Stephanie You (szyou)
15-300 Fall 2018

Intrinsic Evaluation and Comparison of Different Word Embeddings

https://stephanieyou.github.io/research

1 Introduction

Natural language processing (NLP) tasks prevail in many facets of everyday life, from using the Google Translate app for a quick translation to asking Alexa to turn on the lights. To perform tasks like machine translation, applications often use machine learning models trained on large corpora. A common preprocessing step for these models involves word embeddings, which are mappings from words to vectors.

For my 15-400 project, I will be working with Leila Wehbe, an Assistant Professor in the Machine Learning Department at Carnegie Mellon University. The project aims to shed light on the various semantic features that neural word embeddings capture by using novel evaluation methods to compare different embeddings. Preliminary ideas include implementing canonical correlation analysis and regression residuals. The approach is to use quantitative analysis and experiments to postulate qualitative characteristics of word embeddings.

1.1 Word Embeddings

A word embedding is a mapping from words to high-dimensional vectors, and these vectors typically reflect some semantics of the original word. Neural word embeddings are obtained by training a model using an algorithm such as word2vec or GloVe.

Because of the machine learning approach, many aspects of these embeddings, such as why some embeddings perform better with certain downstream tasks, are still a mystery.

1.2 Motivation

Gaining a deeper understanding of how word embeddings work can lead to improvements in algorithms for various NLP tasks, especially if the quality of word embedding becomes a bottleneck of the algorithm.

2 Project Goals

My overall goal is to make a contribution to the collective knowledge of word embeddings and introduce new methods of evaluating them.

2.2 The 75% Goal

If I do not meet all my milestones, I should still be able to replicate existing results and run at least one new analysis, like canonical correlation analysis (CCA).

2.1 The 100% Goal

If all goes expected, I will be able to analyze the results of my experiments and interpret novel hypotheses about the differences between popular word embeddings.

2.3 The 125% Goal

If my project goes more smoothly and faster than expected, then the goal is to work with brain data and see how each word embedding differs when predicting brain patterns. This goal is closely tied to Professor Wehbe's research interests, and my work with embeddings can aid her research directly in this way.

3 Milestones

By the end of the fall semester, I expect to have gained a better understanding of word embeddings and the general differences between the current popular pre-trained embeddings. To accomplish this goal, 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 also have biweekly milestones planned for the spring semester, starting with February 1, 2019 and ending on May 3, 2019.

3.1 Milestone 1: February 1, 2019

By this date, I should choose which four word embedding algorithms to investigate for the remainder of the semester. This requires using the knowledge from the previous semester's technical milestone, as well as any additional research on popular word embeddings, to make an intentional choice of which embeddings will be the most interesting to experiment with.

3.2 Milestone 2: February 15, 2019

My goal is to have a deeper understanding of the chosen word embeddings by reading relevant papers and blog posts.

3.3 Milestone 3: March 1, 2019

I should obtain the chosen word embeddings for a single test corpus. For the experiments, the word embeddings ideally should have been trained on the same training data so that the training data does not introduce a confounding variable.

3.4 Milestone 4: March 22, 2019

My goal is to be able to replicate the results from Schnabel et al. "Evaluation methods for unsupervised word embeddings" by this point in the semester. This paper discusses results from several types of word embedding evaluations.

3.5 Milestone 5: April 5, 2019

After replicating another paper's results, I should be able to confidently use CCA (canonical correlation analysis) and/or regression residuals to find similarities and differences between word embeddings.

3.6 Milestone 6: April 19, 2019

Next, I will interpret the results from the initial CCA/residuals experiment. I should also run more experiments to determine which words lead to the largest differences in word embeddings; the hope is that this reveals more information about the information that these word embeddings capture through the training process.

3.7 Milestone 7: May 3, 2019

Finally, I will gather all the data collected throughout the semester, in addition to my background knowledge in the research area, and make my final analysis.

4 Literature

Khandelwal et al. introduced me to the idea of using ablation studies to experimentally determine how one aspect of a model, in this case linguistic context, affects the quality of the model in the paper "Sharp Nearby, Fuzzy Far Away: How Neural Language Models Use Context."

In “Dynamic Meta-Embeddings for Improved Sentence Representations,” Facebook researchers determined the best word embedding for a task dynamically within the model.

“Evaluation of Word Vector Representations by Subspace Alignment” by Tsetskov et al. and “Evaluation methods for unsupervised word embeddings” by Schnabel et al. both provide background on what methods have already been used to compare word embeddings.

I will continue reading published papers and blog posts to gain a better understanding of the popular word embeddings as well as the current findings related to embeddings.

5 Resources Needed

To conduct my experiments, I will install PyTorch and/or TensorFlow, which are both open source libraries used for machine learning. To obtain word embeddings, I can download publicly available, pre-trained models from sites such as TensorFlow Hub. Since I will be using pre-trained models, I do not need GPUs for training. However, I will need access to the CMU clusters, which are not readily available to students; Professor Wehbe is kindly letting me use hers.