Assessing a Word Similarity Model for Determining Overall Category

Ethan Campbell-Taylor
Linguistics '16

Becky Marvin
Cognitive Science '16

Harvey Xia
Computer Science '16

Abstract

This document contains the methodology and results of a study examining the effectiveness of using WordNet's semantic hierarchy information to classify text documents. We used the semantic similarity information from WordNet, in addition to a hierarchical clustering algorithm to determine the similarity of the nouns in a given text document. We then found the hypernyms of the resulting clusters and compared the hypernyms to the labeled categories of the document. We assessed the accuracy of our algorithm by comparing its judgments of overall categories to the actual labeled categories of documents.

1 Introduction

Text classification has previously been researched using "bag of words" representations of word meanings. We explore a different approach in this paper: that of using WordNet's available semantic similarity information combined with a hierarchical clustering algorithm to determine the overall category of a text document. We believe this approach better models the semantic information in a given document, which could be useful in determining the overall category of a document.

2 Our Model

2.1 The Algorithm

Since our model is attempting to determine the category of a document based on the nouns used in it, the first step in our algorithm is to extract all the document's nouns. After removing punctuation and non-ascii characters, the algorithm tokenizes each line and tags the tokens with nltk's part of speech tagger.

It then stems the relevant nouns, determined by whether their stemmed versions exist in wordnet. We removed the stems of words ending in "ing," "s," "e," "able," "y," and "er." For example, "photography" becomes "photograph," but "agency" remains as is, since "agenc" is not recognized as a word by WordNet. After stemming, the algorithm counts every instance of each token and stores this information in a Python dictionary. For each of these tokens, the algorithm finds the corresponding synset (the set of senses listed in WordNet for the word) if it exists. These synsets are used to compute similarity measures for the purposes of clustering.

Having done this, the algorithm computes a similarity matrix for every pair of nouns in the document: that is, it calculates and stores the similarity of the synsets for each noun pair. We used the fastcluster library to do heirarchical clustering: we exclude clusters smaller than size 2 and larger than size 99, as well as those with a distance of less than 0.5. The clusters are then sorted by size, those with the greatest number of raw tokens appearing first. Finally, for each cluster, the algorithm finds the hypernym (the least common ancestor of all the synsets in the cluster).

The algorithm evaluates performance by finding the item in the hypernyms returned by our algorithm that is most similar to a tag for each labeled tag of the text document. We use Wu-Palmer similarity as a measure of how similar two words are according to WordNet.

3 The Corpus

NPR woooooo