

Stephanie You  
15300 Research and Innovation in Computer Science, Fall 2018  
12/16/2018

## Intrinsic Evaluation and Comparison of Different Word Embeddings

### Project Milestone Report #1

## 1 Work Accomplished

At this point, I have read a few articles related to word embeddings, both general and specific, and tinkered with two word embedding modules on TensorFlow Hub.

### 1.1 Articles

I have read and taken notes on the following articles:

- “[The Current Best of Universal Word Embeddings and Sentence Embeddings](#)” (Thomas Wolf)
- “[7 Types of Artificial Neural Networks for Natural Language Processing](#)” (Olga Davydova)
- “[A Beginner’s Guide to Word2Vec and Neural Word Embeddings](#)”

### 1.2 Word Embedding Modules

So far, I have looked at the [universal-sentence-encoder](#) and [elmo](#) word embedding modules found on TensorFlow Hub.

There is a [Colab notebook](#) for the universal encoder that allows me to easily explore and learn how to interact with the modules within my browser. Although it is only set up to use the universal sentence encoder, I can alter the code so that it uses the elmo module instead (or any other TensorFlow Hub word embedding module).

## 2 First Milestone

My first milestone set in the initial project proposal:

*I will read published papers recommended by Professor Wehbe or found online, and download pre-trained models from Tensorflow Hub to familiarize myself with the usage of word embeddings.*

I have accomplished this goal to an extent. I have only read three articles at this point (in addition to the literature read mentioned in the project proposal), so I hope to find and read at least a few more papers before I move on to the next step in my research timeline.

## 3 Next Steps

The next steps are to obtain access to CMU clusters through Professor Wehbe and decide which word embedding algorithms to use next semester. Since most of the fourteen English word embedding modules on TensorFlow Hub are not trained with the same test corpus, I also need to investigate how to obtain pre-trained models for each of the algorithms I decide to use.