.99     82664
 class B|I-PER       0.81      0.94      0.87      5614
  class B|I-LOC      0.86      0.80      0.83      4026
  class B|I-ORG      0.84      0.70      0.76      4882
 class B|I-MISC      0.90      0.71      0.79      2282

      accuracy                           0.96     99468
     macro avg       0.88      0.83      0.85     99468
  weighted avg       0.96      0.96      0.96     99468
```

Deep learning



```
Epoch 1/3
3168/3168 [==============================] - 486s 153ms/step - loss:
0.3306 - acc: 0.9204 - val_loss: 0.2151 - val_acc: 0.9497
Epoch 2/3
3168/3168 [==============================] - 483s 153ms/step - loss:
0.2082 - acc: 0.9472 - val_loss: 0.1828 - val_acc: 0.9575
Epoch 3/3
3168/3168 [==============================] - 444s 140ms/step - loss:
0.1660 - acc: 0.9558 - val_loss: 0.1829 - val_acc: 0.9597
```



```
           precision    recall  f1-score   support

      ORG       0.71      0.66      0.69       240
      PER       0.87      0.61      0.72       249
      LOC       0.88      0.56      0.68       327
     MISC       0.55      0.35      0.42       121

micro avg       0.78      0.57      0.66       937
macro avg       0.79      0.57      0.66       937
```

Reference:

[Chung et al., 2014] Junyoung Chung, Caglar Gulcehre,, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555, 2014.

[Socher et al., 2012] Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. Semantic compositionality through recursive matrix-vector spaces. In Proceedings of EMNLP, pages 1201–1211, 2012.

[Socher et al., 2013] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over r a sentiment treebank. In Proceedings of EMNLP, 2013.

[Collobert et al., 2011] Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and ´ Pavel Kuksa. Natural language processing (almost) from scratch. The Journal of Machine Learning Research, 12:2493–2537, 2011.

[Kalchbrenner et al., 2014] Nal Kalchbrenner, Edward network for modelling sentences. In Proceedings of ACL, 2014.

[Ashish Vaswani, Noam Shazeer et al., 2017] Illia Polosukhin, Attention Is All You Need, arXiv.org > cs > arXiv:1706.03762, 2017

External links:

https://appliedmachinelearning.blog/2019/04/01/training-deep-learning-based-named-entity-recognition-from-scratch-disease-extraction-hackathon/

https://towardsdatascience.com/named-entity-recognition-ner-meeting-industrys-requirement-by-applying-state-of-the-art-deep-698d2b3b4ede

https://medium.com/@rohit.sharma_7010/a-complete-tutorial-for-named-entity-recognition-and-extraction-in-natural-language-processing-71322b6fb090

https://towardsdatascience.com/named-entity-recognition-and-classification-with-scikit-learn-f05372f07ba2

https://dkedar7.github.io/Named%20Entity%20Recognition.html

https://github.com/Anand-krishnakumar/Information-Extraction-

https://github.com/huglittlecat88/wiktextract

https://medium.com/datadriveninvestor/python-data-science-getting-started-tutorial-nltk-2d8842fedfdd

https://stackabuse.com/python-for-nlp-creating-bag-of-words-model-from-scratch/