Investigating the Comparative Efficacy of Two Bayesian Classifiers in the Task of Text Classification

Research Question: What is the Relative Accuracy of the Multinomial Naive-Bayes Classifier as compared to the Bernoulli Naive-Bayes Classifier in Text Classification Across Languages?

A Computer Science Extended Essay

WORD COUNT: 3998

Table of Contents

1. Introduction

Machine learning is a relatively new field in technology focusing around the usage of large quantities of data to train models to recognize certain patterns [1]. Within this field is the task of text classification: given a document, a model is able to assign a classification based on the words contained within (e.g. label an email as spam or important based on its contents).

The Naive-Bayes (NB) algorithm is used to train models for text classification. In order to assign a classification, each word that occurs in a document is considered on an individual basis. The model is trained to assign the document a label by associating certain words with certain classifications[1]. There are multiple types of the NB algorithm, each utilizing different methods of classification. Namely, the Multinomial NB algorithm considers the amount of times a word occurs as an integer, from 0 to n, while the Bernoulli NB algorithm considers any given word's occurence on a binary basis, either 0 or 1[1]. The family of NB algorithms all completely disregard sentence structure and word order: words are taken from the text and considered in isolation, in association with the document's label. This is called a bag-of-words, or unigram model.

Existing research[2] states the comparative efficacies of the Multinomial and Bernoulli NB algorithms in the English language. It was found that the Bernoulli model generally performs better at smaller vocabulary sizes, while the Multinomial model both consistently outperforms the Bernoulli model and in general performs better at larger vocabulary sizes. Because NB

[1] IBM Cloud Education. "What Is Machine Learning?" *IBM*, IBM Cloud Learn Hub, 15 July 2020, www.ibm.com/my-en/cloud/learn/machine-learning.

[2] McCallum, Andrew, and Kamal Nigam. "A Comparison of Event Models for Naive Bayes Text Classification." *Https://Www.cs.cmu.edu*, Carnegie-Mellon University, 1998, www.cs.cmu.edu/~knigam/papers/multinomial-aaaiws98.pdf.

algorithms disregard word order, the aim of this paper is to investigate the comparative performance between Multinomial and Bernoulli NB across different languages; hypothetically, they should have the same efficiency rates relative to each other. That is to say, the ratio of accuracy between Multinomial and Bernoulli Naive-Bayes in English should be similar to the ratio of Multinomial to Bernoulli Naive-Bayes in Chinese, because a different word order doesn’t affect an algorithm that ignores word order.

2. Background

2.1 Definition and Types of Machine Learning

In machine learning, a model is the file to which algorithms are applied and from which output is gained [3]. There are two types of machine learning [4]: supervised and unsupervised, the distinction being how the model is trained. Supervised machine learning involves training the model with processed data; datasets whose datums have already been classified, so that the model begins to recognize what patterns it should search for between datasets and classes. Unsupervised machine learning involves training the model with unprocessed data, so that the model must find the patterns itself.

2.2 Natural Language Processing (NLP) and Text Classification

One application of machine learning is Natural Language Processing (NLP)[5]: the field of representing and analyzing natural language (i.e. human-like speech) using computers. Common applications of NLP in the real world include word prediction services when typing (such as Google's autocomplete), text-to-speech services, and understanding the intent behind a certain statement.

[3] Radich, Quinn, and Eliot Cowley. "What Is a Machine Learning Model?" *Microsoft Docs*, Microsoft, 24 May 2021, docs.microsoft.com/en-us/windows/ai/windows-ml/what-is-a-machine-learning-model#:~:text=A%20machine%20learning%20model%20is,and%20learn%20from%20those%20data.

[4] IBM Cloud Education. "What Is Machine Learning?" *IBM*, IBM Cloud Learn Hub, 15 July 2020, www.ibm.com/my-en/cloud/learn/machine-learning.

[5] Liddy, Elizabeth D. "Natural Language Processing ." *Surface.syr.edu*, Syracuse University, 2001, surface.syr.edu/cgi/viewcontent.cgi?referer=scholar.google.com.my/&httpsredir=1&article=1019&context=cnlp.

Text classification as a subset of NLP is the ability to assign a body of text with a certain class, such as assigning novels a genre[5]. It's a growing necessity in the modern world as more and more websites and content creators produce more and more text to be consumed. Common applications of text classification include email services tagging incoming emails as spam, or search engines classifying news articles as a certain genre[5].

2.3 Definition and Types of the Naive-Bayes Algorithm

The Naive-Bayes algorithm is a type of supervised machine learning used for classification tasks. Bayes' Theorem (often written as Bayes Theorem), for which NB is named, is a mathematical model used to estimate the likelihood of some final event given prior event likelihoods [6]. For example, the prior event could be the temperature of a certain day, while the final event would be if it rains or not; the prior is assumed to directly influence the final. In other words, Bayes Theorem provides the likelihood of a final event y given a prior event X. The Naive-Bayes algorithm incorporates Bayes Theorem but uses multiple prior events. In this application, the X's are the words contained in a document, while the y is the label, or class of the document. The 'naive' assumption the Naive-Bayes algorithm makes is that the appearance of one X has no effect on the appearance of another: any given X is independent and unaffected by any other X[7].

[6] Brownlee, Jason. "A Gentle Introduction to Bayes Theorem for Machine Learning." *Machine Learning Mastery*, 3 Dec. 2019, machinelearningmastery.com/bayes-theorem-for-machine-learning/.

2.4 Statistical Theory as it Relates to Machine Learning

2.4.1 A Probability Primer

Within the field of statistics, the probability of an event is the quantification of uncertainty behind the outcomes of a random variable; probability is the ratio of a certain outcome's chance to occur compared to all possible outcomes' chance to occur[7].

