Author Identification Problem using Text Classification

Michael Sadler and Siddarth Patel

Abstract

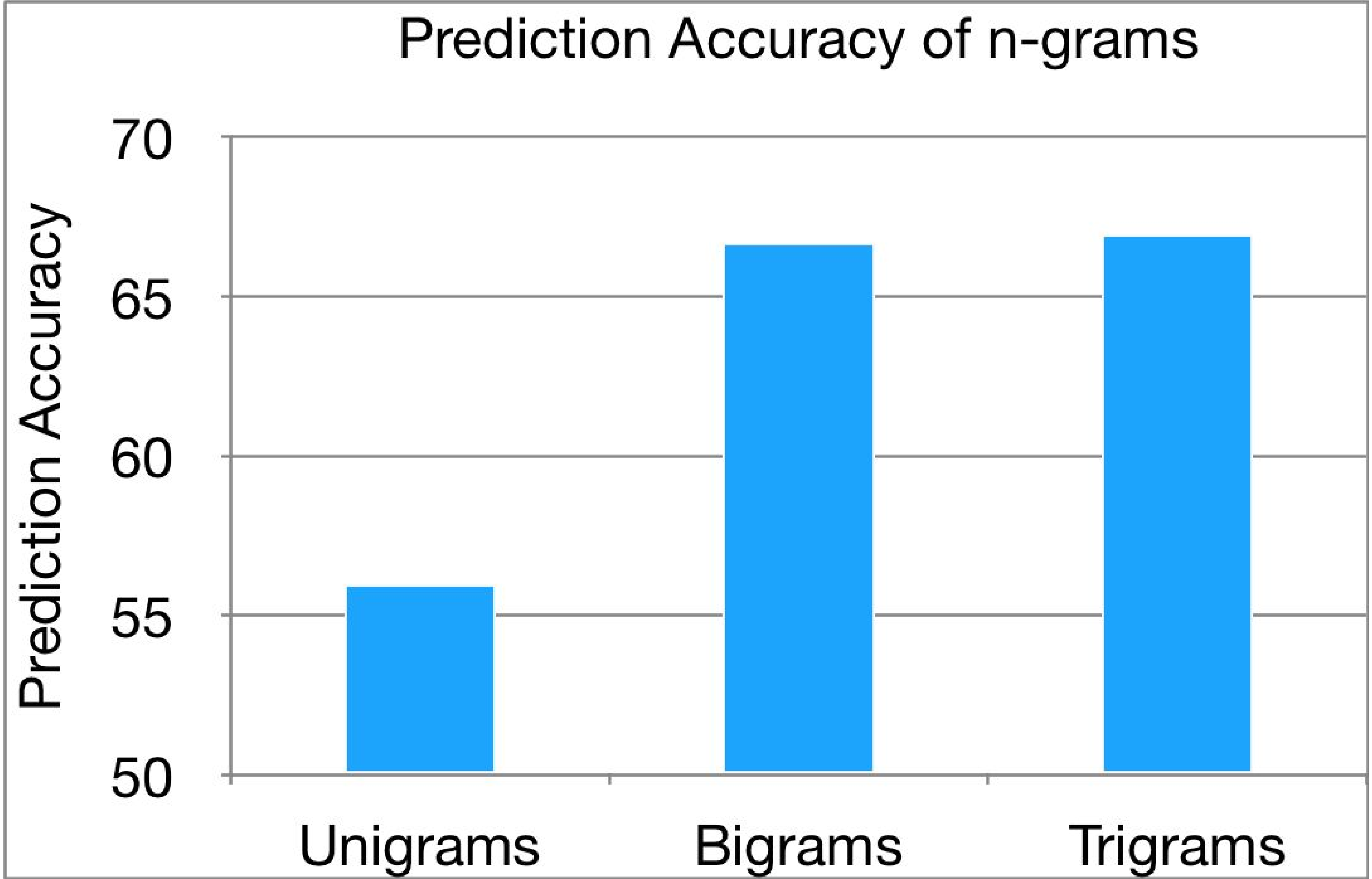
Correctly identifying the author of a work of literature has been a task that we have struggled with throughout history. We still have no proof that William Shakespeare truly wrote all the works he is attributed. Many public figures believe that there are multiple authors (a.k.a. ghost writers) which contributed to these works, but there is no proof or evidence for or against this theory. With the advancement in computers, we could create a model to predict the author of a piece of literature, and possibly even detect plagiarism and/or wrongful appropriation of one's creative ideas.

