CIS 530: Computational Linguistics
Brendan Mahoney

HOMEWORK 2 WRITE-UP

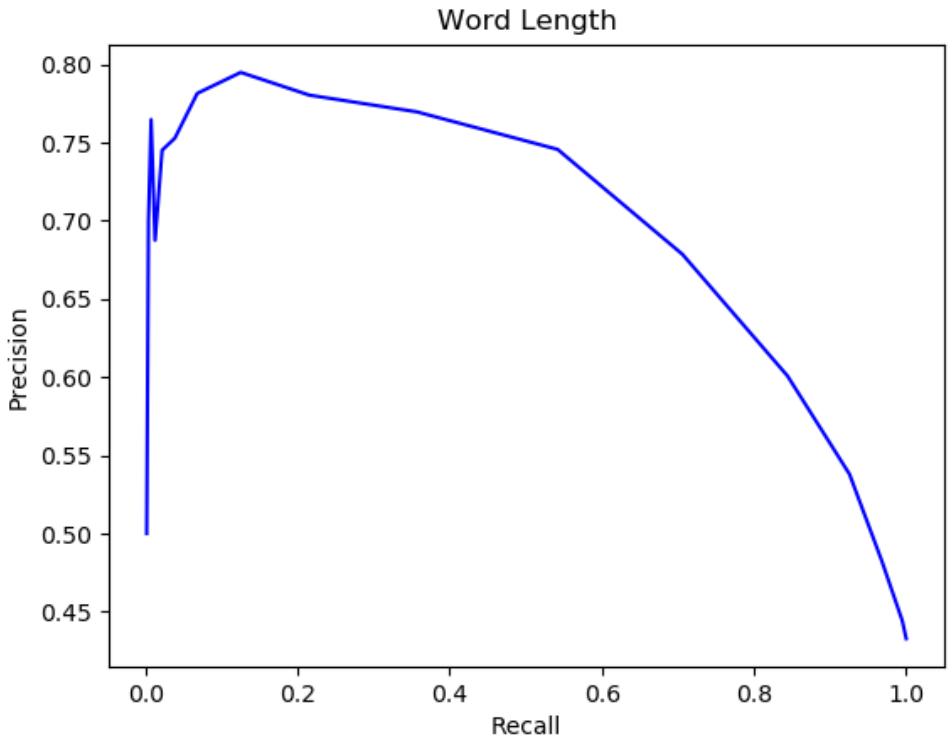
I. BASICS

The summary of my results is as follows:

Classifiers	Train			Dev		
	Precision	Recall	F-score	Precision	Recall	F-score
allcomplex	0.43275	1	0.75318	0.4180	1	0.74179
wordlength	0.53794	0.92547	0.78423	0.52708	0.95454	0.79363
wordfrequency	0.49212	0.97515	0.78302	0.46412	0.95933	0.75731
naivebayes	0.57628	0.90121	0.78987	0.55766	0.91387	0.78803
logistic	0.57628	0.90121	0.78987	0.55766	0.91387	0.78803

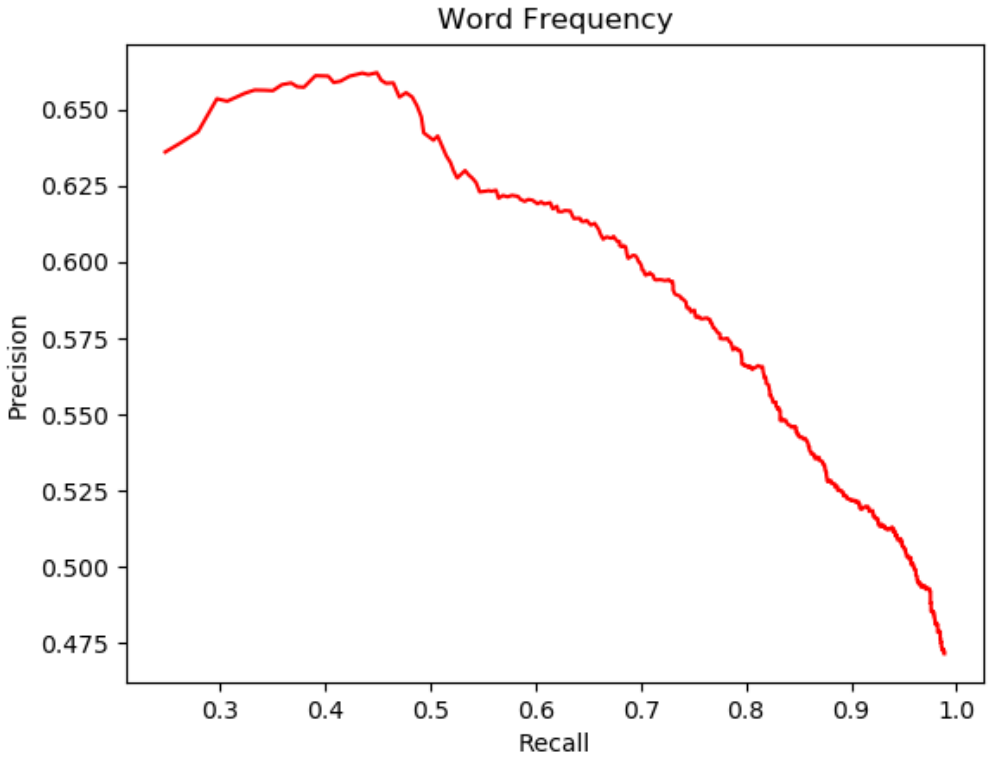
Q 2.2:

In thresholds 1 through 20 the best threshold is 6, which produces an f-score of 0.7842369355036103. The P-R plot is as follows:

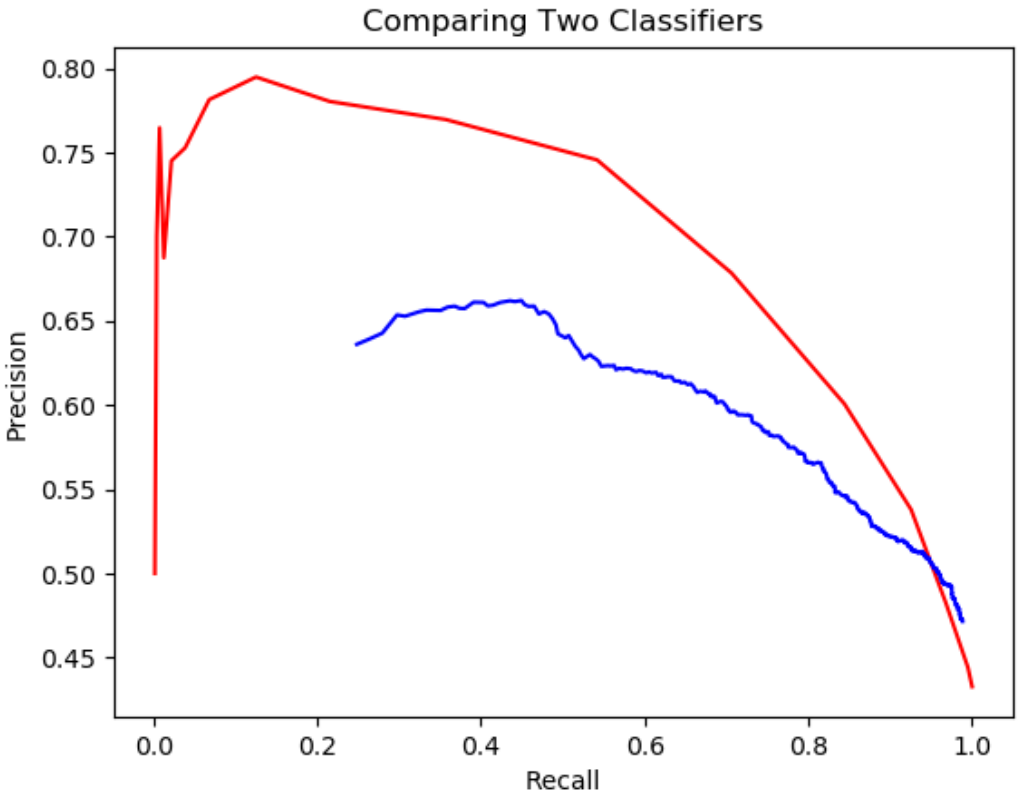


Q 2.3:

In thresholds 1 through 100,000,000 (tested with a granularity of 1,000,000) the best threshold is 69300000, which produces an f-score of 0.783022150063783. The plot is as follows:



Additionally, as can be seen from this plot comparing the two classifiers, word length is more accurate at almost all thresholds:



Q 3.3:

Well my results for these tests are exactly the same so I assume I've done something wrong here, but, as per the assignment, if they WERE different I would probably attempt to explain it by talking about how these two classifiers are looking at different things. The Naive Bayes is a generative classifier whereas the logistic regression is a discriminative classifier, which I would think probably makes the LR better at a task like this.

II. BUILDING A MODEL

I tried to build two models, a Random Forest classifier and a Neural Net, using a fixed set of features. This list of features included word length and word frequency, as well as the syllable count, and Number of Wordnet synsets. I also tested a lemmatized version of each of these. This list of features was selected after a review of related academic work on this task, such as the Complex Word Identification Shared Task 2018.

The NN ultimately outperformed the RF model[1], as can be seen in the following table:

Classifiers	Dev		
	Precision	Recall	F-score
neuralnet	0.43401	0.85074	0.68605
randomforest	0.70121	0.27511	0.32440

[1] Note: I was unable to upload the neural net code, as Gradescope couldn't import the keras module. Although I hashed out the lines in my code so that I could upload it for grading, you can see the function I ran on my own computer. Sorry if this wasn't technically the right way to do the assignment, but I decided to include it because it took me a while to build and I'm proud of it :)