The marginal probability of an event A, written as P(A), is the probability of that event occurring, without any regard to other variables. Given a random variable, X, P(A) is equivalent to X[7].

Joint probability is the probability of two or more events, A and B, occurring simultaneously, given two or more random variables, X and Z. Joint probability is written as P(A and B) or P(A,B)[7].

Conditional probability is the likelihood of one event, A, occurring given the outcome of another event, B. Conditional probability is written as P(A given B) or P(A|B)[7].

Note that conditional probability is directly related to joint probability by the product rule: P(A and B) = P(A|B) * P(B). Note also that joint probability is symmetric: P(A and B) is equivalent to P(B and A).

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

Figure 1: The classical representation of Bayes Theorem [4]

[7] Brownlee, Jason. "A Gentle Introduction to Joint, Marginal, and Conditional Probability." *Machine Learning Mastery*, 6 May 2020, machinelearningmastery.com/joint-marginal-and-conditional-probability-for-machine-learning/.

Bayes Theorem originally came about as a convenient method to calculate the conditional probability of an event when not in possession of the joint probability[7]. In Bayes Theorem, P(y|X) is called the posterior probability, P(X) the prior probability, P(X|y) the likelihood and P(y) the evidence[7]. The posterior probability is the probability of a model given new data[13]: how likely a certain outcome is. Prior probability is an assumed value for the probability of the model, before evaluating new data.

It is possible to get a value of 0 for the likelihood and evidence, thus a 0 posterior probability. This is not helpful in statistical research, so Laplacian smoothing is often used. A value of 1 is added to the numerator and denominator. This prevents a 0 posterior probability while also maintaining the relative integrity of calculations[8].

2.4.2 Naive-Bayesian Probability

$$P(y|x_1, ..., x_n) = \frac{P(x_1|y)P(x_2|y)...P(x_n|y)P(y)}{P(x_1)P(x_2)...P(x_n)}$$

Figure 2: The Naive-Bayes expression [3]

While Bayes Theorem is designed to relate only two variables, X and y, NB expands it to relate X_n variables, via the naive assumption: the variables X_n and X_{n+1} have no effect on each other, and have an equal effect on the final outcome. The naive assumption is rarely true in real life, as variables are almost always related: If the word “United” is present then the word “States” is much more likely to follow than the majority of other words. Regardless of the

[8] Jayaswal, Vaibhav. “Laplace Smoothing in Naïve Bayes Algorithm.” *Medium*, Towards Data Science, 22 Nov. 2020, towardsdatascience.com/laplace-smoothing-in-na%C3%AFve-bayes-algorithm-9c237a8bdece.

assumption, the NB classifier often works well for large data sets, providing a useful approximation in a relatively short time[2].

2.4.2 A Statistics Primer

2.4.2.1 Statistical models

In statistics, a model is a mathematical representation of observed data[9]. Models can be probabilistic or deterministic. Deterministic models determine the single possible outcome for an event (e.g. the conversion of a measurement from metric to imperial; there is only one possible outcome). Probabilistic models provide a range of possible outcomes as a distribution: for example, the Gaussian, or ‘bell-curve’ distribution, provides a range of possibilities and their accompanying probabilities (shown in figure 1).

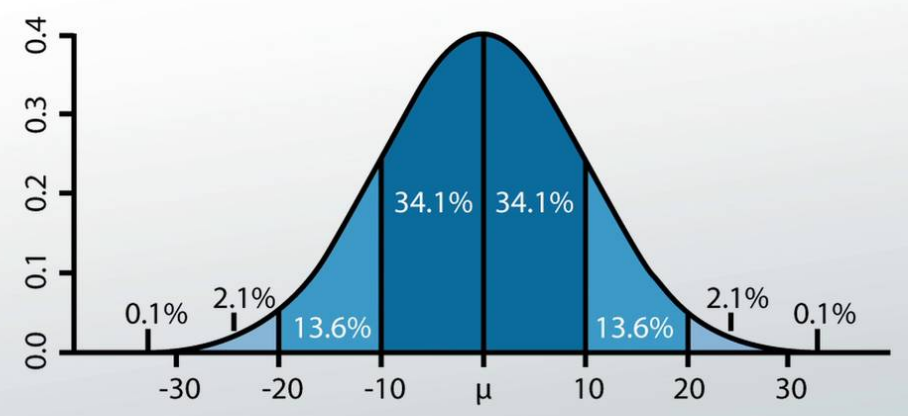

Figure 3: A classic bell-curve. The average value μ is the most likely event to occur. [6]

A mixture model is a model that uses a combination of distributions. Data is said to be multimodal when there are many regions of high probabilities (e.g. many peaks on the bell

[9] Stobierski, Tim. “What Is Statistical Modeling for Data Analysis?” *Northeastern University Graduate Programs*, Northeastern University, 11 Aug. 2021, www.northeastern.edu/graduate/blog/statistical-modeling-for-data-analysis/.

curve). Mixture models are useful for multimodal data as each mode, or collection of high probabilities, can be displayed as a distinct component of the model.

