John Wieting

Shuaichen Chang and Reid Fu

Short Bio

- Started out in Chemistry
- Outstanding TA award
- PhD student of Toyota Technological Institute at UChicago
- Adviser is Kevin Gimpel

Research Interests

- States that main interests are “machine learning, learning theory, optimization, natural language processing and computer vision”
- Main work has been in using sentence embeddings for paraphrase detection
- Old projects include generalization of strongly convex online learning algorithms, lexical entailment, and entity relation identification

Publications

- Learning Paraphrastic Sentence Embeddings from Back-Translated Bitext
  - Co-authors: Jonathan Mallinson (UEdinburgh), Kevin Gimpel
  - EMNLP 2017
- Revisiting Recurrent Networks for Paraphrastic Sentence Embeddings
  - Co-authors: Kevin Gimpel
  - ACL 2017
- Charagram: Embedding Words and Sentences via Character n-grams
  - Co-authors: Mohit Bansal, Kevin Gimpel, Karen Livescu
  - EMNLP 2016
- Towards Universal Paraphrastic Sentence Embeddings
  - Co-authors: Mohit Bansal, Kevin Gimpel, Karen Livescu
  - ICLR 2016

Towards universal paraphrastic sentence embeddings

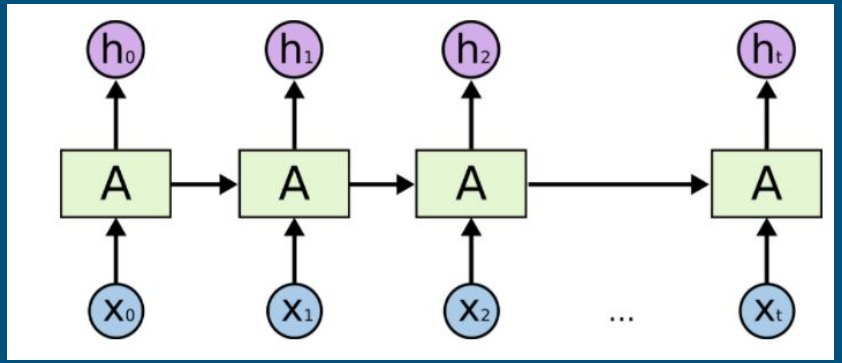
Sentence Embedding:

1. Average of word embedding (AVG)

   Use the average of embedding of every word in a sentence.

   Bag-of-words models (ignore word order)

2. Last hidden state of LSTM (LSTM)



Training with margin-based loss

Results

Textual Similarity:

AVG > LSTM (trained on PPDB phrase pairs, tested on SemEval Textual Similarity)

AVG < LSTM (trained and tested on PPDB phrase pairs)

AVG (trained on PPDB) > AVG (pre-trained embedding)

Sentiment classfication:

AVG < LSTM (trained and tested on Stanford Sentiment Treebank)

Why does AVG outperform LSTM ?

Length?

Not exactly. LSTM performs worse as the length of sentences increases but is always beaten by AVG.

Overfitting?

No. Performances of LSTM and AVG on training dataset are similar.

Only consider the last hidden state?

Next paper!

How to make LSTM better?

Combine the benefit of sequential order and average of different time step

New paper: Revisiting Recurrent Networks for Paraphrastic Sentence Embeddings

Revisiting Recurrent Networks for Paraphrastic Sentence Embeddings

1. LSTMavg: Use the average of all hidden state of LSTM

2. New model: GATED RECURRENT AVERAGING NETWORK (GRAN)

   Use LSTM to generate the weight in Weighted Average of word embedding

$$a_t = x_t \odot \sigma(W_x x_t + W_h h_t + b)$$

3. AVG > LSTMavg/GRAN > LSTM

4. Add more regularizition and train on a sentence-level simplification dataset

   LSTMavg/GRAN > AVG in most datasets.

Questions