Journal of Information Engineering and Applications
ISSN 2224-5782 (print) ISSN 2225-0506 (online)
Vol.5, No.8, 2015

N-gram Based Text Categorization Method for Improved Data Mining

Kennedy Ogada*, Waweru Mwangi, Wilson Cheruiyot
School of Computing and Information Technology, Jomo Kenyatta University of Agriculture and Technology, P. O. Box 62000-00200 Nairobi, Kenya

Abstract
Though naïve Bayes text classifiers are widely used because of its simplicity and effectiveness, the techniques for improving performances of these classifiers have been rarely studied. Naïve Bayes classifiers which are widely used for text classification in machine learning are based on the conditional probability of features belonging to a class, which the features are selected by feature selection methods. However, its performance is often imperfect because it does not model text well, and by inappropriate feature selection and some disadvantages of the Naive Bayes itself. Sentiment Classification or Text Classification is the act of taking a set of labeled text documents, learning a correlation between a document's contents and its corresponding labels and then predicting the labels of a set of unlabeled test documents as best as possible. Text Classification is also sometimes called Text Categorization. Text classification has many applications in natural language processing tasks such as E-mail filtering, Intrusion detection systems, news filtering, prediction of user preferences, and organization of documents. The Naive Bayes model makes strong assumptions about the data: it assumes that words in a document are independent. This assumption is clearly violated in natural language text: there are various types of dependences between words induced by the syntactic, semantic, pragmatic and conversational structure of a text. Also, the particular form of the probabilistic model makes assumptions about the distribution of words in documents that are violated in practice. We address this problem and show that it can be solved by modeling text data differently using N-Grams. N-gram Based Text Categorization is a simple method based on statistical information about the usage of sequences of words. We conducted an experiment to demonstrate that our simple modification is able to improve the performance of Naive Bayes for text classification significantly.
Keywords: Data Mining, Text Classification, Text Categorization, Naïve Bayes, N-Grams

1. Introduction

Automated text classification has been considered as a vital method to manage and process a vast amount of documents in digital forms that are widespread and continuously increasing. There is explosive growth of available information contained in digital documents through online social networks. Naive Bayes is based on Bayes' theorem and an attribute independence assumption. Growing amount of information available on the Internet is raising interest in automatic analysis of text data. Using machine learning techniques on text data (text-learning) is opening many important and interesting research problems (Mladenic and Grobelnik, 1988). Because of the variety of languages, applications and domains, machine learning techniques are commonly applied to infer a classification model from example documents with known class labels. The inferred model can then be used to classify new documents. The automated categorization (or classification) of texts into predefined categories has witnessed a booming interest in the last 10 years, due to the increased availability of documents in digital form and the ensuing need to organize them (Karl-Michael, 2005). Machine learning (ML) is defined as a paradigm according to which a general inductive process automatically builds an automatic text classifier by learning, from a set of preclassified documents, the characteristics of the categories of interest. The advantages of this approach are accuracy comparable to that achieved by human experts, and a considerable savings in terms of expert labor power, since no intervention from either knowledge engineers or domain experts is needed for the construction of the classifier or for its porting to a different set of categories (Fabrizio, 2002).

Typically, most data for genre classification are collected from the web, through newsgroups, bulletin boards, and broadcast or printed news. They are multi-source, and consequently have different formats, different preferred vocabularies and often significantly different writing styles even for documents within one genre (Ikonomakis et al, 2005). "Text mining" is mostly used to represent all the tasks that, by analyzing large quantities of text and identifying usage patterns, try to extract probably helpful (although only probably correct) information. Concentrating on the above opinion, text categorization is an illustration of text mining. The main aim of text categorization is the classification of documents into a fixed number of predetermined categories. Every document will be either in multiple, or single, or no category at all. Utilizing machine learning, the main purpose is to learn classifiers through instances which perform the category assignments automatically. This is a monitored learning problem. The first step in text categorization is to transform documents, which typically are strings of characters, into a representation apt for the learning algorithm and the classification task (Durga & Venu, 2012). Web documents exhibit some specific properties which may require some adjustment or use of proper learning algorithms. The document features inherit some of the properties of the natural language text

from which they are derived. Many features are irrelevant to the classification task and they are often correlated. This violates two basic assumptions that many learning algorithms rely on: the equal importance of all attributes and their statistical independence (Zdravko & Daniel, 2007).