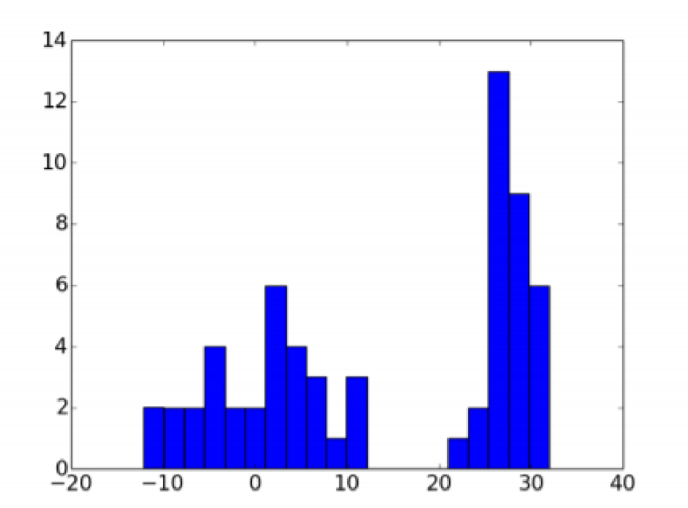

Figure 4: Multimodal data. A histogram of daily high temperatures for two different areas. There are two distinct modes.[10]

In mixture models, each data point is assumed to belong to one of the components, and the distribution of each component is independent of other components[10].

The parameter of a model is the input given. For a model to be parametric means that the parameters, given by the variable theta (Θ), are finite[11]. This is as opposed to non-parametric models, which assume the data distribution to be modeled by an infinite set of parameters[11].

[10] Grosse, Roger, and Nitish Srivastava. “Lecture 16: Mixture Models.” *Www.cs.toronto.edu*, University of Toronto , www.cs.toronto.edu/~rgrosse/csc321/mixture_models.pdf.

[11] Ghahramani, Zoubin. “Parametric vs Nonparametric Models.” *Machine Learning Summer Schools at the Max Planck Institute for Intelligent Systems*, Max Planck Institute for Intelligent Systems, Tübingen, Germany,.

2.4.2.2 Optimal Estimates

It is often helpful to find certain parameters of a population, such as the mean (average). In most situations, the population is too large to calculate the parameter; finding the average age of all american citizens would mean hundreds of millions of datums. The solution is point estimation[12]: take a few random samples from the population, and use those to calculate the desired parameter. There are many methods of point estimation. The most relevant in this instance are Maximum Likelihood Estimation (MLE) and Bayesian Optimal Estimates.

Both MLE and Bayesian Optimal estimation attempt to find the most likely model for a set of data: given different models such as a Gaussian distribution (the aforementioned bell curve), Bernoulli distributions, etc. they attempt to find the most accurate model. Note that the probability of a model is the chance of the possible outcomes while the likelihood of a model is how well it fits the data[13]. MLE posits that if one model has a higher likelihood, it also has a higher probability even though this probability is never calculated. Bayesian Optimal Estimation is more controversial, involving a representation of the probability of a model before it is calculated. Both of these methods will be used further in the paper.

[12] Laura, Simon. " Introduction to Mathematical Statistics." *Penn State Eberly College of Science*, Penn State University , online.stat.psu.edu/stat415/lesson/1/1.1.

[13] Hallinan, NM. "Likelihood: Frequentist vs Bayesian Reasoning ." *Ib.berkeley.edu*, University of California, Berkeley, 2009, ib.berkeley.edu/courses/ib200b/ib200b_2009/Lectures/Nat/Likelihood%20lecture.pdf.

2.4.3 Detailing the Statistics Present in the Specific Naive-Bayes Models

The following approach to the NB algorithms is taken from the McCallum & Nigam study[2] and re-explained here. What follows is the training process for an NB model. Note that classifiers (such as NB) are trained using data that is already classified. Given a text document generated by a mixture model parameterized by Θ, with mixture components c, such that c is in the set C = $\{c_1...c_C\}$. Each component is parameterized by a disjoint subset of Θ (disjoint indicating the subsets are unrelated to each other). Any document, d_i, is therefore generated by selecting a component according to the priors $P(C_j|\Theta)$ and having the mixture component generate a document according to its own parameters, with distribution $P(d_i|c_j;\Theta)$. The likelihood of a whole document being a particular label is then the sum of the likelihood of the words, or the sum over all mixture components.

$$\mathrm{P}(d_i|\theta) = \sum_{j=1}^{|\mathcal{C}|} \mathrm{P}(c_j|\theta)\mathrm{P}(d_i|c_j;\theta).$$

Figure 5 [2]

It's assumed there's a one-to-one correspondence between class labels and mixture model components (because the model is trained with labelled documents), so c_j is both the jth mixture component and the jth class.

Bernoulli Naive-Bayes is represented by figure 6:

$$\mathrm{P}(d_i|c_j;\theta) = \prod_{t=1}^{|V|}(B_{it}\mathrm{P}(w_t|c_j;\theta) + (1 - B_{it})(1 - \mathrm{P}(w_t|c_j;\theta))).$$

Figure 6 [7]

In a set of vocabulary, *V*, of elements *t* such that *t* ranges from {1… |V|}, each original word *w* in the body of text is assigned a unique index in the set *V*, such that any given word *w* can be represented numerically as w_t. This approach provides a means of translating words to numbers, to be worked with mathematically. *t* is represented in the equation by B_{it}, which can be either 0 or 1, indicating in binary if the word w_t occurs in that document (B_{it}=1) or does not (B_{it}=0). Note that this methodology means each word is considered individually, and the final probability of a document is the sum of these independent word-driven equations: this is the naive assumption, that the words are totally independent. Therefore the probability of an individual word is given by its component $P(w_t|c_j;\Theta)$.

