Deep RNNs Encode Soft Hierarchical Syntax

Amir Veyseh
Winter 2020
Introduction

- Syntactic features are important features for feature based models
  - Part-of-speech
  - Dependency tree
  - Constituency tree
- Syntactic information are important for deep models too:
  - Considered as input feature
  - Considered as supervision signal
- However, deep architectures are able to recover some important syntactic features
  - Similar to vision, different layers in deep architectures recover syntactic features in different levels
Syntactic Features

- Part of speech:
  - Classes of words sharing syntactic properties:
    - Noun,
    - Verb,
    - Adjective

- Hierarchical structure of language:
  - Constituency tree:
    - Hierarchy of the phrases in the sentence
  - Dependency tree:
    - Syntactic dependencies between words of the sentence
Retrieving Syntactic Features

- Train Recurrent Neural Net on a specific task to generate word representations
- Employ word representations generated by RNN as input feature into a feed forward classifier
- Predict different syntactic features for each word using the feed forward classifier:
  - Part of speech tag
  - Parent constituency tag
  - Grandparent constituency tag
  - Great-grandparent constituency tag
- Predict dependency arc between two words using a feed forward classifier and word representations generated from RNN
Tasks for RNN

- Different RNN architectures are used to be trained on different tasks:
  - Dependency parsing: Generate dependency parse tree for the given sentence
  - Semantic Role Labeling: Answering who did what to whom. Find predicates (verbs) and their arguments (roles of words)
  - Machine Translation: Encode the source language and decode it in another language
  - Language Modeling: Predicate the next word in a sentence

- After training RNN on above tasks, the representations are fine tuned to extract syntactic features for words in different levels using feed forward classifier:
  - Use development set of syntactic feature corpus to fine tune the parameters
Hypothesis

- Representations generated by RNN should have enough information about syntactic tree if they are used in a simple classifier (e.g. feed forward classifier) trained on a small dataset for extracting syntactic features.
- Different levels of RNN would capture different syntactic features:
  - More word-level features are encoded in first layers.
  - More abstract features are encoded in deeper layers:
    - Higher order constituency tags which covers more parts of the sentence.
- Deep architectures are able to capture more useful features out of word embeddings.
Methodology

- Train a multi-layer RNN on a task and use the representations of the words from different layers to predict some constituency based features for the word:

```
S
  NP
    JJ NN NNS
  ADVP
    RB VBD
  VP
    PP
      IN NP
        NNP
Other stock indexes also fell on Monday
```
Experimental Setup

- Predict the syntactic feature using a feed forward neural net with 300 hidden dimensions and ReLU activation function:

  \[ y_i^l = \text{SoftMax}(W_2\text{ReLU}(W_1x_i^l)) \]

- Train the classifier on the development set of CoNLL-2012 and evaluate it on test set of CoNLL-2012

- Each layer is trained separately:
  - There are L classifiers.
Baselines

- **Majority:**
  - Assign the label with the highest frequency to the word
    - Cat → Noun
    - Walks → Verb

- **Use GloVe embedding to predict the syntactic label for each word:**
  - It is weaker than the majority baseline

- **Contextual:**
  - Concatenate the word embedding with the average of its context embedding
  - It is weaker than the majority baseline
Analyzed Model

- Dependency parser:
  - Generate dependency parse tree for a given sentence
- Model:
  - 4-layer bidirectional LSTM
- Dataset:
  - Universal Dependencies English Web Treebank
- Performance on CoNLL 2017:
  - UAS 91.5
  - LAS 82.18
- POS is given as input feature to the model
Analyzed Model

- Semantic Role Labeling:
  - Given a sentence find the predicates and the arguments (verbs and the role of the other words toward that verb)
  - Mary sole the book to John
- Model:
  - 8-layer alternating forward and backward LSTM
- Dataset:
  - CoNLL-2012 training set
- Layer Representation:
  - Concatenation of the forward and backward to represent the word representation in that layer
Analyzed Model

- Machine Translation:
  - Encode the sentence in source language using neural nets to generate the sentence in the destination language
- Model:
  - 4-layer bidirectional LSTM
- Dataset:
  - WMT-14 English-German
- Performance:
  - BLUE 21.37
Analyzed Model