Formally, Fabrizio (2002) define Text Categorization as the task of assigning a Boolean value to each pair $(d_j, c_i) \in D \times C$, where D is a domain of documents and $C = \{c_1, ..., c_{|C|}\}$ is the set of predefined categories. A value of T assigned to (d_j, c_i) indicates a decision to file d_j under c_i, while a value of F indicates a decision not to file d_j under c_i. More formally, the tasks is to approximate the unknown target function $\overset{v}{\Phi} : D \times C \to \{T, F\}$ (that describes how documents ought to be classified) by means of a function $\Phi : D \times C \to \{T, F\}$ called the classifier (aka rule, or hypothesis, or model) such that $\overset{v}{\Phi}$ and Φ "coincide as much as possible. Ramasundaram and Victor (2013) define a classifier as a function that maps an input attribute vector, $x = (x_1, x_2, x_3, ...x_n)$, to a confidence that the input belongs to a class- that is, $f(x) = confidence(class)$. In case of text classification, the attributes are words in the document and the classes correspond to the text categories.

The objective of text classification systems is to create a mapping (also called a model or hypothesis) between a set of documents and a set of class labels. This mapping is then used to determine automatically the class of new (unlabelled) documents. The general framework is usually called supervised learning (also, learning from examples, concept learning) and includes the following steps (Zdravko & Daniel, 2007):

- Step 1: Data Collection and Preprocessing. At this step, documents are collected, cleaned, and properly organized, the terms (features) identified, and a vector space representation created. Documents are labeled according to their topic, user preference, or any other criterion that may be used to organize documents in classes (categories). At this step the data may be divided into two subsets:
  - Training set. This part of the data will be used to create the model. In some cases this set is then split into two: the actual model construction subset and a model validation subset needed to tune the learner parameters.
  - Test set. This part of the data is used for testing the model.
- Step 2: Building the model. This is the actual learning (also called training) step, which includes the use of the learning algorithm. It is usually an iterative and interactive process that may include other steps and may be repeated several times so that the best model is created:
  - Feature selection
  - Applying the learning algorithm
  - Validating the model (using the validation subset to tune some parameters of the learning algorithm).
- Step 3: Testing and evaluating the model. At this step the model is applied to the documents from the test set and their actual class labels are compared to the labels predicted. At this step, the document labels are used for evaluation only, which is not the case at the earlier validation step, where the class labels are actually used by the learning algorithm.
- Step 4: using the model to classify new documents (with unknown class labels).

The rest of the paper is organized as follows: Section 2 describes Naïve Bayes text classifier and its limitations; section 3 introduces the concept of n-grams. In section 4, we analyze relevant research which has been done about applications of n-grams, and section 5 presents experiment and data results. Finally, section 6 gives the conclusion.

2. Naïve Bayes

Many text classifiers use machine learning techniques. Naive Bayes is based on Bayes' theorem and an attribute independence assumption. Statistical processing based on the Bayes decision theory is a fundamental technique for pattern recognition and classification. It is based on the assumption that the classification of patterns (the decision problem) is expressed in probabilistic terms. The statistical characteristics of patterns are expressed as known probability values that describe the random nature of patterns and their features. These probabilistic characteristics are mostly concerned with a priori probabilities and conditional probability densities of patterns and classes. The Bayes decision theory provides a framework for statistical methods for classifying patterns into classes based on probabilities of patterns and their features (Krzysztof et al, 2007).

Ronen & James (2006) describe Bayes theorem as follows. Probabilistic classifiers view the Categorization Status Value CSV (d, c) as the probability $P(c \mid d)$ that the document d belongs to the

category c and compute this probability by an application of Bayes' theorem (Eq. 1):

$$P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)} \quad \text{.....(1)}$$

The marginal probability $P(d)$ need not ever be computed because it is constant for all categories. To calculate $P(d \mid c)$, some assumptions have to be made about the structure of the document d. With the document representation as a feature vector $d = (w_1, w_2, ...)$, the most common assumption is that all coordinates are independent, and thus (Eq. 2)

$$P(d \mid c) = \prod_i P(w_i \mid c) \quad \text{.....(2)}$$

The classifiers resulting from this assumption are called Naïve Bayes (NB) classifiers. They are called "naïve" because the assumption is never verified and often is quite obviously false (Ronen & James, 2006). Although the naive Bayes is frequently used in text classification, there exist a lot of problems to improve. One systemic problem is that the training sample may not be uniform, some researchers call this occasion as skewed data, when one class has more training examples than another, or the training documents in one class are longer than another, naive Bayes usually prefers one class over the other. Another problem is that the naive Bayes assumption ignores the relationship between words, meanwhile the parameter estimation is too rough (Yuqian et al., 2011).