When given a set of labeled training documents D such that D={$d_1,\ldots,d_D$}, learning the parameters of a model is the equivalent of estimating each of the class-conditional word probabilities (the probability of a word occuring a certain class). The mixture component's parameters are written as $\Theta_{wt|cj}=P(w_t|c_j;\Theta)$, where $0 \leq \Theta_{wt|cj} \leq 1$. Using the number of events and assuming a prior probability, the Bayesian optimal estimate is found. Additionally, Laplacian smoothing is performed, to prevent the case where a word occurs 0 times.

$$\hat{\theta}_{w_t|c_j} = \mathrm{P}(w_t|c_j;\theta) = \frac{1 + \sum_{i=1}^{|\mathcal{D}|} B_{it}\mathrm{P}(c_j|d_i)}{2 + \sum_{i=1}^{|\mathcal{D}|} \mathrm{P}(c_j|d_i)}.$$

Figure 7 [2]

In this instance, the prior probabilities are calculated using MLE:

$$\hat{\theta}_{c_j} = \mathrm{P}(c_j|\hat{\theta}) = \frac{\sum_{i=1}^{|\mathcal{D}|} \mathrm{P}(c_j|d_i)}{|\mathcal{D}|}.$$

Figure 8 [2]

In summary, the components of the mixture model (i.e. the words) are represented as independent subsets of the variable Θ. The carat above theta, '^', indicates that it is an estimate. Laplacian smoothing is performed by adding 1 to the fraction, to prevent values of 0. For any given word, w_t, the probability of an entire document based on that word, $P(w_t|c_j;\Theta)$, is equivalent to the sum (Σ) of the word's occurrence B_{it}, multiplied by the conditional probabilities of the component in the document. The end goal in this training phase is to associate certain words with certain classes.

Multinomial Naive-Bayes is represented by the following:

$$\mathrm{P}(d_i|c_j;\theta) = \mathrm{P}(|d_i|)|d_i|!\prod_{t=1}^{|V|}\frac{\mathrm{P}(w_t|c_j;\theta)^{N_{it}}}{N_{it}!}.$$

Figure 9 [2]

From the same vocabulary V, a document is a sequence of words, w, indexed by t. Document length is assumed to be independent of class, and the naive assumption is made: context/placement of words has no effect on their probability. A document, d_i, is drawn from a multinomial distribution of words, with as many trials as the length of d_i. N_{it} is the count of how many times the word w_t occurs in d_i. The Bayesian Optimal Estimate is calculated by the probability of word w_t in class c_j.

$$\hat{\theta}_{w_t|c_j} = \mathrm{P}(w_t|c_j;\hat{\theta}_j) = \frac{1+\sum_{i=1}^{|\mathcal{D}|} N_{it}\mathrm{P}(c_j|d_i)}{|V|+\sum_{s=1}^{|V|}\sum_{i=1}^{|\mathcal{D}|} N_{is}\mathrm{P}(c_j|d_i)}$$

(6

Figure 10 [2]

3. Experimental Methodology

3.1 Methodology Overview

This paper aims to investigate the research question by creating rudimentary Naive-Bayesian Bernoulli and Multinomial models via the publicly available scikit-learn packages for Python, using open-source Chinese and English datasets and running trials to find the performance of each algorithm. Chinese is a language extremely different from English grammatically, but just as commonly used on the internet [14], therefore serving as a good comparator (note Chinese was arbitrarily selected; any language that is far removed from English's syntactical structure, such as Japanese, Arabic, or Russian, would suit the needs of the study as well). NB efficacy data will be collected for both Chinese and English because of the novel nature of this research: adequate resources are not at hand to create data for only Chinese at a comparable level to the original McCallum & Nigam study, thus new data will be created for both languages, so the results are comparable in size and quality.

3.2 Preprocessing

3.2.1 Stop Words

Machine learning revolves around analysis of large datasets. Data is often in different/invalid formats, and thus before processing it must be pre-processed [15]. One component

[14] Johnson, Joseph. "Most Common Languages Used on the Internet 2020." *Statista*, Statista, 27 Jan. 2021, www.statista.com/statistics/262946/share-of-the-most-common-languages-on-the-interne.

[15] Kumar, Dhairya. "Introduction to Data Preprocessing in Machine Learning." *Medium*, Towards Data Science, 20 June 2021, towardsdatascience.com/introduction-to-data-preprocessing-in-machine-learning-a9fa83a5dc9d.

of pre-processing in text classification is the removal of stop words. Stop words are words that are extremely common, and don't necessarily indicate anything unique about a document (e.g. "the", "a", etc.). There are two approaches to removing stop words[16]: either go through all documents and find the most common words, thus creating a stop word list unique to that dataset, or use a pre-existing dictionary of common stopwords in that language. In this instance, pre-existing dictionaries were used for both the English and Chinese datasets. The dictionaries came from the same source[17], so as to remain a constant in the experiment. In the context of the English and Chinese documents, any words in the dictionaries were removed from the documents.

3.2.2 Punctuation

Another important step in preprocessing data is removing any punctuation. This is not always true for machine learning, but in this case, because a unigram model is applied (i.e. only individual words matter), punctuation is irrelevant. This step is shown in the code attached in the appendix.

3.3 Resources used

3.3.1 Datasets used

Three English datasets and three Chinese datasets were gathered. The vocabulary sizes in the datasets were somewhat unequal: the Chinese datasets' vocabulary size ranged from 3,000 to 52,000 unique words, while the English datasets' vocabulary size ranged from 46,000 to 155,000 unique words. Ideally, more datasets with equal vocabularies would be used, but adequate

[16] Teja, Sai. "Stop Words in Nlp." *Medium*, Medium, 12 June 2020, medium.com/@saitejaponugoti/stop-words-in-nlp-5b248dadad47.

[17] Genediazjr. "Stopwords-Iso/Stopwords-Iso: All Languages Stopwords Collection." *GitHub*, Sept. 18AD, 2020, github.com/stopwords-iso/stopwords-iso.

funding was not available to purchase these datasets, so open-source datasets had to be used from Kaggle, a platform for open-source data science projects.

The ‘chinese-news.csv’ file was sourced from Kaggle[18], and contains 20,378 unique entries from the Chinese Official Daily News. Each entry has a date, a tag (class), a headline, and content. This file has only 3 classes.

