ASHESI UNIVERSITY COLLEGE

ANTHONY KAFUI KWAWU

UNDERGRADUATE THESIS

B.Sc. Computer Science

Anthony Kafui Kwawu

2016

ASHESI UNIVERSITY COLLEGE

Automatic Classification of News Stories – A Machine Learning Approach

UNDERGRADUATE THESIS

Undergraduate Thesis submitted to the Department of Computer Science, Ashesi University College in partial fulfilment of the requirements for the award of Bachelor of Science degree in Computer Science

Anthony Kafui Kwawu

April 2016

DECLARATION

I hereby declare that this undergraduate thesis is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

……

Candidate's Name:

……

Date:

……

I hereby declare that preparation and presentation of this undergraduate thesis were supervised in accordance with the guidelines on supervision of undergraduate thesis laid down by Ashesi University College.

Supervisor's Signature:

……

Supervisor's Name:

……

Date:

……

Acknowledgement

I would like to express my deepest gratitude to Dr. Ayorkor Korsah of Ashesi University College for all her patience and invaluable advice in fine-tuning my ideas and in my writing of this Thesis. I would also like to thank Mr. Ehizogie Binitie of Rancard Solutions Limited for his guidance, the dataset used for this study and making available a team of software engineers to answer all the questions that I had.

Abstract

Humans are good at classifying things because our brains are adept at understanding contextual nuances. Machines, however, need to be fed the right features to achieve reasonably good levels of classification. Classifying text manually is a time-consuming and expensive process especially in the information age where a combination of the success of cloud computing, big data and the resurgent trend of the internet of things as well as unprecedented population growth have led to an explosion in the amount of data that we have to deal with – approximately 2.5 quintillion bytes every 24 hours (Walker, 2015). This Thesis explores the efficiency of two well-known machine learning classification algorithms; Naïve Bayes and Support Vector Machines in classifying news stories - an important subset of the global repositories of information. The findings in this study report that using machine learning to classify news stories is not easy but is feasible and if done properly can yield accuracy rates of at least 70%. These results translate into significant time savings that cannot be achieved by manual classification and are a precursor to other machine learning techniques such as recommendation, clustering and sentiment analysis.

Table of Contents

Chapter 1: Introduction **1**

Section 1.1: Research Questions 4

Chapter 2: Literature Review **5**

Section 2.1: Relevance to Thesis 15

Section 2.2: Limitations 16

Chapter 3: Methodology & Implementation **17**

Section 3.1: Naïve Bayes 18

Section 3.2: Support Vector Machines 19

Section 3.3: Persisting the Classifier 20

Chapter 4: Results **21**

Section 4.1: Validation Score for Multinomial Naïve Bayes 21

Section 4.2: Validation Scores for Support Vector Machine 22

Chapter 5: Conclusion & Future Work **24**

References **26**

Chapter 1: Introduction

Today, there are large volumes of digitized data due to the ubiquity of the internet and the overwhelming success of cloud computing. Corporations as well as billions of ordinary people have moved from traditional sources of information such as radio and print because the internet offers an instantaneous, cheaper alternative. The conquest of data storage and processing by cloud computing has led to a further reduction in the costs of storing, maintaining and processing these caches of information – big data has ceased to be the sole prerogative of monolithic companies. Out of these volumes of data, news stories form a significant percentage and are growing increasingly relevant because, essentially, they are stories about daily occurrences in different parts of the globe and thus offer a rich narrative of the interactions between human beings and their environment. This makes news stories the popular choice for people who want to be abreast with happenings around the world; whether it is catching up on the latest sports stories, tracking Donald Trump's progress during the presidential race, keeping up to date with the European refugee crisis or savouring the latest lifestyle, technological and financial news, online news stories offer a rich and vast repository. Online news readership grew by 47% in 2004 and more than 40% in 2005. Many users have indicated online news repositories as their primary source of information (47%) compared to conventional news sources like radio (16%), TV (18%) and printed newspapers (12%) (Barthel, 2015). Additionally, other sources of information including blogs and RSS feeds also add to the rich corpus of stories available for mining. Owing to this explosion in online data consumption, it is increasingly important to be able to detect patterns in data, classify it into meaningful categories, make meaningful, tailor-made recommendations, detect outliers and make reasonable predictions - quickly. The aforementioned operations are especially valuable in commercial settings because they give corporations the ability to make better and, more often than not, split-second, decisions and

thus bestow on them, a competitive advantage. Furthermore, these enormous amounts of data resident on countless servers around the world and which are constantly being created has made it impractical to manually classify content into appropriate and relevant categories. This trend has exacerbated the already time-consuming task of placing items into distinct, recognizable categories.