3. N-Grams

Bag-of-words is the most widely used model for text representation. The method consists of counting word occurrences in the text. This produces a vector in the space of features, which correspond to the words found in the text (Artur et al, 2007).

Generally, information about sentiment is conveyed by adjectives or more specifically by certain combinations of adjectives with other parts of speech. Models that assign probabilities to sequence of words are called language models (LMs). The simplest model that assigns probabilities to sentences or sequence of words is called N-gram. Whether estimating probabilities of next words or the whole sentence, the n-gram model is one of the most important tools in speech and language processing (Daniel and James, 2014). Bing (2007) defines an n-gram as simply a consecutive sequence of words of a fixed window size n. For example, the sentence, "John went to school with his brother," can be represented with five 3-gram phrases "John went to", "went to school", "to school with", "school with his", and "with his brother". Note that 1-gram is simply the individual words N-grams have been used in duplicate detection in the Web. There are different types of duplication of pages and contents on the Web. Copying a page is usually called duplication or replication, and copying an entire site is called mirroring. Duplicate pages and mirror sites are often used to improve efficiency of browsing and file downloading worldwide due to limited bandwidth across different geographic regions and poor or unpredictable network performances. Of course, some duplicate pages are the results of plagiarism. Detecting such pages and sites can reduce the index size and improve search results. One efficient duplicate detection technique is based on N-grams (also called Shingles) (Bing 2007).

Let $S_n(d)$ be the set of distinctive n-grams (or shingles) contained in a document d. Each n-gram may be coded with a number or a MD5 hash (which is usually a 32-digit hexadecimal number). Given the n-gram representations of two documents d_1 and d_2, $S_n(d_1)$ and $S_n(d_2)$, the Jaccard coefficient can be used to compute the similarity of the two documents (Eq. 3):

$$Sim(d_1, d_2) = \frac{| S_n(d_1) \cap S_n(d_2) |}{| S_n(d_1) \cup S_n(d_2) |} \quad \text{.....(3)}$$

A threshold is used to determine whether d_1 and d_2 are likely to be duplicates of each other (Bing 2007).

Some researchers define n-gram differently. Canvar and Trenkle (1994) define an N-gram as an N-character slice of a longer string. Term can include the notion of any co-occurring set of characters in a string (e.g. an N-gram made up of the first and third character of a word). Artur et al (2007) define n-gram as a sequence of nonconsecutive characters that appear in the text. For example, the string "new shipment" yields the following set of 3-grams: "new", "ew ", "w s", " sh", "shi", "hip", "ipm", "pme","men", "ent".

4. Related Work

N-grams, as a language model, have been used in a variety of applications. N-grams were first mentioned by

Shannon in 1948 in a context of theory of communication. Since then, n-grams have found their use in many applications such as spell checking, string searching, prediction, and speech recognition (Artur et al, 2007).

Christian et al (2009) investigated the utility of n-gram analysis in predicting operator sequences in plans. Given a set of sample plans, they performed n-gram analysis to predict the likelihood of subsequent operators, relative to a partial plan. They identified several ways in which this information might be integrated into a planner. They investigated one of these directions in further detail. Preliminary results demonstrated the promise of n-gram analysis as a tool for improving planning performance. N-gram analysis provides a means of probabilistically predicting the next item in a sequence. Due originally to Shannon, it has proven an effective technique for word prediction in natural language processing and for gene sequence analysis. They divided the presentation of their approach in two parts. The first corresponded to the learning phase, in which they extracted statistical information from a set of plans, and then compiled this information into patterns that can be exploited by the planner. The second phase was the planning phase.

Show-Jane et al (2010) researched on automatic Chinese text classification using n-gram model which works with letters level feature. They proposed an N-gram-based language model for Chinese text classification which considers the relationship between words. To prevent from the out-of-vocabulary problem, they proposed a novel smoothing method based on logistic regression to improve the performance. They also proposed a novel feature selection which fits N-gram model for Chinese text classification. Their experimental result showed that their approach outperformed the previous N-gram-based classification model above 11% on micro-average F-measure.

