A Vector Space Model of Language Using Semantic Role Structures

Charley Wu1 (charleymswu@gmail.com), Marcin Skowron2, and Paolo Petta3
1MEi: CogSci, Universität Wien
2Austrian Research Institute for Artificial Intelligence (OFAI)
3Institute of Cybernetics, Tallinn University of Technology

Introduction

In this poster, we present several methods for creating a compositional Vector Space Model (VSM) that can capture the semantic difference between texts based on opposing ideological positions. We compare the effectiveness of these methods by training a Support Vector Machine (SVM) classifier on the vector outputs produced by the model. Traditionally, VSMs have been trained using Latent Semantic Analysis (LSA), which uses cross-occurrence words to learn a representation about their meaning. This is based on the notion that words that occur in the same contexts have a similar meaning. We add to this framework by using Fisher's Projections (Fisher Projections: Latent Semantic Analysis) to provide structured data for the creation of a compositional VSM. The goal is to use the semantic information of an entire document.

Process

Part 1. Word2Vec - from words to vectors

This is an LSA model we trained on the entire English Wikipedia. Word2Vec allows us to query any word it has learned and return a 300-dimensional word vector. Word vectors are composed in such a way that Cosine Distance is a measure of semantic similarity. (Complexity: 48hrs, 60 GB)

Part 2. Semantic Role Labeling - adding structure

Using a Deep Neural Network trained on the Wall Street Journal corpus (SENNA), we interpret the relationships between semantic arguments in a text. This provides the structure about how individual words are related to each other. (Complexity: 24 hrs, 2 GB)

Part 3. TF-IDF Model - which words are important

Term Frequency-Inverse Document Frequency is a measure of how important a given word is to a document in a given corpus. The model was trained on a subset of the CORPS corpus and is used to scale the magnitude of the word vectors to reflect the amount of information it contributes to the document. (Complexity: 1 min, 3 MB)

Part 4. Compositionality - putting it together

For each semantic role structure in a document, we translated individual words are related to each other. This is based on the notion that words that occur in the same contexts have a similar meaning. We add to this framework by using Fisher's Projections (Fisher Projections: Latent Semantic Analysis) to provide structured data for the creation of a compositional VSM. The goal is to use the semantic information of an entire document.

Part 5. Classification - evaluating the methods

The corpus was divided into training and testing data. We used the training data to train the SVM, and the testing data to evaluate the performance of the model. The evaluation was done using 5-fold cross-validation. (Complexity: 10 mins, 2 GB)

Discussion

We set Method 1 as a baseline because it is essentially a non-compositional VSM. It makes almost no use of the semantic structure provided by SENNA and is equivalent to a vanilla Word2Vec approach with the addition of TF-IDF vector scaling. Methods 2 and 3 also make little use of SRL structure but have different mathematical approaches. Methods 4-6 were not significantly different from the non-compositional VSM. It makes almost no use of the semantic structure provided by SENNA and is equivalent to a vanilla Word2Vec approach with the addition of TF-IDF vector scaling. Methods 2 and 3 also make little use of SRL structure but have different mathematical approaches. Methods 4-6 were not significantly different from the non-compositional VSM. It makes almost no use of the semantic structure provided by SENNA and is equivalent to a vanilla Word2Vec approach with the addition of TF-IDF vector scaling.

Acknowledgements

CORPS was provided for research purposes by Marco Guerini. OFAI is supported by the Austrian Federal Ministry for Transport, Innovation, and Technology. SENNA was written by Ronan Collobert and is available through non-commercial licence from NEC Labs America. Word2Vec was implemented through Gensim, which is written by Radim Rehurek.