LING 83600/CSCI 76000
Prof. Levitan

HW2: Text Classification

The goal of this HW is to gain experience using NLTK to preprocess text and extract features and use scikit-learn to classify text.

Deadline: Thursday September 22, 11:59pm

Submission instructions: Submit a document with your answers to the HW questions (.pdf or .doc) as well as your code (.ipynb or .py). Your code must run with python3.

Motivation

There is an overwhelming amount of news information available online. Some news headlines are known as clickbait – they aim to attract users to click on a link but the articles that they link to may not be of value or interest to the reader. In this HW we aim to answer the following question: C*an we automatically distinguish between clickbait and non-clickbait headlines, using a variety of linguistic cues?*

Data

You are provided with a corpus of clickbait and non-clickbait article headlines. The corpus is described in detail in this paper: https://arxiv.org/pdf/1610.09786.pdf
The data is stored in two files:
Non-clickbait: http://www.cs.columbia.edu/~sarahita/CL/non_clickbait_data.txt
Clickbait: http://www.cs.columbia.edu/~sarahita/CL/clickbait_data.txt
Note: this data is an extended version of the data in the paper – there are 16,000 headlines in each file.

Task

You are provided with starter code that loads the data, extracts a set of features, and then trains a Naïve Bayes classifier using those features and outputs the classifier accuracy. Your job is to extract additional feature sets using NLTK and report the classifier performance for each set of features. (Note: the starter code is optional – you can write your own)

Starter code:
https://colab.research.google.com/drive/1fpmrn9eXJ0mbk02tmMafO5ESegnr0Nok?usp=sharing

Extract the following features:

1. Stop words: counts for each function word (from the NLTK stopwords list)

    ***this function is already implemented for you!**

2. Syntactic: counts for the following 10 common POS tags -- ['NN', 'NNP', 'DT', 'IN', 'JJ', 'NNS','CC','PRP','VB','VBG']
3. Lexical: counts for 30 most common unigrams in entire corpus (remove stopwords and punctuation for unigram count)
4. Punctuation: Counts for each punctuation mark in string.punctuation
5. Complexity:
    - average number of characters per word
    - #unique words/#total words
    - number of words
    - Count of “long” words - words with >= 6 letters
6. Your own proposed feature set: Think about what other features may be useful for clickbait identification and implement them. You can get ideas from this paper: https://arxiv.org/pdf/1610.09786.pdf

You are encouraged to play around with preprocessing and normalization that might help performance. You can also try other classification models in addition to Naive Bayes.

Record your results and answer the questions below.

1. Results (5pts)

Report the accuracy obtained using the following individual feature sets, as well as using all features combined.

Feature Set	Accuracy
Function words	0.8736
Syntax	0.7672
Lexical	0.7349
Punctuation	0.5013
Complexity	0.7004
Your feature set	0.6133
All features combined	0.9186

2. Based on the features you extracted and the results, what observation(s) can you make about the nature of clickbait headlines? (2pts)

The features extracted were the counts for each stopword, the counts for common POS tags, counts for common unigrams, counts for punctuation, and complexity, which is based on the average number of characters per word, the percent of unique words, the total number of words, and the number of long words.

Based on my results, extracting the counts for each stopword produced the highest accuracy at 87%. It seems one main difference between clickbait and non clickbait headlines is that clickbait headlines are typically complete sentences, so they require stopwords to make up the sentence structure, non clickbait headlines on the other hand typically contain more important context words to give the reader an idea of what the article is about, but leave out stopwords.

According to my results punctuation returned the least accuracy, only 50%. Which is interesting since I would assume clickbait headlines are more likely to use excessive punctuation in order to intrigue a reader and capture their interest. It is possible it is only 50% because while clickbait headlines are more likely to use exclamation marks and question marks, non clickbait headlines are just as likely to use commas, periods, and of course quotation marks in quotes.

Common POS tags had an accuracy of 77% which is the second to highest, because the types of words used in clickbait and non clickbait headlines differ. Non clickbait headlines contain more proper nouns which would be context words while clickbait headlines contain more adverbs and determiners and personal and possessive words since clickbait headlines use more descriptive words and more words related to the readers to capture their attention.

Counts of common unigrams returned a decent accuracy at 73%, but I was a bit surprised the accuracy was not higher since it is related to counts of stopwords, which produced the highest accuracy. Since unigrams is only one word in length, clickbait and non clickbait titles will differ in this category since clickbait headlines are more likely to use the same words more often especially stopwords to help with sentence structure while words used in non clickbait titles vary more.

Complexity returned an accuracy of 70% which is reasonable. It is made up of four features: average length of words, percent of unique words, count of words, and count of long words. In clickbait headlines, the average length of words will likely be shorter because they use simpler words and include many stop words; the percentage of unique words will also be higher for this reason. Count of words will be longer in clickbait headlines because they are complete

structured sentences. Count of long words will be higher in non clickbait headlines because they use more complex words.

3. Describe the additional feature set that you implemented and the motivation for choosing that set of features. Were these features useful for clickbait classification? (2pts)

The additional feature set I implemented was count of Possessive words. I used a list of possessive nouns, possessive pronouns, and possessive determiners and calculated their counts. The reason I choose this feature is because clickbait headlines, when trying to intrigue their readers, will often address them in first or second person. Making an article more personal and using possessive words makes readers believe the article is more relevant to them. On the other hand non clickbait headlines often use third person.

I was surprised the accuracy for this feature was only 61% which is the second to lowest which is not very useful for clickbait classification when used on its own.

4. What are some ideas you could try to further improve performance? (1pt)

Something I could try to further improve performance would be to add more features such as detecting hyperbolic words, internet slang, and common clickbait phrases which will all be significantly higher in clickbait headlines. I believe these features will significantly help performance. Of course, as seen in the results and as would be expected, combining features together, even if individually they are not very strong and providing high accuracy, together they are stronger and produce higher accuracy. All of the features I used combined provided an accuracy of 92 percent which is great!