Thomas et al (2015) carried out an experiment on web narratives of breast cancer patients as a valuable source of information for the study of health-related quality of life. They focused on posts from a web forum related to breast cancer. Based on a large descriptor set including n-grams, temporal expressions and forum metadata, they first conducted descriptor pre-selection before developing a supervised classifier for identifying the following three features: 1) speaker: the role of a post's author (patient, expert, ...) 2) surgery: temporality with respect to surgery for the removal of cancerous breast tissues (before vs. after) 3) relapse (yes vs. no). Relevance and redundancy of certain descriptor groups was discussed, indicating that n-grams of lemmas and grammatical tags should be of principal interest.

Statistical machine translation (SMT) is today one of the research areas that, together with speech recognition, is pushing mostly toward the use of huge n-gram LMs. Marcello and Mauro (2007) carried out an experiment on the use of n-grams on SMT. Results showed halving of memory requirements, at the cost of 44% slower decoding speed. In addition, loading the LM on demand permitted to keep the size of memory allocated to the decoder nicely under control. Liangyou and Zhengxian (2013) researched on Fuzzy Matching for n-gram-based Machine Translation evaluation. They proposed to view lexical semantic similarity, which is based on existing linguistic resource, WordNet, as a way of fuzzy matching and used it during the process of matching n-grams. Their method took synonym into consideration when matching two n-grams. In their experimental results, the performance of BLEUwn when evaluating segment and considering only unigram, declined. Their reason for that was that without high-order n-grams which can handle local context and with a relative short length of segment, the ambiguity of words and fewer synonyms bring much "noise" into evaluation. However, results showed that as N increases, evaluation performance of BLEUwn was improved gradually while BLEU develops in the opposite direction. This indicated that the method proposed in their paper was more suitable for high-order n-grams. They suggested that this may be because high-order n-grams can better regulate WordNet-based lexical similarities with the aid of close words. Experimental results showed that their method can significantly improve the correlation coefficient between automatic and manual evaluation scores, and fuzzy matching is more suitable for high-order n-grams and document-level evaluation, and WordNet also performs more effectively at document-level evaluation. Moreover, experimental results showed fuzzy matching can also achieve promising performance when multiple references are not available.

Myunggwon et al (2011) conducted an experiment on published n-gram data which was constructed from huge document set gathered until 2005 by Google. They proposed a method to construct domain n-gram data in which a specific domain group is interested and apply the data to text editor for practical efficiency in evaluation. Results finding pointed out the limitation of Google n-gram to be used in real world application due to its hugeness and suggested a method constructing domain n-gram which is related to a specific subject. To construct domain n-gram, the work contained a selection method of domain words and a construction method of domain n-gram data containing the domain words. In the evaluation, a few text editors based on manual typing, Google trigram, domain each-gram, and Domain n-gram were examined respectively. And the results showed strengths of domain n-gram through analyses in aspects of time efficiency, count of reduced typing, count of recommended typing, and capacity efficiency. Especially, domain unigram, bigram, and trigram can be expected to show high performance. However, it was not the case for 4- and 5-grams because it could not cover tokens which people would input and it rather wasted much time for searching.

Asaf et al (2012) conducted a research on detecting unknown malicious code by applying classification

techniques on OpCode patterns. They used a methodology for malware categorization by implementing concepts from the text categorization domain. While most of the previous studies extracted features which are based on byte n-grams, in this study, they used OpCode n-gram patterns, generated by disassembling the inspected executable files, to represent the files. Unlike byte sequence, OpCode expressions, extracted from the executable file, were expected to provide a more meaningful representation of the code. In the analogy to text categorization, using letters or sequences of letters as features is analogous to using byte sequences, while using words or sequences of words is analogous to the OpCode sequences. Evaluation results indicated that the evaluated methodology achieved a level of accuracy higher than 96% (with TPR above 0.95 and FPR approximately 0.1), which slightly improved the results in previous studies that used byte n-gram representation. The chronological evaluation showed a clear trend in which the performance improves as the training set is more updated.

The use of n–grams is quite prevalent in the field of pattern recognition. Preety et al (2010) presented a speaker-independent lipreading system, based on n–gram modeling of visual features. Lipreading systems identify utterances using only the visual information extracted from images of the mouth. A combination of visual features extracted from the lip boundary was used to build n–gram models for recognition of the spoken word. Their proposed n–gram model showed encouraging and comparable results even when the training dataset is not very large. The proposed lipreading system made the following contributions: 1) It detected the outer lip contour from the image and extracted geometrical parameters from the contour. 2) It built n–gram models from the visual parameters to form new feature vectors. 3) The n–gram feature vector was used for speech classification using the Hidden Markov Model (HMM).

