Natural Language Processing (Almost) From Scratch
Collobert, Weston, Bottou, Karlen, Kavukcuoglu and Kuksa
Overview

1. Introduction
   - Overview
   - Motivation
   - Tasks

2. Approach
   - Window based
   - Sentence based

3. Evaluation
   - Supervised
   - Unsupervised Model
   - Results

4. Summary
Motivation

- Aim is to create a model that is capable of performing well on several NLP tasks.
- Initially ignore most linguistic knowledge.
- Minimise amount of pre-processing – allow the model to create features.
Four tasks chosen were:

- Part of Speech Tagging
- Chunking
- Named Entity Recognition
- Semantic Role Labelling

All of these tasks can be thought of as mapping words to tags.

Benchmark systems chosen:

- POS Tagging: Toutanova et al. (2003)
- NER: Ando and Zhang (2005)
- SRL: Koomen et al. (2005)
Multi-layer Neural Networks

- A multi-layer neural network can be thought of as a series of functions.
- Including non-linear layers, such as the hard tanh function, allows more complex features to be modelled.
- Trained by backpropagation.
Words are converted into vectors of real numbers by the lookup table layer.

The length of these vectors are a parameter of the model.

The values of the vector are set during training.

Features other than words can also be represented in this manner.
Window-based Model

- The window based version of the model attempts to determine the tag of a word based on a $T$ word window around the word of interest.
- $T$ is a parameter of the model.
- After converting the words to a matrix of representation vectors, the columns are concatenated.
- This vector is then passed into a linear layer,
- then a hard tanh layer to introduce non-linearity,
- then a final linear layer, which is task-dependent, and has as many outputs as there are tags for that task.
Sentence-based Model

- SRL needs to look at the whole sentence due to interactions between distant words.
- Slightly more complicated, as the length of sentences can vary but the length of the representation vector needs to remain constant.
- Requires a convolutional neural network.
- Since the word of interest is no longer always the one in the centre, another feature is required to represent this.
Sentence-based Model

- After converting the words to a matrix of representation vectors, instead of concatenating, use a convolution layer.
- This convolution layer creates a number of representation vectors by using a sliding window along the sentence.
- A ‘max’ layer then selects the maximum value from each row.
- This representation is then passed into the same last three layers as for the window-based model.
Initially, both models were trained in a supervised manner.

Dictionary consisted of 100,000 most common words in WSJ.

pre-processing: lowercase, capitalisation encoded as feature, numbers replaced with ‘NUMBER’ keyword.

This gave performance slightly worse than the benchmark systems.
As the goal is to learn a generalizable model, it is useful to look at the word representations.

Ideally, words with similar meanings would have similar representations.
Unsupervised Model

- Trained language models on large amounts of unlabelled data.
- The language models used the same structure as the window based network described above.
- Uses a ranking criterion to score the window.
  - LM1 was trained on 631 million words from Wikipedia, pre-processing was the same, dictionary: 100,000 most common words from WSJ.
  - LM2 was trained on the Wikipedia dataset, as well as 221 million words from Reuters, pre-processing was the same, dictionary: 100,000 most common words from WSJ, and 30,000 most common words from Reuters.
Unsupervised Model

