Word Embeddings with Multiple Word Prototypes

Anurendra Kumar[1] and Nishant Rai[2]

[1] Department of Electrical Engineering
[2] Department of Computer Science

October 4, 2015

1 Introduction

The ability to accurately represent word vectors to capture syntactic and semantic similarity is central to Natural language processing. Thus, there is rising interest in vector space word embeddings and their use especially given recent methods for their fast estimation at very large scale. However almost all recent works assume a single representation for each word type, completely ignoring polysemy which eventually leads to errors.

Problems caused by ignoring polysemy can be seen in the following sentences,

- "I can hear bass sounds"
- "They like grilled bass"

Here, the word bass represents two different meanings: tones and a type of fish, respectively. Making this distinction is not possible unless we account for polysemy in our models.

2 Motivation

Inspiration to compute multiple embeddings comes from the following observation, a word with two distinct senses A and B will have neighbors belonging to two distinct clusters. This can be termed as violation of triangle inequality. Many approaches exploit the property to compute better representations of word types.

Notice that word embeddings play a crucial role in construction of multiple embeddings, thus we should also aim at improving the word embeddings. In nearly all approaches, word vectors are built from local contexts. But in work done by Huang et al. [?], it is shown that including global context helps improve the quality of the embeddings.

In our project, we aim at improving the initial word embeddings using both local and global context. Also, we try to compute multiple senses of a given word type using algorithms discussed in [?].

3 Related Work

Mooney et al. [?] introduce a method for constructing multiple sparse, high-dimensional vector representations of words. Huang et al. [?] extend this approach incorporating global document context to learn multiple dense, low-dimensional embeddings by using recursive neural networks.

Both the methods perform word sense discrimination as a preprocessing step by clustering contexts for each word type, making training more expensive. Improvements are suggested in the methods proposed by Neelakantan et al. [?].

4 Proposed Approach

The work we aim to complete can be roughly described in the following steps,

1. Construction of word embeddings using the following approaches:
    - Consider both Local and Global features, termed as Global Context Aware Neural Language in [?]. (Try to find alternatives or improvements to the global features described in [?])
    - Using a Skip Gram model, as done in [?], [?]. Existing implementation at [?], Alternative explanation in [?].
2. Initial Context-based clustering of words.
3. Compute multiple senses using both Parametric and Non Parametric models.
4. Compute the final clusters using AdaGrad Stochastic Gradient Descent [?].
5. Comparision on both isolated and context-supported pair of words.

5 Datasets

- WordSim-353 dataset: Associate human judgments on similarity between pairs of words, but similarity scores given on pair of words in isolation
- Stanford's Contextual Word Similarities (SCWS): Consists of a pair of words, their respective contexts, the 10 individual human ratings, as well as their averages
- Training Corpus: Same as the one used in [?]