Canvar and Trenkle (1994) used N-gram-based approach to text categorization on language classification. They collected 3713 language samples from the soc.culture newsgroup hierarchy of the Usenet. These newsgroups are devoted to discussions about topics relevant to particular countries or cultures. First, they kept track of whether the original article was over or under 300 bytes in length. Their initial hypothesis was that the system would have more problems classifying shorter messages because there would be a smaller amount of text from which to compute N-gram frequencies. On the whole, their system was only slightly sensitive to length. Second, they also varied the number of the N-gram frequencies available in the profile for the match, by limiting it to statistics for 100,200,300 or 400 N-grams. This variable did have an impact on match performance, although by the 400 N-gram level language classification was almost perfect. Their system worked very well for language classification, achieving in one test a 99.8% correct classification rate on Usenet newsgroup articles written in different languages. The system also worked reasonably well for classifying articles from a number of different computer-oriented newsgroups according to subject, achieving as high as an 80% correct classification rate. Their results showed some interesting patterns. First, the classification procedure worked a little better for longer articles, but not quite as much as they expected. Secondly, for the most part, the classification procedure worked better the longer the category profile it has to use for matching. However, there were some interesting anomalies. These represented combinations of test variables that did worse than similar combinations with shorter profiles. In other words, for these cases, using more N-gram frequencies actually decreased classification performance. Post mortem examination of the problematic articles showed that at least part of the difficulty was that, in spite of the manual truthing efforts to remove mixed text, some articles still had passages from two languages.

Artur et al (2007) compared N-grams and morphological normalization. They experimented on a Croatian-English parallel corpus. The source data was Croatia Weekly (CW), a newspaper that was published in English and translated to Croatian from 1998 to 2000 by the Croatian Institute for Information and Culture. Each article was assigned to one of four classes: politics, economy, nature, and events. They used SVM classification method that was introduced by Vladimir Vapnik and his colleagues. Their classification performance showed that for both languages, 3-grams and 4-grams achieved similar result as words, whereas 2-grams performed significantly worse. For the Croatian documents, morphological normalization improved the classification quality of words from 1% up to 2%. In terms of the F1 measure, words were worse than 3-grams or 4-grams on an equal feature set size up to the size of about 45k features. At higher feature set sizes, words achieved quite good results similar to words-id. This was because although the Croatian morphology allows many words to have more than 30 forms, not all of these forms are used often. Besides that, if a corpus and the documents are large enough, all word forms will be covered much better and the classification with words will outperform 3-grams and 4-grams. The 3-grams achieved better results than 4-grams on low to middle sized feature sets. That was because 3-grams are shorter thus yielding more intersecting features among documents at lower numbers of features. For the English documents, words performed better than the n-grams for all feature set sizes. Normalized set words-p performed just slightly better than the words, but not as significantly as in Croatian. This is due to morphology richness. N-gram performance is comparable to words and words-p. The figures showed that n-grams are less sensitive to language morphology than the bag-of-words approach. On the English set, the classification quality obtained by n-grams was close to the classification quality obtained by words, but on the

Croatian corpus n-grams were closer to the morphological normalizing method. Limitations to their work included the following questions: Is there a way the n-grams could outperform words in a morphologically rich language? Their answer was probably yes, if they had shorter documents having disjunctive sets of word forms. The data set size could affect this as well; classifiers trained on smaller sets would be more sensitive to morphology of the language. They recommended future work should try to answer these questions and compare n-grams with string kernels.

Alberto et al (2010) researched on the use of word length n-grams for text re-use detection. They performed experiments to compare a common word n-gram encoding to their proposed length n-gram encoding. Evaluation in terms of Recall was carried out by considering two corpora including cases of simulated plagiarism and text co-derivatives. They approached the problem of the preliminary selection of closely related texts, the first step for text-reuse, co-derivatives analysis, and automatic plagiarism detection. They proposed a method to solve this task which encodes the documents on the basis of their word lengths. They recognized that some efficient methods, such as fingerprinting, imply a loss of information between the actual document and its fingerprint, their method reduced such loss, as they empirically showed. Their retrieval experiments were performed on two corpora: the first one of simulated plagiarism and the second one of text co-derivatives. The obtained results showed that representing the documents by the length encoding: (i) does not affect the performance of the retrieval process; (ii) favours a flexible comparison of documents as n-grams of any level are available in the text representation; and (iii) the entire encoding and comparison process is speeded up, an important factor when the amount of comparisons to perform is significant.

