Rosalie Day – Milestone Report, Capstone 2
Prediction of Category in News Text
February, 2020

The Purpose

Natural language processing is ubiquitous for the internet and the IOT universe. Classification provides an important tool for identifying subject matter of content. Results of classification models may provide information for composition that is easily classified and potentially insights on text that is not easily classified.

This analysis compares standard Scikit Learn models and extractors of the text for full news articles. The focus is on the discrete words, not parts of speech, sequences or contexts in which they are used.

The Data

The BBC news data set composed of 2225 full-text articles and appears in the Kaggle competition data sets. Downloaded as a comma separated file, the set has two variables: news category ('category') and full text of the articles ('text'). The text was already converted to lower case with punctuation removed. Data cleaning included removing duplicates. There remained 2126 articles and five categories. (*See chart of categories and percentages in final report*).

The article text was prepared for tokenization by removing special characters. The list of "stopwords," articles, and other often used, meaningless words when taken out of context, was provided in the Natural Language Toolkit package. The stopwords were removed in separate steps so the length of the articles, in this case, number of tokens, could be compared with and without stopwords. Stopwords could be generated according to frequency for this corpus (or hand compiled) as a parameter tuning for the Bag of Words and Term Frequency-Inverse Document Frequency (TF-IDF) extractors in SciKit-Learn.

The 20 most common words for each category were produced. (*See charts in final report*).

Preliminary Findings

In preparation for the classification models, the tokens were extracted and vectorized, converted to numeric coding, for machine learning models. The two extraction outputs were: a "Bag of Words," word counts in the entire body of texts, the "corpus;" and Term Frequency-Inverse Document Frequency ("TF-IDF"), words weighted for importance.

Bag of Words

CountVectorizer was used to extract a Bag of Words, words pooled with no use or sequence information, for the corpus. This extractor creates a feature for each token. In its default setting, "ngram_range; tuple (min_n, max_n), default=(1, 1)" *from Scikit-Learn documentation*, a token is one word. Token, word and term will be used interchangeably, as will document and article because one refers to the other in this analysis.

The Bag of Words, in its transformed output, is a document-term matrix, composed of word features in the columns – a vocabulary of 29,241 and rows corresponding to the 2,126 articles. The cells in the matrix are filled with the count of how many times the word (feature column) occurred in the article designated by the row.

In the final report, a table shows a glimpse of the numeric code for the word and the sum of the frequency the words occurred for all the articles. The second column below corresponds to the sum of every column in the matrix above.

One intuitive parameter to tune in CountVectorizer is directing the extractor with respect to the minimum occurrence of that term. Called the minimum document frequency, "min_df," or the cut-off, the default is one time (min_df = 1).

A plot of cumulative distribution of document frequencies is included in the final report.

By inspecting the cumulative distribution of word frequencies by number of documents, the minimum probably is greater than one. The fitted and transformed text with CountVectorizer with the minimum document frequency of 2 (min_df= 2). This reduces the total number of features, the vocabulary, to 17240.

For optimal tuning, the model is considered. A combination of trial parameters for the extractor and the respective preferred model(s) may be needed for optimization.

Next steps - Prediction Models

The next step will perform classification performance of supervised machine learning classification models on the Bag of Words vectorization. The final step will be use of the top performing models on the TF-IDF vectorized text.

More research
Performance of unsupervised models, cluster analysis and principal component analysis, would be interesting for comparison.