Two English datasets were sourced from Kaggle. The ‘merged_data_file.csv’, is composed of 46,491 YouTube videos, with 9 possible classes.. Each entry contains the video’s category, description, title, and ID. The ‘News_Category_Dataset_v2.json’ file is composed of 200,786 Huffington Post articles, with 41 classes. Each entry has a category, a headline, authors, the url, a short description, and the date. The third English dataset is a staple in NLP, and a version of it was actually used in the original McCallum & Nigam study: 20 Newsgroup. This file is contained in the sklearn module, and is composed of approximately 18000 Newsgroup posts, split in 20 classes.

Two Chinese datasets were sourced from the website Github. ‘news.json’ was taken from the news section of Toutiao Today. It has 15 categories, 10,000 entries, each with a label, label description, and a sentence. ‘long_text.json’ contains data about app descriptions, with 119 categories and 14,897 entries. Each entry has a label, label description, and sentence.

[18] Erikcabeza. “Chinese Daily News-EDA.” *Kaggle*, Kaggle, 3 Sept. 2020, www.kaggle.com/erikcabeza/chinese-daily-news-eda?scriptVersionId=39499550.

3.3.2 Libraries Used

Several native Python libraries were used: CSV, JSON, pickle, sys, collections, and unicodedata. Additionally, several external libraries were used: pandas, sklearn, pynlpir, stopwordsiso, and flashtext.

The CSV, JSON, and pickle libraries were used as a means of handling the data files: all the datasets originally came in either CSV or JSON formats. The pickle library employs the method of pickling, or marshalling, data, an efficient way to transfer large amounts of data.

Stopwordsiso was the centralized source of stopwords for Chinese and English. The flash text library provided a module containing an application of the flashtext algorithm, which is an extremely fast method of finding and replacing words or characters[19].

Pandas and sklearn (short for sci-kit learn) were used for their unique modules: pandas includes the 'dataframe' data structure, which is able to represent large arrays of data and is able to interact with sklearn, which includes several machine learning models (including NB).

3.4 Procedure

The first step was to source datasets. There were several possible sources, the primary being the Linguistic Data Consortium, which hosts many high quality datasets for the explicit intent of machine learning applications. The issue was funding; many datasets cost between 250 and 500 USD. Thus, datasets were sourced from Kaggle.

[19] Singh, Vikash. "Replace or Retrieve Keywords In Documents At Scale." *Arxiv.org*, Cornell University, 31 Oct. 2017, arxiv.org/abs/1711.00046.

Once datasets were obtained, they were preprocessed: using Visual Studio Code's Python IDE, a dedicated file was created to preprocess each dataset. In the English datasets, formatting the data was done using an inclusion by exclusion approach: a whitelist was made using the 26 letters of the English alphabet. If the characters in a certain entry were not a letter (i.e. punctuation), they were removed.. In the Chinese datasets, a blacklist was made of all possible non-valid characters: numbers, punctuation, English letters, etc. If a character in an entry was on this blacklist, it was removed. Stop words were obtained through the stopwordsiso module and filtered from all files.

Throughout the overall research process, more efficient methods were discovered. While there initially was a dedicated preprocessing file for every dataset, it was identified that it was possible to complete both the preprocessing and processing of a dataset within a single file. This made the need for the pickle module obsolete.

Preprocessing output the data as two parallel Python lists: y, a list of classes, and docs, a list of each individual word that occurred in a certain class. The data was two dimensional: for a single class, all words were combined because of the unigram model. A Pandas dataframe was used to combine the two lists into a sparse array of shape (y, X). Note that a sparse array is simply a data structure with many null entries, used in this case because many words would not occur in a given document.

Once in the sparse array, a sklearn Countvectorizer was used. This module is used to convert words into numbers. Note that the rows of the sparse array indicate the number of classes, while the columns indicate the number of unique words; each column represents a single word. Therefore, the datum at the coordinate y, X would be the number of times word X occurred in class y.

After numerization, the data was divided into a train and a test split. With supervised machine learning, it's common practice to randomly divide data, such that one component of the data is used to train the model while one component of data is used to test the model (i.e. have the model classify the data). A typical train test split of 4:1 was used, meaning that 25% of the data was tested.

Once split, the sklearn Multinomial and Bernoulli models were trained and tested and output was recorded. The sklearn train test split function allows for a random seed to be input, so that results can be replicated. Each dataset was tested with 3 different seeds to ensure accurate results.

4. Experimental Results

Full data tables are available in the appendix. The following graphs are intended to summarize trends in the data. Note that the same color indicates the same document, and the data is split into Multinomial and Bernoulli classification. Figure 11 shows the classification accuracies of Bernoulli and Multinomial NB in Chinese.