Shereen et al (2013) researched on semantic enrichment in text supervised classification. They focused on two strategies for semantic enrichment of conceptualized text representation. The first one was based on semantic kernel method while the second one was based on enriching vectors method. These two semantic enrichment strategies were evaluated through experiments using Rocchio as the supervised classification method in the medical domain, using UMLS ontology and Ohsumed corpus. The first enrichment strategy, before training, was based on Semantic Kernels method. Obtained results showed deterioration in the performance of Rocchio and its variants after applying Semantic Kernels on vectors that represented corpus documents. Thus, Semantic Kernels seemed to introduce noise to text representation and weakens its capability to distinguish classes. The second enrichment strategy, after training and before prediction, was based on Enriching Vectors method. They reported better results than those obtained without enrichment as well as those obtained after applying Semantic Kernels. Results showed that this improvement depends on both the semantic similarity measure used in enrichment and the similarity measure used in prediction.

5. Experiment and Results

The experiments were performed over hate speech text data in Kenya written in English language. The data used in our study includes text from online social media and some text data were generated by the researchers. Purposive sampling method was used to select data for testing and training. This sampling technique was guided by National Cohesion and Integration Commission (2010). In this paper, all experiments were conducted for the corpus of 63 sentiments. For evaluation of the experiments, we used 53 sentiments for training, and the rest for classification. We used WEKA implementation of Naïve Bayes (NB), K-Nearest Neighbour (KNN), and Support Vector Machines (SVM) classifiers. Figure 5.1 below shows WEKA Graphical User Interface.

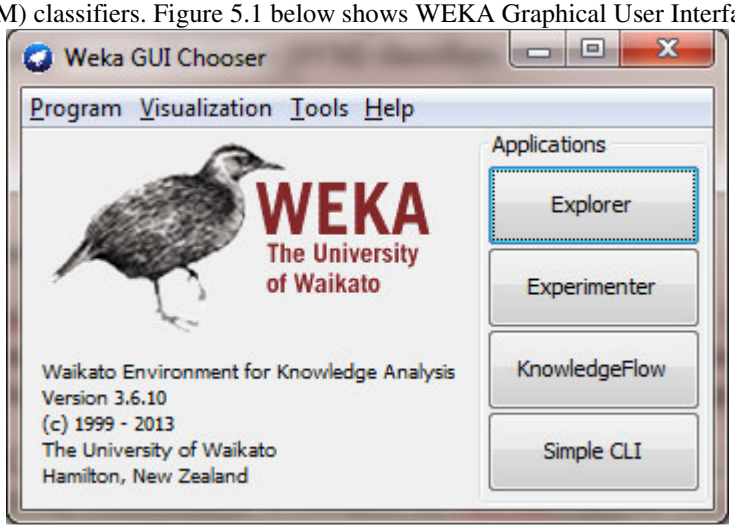

Figure 5.1 WEKA Graphical User Interface

We used two features as baseline: word based features and word based n-gram features. The experiments were first conducted using word based features and later using word based n-gram features. Figure 5.2 below shows a

sample screen of Naïve Bayes training results using word based features. Accuracy, Precision, Recall and F-Measure were used to evaluate three classifiers namely Naïve Bayes (NB), K-Nearest Neighbour (KNN), and Support Vector Machines (SVM).

