A Unified Architecture for Natural Language Processing

(Deep Neural Networks with Multitask Learning)

Ronan Collobert & Jason Weston
ronan@collobert.com    jaseweston@gmail.com

NEC Laboratories America
The Big Picture

Blah Blah Blah

Embedding

Local features

Global features

Tags

Deep architecture

Unification of NLP tasks
NLP Tasks

Part-Of-Speech Tagging (POS): syntactic roles (noun, adverb...)

Chunking: syntactic constituents (noun phrase, verb phrase...)

Name Entity Recognition (NER): person/company/location...

Semantic Role Labeling (SRL): semantic role

[John]_{ARG0} [ate]_{REL} [the apple]_{ARG1} [in the garden]_{ARGM-LOC}

Labeled data: Wall Street Journal (∼1M of words)
## The Shallow System Way

### (1/2)

Choose some good **hand designed features**

<table>
<thead>
<tr>
<th><strong>Predicate and POS tag of predicate</strong></th>
<th><strong>Voice</strong>: active or passive (hand-built rules)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Phrase type</strong>: adverbial phrase, prepositional phrase, ...</td>
<td>** Governing category**: Parent node's phrase type(s)</td>
</tr>
<tr>
<td><strong>Head word and POS tag of the head word</strong></td>
<td><strong>Position</strong>: left or right of verb</td>
</tr>
<tr>
<td><strong>Path</strong>: traversal from predicate to constituent</td>
<td><strong>Predicted named entity class</strong></td>
</tr>
<tr>
<td><strong>Word-sense disambiguation of the verb</strong></td>
<td><strong>Verb clustering</strong></td>
</tr>
<tr>
<td><strong>Length of the target constituent (number of words)</strong></td>
<td><strong>NEG</strong> feature: whether the verb chunk has a &quot;not&quot;</td>
</tr>
<tr>
<td><strong>Partial Path</strong>: lowest common ancestor in path</td>
<td><strong>Head word replacement</strong> in prepositional phrases</td>
</tr>
<tr>
<td><strong>First and last words and POS in constituents</strong></td>
<td><strong>Ordinal position</strong> from predicate + constituent type</td>
</tr>
<tr>
<td><strong>Constituent tree distance</strong></td>
<td><strong>Temporal cue words</strong> (hand-built rules)</td>
</tr>
<tr>
<td><strong>Dynamic class context</strong>: previous node labels</td>
<td><strong>Constituent relative features</strong>: phrase type</td>
</tr>
<tr>
<td><strong>Constituent relative features</strong>: head word</td>
<td><strong>Constituent relative features</strong>: head word POS</td>
</tr>
<tr>
<td><strong>Constituent relative features</strong>: siblings</td>
<td><strong>Number of pirates existing in the world...</strong></td>
</tr>
</tbody>
</table>

Feed them to a **shallow classifier** like SVM
Cascade features: e.g. extract POS, construct a parse tree

Extract hand-made features from the parse tree

Feed these features to a shallow classifier like SVM
The Deep Learning Way (1/2)
The Deep Learning Way

(2/2)

Input Sentence
the cat sat on the mat
word of interest
verb of interest
s(1) s(2) s(3) s(4) s(5) s(6)
-1 0 1 2 3 4

Lookup Tables

LT_w
LT_pw
LT_pv

Convolution Layer

Max Over Time

HardTanh
Linear
Softmax

Tags

Local features

Global features

Embedding

Blur Blah Blah
Convolutions

Extract local features – share weights through time/space

Used with success in image (Le Cun, 1989) and speech (Bottou & Haffner, 1989)

Lookup-table is a special case: convolution with kernel size of 1 and input $i^{th}$ word

$(0, 0, \ldots, 1, 0, \ldots, 0)$ 1 at position $i$

Bengio et al (2001)
yesterday, after microsoft bought google, the dollar went down under half a euro and the fish market exploded.
Removing The Time Dimension (2/2)

yesterday, after Microsoft bought Google, the dollar went down under half a euro and the fish market exploded.
Multi-Task Learning

Good overview in Caruana (1997)
Improving Word Embedding

- Rare words are not trained properly
- Sentences with similar words should be tagged in the same way:
  - The cat sat on the mat
  - The feline sat on the mat

Wordnet
  - pull together linked words
  - push apart other pair of words
Language Model: Think Massive

Language Model: “is a sentence actually english or not?”
Implicitly captures: * syntax * semantics

