End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF

Paper Reimplementation

bacc. Bartol Freškura[1] Assoc. prof. dr. sc. Jan Šnajder[2]

[1]Author
Faculty of Electrical Engineering and Computing

[2]Mentor
Faculty of Electrical Engineering and Computing

Masters Seminar, 2017

Introduction

- Sequence labelling problems
- POS - *Part of Speech* tagging and NER - *Named Entity Recognition*
- Standard approach: graphical models like HMMs and CRFs
- New approach: deep neural architectures - RNNs and CNNs

Paper Abstract

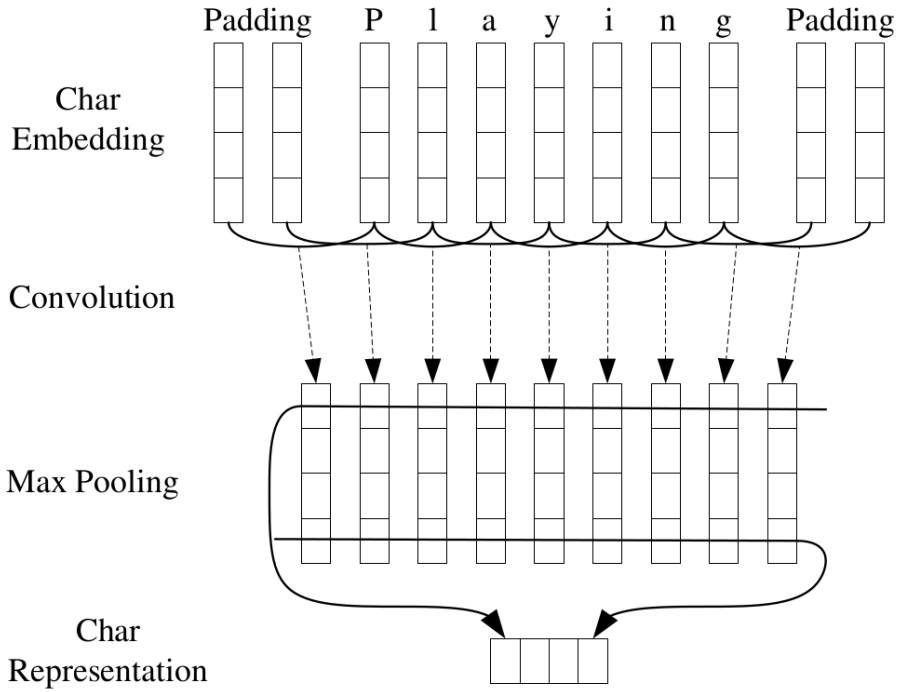
- Authors: Xuezhe Ma and Eduard Hovy from Carnegie Mellon University (2016)
- Sequence labelling via CNN-Bi-LSTM-CRF architecture
- State of the art results on the Penn Treebank WSJ and CoNLL 2003 datasets
- No task-specific resources, feature engineering, or data pre-processing is needed, except for word embeddings

Architecture

- 1-D Convolutional layer for character embeddings
- Pre-trained Glove word embeddings
- Bi-directional Long-short Term Memory (LSTM) for capturing word dependencies
- Conditional Random Fields layer for final sequence labelling

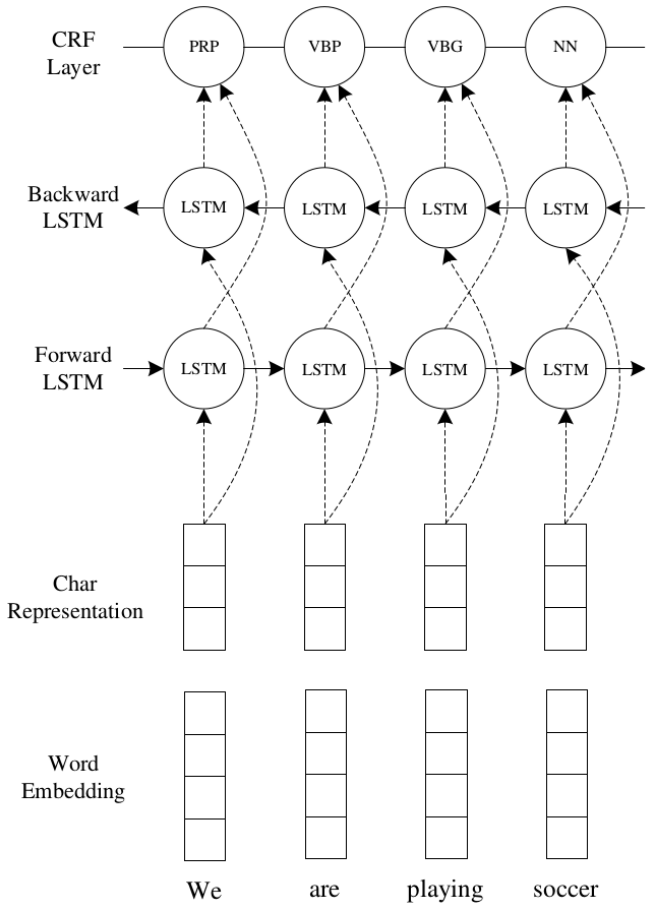
Character embeddings layer

Figure: Character embeddings layer followed by a 1-D convolutional layer. Max pool layer with stride=2 and size=2 is applied after the convolution.



Pipeline

Figure: Structure of the Bi-LSTM and CRF network layers.



Parameters

- 30 filters for the convolutional layer
- 100-dimensional word embeddings
- Single layer Bi-LSTM with the hidden state size of 200
- Dropout layers with the dropout rate of 0.5
- Adam optimizer
- Early stopping for overfit prevention

Datasets

- Sample of WSJ Treebank from the NLTK library
- 45 unique POS tags
- Complete CoNLL 2003 dataset
- Categories: Persons, locations, organizations, and names of misc. entities
- Train set (60%), Development set (20%), Test set (20%) with random shuffling

Results

		POS	NER
Development	Accuracy	96.73	**98.80**
	Precision	93.77	93.73
	Recall	93.91	94.02
	F1	93.52	93.56
Test	Accuracy	96.71	98.23
	Precision	93.73	91.45
	Recall	93.80	91.90
	F1	93.45	91.30

Table: Results with dropout layers

Results

		POS	NER
Development	Accuracy	**97.08**	98.78
	Precision	**94.24**	**93.69**
	Recall	**94.28**	**94.13**
	F1	**93.97**	**93.63**
Test	Accuracy	**96.98**	**98.32**
	Precision	**94.14**	**91.73**
	Recall	**94.19**	**92.22**
	F1	**93.87**	**91.63**

Table: Results without dropout layers

Results

		POS	NER
Development	Accuracy	77.63	51.89
	Precision	84.80	63.25
	Recall	83.89	53.90
	F1	82.42	49.09
Test	Accuracy	78.16	48.32
	Precision	84.68	61.47
	Recall	83.95	50.84
	F1	82.65	46.12

Table: Results without the CRF layer

Conclusion

- Near identical results with the full blown architecture
- Huge differences when the CRF layer is removed
- Approach which can be generalized to any sequence labelling taks in NLP
- Poor character embeddings layer description
- Inverse dropout effect
- Further work: fastText and learning seperate character embeddings

Q&A

Questions?