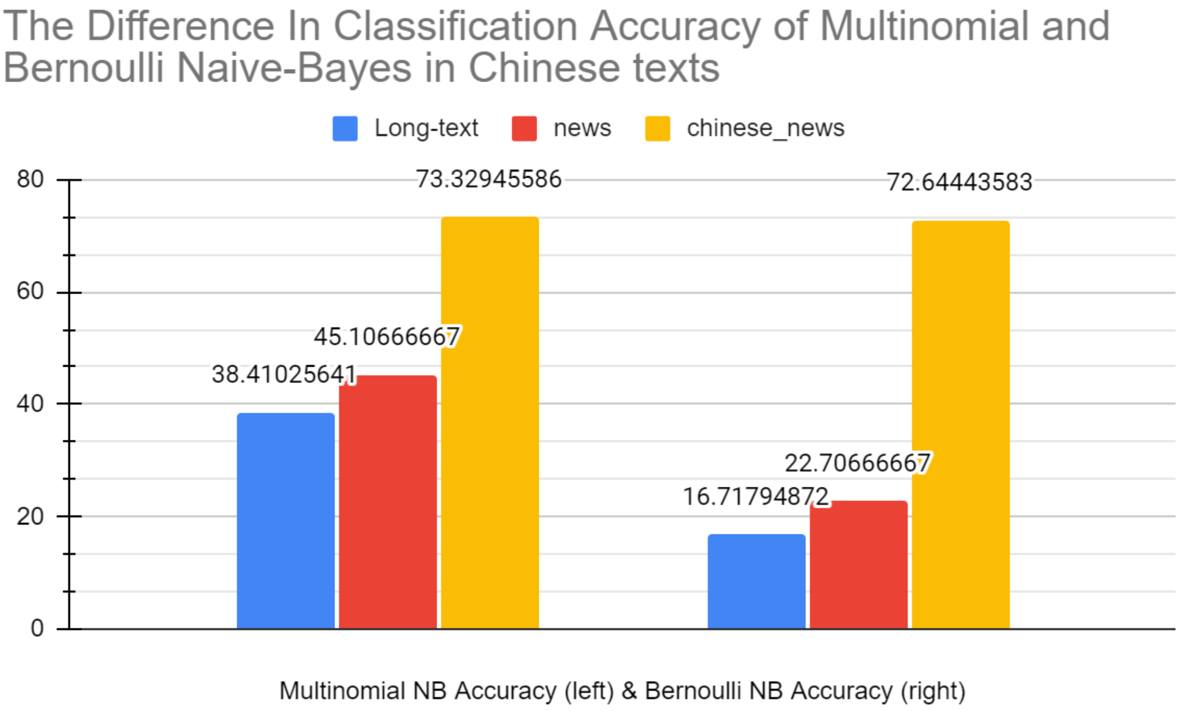

Figure 11

Figure 12 shows the classification accuracies of Bernoulli and Multinomial NB in English.

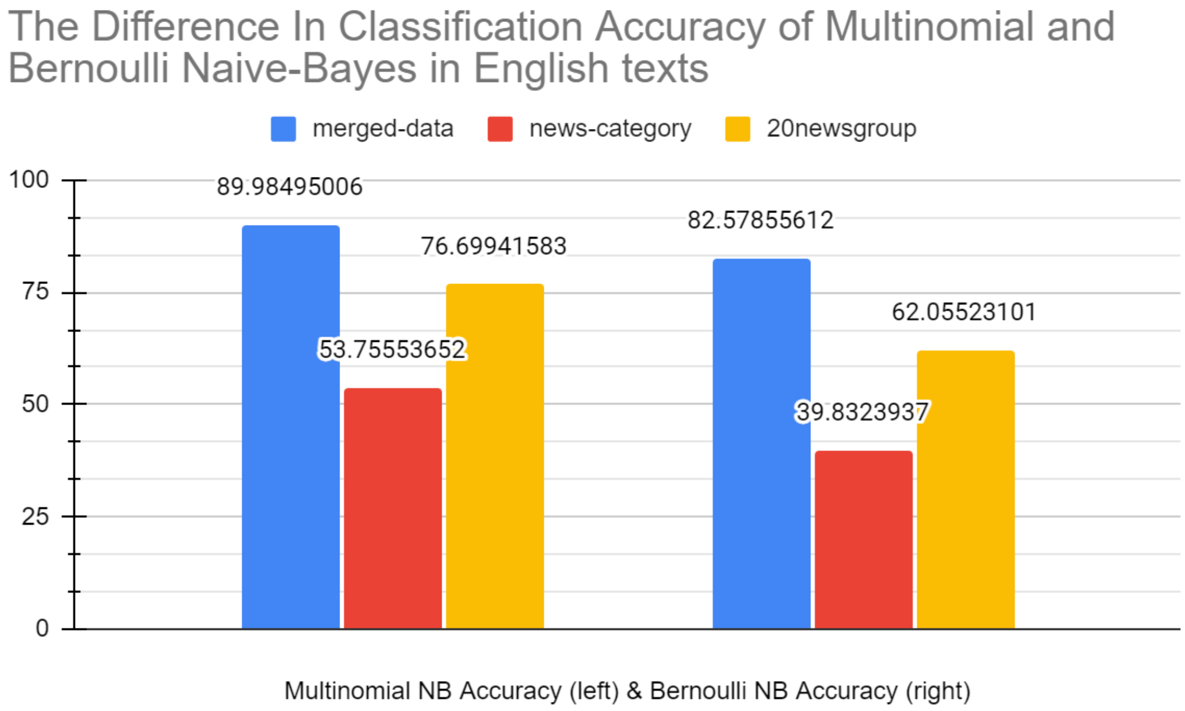

Figure 12

In both cases, the vocabulary size is arranged from least to greatest; in English merged-data (blue) contains ~46,000 words while 20newsgroup (yellow) contains ~150,000. In Chinese, long-text (blue) contains ~3,000 words, while chinese_news (yellow) contains ~52,000 words.

5. Conclusion

The data trends generally support the existing axiom that Multinomial NB performs better at higher vocabulary sizes and consistently outperforms Bernoulli NB. Given that Multinomial NB tends to perform better than Bernoulli NB, this study originally set out to ascertain whether that comparative efficacy remained constant in different languages, as different languages should have no effect because of the bag-of-words model. Figure 13 shows the difference in accuracy between Multinomial and Bernoulli NB, for each examined dataset. The Chinese datasets are in red, while the English datasets are in green.

