SEMANTIC LEVEL RELEVANCE EVALUATION

===== MOTIVATION =====

Semantic Level Relevance Evaluation is a sub-problem of natural language understanding. Some applications can scale down the complexity of search tasks. It would become possible for a machine to gather relevant news, articles, and relevant data about a particular topic more efficiently and accurately. By solving the problem of semantic level relevance (and its family), a computer will have a more extensive grasp of the semantic of words, bringing it one step closer to the ability to understand features of human language.

===== OUTLINE =====

There are several levels to which we address this project, in the following order:

BASIC
Level 0 : Given Wikipedia articles of certain length, evaluate relevance.

INTERMEDIATE
Level 1 : Given shorter articles (maybe as short as a sentence), evaluate relevance.

ADVANCED
Level 2 : Given an article, produce a set of most relevant words (auto-tagging)
Level 3 : Given an article, produce a "short summary" of that article of a CERTAIN length.

Our primary goal is to finish level 2 at least, that is, to make an automatic tagging algorithm that produce only key words or phrases, and also evaluate the relevance (similar to the level of "confidence"). Level 3 goal is also within our roadmap, and we will also try to expand our project to those levels as well. However, only in case of emergency, we will drop the harder problems and tackle the preliminary goals.

===== INPUT / OUTPUT BEHAVIOR =====

Input = names of two Wikipedia articles
Output = relevance ranging from 0 to 1 (0 is least relevant, 1 is most relevant)

===== METRIC OF SUCCESS =====

We can measure our algorithm by comparing the result against manual relevance evaluation. This can be done by randomly selecting different sentences from same paragraphs in the same source, pick existing tags in question answering websites, or selecting some Wikipedia categorization of articles, and use these keywords (along with corresponding articles) and run our relevance test on inputs. The better algorithm should give high corresponding factor, as compared to more random articles.

===== ROADMAP =====

__Getting started__
We have implemented a parser to process raw Wikipedia articles, punctuation handler, frequency analysis, and basic relevance evaluator.
__Baseline algorithm/models__
We implement a baseline algorithm ("Angular Analysis") by create a feature vector that contains word counts for each article with particles and other unrelated words

removed. Then we evaluate the relevance of two articles by taking a dot product and normalize it by the norm of two vectors.The relevance value therefore will range from 0 to 1. The baseline algorithm performs moderately on the dataset but does not capture some advance linguistic features. For example, some relevant articles
yield relevance such as 'Sushi' and 'Tempura', but 'Roti' and 'India' yields 0.188.

__ More complex language modeling and search__
1. We will use Naive-Bayes and try to capture some syntactic features (part of speech tagging / PCFG),
eliminating words like "is" "I" "on" "maybe" (guessing important words like "science" > "physics" > "thermodynamics" > entropy) (Eliminative frequency analysis + Maximum Entropy parsing)
2. We might use some variation of PageRank to help assign preliminary score to words as well.
3. Some language modeling might be more helpful. We may try to construct Lexical Semantic graph, or
word vector model, then implement some search algorithm
4. We might end up using some deep learning / neural network if we have enough time.

===== PRELIMINARY DATA =====

```
'Pad Thai', 'Sushi':                                0.257
'Pad Thai', 'Thailand':                             0.428
'Statue of Liberty', 'American Revolution':         0.363
'Elephant', 'Dog':                                  0.331
'Elephant', 'Detergent':                            0.139
'Elephant', 'Cat':                                  0.389
'Elephant', 'Car':                                  0.289
'Sushi', 'Tempura':                                 0.421
'Sushi', 'Hamburger':                               0.333
'Stanford University', 'University':                0.307
'Stanford University', 'Massachusetts Institute of
Technology':                                        0.307
'Stanford University', 'Tree':                      0.150
'Roti', 'Dosa':                                     0.281
```

===== CHALLENGE, GAP, ORACLE =====

The baseline algorithm performs moderately on the dataset but does not capture some advance linguistic features. For example, some relevant articles yield high relevance such as 'Sushi' and 'Tempura', but 'Roti' and 'Dosa' yields lower relevance. In addition, we believe that the baseline algorithm will perform poorly on short inputs (a single sentence or a word), although we have not tested it yet. We would like to have oracles which can be language models where all words are represented by nodes in a graph. The costs of edges can represent the inverse of relevance between two words.

===== RELEVANT WORKS =====

"Assessing Relevance"
Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze,
Introduction to Information Retrieval, Cambridge University Press. 2008.

Alessandro Presta's Github repository, Tagger Project
https://github.com/apresta/tagger