Classification, in basic terms, is assigning categories to objects based on well-defined features. For instance, a vehicle may be classified as a car based on its having four wheels, a circular steering wheel and a windscreen wiper. Similarly, an animal may be classified as a cat based on such features as agility, dietary patterns and physiology. According to a survey on text classification algorithms (Aggarwal & Zhai, 2012), classification is mathematically defined when we have a set of records D = $\{X_1 \ldots X_N\}$, such that each record is assigned a class label drawn from a set of different discrete values indexed by $\{1 \ldots k)$. The training data is then used to generate a classification model which matches the features of in the underlying data to one of the class labels. In testing the algorithm, previously unseen data is fed to the classification model in order to determine if the model can correctly assign a label to the data inputs. Consequently, in the definition of our problem, the training and test records comprise summaries of curated news articles from a dataset that was provided by Rancard Solutions Limited, a Technology firm in Accra Ghana. Each article in the dataset has an appropriate class label (e.g. Sports, Business, Health etc.). Text classification has gained considerable traction because classification is a fundamental, computational activity and it has many applications in the real world: some of which include email classification and spam filtering, opinion mining, target marketing, medical diagnosis, document organization and filtering and, with especial relevance to this thesis, **news filtering and organization**. This paper seeks to produce an efficient classification algorithm, using a machine learning approach, in order to assign meaningful

categories to news stories that have been collected and curated by Rancard Solutions. The choice of a machine learning approach has as its basis, the benefits of utilizing low-cost open source tools, and algorithms that have reasonable computational processing needs. Machine learning is a key component of modelling varying forms of artificially intelligent behaviour. Two categories of learning are usually employed; supervised and unsupervised. A learning process is supervised if the inputs of the learning algorithm are well labelled, its similarities are well defined and matched to the appropriate outputs and if in general, a program can be written to handle each case of the relation (connection between input and output). Supervised learning methods are trained on datasets to match inputs to the correct outputs and then tested on new datasets. In stark contrast, unsupervised learning algorithms are used if there exists little or no information about correct outputs: the algorithm essentially has to detect patterns within the structure of the data, on its own. This is useful in instances that we cannot write programs to handle each relation. For this project, a supervised approach is pursued because the news stories data that is used for this project consists of inputs that have well-defined class labels; the algorithm will be required to learn what category a story falls in based on what class label it is associated with. **The objective of this thesis is to explore an efficient, supervised procedure to classify news stories.** Moreover, classifying news stories provides an efficient way to supply personalized content to users – the practice of overwhelming different readers who have disparate interests with unappreciated generic information is not a welcome one neither is it a commercially viable.

1.1 Research questions

1. Can a machine learning algorithm produce significant (at least 70%) rates of precision during classification?
2. Does a high precision rate go in tandem with a high recall rate?

Chapter 2: Literature Review

In their research, Masand, Linoff and Waltz (1992) describe a method to classify news story that is reliant on memory-based reasoning. Memory based techniques are modifications of nearest neighbour techniques in that new tasks are solved by looking up examples of tasks that are identical to the new task and using remembered solutions to determine its solution. Prior to implementing the solution to their problem, they noted that some previous approaches relied on creating topic definitions that require the selection of relevant phrases and words and the use of other Natural Language Processing (NLP) techniques that are meant for more than classification e.g. for extracting relational information from text. They also noted that other approaches developed statistical methods such as conditional probabilities in order to create summary representations text. However, the problem of high dimensionality of training space that is associated with statistical approaches (at least 150000 unique words) makes it hard to compute probabilities involving conjunctions or co-occurrence of features and consequently, the use of neural networks in this case is irritating at best. To better understand the context of the problem that they solved, a little background is needed. Editors of the Dow Jones assign codes to stories collected from a vast range of sources such as newspapers, magazines and press releases. Each editor must work with, at least, 350 distinct codes grouped into seven categories: industry, market sector, product, subject, government agency and region. The huge volume of stories makes manually coding them with high precision and timeliness infeasible. The features of a news story that were relevant to the memory based reasoning approach were the headline, author, and main text, amongst others. The news stories dataset used for training consisted of 49652 stories wherein, on average, each story is made of about 2700 words and 8 codes. The results reported in the paper were outputs of an n-way validation which involved excluding each test example one at a time and performing the classification on it – 1000 randomly chosen

articles were used for testing. Although the classifier performed better on code categories with a fewer number of codes, the overall recall and precision rates were 83% and 88% respectively.

In another pertinent study, (Billsus & Pazzani, 1999) designed a hybrid model for news story classification. Their research, much like this thesis, seeks to combat information overload. Moreover, it features an intelligent information agent that compiles a daily news program for a user and offers news content that is correctly classified and relevant to a specific individual. The intelligent system is capable of adapting to user's interests over time; the challenge lies in distinguishing between short-term interest and long-term interests. The designed system was incorporated into a Java applet which learns based on feedback from a user. The software keeps a score of the ratings that the user gives to different kinds of content fine-tunes the content it offers based on if the users deem the content as interesting, somewhat interesting or irrelevant. For instance, let pl be the proportion of the story that the user has read or heard, if the user indicates that the story is not interesting by clicking the appropriate button on the applet, the score = 0.3 * *pl*, if the user finds it interesting the score = 0.7 + 0.3 * pl, if the user asks for more information the score becomes 1.0. (The constants 0.7 and 0.3 are arbitrarily chosen). Obviously, news stories that are found to be uninteresting have lower scores and are not likely to be reoffered to a user. The function of the short-term model is to store information about recently rated stories so that stories that belong to the same thread of events can be identified. Secondly, the model should allow the users to identify stories that they are already familiar with. The nearest neighbour algorithm is used to achieve this by storing all the rated news stories in memory. To classify a previously-unseen instance of a news story, the algorithm compares the story to all stored stories, given some similarity measure, and determines the nearest neighbours or the nearest *k* neighbours. The class labels for this new instance can be