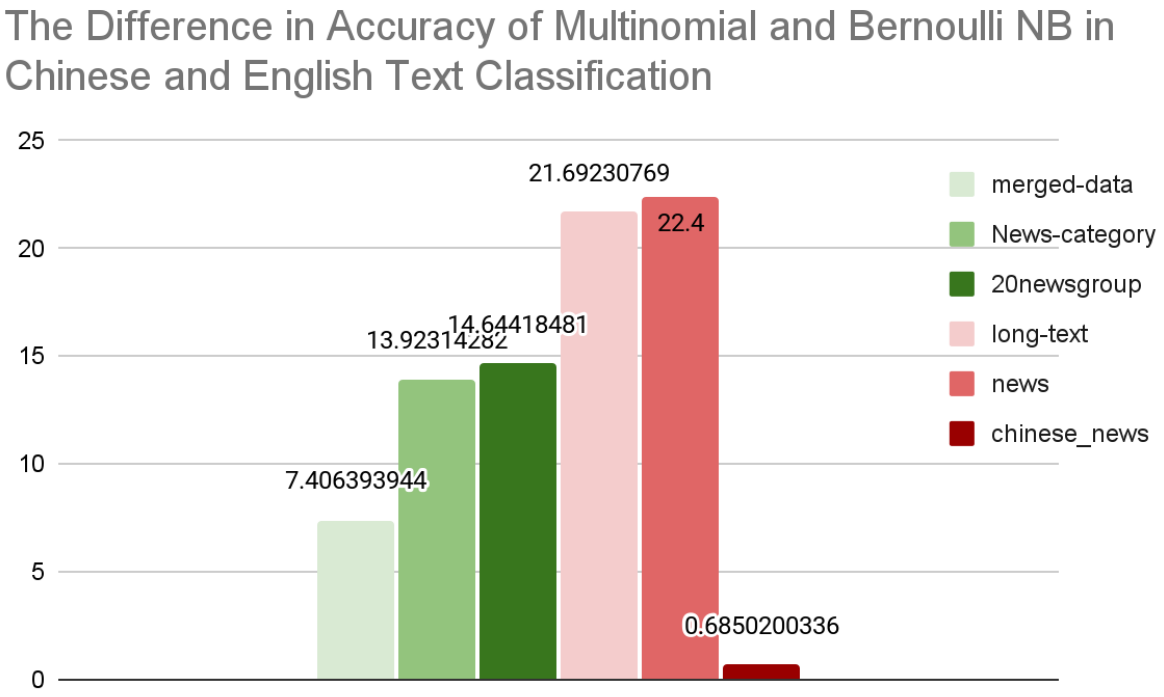

Figure 13

The average difference in Chinese is 14.9%, while the average difference in English is 11.9%. Therefore, the hypothesis that the comparative efficacies would remain consistent across languages is supported; the difference comparative efficacy between Chinese and English detours by 3%.

6. Extending Research

6.1 Limitations

The primary flaws in the current methodology is twofold: the vocabulary size of a given dataset cannot be manipulated, as in the original McCallum & Nigam study, and the vocabulary sizes are not consistent across different languages (i.e. the Chinese datasets' vocabulary was smaller than that of the English). However, the latter flaw is not significant: the aim of this study was to measure the comparative efficacy, between Multinomial and Bernoulli NB, across languages; the variable being measured was how Multinomial and Bernoulli NB compared to each other in a single dataset. The variation of vocabulary size does present an issue, though, as it's entirely possible the Bernoulli NB accuracy would increase as vocabulary size decreased.

6.2 Improvements to be Made

In the preprocessing stage of research, the process of lemmatizing was skipped due to time and capability constraints. Lemmatizing is when words that have been modified are made constant (e.g. 'run','ran','runs' all become 'run'). Additionally, the process of removing Chinese stopwords may have inadvertently removed components of proper nouns (this isn't an issue though, as all proper nouns would've been affected equally).

Higher quality and a greater quantity of data might also change the results of the research: for example, the Chinese long-text and merged-data datasets contained a label and a sentence, and the chinese_news dataset contained a headline and description. Higher quality data might include data that has more words associated with a certain label. More data might also influence the results of the study.

TF-IDF, or term frequency-document frequency, is a method used for weighting words based on how often they occur: in the current system, every word plays an equal part in determining a document's class, but in reality words that occur more often should, on an individual basis, have less influence. Thus, implementation of TF-IDF could influence results by decreasing the effect of common words that aren't part of the stop word list.

7. Works Cited

Brownlee, Jason. “A Gentle Introduction to Bayes Theorem for Machine Learning.” *Machine Learning Mastery*, 3 Dec. 2019, machinelearningmastery.com/bayes-theorem-for-machine-learning/.

Brownlee, Jason. “A Gentle Introduction to Joint, Marginal, and Conditional Probability.” *Machine Learning Mastery*, 6 May 2020, machinelearningmastery.com/joint-marginal-and-conditional-probability-for-machine-learning/.

Chauhan, Nagesh Singh. “Naïve Bayes Algorithm: Everything You Need to Know.” *KDnuggets*, June 2020, www.kdnuggets.com/2020/06/naive-bayes-algorithm-everything.html.

Erikcabeza. “Chinese Daily News-EDA.” *Kaggle*, Kaggle, 3 Sept. 2020, www.kaggle.com/erikcabeza/chinese-daily-news-eda?scriptVersionId=39499550.

Genediazjr. “Stopwords-Iso/Stopwords-Iso: All Languages Stopwords Collection.” *GitHub*, Sept. 18AD, 2020, github.com/stopwords-iso/stopwords-iso.

Glen, Stephanie. “Probabilistic: Definition, Models and Theory Explained.” *Statistics How To*, 14 Dec. 2020, www.statisticshowto.com/probabilistic/.

Grosse, Roger, and Nitish Srivastava. “Lecture 16: Mixture Models.” *Www.cs.toronto.edu*, University of Toronto , www.cs.toronto.edu/~rgrosse/csc321/mixture_models.pdf.