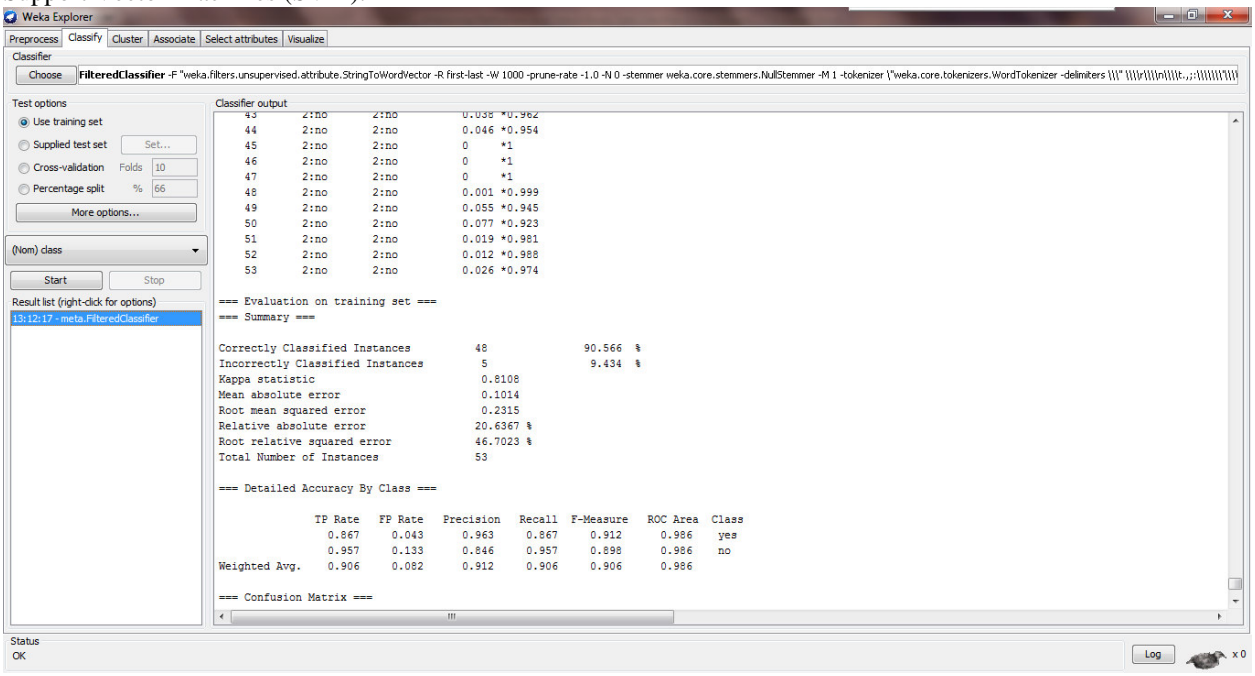

Figure 5.2 Naïve Bayes Classifier Training Results

Table 5.1 shows summary of accuracy results observed by using training data on the three classifiers using word based features over a range of n-grams. Using 1-gram (unigrams), KNN produced the highest accuracy value of 98.1%, followed by SVM at 96.2%, and NB at 90.6%. Using 2-grams (bigrams), all the three classifiers produced equal accuracy value of 98.1%. Compared to 1-gram, NB had the largest improvement of about 7%, followed by SMV at 2%. KNN did not show any improvement.

Table 5.1 Accuracy Values of Different and Varying n-grams

	1-gram	2-grams	3-grams	4-grams	5-grams	6-grams	7-grams	8-grams	9-grams	10-grams
NB	90.6%	98.1%	92.5%	77.4%	66.0%	58.5%	56.6%	56.6%	56.6%	56.6%
KNN	98.1%	98.1%	98.1%	79.2%	67.9%	66.0%	60.4%	56.6%	56.6%	56.6%
SVM	96.2%	98.1%	98.1%	69.8%	67.9%	58.5%	60.4%	56.6%	56.6%	56.6%

When 3-grams (trigrams) was used, KNN and SVM had the same accuracy value they had in bigrams at 98.1% while NB accuracy value reduced by about 6%. From 3-grams, all classifiers showed reducing accuracy values. Figure 5.3 is a line graph showing the different accuracy values with the corresponding classifiers and n-grams.
The highest recall and F-Measure recorded by the classifiers was 0.981. KNN had this value for unigram, bigram and trigram while NB had this value for bigram only. SVM achieved this value for bigram and trigram. The highest precision value recorded was 0.982. KNN had this value with unigram, bigram and trigram. SVM had this precision with bigram and trigram while NB has this value for bigram.