extracted from the class labels of its nearest neighbours. In order to correctly represent natural language text, it is converted to Term Frequency Inverse Document Frequency (TF-IDF) vectors. The TF-IDF weighting scheme that mirrors how important a word is to a document in a corpus. The cosine similarity measure is then used to compute the similarity between two vectors. After a story is converted to a TF-IDF representation, it is stored in a user model. To calculate a score prediction for a new instance, stories that are closer than a threshold *t_min* to the new instance become voting stories. The score is then computed as the weighted average over all the voting stories' scores; here the weight is the similarity between a voting story and the new story. If a voting story is closer than the threshold *t_max* to the new story, the story is labelled as *known* and its calculated score is multiplied by a factor $k<<1.0$ – the system makes an assumption that the user is familiar with the story. In a situation where the story has no voters, it cannot be classified by the short-term model, it is passed on to the long-term model. The nearest neighbour-based short-term model is able to represent a user's multiple interests and can adapt to a user's novel interests. The chief advantage of the nearest neighbour approach is that only a single topic of a new story is required to enable the algorithm determine future follow-up stories from the same thread. Furthermore, the long-term model is useful when modelling a user's general news story preferences and in generating predictions for stories that fail to be classified by the short-term model. The Naïve Bayes classifier, a probabilistic algorithm, is used in this instance. News stories are represented as Boolean features where each feature indicates the absence or presence of a word. Not all words are used as features: words that are likely to appear in commonly recurring themes are chosen. 200 words ranging from disaster, politics, sports, technology, business, crime and countries related terms are curated for this study. The naïve assumption made here are that words (features) are independent given the class label (interesting vs uninteresting), when in reality they are not. For instance, in spam filtering,

the words “erectile” and “dysfunction” are dependent on each other. The probability of a story belonging to a class *j* given a set of features, $p(class_j | f_1, f_2, \ldots, fn)$ is proportional to $p(class_j \prod_i^n p\ p(f \mid class_j)$; $p(class_j)$ and $p(f_i \mid class_j)$ can easily be appraised from training data. Precisely, a multi-variate Bernoulli event model formulation of naïve Bayes to compute Bayes-optimal estimates of $p(class_j)$ and $p(f_i \mid class_j)$ by counting the occurrences of words and class in training data. In order to prevent zero probabilities for infrequency occurring words, Laplace smoothing is utilized. This culminates in a new instance of a news story being able to be labelled with the probability of belonging to an interesting class. Furthermore if a story can neither be classified by the short-term or long-term model, a default score is assigned to it e.g 0.3 (this score, again, is arbitrarily chosen). In evaluating the performance of the model, a web-based prototype is used to collect user data. Ten users trained the data, daily, over a period ranging from 4 to 8 days: 3000 total rated news stories (300 stories, on average, per user). In addition to the classification accuracy of the system, common performance measures such as precision, recall and F-measure are considered. As regards this study, the precision describes the percentage of interesting stories that are correctly classified as interesting whereas recall is the percentage of stories that are classified as interesting. The F-measure (F_1) is a weighted combination of precision and recall that yields results ranging from 0 to 1. Mathematically,

$$F1 = \frac{2\,.precision.recall}{precision + recall}$$

The short-term model had a high precision but low recall rate, whereas the long-term model had a high recall but low precision rate. Combining the two models into a hybrid one combines the strength of the two models. This yielded a higher F-measure and, accordingly, a better classification accuracy.

Furthermore, (Chase, Genain, & Karniol-Tambour) worked on classifying the topic of news articles for which there are multiple relevant class topic labels. In addition to this, the shortcomings of a number of algorithmic approaches are discussed. A challenging factor in multi-label classification is that a news item can fall in multiple categories. For instance, a news story about the ownership of a football club can fall, simultaneously, into both sports and business categories. This task is straightforward for humans but can the same be said for computers? This study uses the publicly available New York Times dataset. A small portion of this dataset, with 9 class labels - business, arts, technology, health, style, books, home and garden, books and science - is selected. In using Naïve Bayes, let us consider the following probabilities:

Table 1.1: A list of word probabilities

Label	Article Content	Priors
(1)	hello hello world	P(1) = 0.5
(2)	greeting word	P(2) = 0.5
(3,2,1)	hello speaker	P(3) = 1/3

