CS6370: Natural Language Processing

Team 5: Project Proposal

29th September 2016

Polysemy Resolution in Word Embeddings

Word Embeddings

Word embedding is the collective name for a set of language modeling and feature learning techniques in natural language processing (NLP) where words or phrases from the vocabulary are mapped to vectors of real numbers in a low-dimensional space relative to the vocabulary size.
Methods to generate this mapping include neural networks, dimensionality reduction on the word co-occurrence matrix, probabilistic models, and explicit representation in terms of the context in which words appear.
Word and phrase embeddings, when used as the underlying input representation, have been shown to boost the performance in NLP tasks such as syntactic parsing and sentiment analysis.

The Problem

Traditional word embeddings like Mikolov et. al's Word2Vec result in vectors for each string that is traditionally considered to be a word in the vocabulary.
One glaring drawback of this approach is that while most words are polysemous in languages like English, each word is represented by a single vector.
After applying the Word2Vec approach, we obtain vectors for all the words in the corpus. This might include highly polysemous words like ”light” (electro-magnetic waves or to set on fire or the opposite of heavy) ”bank” (a riverside or a financial institution) or ”top” (a spinning toy, or the upside, or an upper garment).
It is clear that the quality of the embeddings will be detrimentally affected because of the multiple meanings we can ascribe to each word, depending on the context. We cannot reasonably hope that a single vector can effectively capture the ”correct” meaning in all the contexts.

Proposed Solution

Our proposed solution is to assign vectors to senses (i.e. meanings) instead of words. One way to enumerate the various senses is to use the synsets defined in WordNet. We list the abstract steps involved in our approach below:

1. Given a corpus, we process it to tag the words with the appropriate synsets using some form of Word Sense Disambiguation.
2. We use traditional word embedding approached (say Word2Vec) to obtain vectors for these synsets.
3. In order to use these vectors for various tasks involving text, we again need to use Word Sense Disambiguation to tag the test phrases with the appropriate synsets and then use the corresponding vectors.

Evaluation Measures

Since there is no obvious way of judging the quality of word embeddings obtained, we propose that we perform a set of common NLP tasks using traditional word embeddings, and using the ”sense embeddings”. A comparison between the metrics used to judge these tasks can reflect how much advantage sense embeddings deliver (if any).
The baseline to compare would be the results based on word embeddings trained using Word2vec on the same corpus and with the same settings.
We list some sample tasks below:

- Sentiment Analysis
- Semantic Text Similarity
- Relations between senses (E.g. Does V(King)-V(Queen) correspond to V(Man)-V(Woman))

Datasets

- **Training:** Any large corpora like Brown Corpus, Wikipedia dumps, news articles collections can be used to train the word embeddings.
- **Testing:** For testing we can use any popular datasets for evaluating one of the given tasks. For e.g. Stanford’s Contextual Word Similarities (SCWS) Dataset or WordSim-353 dataset.

References

1. Huang, Eric H., et al. ”Improving word representations via global context and multiple word prototypes.” Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1. Association for Computational Linguistics, 2012.

2. Liu, Yang, et al. ”Topical Word Embeddings.” AAAI. 2015.

3. Tian, Fei, et al. ”A Probabilistic Model for Learning Multi-Prototype Word Embeddings.” COLING. 2014.

4. Mikolov, Tomas, et al. ”Efficient estimation of word representations in vector space.” arXiv preprint arXiv:1301.3781 (2013).

5. Liu, Pengfei, Xipeng Qiu, and Xuanjing Huang. ”Learning context-sensitive word embeddings with neural tensor skip-gram model.” Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence. 2015.