Hallinan, NM. “Likelihood: Frequentist vs Bayesian Reasoning .” *Ib.berkeley.edu*, University of California, Berkeley, 2009, ib.berkeley.edu/courses/ib200b/ib200b_2009/Lectures/Nat/Likelihood%20lecture.pdf.

IBM Cloud Education. “What Is Machine Learning?” *IBM*, IBM Cloud Learn Hub, 15 July 2020, www.ibm.com/my-en/cloud/learn/machine-learning.

IKONOMAKIS, M., et al. “Text Classification Using Machine Learning Techniques.” *www.researchgate.net*, WSEAS TRANSACTIONS on COMPUTERS, Aug. 2005, www.researchgate.net/profile/V-Tampakas/publication/228084521_Text_Classification_Using_Machine_Learning_Techniques/links/0c96051ee1dfda0e74000000/Text-Classification-Using-Machine-Learning-Techniques.pdf.

Laura, Simon. “ Introduction to Mathematical Statistics.” *Penn State Eberly College of Science*, Penn State University , online.stat.psu.edu/stat415/lesson/1/1.1.

Jayaswal, Vaibhav. “Laplace Smoothing in Naïve Bayes Algorithm.” *Medium*, Towards Data Science, 22 Nov. 2020, towardsdatascience.com/laplace-smoothing-in-na%C3%AFve-bayes-algorithm-9c237a8bdece.

Johnson, Joseph. “Most Common Languages Used on the Internet 2020.” *Statista*, Statista, 27 Jan. 2021, www.statista.com/statistics/262946/share-of-the-most-common-languages-on-the-interne.

Kumar, Dhairya. “Introduction to Data Preprocessing in Machine Learning.” *Medium*, Towards Data Science, 20 June 2021, towardsdatascience.com/introduction-to-data-preprocessing-in-machine-learning-a9fa83a5dc9d.

Liddy, Elizabeth D. “Natural Language Processing .” *Surface.syr.edu*, Syracuse University, 2001, surface.syr.edu/cgi/viewcontent.cgi?referer=scholar.google.com.my/&httpsredir=1&article=1019&context=cnlp.

McCallum, Andrew, and Kamal Nigam. “A Comparison of Event Models for Naive Bayes Text Classification.” *Https://Www.cs.cmu.edu*, Carnegie-Mellon University, 1998, www.cs.cmu.edu/~knigam/papers/multinomial-aaaiws98.pdf.

Ghahramani, Zoubin. “Parametric vs Nonparametric Models.” *Machine Learning Summer Schools at the Max Planck Institute for Intelligent Systems*, Max Planck Institute for Intelligent Systems, Tübingen, Germany,.

Radich, Quinn, and Eliot Cowley. “What Is a Machine Learning Model?” *Microsoft Docs*, Microsoft, 24 May 2021, docs.microsoft.com/en-us/windows/ai/windows-ml/what-is-a-machine-learning-model#:~

:text=A%20machine%20learning%20model%20is,and%20learn%20from%20those%20data.

Singh, Vikash. “Replace or Retrieve Keywords In Documents At Scale.” *Arxiv.org*, Cornell University, 31 Oct. 2017, arxiv.org/abs/1711.00046.

Stobierski, Tim. “What Is Statistical Modeling for Data Analysis?” *Northeastern University Graduate Programs*, Northeastern University, 11 Aug. 2021, www.northeastern.edu/graduate/blog/statistical-modeling-for-data-analysis/.

Teja, Sai. “Stop Words in Nlp.” *Medium*, Medium, 12 June 2020, medium.com/@saitejaponugoti/stop-words-in-nlp-5b248dadad47.

Appendix

Appendix A: English data (raw)

Multinomial	20newsgroup	merged-data	News-category
Vocabulary size	155,448	46,391	108,559
# of docs:	18,846	29,233	200,786
Classes:	20	9	41
Accuracy with Seed 1:	0.7669941582580988	0.902996305924203	0.5351315815686196
Accuracy with Seed 20:	0.7669941582580988	0.8945136133534	0.5368249098551706
Accuracy with Seed 42:	0.7669941582580988	0.9020385825694349	0.5407096041596111

Bernoulli	20newsgroup	merged-data	news-category
Vocabulary size	155,448	46,391	108,559
# of docs:	18,846	29,233	200,786
Classes:	20	9	41
Accuracy with Seed 1:	0.6205523101433882	0.832672048159803	0.39725481602486207
Accuracy with Seed 20:	0.6205523101433882	0.8150225749076481	0.39701575791381954
Accuracy with Seed 42:	0.6205523101433882	0.829662060473389	0.40070123712572464

Appendix B: Chinese data (raw)

Multinomial	chinese_news	long-text	news
Vocabulary size	52,496	2,599	14,165
# of docs:	20,631	14,897	10,000
Classes:	3	119	15
Accuracy with Seed 1:	0.7402093834819697	0.39384615384615385	0.4484
Accuracy with Seed 20:	0.7318728189220628	0.3646153846153846	0.4492
Accuracy with Seed 42:	0.7278014734393176	0.39384615384615385	0.4556

Bernoulli	chinese_news	long-text	news
Vocabulary size	52,496	2,599	14,165
# of docs:	20,631	14,897	10,000
Classes:	3	119	15
Accuracy with Seed 1:	0.7341993020550601	0.17076923076923076	0.2088
Accuracy with Seed 20:	0.7241178751454052	0.15384615384615385	0.234
Accuracy with Seed 42:	0.7210158976347422	0.17692307692307693	0.2384

Appendix C: GitHub repository with code:

https://github.com/jacobshk/Naive-Bayes-Research-V2