Due to the fact the sets above are not mutually exclusive, $p(1) + p(2) + p(3) > 1$. This makes computing prior probabilities with Naïve Bayes impractical. To circumvent this, the inclusion-exclusion principle was used to create new, mutually exclusive classes. This also allows the data to be represented as a set of non-overlapping sets – this also allows the calculation of class priors that sum to 1. However, the addition of new classes greatly increases complexity; the number of classes for consideration increases to $2^9 - 1$, which means that the runtime of the algorithm becomes exponential. Also, the addition of new classes splits the dataset across the news set of priors thereby reducing the number of

training examples to each of the new mutually exclusive priors. To counteract this, a larger dataset would have to be used for training. One method of solving this mutual exclusivity problem is to create independent binary classifiers for each class represented in the dataset. In this implementation, class labels are learned for each class and intersections between labels are disregarded. A prior is calculated for each binary classifier on the class that it classifies. Also posterior conditional probabilities are constructed for each word – this computation is done for two classes at a time and it reflects the probability that a word was drawn from an arbitrary class or from any class outside that class. The results were quite accurate even in scenarios where only the headlines of the articles were supplied. The classification model's average labelling errors across each feature class were 3.1%, 4.25% and 6.2% respectively. In spite of this, there were two major limitations of the model that called for need to pursue a different approach. The first of which is that, it is infeasible to directly modify or enhance a binary classifier. Doing this would necessitate the implementation of a more sophisticated algorithm such as a Support Vector Machine (SVM). The other is that binary classifiers do not come close to approximating the manner in which humans tackle multi-label classification. Humans are adept at assigning classes by identifying key words in text. For instance, an article with the word "Obama" is very likely to be about politics. Furthermore, the same article with additional words like "policy" and "parliament" increases the likelihood of the article being about politics. In developing a solution that relies on finding high-information words to facilitate multi-label classification, the TF-IDF weighting scheme is used. The first element (tf) the term frequency, is the number of times token *w* occurs in article a summed across all articles in a particular class.

$$\forall w, tf\ (w) = \sum_{articles\ a} \frac{\sum_{j=1}^{n} 1\{w = x_j^{(a)}\}}{\sum_{j=1}^{n} x_j^{(a)}}$$

The tf-score accounts for the frequency of a word in each article and the frequency of a word across articles in a given class, with words that appear frequently being assigned high tf-scores. The inverse document frequency measures the commonality or rarity of a particular token across the entire corpus of articles. It is calculated by dividing the total number of words in a corpus by the count of the number of instances of a particular word and then finding the logarithm of that quotient.

$$\forall w, idf\ (w) = \log\left(\frac{\sum_{articles\ a,\ \ words\ j} x_j^{(a)}}{\sum_{articles\ a,} x_w^{(a)}}\right)$$

Words that appear frequently in a given class but less so across the corpus can be construed as high-information words whereas those that appear frequently across the corpus have low information. In order to takes these two factors into account, the tf-score is multiplied by the idf-score, which gives us the tf-idf score, for each token.

$$\forall w, tfidf\ (w) = tf\ (w) * idf\ (w)$$

In testing the TF-IDF weighting scheme, the 100 most relevant (highest information) were sampled. Additionally, each time a word in the test article appears in the category that is a subset of these 100 words, the score of word is added to the total score for that category.

$$score(category\ c) = \sum_{word\ in\ w} tfidf(w)$$

The results of the test showed that when a class contains few words, high information about it can be captured. Moreover, to determine whether an article belongs to a class or not, the K-means machine learning technique was implemented. The goal of this technique is to cluster the TF-IDF scores into two categories; scores with high information and scores with low information. The integers 1 and 0 are used to predict that an article belongs to a class

with high information and low information respectively. A useful feature of the K-means technique is that reasonable predictions can be made, even for the first article, because selecting thresholds for each article is an independent activity. Moreover, the K-means algorithm, in a sense, does a relatively good approximation of the manner in which humans classify objects – Humans have the ability to project scores, independently of previously computed thresholds for articles. However, a drawback of the K-means technique is that categories with a relatively large number of articles tend to have higher tf-idf scores. This really imperils the premise of this technique - that scores are analogous; larger categories such as Business and sports report the most false positives due to this. In analyzing the incidence of false positives, it was found out that high-frequency but low-information words were the culprits. For instance, in the business articles, words such as "company", "market" and "share" occurred frequently but is obvious that they do not convincingly determine is an article should be classified as business or otherwise. In the same vein, words such as "year" and "month", which had high tf-idf scores and accounted for a considerable proportion of false positives, have insignificant semantic value. In solving this problem, the authors modified the original tfidf function to map words with low idf scores to low tfidf scores. This evenly spread the occurrences of false positives and slightly ameliorated the problem but failed to produce a model that was competitive with the binary classification model.

To add to the aforementioned research on the classification of news stories, a study conducted by (Cooley, 1999) revealed that text classification with SVMs does not need feature space reduction. In text classification, documents are often represented by a variant of the vector space model to facilitate the computation of similarity (as cited in Salton., 1971) Nevertheless, the dimensionality of this model is significantly high because the number of vector dimensions is equal to the number of unique words in a corpus. SVMs are