<table>
<thead>
<tr>
<th>Country</th>
<th>Word</th>
<th>Word</th>
<th>Word</th>
<th>Word</th>
<th>Word</th>
<th>Word</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRANCE</td>
<td>JESUS</td>
<td>XBOX</td>
<td>REDDISH</td>
<td>SCRATCHED</td>
<td>MEGABITS</td>
<td></td>
</tr>
<tr>
<td>AUSTRIA</td>
<td>GOD</td>
<td>AMIGA</td>
<td>GREENISH</td>
<td>NAILED</td>
<td>OCTETS</td>
<td></td>
</tr>
<tr>
<td>BELGIUM</td>
<td>SATI</td>
<td>PLAYSTATION</td>
<td>BLUISH</td>
<td>SMASHED</td>
<td>MB/S</td>
<td></td>
</tr>
<tr>
<td>GERMANY</td>
<td>CHRIST</td>
<td>MSX</td>
<td>PINKISH</td>
<td>PUNCHED</td>
<td>BIT/S</td>
<td></td>
</tr>
<tr>
<td>ITALY</td>
<td>SATAN</td>
<td>IPOD</td>
<td>PURPLISH</td>
<td>POPPED</td>
<td>BAUD</td>
<td></td>
</tr>
<tr>
<td>GREECE</td>
<td>KALI</td>
<td>SEGA</td>
<td>BROWNISH</td>
<td>CRIMPED</td>
<td>CARATS</td>
<td></td>
</tr>
<tr>
<td>SWEDEN</td>
<td>INDRA</td>
<td>psNUMBER</td>
<td>GREYISH</td>
<td>SCRAPED</td>
<td>KBIT/S</td>
<td></td>
</tr>
<tr>
<td>NORWAY</td>
<td>VISHNU</td>
<td>HD</td>
<td>GRAYISH</td>
<td>SCREWED</td>
<td>MEGAHERTZ</td>
<td></td>
</tr>
<tr>
<td>EUROPE</td>
<td>ANANDA</td>
<td>DREAMCAST</td>
<td>WHITISH</td>
<td>SECTIONED</td>
<td>MEGAPIXELS</td>
<td></td>
</tr>
<tr>
<td>HUNGARY</td>
<td>PARVATI</td>
<td>GEFORCE</td>
<td>SILVERY</td>
<td>SLASHED</td>
<td>GBIT/S</td>
<td></td>
</tr>
<tr>
<td>SWITZERLAND</td>
<td>GRACE</td>
<td>CAPCOM</td>
<td>YELLOWISH</td>
<td>RIPPED</td>
<td>AMPERES</td>
<td></td>
</tr>
</tbody>
</table>

- Initialising the supervised model with the word representations learnt by the language model improves the final word representations.
### Results

<table>
<thead>
<tr>
<th>Approach</th>
<th>POS (PWA)</th>
<th>CHUNK (F1)</th>
<th>NER (F1)</th>
<th>SRL (F1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Benchmark Systems</td>
<td>97.24</td>
<td>94.29</td>
<td>89.31</td>
<td>77.92</td>
</tr>
<tr>
<td>NN+WLL</td>
<td>96.31</td>
<td>89.13</td>
<td>79.53</td>
<td>55.40</td>
</tr>
<tr>
<td>NN+SLL</td>
<td>96.37</td>
<td>90.33</td>
<td>81.47</td>
<td>70.99</td>
</tr>
<tr>
<td>NN+WLL+LM1</td>
<td>97.05</td>
<td>91.91</td>
<td>85.68</td>
<td>58.18</td>
</tr>
<tr>
<td>NN+SLL+LM1</td>
<td>97.10</td>
<td>93.65</td>
<td>87.58</td>
<td>73.84</td>
</tr>
<tr>
<td>NN+WLL+LM2</td>
<td>97.14</td>
<td>92.04</td>
<td>86.96</td>
<td>58.34</td>
</tr>
<tr>
<td>NN+SLL+LM2</td>
<td>97.20</td>
<td>93.63</td>
<td>88.67</td>
<td>74.15</td>
</tr>
</tbody>
</table>
## Results

<table>
<thead>
<tr>
<th>Approach</th>
<th>POS (PWA)</th>
<th>CHUNK (F1)</th>
<th>NER (F1)</th>
<th>SRL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Benchmark Systems</td>
<td>97.24</td>
<td>94.29</td>
<td>89.31</td>
<td>77.92</td>
</tr>
<tr>
<td>NN+SLL+LM2</td>
<td>97.20</td>
<td>93.63</td>
<td>88.67</td>
<td>74.15</td>
</tr>
<tr>
<td>NN+SLL+LM2+Suffix2</td>
<td>97.29</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>NN+SLL+LM2+Gazetteer</td>
<td>–</td>
<td>–</td>
<td>89.59</td>
<td>–</td>
</tr>
<tr>
<td>NN+SLL+LM2+POS</td>
<td>–</td>
<td>94.32</td>
<td>88.67</td>
<td>–</td>
</tr>
<tr>
<td>NN+SLL+LM2+CHUNK</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>74.72</td>
</tr>
</tbody>
</table>
Aim:
- Create a general purpose model for NLP tasks.
- Avoid using linguistic knowledge to create features.

The model performs close to state-of-the-art for all four tasks.

Using linguistic knowledge to build extra features for the model further improves performance.

The model is quicker and uses less memory than the state-of-the-art systems, as it doesn’t need to calculate and store many complex features.