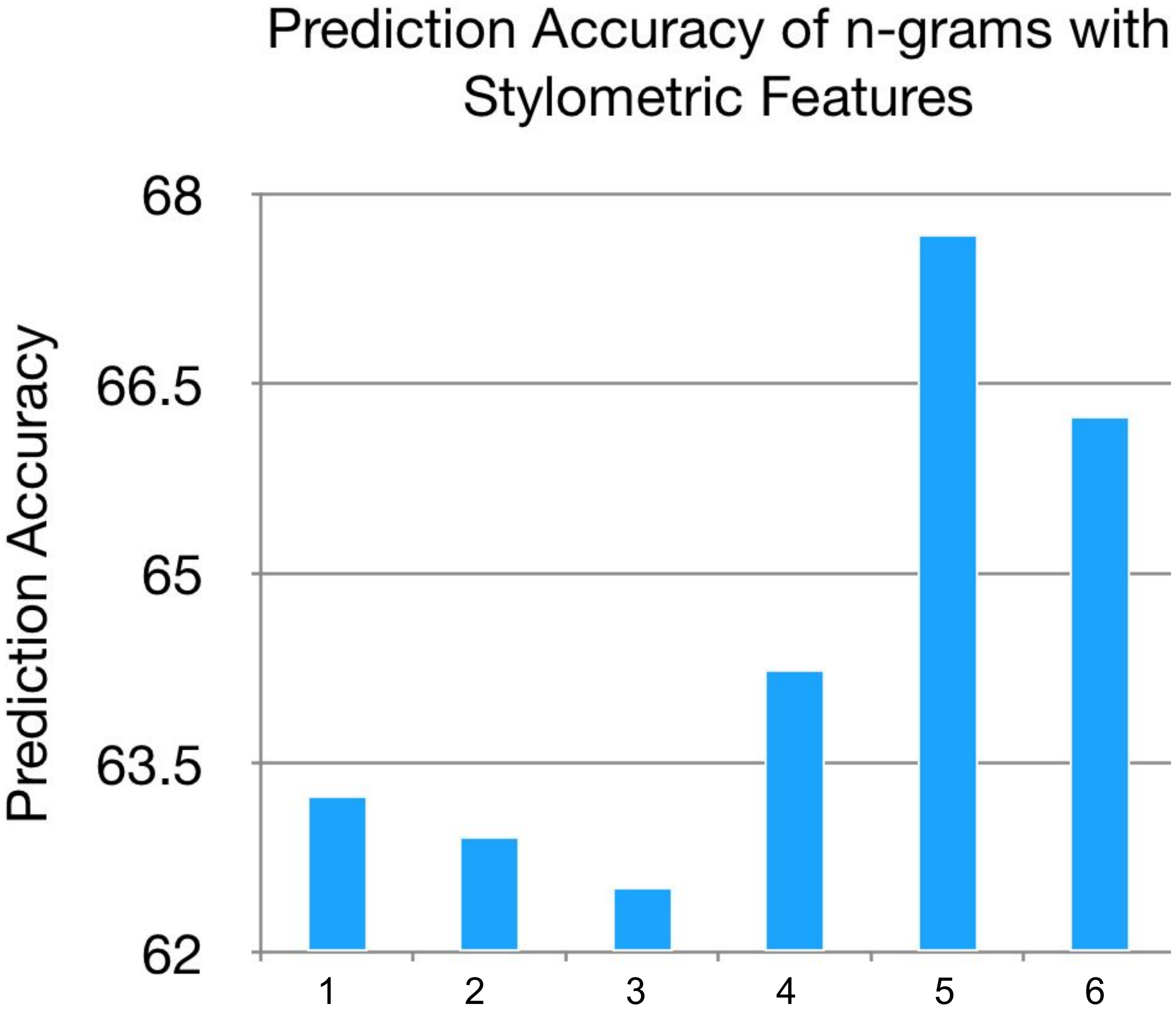
Why we chose this problem?

- Author Identification has been a very important and practical problem in NLP
- Many of the methods involved use deep learning
- Inspired from hw 1 in class, we decided to tackle this problem using text classification
- The baseline for this project is a unigram prediction model using Naives Bayes as the classifier (which gave a result of 55.96% accuracy)
- Our first step was to see how other n-grams compared to this result (including combinations of n-grams) in hopes of exceeding the baseline accuracy.
- After trying multiple combinations of n-grams to build a prediction model, we then included different combinations of stylometric features (stemming, stop words, punctuation, sentence/word length, etc..) when building our prediction model to try and further improve the overall accuracy

Similar approaches to the problem

- W. Cavnar and J. Trenkle's paper on N-gram based text categorization introduces a baseline solution for the Author Identification Problem by applying an N-gram approach. This approach applies Zipfs law to identify authors by their repetitious use of unique tokens (words and/or phrases) which are distinct from other authors.
- A. Aizawa's paper on Linguistic techniques to improve the performance of automatic text discusses simple techniques (such as stemming and stop word removal) and other techniques (such as morphological analysis and probablistic language modelling).
- Author Identification task by Devansh Dalal. He approaches this problem by developing a model using a text classifier based on logistic regression which includes n-grams, style markers and document finger-printing as features.


Prediction Accuracy of n-grams
Prediction Accuracy
70
65
60
55
50
Unigrams
Bigrams
Trigrams



Combinations from left to right:

1.Unigrams + Bigrams

2.Unigrams + Bigrams + Stemming

3.Unigrams + Bigrams + Stemming + Filtering Punctuation

4.Unigrams + Bigrams + Stemming + Punctuation Counting

5.Bigrams + Punctuation Counting

6.Bigrams + Stemming + Punctuation Counting

Summary of the Results

From the first graph, we see that bigrams(66.68) and trigrams(66.96) have a much better prediction accuracy compared to our benchmark (unigrams, 55.96).

From the second graph, there were a few things to note:

- Removing stop words has a **negative impact** on accuracy
- Performing stemming has a **negative impact** on accuracy
- Removing punctuation has a **negative impact** on accuracy
- Combining different n-grams appears to have a **negative impact** on accuracy
- Changing punctuation into unique tokens has a **positive impact** on accuracy
- Running more complex tests results in memory overload on 16 GB RAM