ideally suited for text classification problems because although text classification problems are linearly separable (as cited in Joachim, 1998) and highly dimensional, SVMs are capable of learning models whose complexities are independent of the dimensionality of the feature space. In spite of the fact that SVMs may not require feature space reduction, the huge amounts of data and the difficulty of storing and managing them feature space reduction relevant. Feature reduction methods such as information gain (as cited in Mitchel, 1996) have been proven to have the best text categorization results (as cited in Yang & Pederson, 1997). Other NLP techniques such as the identification of proper nouns can also effectively reduce feature space. Although the removal of stopwords, words that are too generic to have any influence on text – and which mostly include prepositions, definite and indefinite articles such as "the" - is not, in the truest sense, a feature reduction technique, this approach, notwithstanding, reduces the number of unique terms in a vector space. Stemming, the removal of suffixes from words, is also another approach to check feature expansion. For example the words "walking" and "walks" may be reduced to walk. Furthermore, in using support vector machines for news story classification, TF-IDF can be used as the term weighting scheme. The support vector machine was developed by Vapnik based on the structural risk minimization principle and Vapnik's own statistical theory (as cited in Vapnik, 1995). The underlying principle of a Support Vector Machine is to find the optimal separating hyperplane between positive and negative examples. Simply put, the optimal hyperplane is the line that finds the maximum margin between training examples that are closest to the hyperplane. These examples are referred to as the support vectors. In this study, six media sources were used to collate data from January to April in the year 1998; two print sources (New York Times & Associated Press Wire), two television sources (ABC World News Tonight & CNN Headline News) and two radio sources (Public Radio International and Voice of America). The results of applying the Support Vector Machine

showed that using full text without stopwords and with stemming outclassed text representations that were developed with techniques such as information gain and named entities. Also, in comparing the output of the classifier using the TF-IDF weighting scheme to using only TF weights, it was observed that the latter approach performed just as well as the former and with a lower computational cost.

Additionally, Dalal and Zaveri (2011) formulated a generic strategy for text classification. This specification can be distilled into these major phases:

1. Pre-processing the data (removal of html tags and stopwords as well lemmatization)

2. Extracting features with techniques such as the TF-IDF, Latent Semantic Indexing (LSI) and the multiword approach.

3. Choosing a machine learning algorithm such as a neural network, SVM or Naïve Bayes for classification.

4. Training a classifier

5. Testing the classifier

Moreover, this study put forward some of the challenges of text classification. The first among this is that some type of documents such as scientific papers are well-structured hence they are relatively easy to classify due to the positional information of attributes. However, a huge majority of text is unstructured and therefore have to be classified on the basis of the presence or absence of keywords. Furthermore, some words in the feature space for text classification are irrelevant. Researchers have laboured to find ways (stopword removal, TF-IDF, LSI, etc.) to overcome two fundamental problems in text classification: polysemy (one word having different meanings) and synonymy (different words having the same meaning). These semantics-oriented techniques do not only alleviate the

aforementioned problems but help us extract more contextual meaning from text. Additionally retrieving metadata such as keywords, proper nouns, document titles, names of author(s) and places are important but difficult in classification; web pages conveniently have META tags to help with metadata extraction. A decision tree-based model has also been formulated to address the spatial and contextual extraction of metadata (Changuel, 2009). In concluding the study, it was noted that there is a cornucopia of machine learning algorithms with none being superior to another. Naïve Bayes classifiers are founded, ingenuously, on the conditional independence among attributes perform poorly when there are a small number of attributes, however, they are purely statistical models and hence have significantly lower learning times. SVMs perform better with categories that are defined by low-information features but are ideally suited for binary classification problems. Decision trees take into account the dependence between words in a corpus but are do not perform well when the number of features increase (Li & Belford, 2002). The best approach is hybrid one that combines the strengths of all these algorithms.

2.1 Relevance to Thesis

The approaches above describe ways to achieve my research objective and answer my research questions. The solution offered by Bilisus & Pazzani. fits into the context of my problem because it addresses the importance of modifying the learning algorithm over time. Chase et al. also worked on multi-label classification which is at the heart of this project. The Rancard dataset for this project is composed of these categories: business, sports, house and healthy living. Using a Naive Bayes classifier provides sufficient results for a training corpus with multi-labels.

2.2 Limitations

Binary classifiers for multi-label classification are infeasible to modify over time and do not fit into the context of commercializing a classifier. The datasets from Rancard Solutions are fluid and constantly evolving and in this regard, Naïve Bayes binary classifiers are not the optimal solution.

In the study by Bilisus & Pazzani, the intelligent agent is limited in that n-fold cross validation techniques cannot be applied. The chronological order of the data used for training cannot be randomly fed to the training algorithm without unfairly skewing the results. N-fold cross validation is important as it ensures that the different categories of articles are evenly distributed during training, an important step in measuring how good a classifier is. Moreover, a user's interactions with news content cannot be assumed as static or unswerving. The same user can assign completely different labels to the same set of stories at different points in time in the same day. Furthermore, the high dimensionality of data makes memory based techniques highly expensive and leads to the occurrences of a significant number of false positives in statistical methods such as Naïve Bayes classifiers. Hybrid approaches should in theory be the best way forward but they require great effort to implement. Support vector machines, although ideal for binary classifications, are easier to fine-tune over time and handle high dimensionality extremely well. They also do not require complex feature space reduction techniques. Dalal and Zaveri's work is the most relevant to this thesis because it outlines an intuitive format for approaching text classification. The steps in this study are well defined and straightforward.