- **Language Modeling:**
  - Predict the next word in a sentence

- **Model:**
  - Two separate models for forward and backward language model
    - 4-layer LSTM for each model
  - Concatenation of the forward and backward language model represents the word

- **Dataset:**
  - CoNLL-2012 training set

- **Performance:**
  - Perplexities 50.6 (forward)
  - Perplexities 51.24 (backward)
Constituency Label Prediction

Dependecy Parsing

Semantic Role Labeling

Accuracy

Layer 0  Layer 1  Layer 2  Layer 3  Layer 4

Layer 0  Layer 1  Layer 2  Layer 3  Layer 4
Constituency Label Prediction

Machine Translation

Language Modeling
Constituency Label Prediction

- **Findings:**
  - All models outperform the baseline and its input embedding in predicting the constituency label.
  - RNN will extract syntax-based features from embeddings.
  - Different configurations of RNN layers are able to extract syntax-based features.

- **Exceptions:**
  - Dependency parsing cannot outperform its input embedding in POS tag prediction:
    - The POS tag is given as input feature and the architecture cannot outperform this.
Analysis on Number of Layers

- Finding:
  - Deeper layers capture higher-order syntax feature
- 11 out of 16 cases the performance improves up to a certain layer then declines
  - Deeper layers encode non-syntax-based features
- Higher-level syntactic features peaks at deeper layers
  - In SRL: POS $\rightarrow$ Parent $\rightarrow$ Grandparent, Great-grandparent
- Deeper layers consume the features extracted from previous layers to encode more abstract features
Language Model and Syntactic Features

- The finding of extracting higher-order syntactic tree in deeper layers is consistent with ELMo.
- ELMo uses the entire layers of encoder in a language model to represent each word.
- Final representation is a task-specific weighted sum over all layers.
- More syntax-based task benefit from first layers and more abstract tasks benefit from deeper layers.
- Other contextualized word embedding like BERT show the same capability to extract syntactic features out of word embedding.
Language Model and Syntactic Features
Dependency Arc Prediction

- Train classifier to predict two words are connected to each other in the dependency tree or not using the representations from different layers of RNN
- Input to the classifier:

\[ [w_c; w_p; w_c \circ w_p] \]

- Dataset:
  - Train: development set of Universal Dependencies dataset
  - Test: Test set of Universal Dependencies dataset
  - For each word in the dataset two positive and negative examples are generated:
    - Positive: One of the neighbors of the word
    - Negative: A random word in the sentence
Dependency Arc Prediction

- Dependency parser has the highest accuracy in the last layer.
- Other models improve the performance in comparison to the first layer up to 12 to 20 points:
  - RNN learns dependencies between words.
- Deeper layers have higher performance:
  - Abstract syntactic features are extracted in deeper layers.

<table>
<thead>
<tr>
<th>Source Model</th>
<th>GloVe</th>
<th>L0</th>
<th>L1</th>
<th>L2</th>
<th>L3</th>
<th>L4</th>
</tr>
</thead>
<tbody>
<tr>
<td>DP</td>
<td>0.50</td>
<td>0.68</td>
<td>0.77</td>
<td>0.81</td>
<td>0.88</td>
<td><strong>0.95</strong></td>
</tr>
<tr>
<td>SRL</td>
<td>0.50</td>
<td>0.58</td>
<td>0.69</td>
<td>0.76</td>
<td><strong>0.79</strong></td>
<td>0.74</td>
</tr>
<tr>
<td>MT</td>
<td>0.50</td>
<td>0.61</td>
<td><strong>0.73</strong></td>
<td>0.63</td>
<td>0.63</td>
<td>0.63</td>
</tr>
<tr>
<td>LM</td>
<td>0.50</td>
<td>0.62</td>
<td>0.74</td>
<td>0.78</td>
<td><strong>0.80</strong></td>
<td>0.73</td>
</tr>
</tbody>
</table>
Conclusion

- RNNs are able to extract syntactic features without direct syntactic features as input.
- Performance of deeper layers is better for higher order syntax based task:
  - Deeper layers extract more abstract features from the representations learned from the first layers.
- Future work:
  - Compare the syntax induced by these models with the gold syntax features.
  - Analyze the syntactic features that RNNs are not able to capture.