Bengio & Ducharme (2001) Probability of next word given previous words. Overcomplicated – we do not need probabilities here

English sentence windows: Wikipedia (∼ 631M words)
Non-english sentence windows: middle word randomly replaced

Multi-class margin cost:

\[
\sum_{s \in S} \sum_{w \in D} \max(0, 1 - f(s, w^*_s) + f(s, w))
\]

\(S\): sentence windows \(D\): dictionary
\(w^*_s\): true middle word in \(s\)
\(f(s, w)\): network score for sentence \(s\) and middle word \(w\)
<table>
<thead>
<tr>
<th>Country</th>
<th>Word</th>
<th>Frequency</th>
<th>Word</th>
<th>Frequency</th>
<th>Word</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>France</td>
<td>jesus</td>
<td>454</td>
<td>xbox</td>
<td>6909</td>
<td>reddish</td>
<td>11724</td>
</tr>
<tr>
<td>Spain</td>
<td>christ</td>
<td>1973</td>
<td>playstation</td>
<td>psNUMBER</td>
<td>yellowish</td>
<td>29869</td>
</tr>
<tr>
<td>Italy</td>
<td>god</td>
<td></td>
<td>dreamcast</td>
<td></td>
<td>greenish</td>
<td></td>
</tr>
<tr>
<td>Russia</td>
<td>resurrection</td>
<td></td>
<td></td>
<td>snes</td>
<td>brownish</td>
<td></td>
</tr>
<tr>
<td>Poland</td>
<td>prayer</td>
<td></td>
<td></td>
<td>wii</td>
<td>bluish</td>
<td></td>
</tr>
<tr>
<td>England</td>
<td>yahweh</td>
<td></td>
<td></td>
<td>nes</td>
<td>hurled</td>
<td></td>
</tr>
<tr>
<td>Denmark</td>
<td>josephus</td>
<td></td>
<td></td>
<td>nintendo</td>
<td>grabbed</td>
<td></td>
</tr>
<tr>
<td>Germany</td>
<td>moses</td>
<td></td>
<td></td>
<td>gamecube</td>
<td>creamy</td>
<td></td>
</tr>
<tr>
<td>Portugal</td>
<td>sin</td>
<td></td>
<td></td>
<td>psp</td>
<td>whitish</td>
<td></td>
</tr>
<tr>
<td>Sweden</td>
<td>heaven</td>
<td></td>
<td></td>
<td>amiga</td>
<td>blackish</td>
<td></td>
</tr>
<tr>
<td>Austria</td>
<td>salvation</td>
<td></td>
<td></td>
<td></td>
<td>silvery</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>blasted</td>
<td></td>
</tr>
</tbody>
</table>

Dictionary size: 30,000 words. Even rare words are well embedded.
MTL: Semantic Role Labeling

We get: 14.30%. State-of-the-art: 16.54% – Pradhan et al. (2004)

250× faster than state-of-the-art. ∼0.01s to label a WSJ sentence.
MTL: Unified Network for NLP

Improved results with Multi-Task Learning (MTL)

<table>
<thead>
<tr>
<th>Task</th>
<th>Alone</th>
<th>MTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRL</td>
<td>18.40%</td>
<td>14.30%</td>
</tr>
<tr>
<td>POS</td>
<td>2.95%</td>
<td>2.91%</td>
</tr>
<tr>
<td>Chunking – error rate</td>
<td>5.4%</td>
<td>4.9%</td>
</tr>
<tr>
<td>Chunking – F1-score</td>
<td>91.5%</td>
<td>93.6%</td>
</tr>
</tbody>
</table>

**POS:** state-of-the-art ~ 3%

**Chunking:** Best system had 93.48% F1-score at CoNLL-2000 challenge [http://www.cnts.ua.ac.be/conll2000/chunking](http://www.cnts.ua.ac.be/conll2000/chunking). State-of-the-art is 94.1%. We get **94.9%** by using POS features.
Summary

We developed a deep neural network architecture for NLP

Advantages
- General to any NLP tagging task
- State-of-the-art performance
- No hand designed features
- Joint training
- Can exploit massive unlabeled data
- Extremely fast: 0.02s for all tags of a sentence

Inconvenients
- Neural networks are a powerful tool: hard to handle

Early Impacts
- Easy to apply to other tasks or languages: extending to Japanese
- Fast: developed a semantic search system