Chapter 3: Methodology

From the related works in literature that have been reviewed in chapter two, two common supervised machine learning algorithms (classifiers) are going to be implemented: the Naïve Bayes and support vector machine classification algorithms. Some of the strengths and limitations of these approaches have been discussed above. On a high level, the approach to solving this problem can be represented by this diagram:

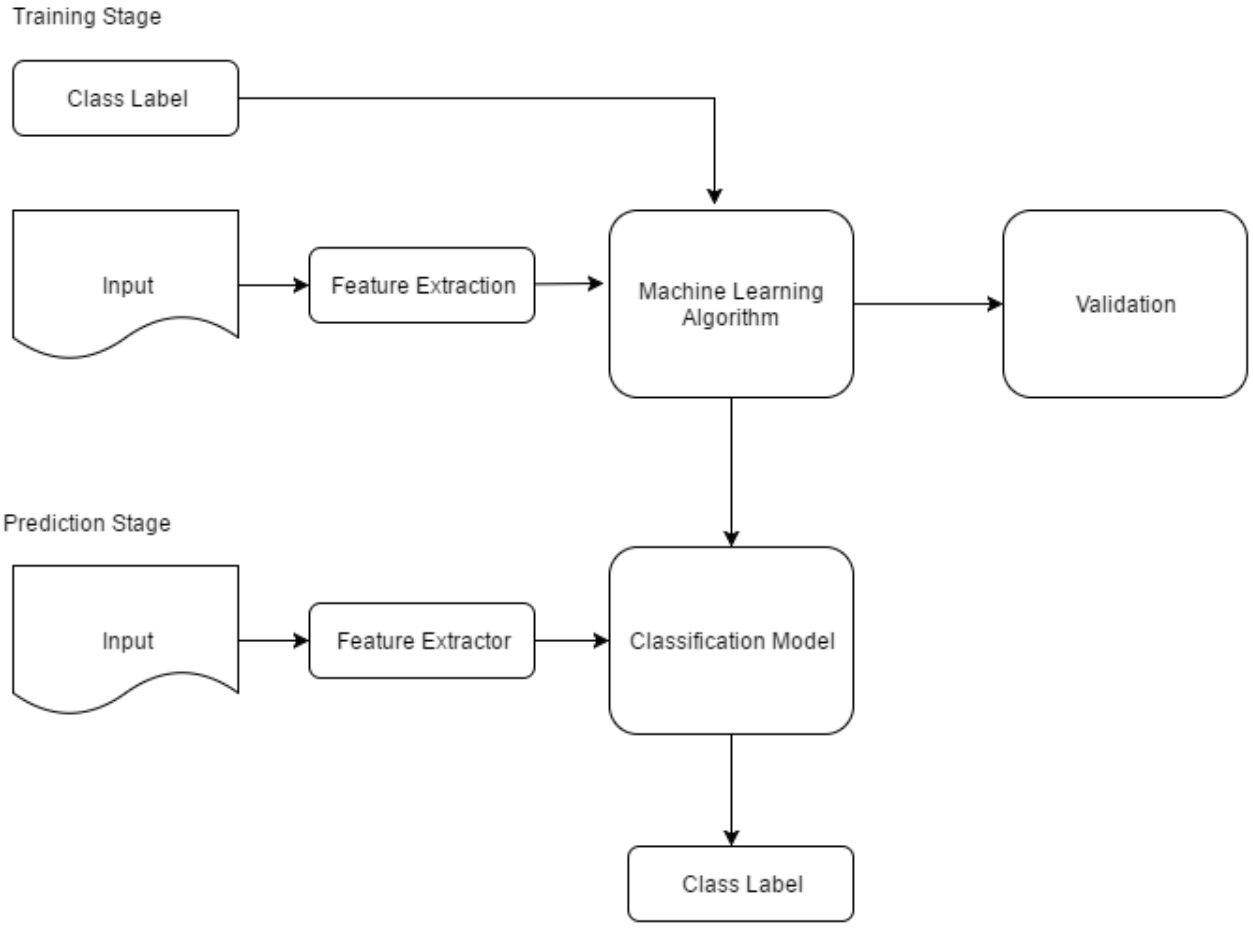

Figure 3.1: High-level Architecture

These classifiers were sourced from Python's extensive collection of the Scikit learn machine learning library. The key components of the implementation of the classification of the Rancard news stories dataset are fairly simple. There are three key stages – creating a data model (in this study, a bag of words approach is used) that represents the news articles,

training the model and performing a cross validation to assess the performance of the classifier. High precision and recall scores during the validation stage indicate that the classifier can be used, with reasonable success, to classify previously unseen data. Before any significant computation was conducted, the data, which was in a csv format, was read and stored using the python pandas text processing library. Furthermore, to create the data model, the first action that was undertaken was cleaning the dataset; a process that ensures that extraneous characters such as HTML tags and punctuation are removed from the data. After ensuring that the data was clean, it was tokenized. The tokenization was achieved by using Python's Textblob text processing library. Stop words were then expunged in order to reduce the size of the feature space. Lemmatization then takes place in order to reduce samples of text to their root form. This was important to prevent different forms of the same word from unnecessarily increasing the feature space – as discussed earlier Naïve Bayes classifiers do not particularly excel when there are a high number of dimensions in data. Stop word removal and lemmatization were achieved with python's nltk, NLP library Next the data is weighted and normalized with the TF-IDF algorithm. After the data is processed and converted into the aforementioned model, it is fed into the classifier, the specific choice of classifier is the Scikit- Learn multinomial Naïve Bayes classifier and the Linear Support Vector Machine Classifier.

