CS5740: Natural Language Processing

Contextualized Word Representations

Instructor: Yoav Artzi

Slides adapted from Graham Neubig
Overview

• Contextualized word representations
• Models
  – context2vec
  – ELMo
  – BERT
Contextualized Word Representations

Hey ELMo, what's the embedding of the word “stick”?

There are multiple possible embeddings! Use it in a sentence.

Oh, okay. Here: “Let's stick to improvisation in this skit”

Oh in that case, the embedding is: -0.02, -0.16, 0.12, -0.1 ... etc

http://jalammar.github.io/illustrated-bert/
Contextualized Word Representations

• word2vec, GloVe
  – Learn a vector for every word type
  – Always the same vector

• Instead: learn a different vector for word type in every usage
  – But: how do we define the space of uses? Isn’t it too large?
  – Solution: use sentence encoders to create a custom vector for every instance of a word
Central Word Prediction Objective (context2vec)

- Model: bi-directional LSTM
- Objective: predict the word given context
- Data: 2B word ukWaC (English data) corpus
- Downstream: use vectors for sentence completion, word-sense disambiguation, etc.

[Melamud et al. 2016]
Bi-directional Language Modeling Objective (ELMo)

• Model: multi-layer bi-directional LSTM
• Objective: predict the next word left→right and next word right→left independently
• Data: 1B word benchmark LM dataset
• Downstream: fine-tune the weights of the linear combination of layers per task

[Peterś et al. 2018; figure from Devlin et al. 2018]
Masked Word Prediction (BERT)

- Model: multi-layer self-attention (Transformer), input sentence (or pair w/[CLS] token) and subword representation
- Objective: masked word prediction + next-sentence prediction
- Data: BookCorpus + English Wikipedia
- Downstream: fine-tune weights per task

[Devlin et al. 2018]
Masked Word Prediction

• Predict a masked word
  – 80%: substitute input word with “[MASK]”
  – 10%: substitute input word with random word
  – 10%: no change

• Like predicting the next word, but adapted for multi-layer self attention

[Devlin et al. 2018]
Consecutive Sentence Prediction

• Classify two sentences as consecutive or not
  – 50% of training data is consecutive

\[
\text{Input} = \text{[CLS]} \text{ the man went to [MASK] store [SEP]}
\]

\[
\text{he bought a gallon [MASK] milk [SEP]}
\]

\[
\text{Label} = \text{IsNext}
\]

\[
\text{Input} = \text{[CLS]} \text{ the man [MASK] to the store [SEP]}
\]

\[
\text{penguin [MASK] are flight ##less birds [SEP]}
\]

\[
\text{Label} = \text{NotNext}
\]

[Devlin et al. 2018]
Using BERT for Downstream Tasks

- Use the pre-trained model as the first “layer” of your final model
- Train with fine-tuning using your supervised data

(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

(b) Single Sentence Classification Tasks: SST-2, CoLA

(c) Question Answering Tasks: SQuAD v1.1

(d) Single Sentence Tagging Tasks: CoNLL-2003 NER
Using BERT for Feature Extraction

Generate Contextualized Embeddings

The output of each encoder layer along each token’s path can be used as a feature representing that token.

But which one should we use?

http://jalammar.github.io/illustrated-bert/
Using BERT for Feature Extraction

What is the best contextualized embedding for “Help” in that context?
For named-entity recognition task CoNLL-2003 NER

<table>
<thead>
<tr>
<th>Method</th>
<th>Embedding</th>
<th>Dev F1 Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Layer</td>
<td></td>
<td>91.0</td>
</tr>
<tr>
<td>Last Hidden Layer</td>
<td></td>
<td>94.9</td>
</tr>
<tr>
<td>Sum All 12 Layers</td>
<td></td>
<td>95.5</td>
</tr>
<tr>
<td>Second-to-Last Hidden Layer</td>
<td></td>
<td>95.6</td>
</tr>
<tr>
<td>Sum Last Four Hidden</td>
<td></td>
<td>95.9</td>
</tr>
<tr>
<td>Concat Last Four Hidden</td>
<td></td>
<td>96.1</td>
</tr>
</tbody>
</table>

http://jalammar.github.io/illustrated-bert/