3.1 Naïve Bayes

The Naïve Bayes approach, as described earlier, assumes that the features in a corpus are conditionally independent off one another given some class. Mathematically, it involves the computation of observing features f_1 through f_n given a class such as Business, sports, house or healthy living:

$$p(f_1, \ldots, f_n|c) = \prod_{i=1}^{n} p(f_i\,|c)$$

This makes it convenient to work with posterior probabilities:

$$p(c|f1, \ldots, fn) \propto p(c)p(f1|c) \ldots p(fn|c)$$

The Multinomial Naïve Bayes is a variant of the basic Naïve Bayes model that simply tells us that *p(fi|c)* is a multinomial distribution, rather than some another distribution such as the Gaussian distribution (Norvig & Russel, 2003).

For example in choosing if a class belongs to the business category or sports category, these probabilities are considered.

p(Business | word₁ word₂ ...) & p(Sports | word₁ word₂ ...)

The key concept here is that if *p(Business | word₁ word₂ ...) > p(Sports | word₁ word₂ ...)* the article is more likely to be an about business. This concept is at the heart of the maximum a posteriori rule (MAP). The assumption of independence of words, although mostly false, produces astonishing results. Naïve Bayes classifiers have performed well in spite of this erroneous conjecture.

3.2 Support Vector Machine (SVM)

Intuitively, an SVM is a linear separator that finds the maximum margin between two classes (For example, Business and Sports) that are being considered for classification. In situations where the data points are such that a curved line is required to separate them, curves are not drawn but rather the features are "lifted" into higher dimensions. For instance, in the picture below if a line cannot be drawn in the space (x1, x2) to separate articles

belonging to business from those belonging to sports, a third dimension (x1, x2,x1 * x2) is added. If this separating, higher-dimensional line, which is referred to as a hyperplane, is projected into the previous dimension, it assumes the shape of a curve. This complexity as cited in the literature review, is achieved with the Kernel trick. The optimal separating line is concerned only with the points closest to it – the support vectors.

3.3 Persisting the SVM Classifier

In this implementation, the scikit-learn Linear SVM library was used (Pedregosa, Varoquaux, Gramfort, & Michel, 2011). This library has a runtime of $O(n^4)$ hence, the Rancard Solutions news story dataset used for this study, which has a combined validation set (training and testing data separate from the 14702 rows used for the final testing stage) of 39136 rows (approximately 391360 features), was serialized and stored to a disk using pythons Pickle library in order to make future testing much quicker (less than 2997.6 seconds that this implementation took). The goal of this is to learn the classifier once so that it can be used multiple times without having to re-learn the classifier every time an experiment or a test is carried out. The runtime of the algorithm and the large amount of features means that persisting the classifier saves time.

Chapter 4: Results

In order to advance to the testing phase, the two implementations had to pass the validation phase. To ensure that the algorithm produced accurate results, approximately 27% of the dataset which consisted of 53838 rows (with three columns representing the summary of the article, its class label and publisher), was reserved for the final testing phase. The rest of the data was randomly split into a validation dataset whereby 80 % of the data represented training samples and 20 %, testing. N-fold cross validation was then implemented to ensure that the class labels were fairly represented irrespective of the differences in the number of articles in each news story category. These experiments were conducted with the SVM because the Naïve Bayes Classifier failed an initial validation phase after 80% of all the data was used in training and the rest in testing – it was counterintuitive to advance it to the testing stage.

4.1 Validation Scores for Multinomial Naïve Bayes Classifier

Table 4.1: Confusion Matrix for Multinomial Naïve Bayes Classifier

	Business	Healthy Living	House	Sport
Business	4949	0	1	119
Healthy Living	199	24	0	22
House	55	0	409	11
Sport	55	0	0	4924

Table 4.2: Classification Report for Multinomial Naïve Bayes Classifier

	Precision	Recall	F1-score	Support
Business	0.94	0.98	0.96	5069
Health Living	1.00	0.10	0.18	245
House	1.00	0.86	0.92	475
Sport	0.97	0.99	0.98	4975
Avg / Total	0.96	0.96	0.95	10768

The classification report above shows that, although the Multinomial Naïve Bayes classifier had, on average, high rates of precision, it had low recall rates. The overwhelming presence of business and sports articles further validated the theory that a simple Naïve bayes classifier is not sufficient for multi-label classification. The exception to this, is to create a one-versus-all classifier for each class. However, the limitations of this approach have been discussed in the literature review.

4.2 Validation Scores for Multinomial Support Vector Machine

Out of the 39136 rows, 31308 were used during the validation process as training data. The remaining 7828 were used to test the trained model. Intuitively, the SVM was expected to perform better than the previously implemented probabilistic approach. These were the results after validation:

Table 4.3: Classification Report for SVM after Validation

	Precision	Recall	F1-score	Support
Business	0.96	0.98	0.97	3658
Health Living	0.89	0.75	0.81	237
House	0.99	0.95	0.97	334
Sport	0.98	0.98	0.98	3599
Avg / Total	0.97	0.97	0.97	7828

4.3 Testing

Although, the SVM perform significantly better by achieving higher rates of recall and precision, the validation scores did not reflect a real world test. To ensure that it did, the classifier was used to classify previously unseen data and these were the scores:

Table 4.4: Classification Report for SVM after Testing

	Precision	Recall	F1-score	Support
Business	0.98	0.95	0.96	7000
Health Living	0.72	0.78	0.75	201
House	0.97	1.00	0.99	501
Sport	0.96	0.98	0.97	7000
Avg / Total	0.96	0.96	0.96	14702

These results show lower recall and precision rates than was observed during validation but these results were significantly better than the Naïve Bayes Approach and met the objective of this study.

Chapter 5: Conclusion & Future Work

In the pursuit exploring an efficient classification algorithm for the News Articles Dataset, the objective of this Thesis was met and the research questions were answered. Precision and recall rates do not always change proportionately and the choice of a classification algorithm greatly influences that change. The implications of the findings in this study are phenomenal in that the amount of time required to manually classify news stories has been significantly reduced. More time can now be spent on generating more revenue for not only Rancard Solutions Limited but any other corporation that elects to implement the solutions that have been uncovered in these findings. More importantly, however, this research has offered a way to deal with classifying the growing size of information that is being generated around the world today especially given that the latest technological trends such as the Internet of Things and Big Data are gaining considerable traction. These technologies are going to spearhead a never-before-seen interconnectedness with machines and humans that will contribute to the production of data on a huge scale. This means that if there are no efficient ways to classify that data, corporations will miss out on time savings and a better management their storage capacities. Additionally, the approach described in this paper describes a classification algorithm, but classification algorithms are fundamental precursors to other important machine learning techniques that add to our understanding of data. Any individual or corporation can build on this work to exploit those other techniques in lieu of starting from scratch. The future of this project involves using more techniques such as LSI to optimize the feature space and enhance the way the algorithm deals with different contexts and the presence of synonymy and polysemy. Furthermore, machine learning algorithms all have strengths and weaknesses hence pursuing a hybrid approach to news classification makes the most sense. In this case, a Neural Network -SVM hybrid approach will be the best way forward to take advantage of

the power of novel database systems such as graph databases that are in use at corporations such as Rancard Solutions and to move a step closer to replicate the brain's process of object classification.

References

Aggarwal, C., & Zhai, C. (2012). *Mining Text Data.* New York: Springer.

Barthel, M. (2015, April 29). *Newspapers: Fact Sheet*. Retrieved from Pew Research Center - Journalism & Media: http://www.journalism.org/2015/04/29/newspapers-fact-sheet/

Billsus, D., & Pazzani, M. (1999). A Hybird User Model for News Story Classification. *CISM International Centre for Mechanical Sciences UM99 User Modeling*, 99-108.

Changuel, S. (2009). A General Learning Method for Automatic Title Extraction from HTML Pages. *Sixth International Conference on Machine Learning and Data Mining*.

Chase, Z., Genain, N., & Karniol-Tambour, O. (n.d.). Learning Multi-Label Topic Classification of News Articles.

Cooley, R. (1999). Classification of News Stories Using Support Vector Machines. *Sixteenth International Joint Conference on Artificial Intelligence*.

Dalal, M., & Zaveri, M. (2011). Automatic Text Classification: A Technical Review. *International Journal of Computer Applications IJCA, 28*(2), 37-40. doi:10.5120/3358-4633

Joachims, T. (1998). Text categorization with Support Vector Machines: Learning with many relevant features. *Machine Learning: ECML-98 Lecture Notes in Computer Science,* 137-142. doi:10.1007/bfb0026683

Li, R.-H., & Belford, G. (2002). Instability of decision tree classification algorithms. *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '02*. doi:10.1145/775047.775131

Masand, B., Linoff, G., & Waltz, D. (1992). Classifying News Stories Using Memory Based Reasoning. *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '92*. doi:10.1145/133160.133177

Mitchell, T. (1997). *Machine Learning.* New York, USA: McGraw-Hill.

Norvig, P., & Russel, S. (2003). *Artificial Intelligence: A Modern Approach.* Pearson.

Pedregosa, F., Varoquaux, G., Gramfort, A., & Michel, V. (2011). Scikit-learn: Machine Learning in Python. *JMLR 12*, 2835 - 2830.

Salton, G. (1971). *The SMART Retrieval System—Experiments in Automatic Document Processing.* New Jersey, USA: Prentice Hall.

Vapnik, V. (1995). Constructing Learning Algorithms. *The Nature of Statistical Learning*, 119-116. doi:10.1007/978-1-4757-2440-0_6

Walker, B. (2015, April 5). *Every Day Big Data Statistics - 2.5 Quintillion Bytes of Data Created Daily*. Retrieved from Virtualization & Cloud News: http://www.vcloudnews.com/every-day-big-data-statistics-2-5-quintillion-bytes-of-data-created-daily/

Yang, Y., & Pederson, J. (1997). A Comparative Study on Feature Selection in Text Categorization. *In Proceedings of the Fourteenth International Conference on Machine Learning (ICML '97)*, 412-420.