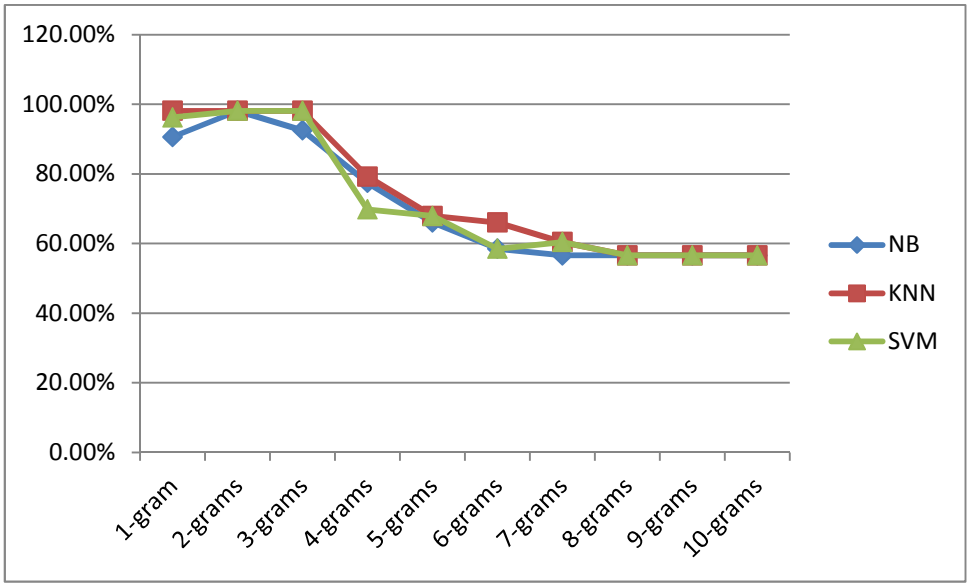

Figure 5.3 Graph of Accuracy of Different Classifiers using Varying n-grams

After training phase, testing phase was conducted using test data set. The table 5.2 shows the accuracy results that were displayed using various feature sets and machine learning algorithms.

Table 5.2 Accuracy Values for Test Data

	1-gram	2-grams	3-grams	4-grams	5-grams	6-grams	7-grams
NB	80%	90%	80%	80%	80%	80%	80%
KNN	100%	90%	80%	80%	20%	20%	20%
SVM	100%	90%	80%	20%	20%	80%	20%

The above results show that there was an increase in accuracy by 10% when bigrams were used as features of NB. Thereafter, the accuracy value remained constant at 80% for tri-grams, 4-grams, 6-grams and 7-grams respectively. Figure 5.4 is a line graph of the accuracy values for testing data set.

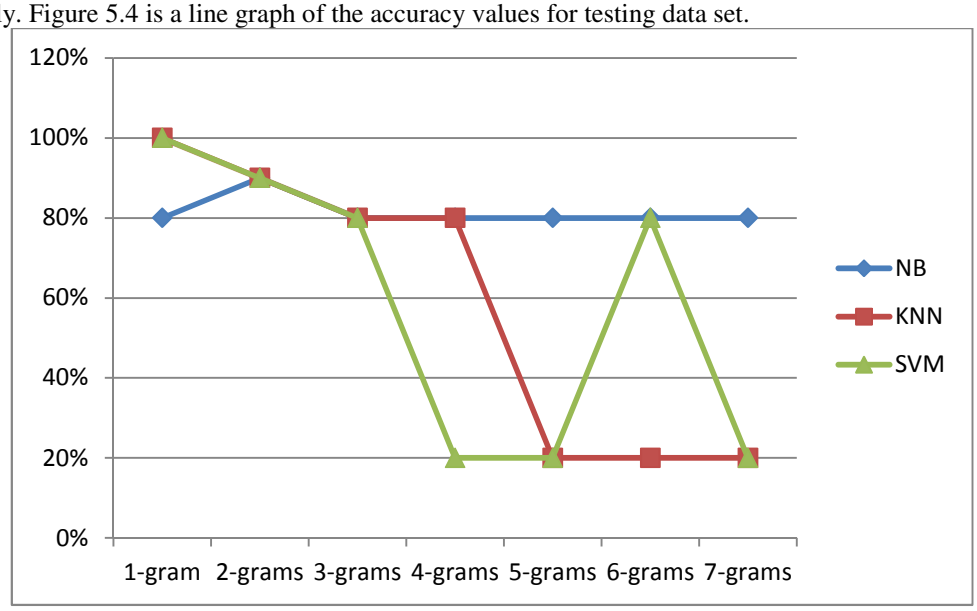

Figure 5.4 Graph of Accuracy Values For Testing Phase

6. Conclusion

Text Categorization is an active research area in Natural Language Processing. Many methods have been used to get better automated text categorization performance. Some methods deals with modifying text document representation while other deal with improving text categorization algorithms. N-gram based text categorization is among the methodologies used in English language for text categorization, having good performance. In this paper, we have presented an n-gram-based model for English text classification for English datasets about hate speech in Kenya. For our dataset, analyzing the efficiency of n-grams shows that bi-grams have much better

performance for text categorization for NB. KNN had the same accuracy value (98.1%) for unigram, bigram, and trigram when training data was used. SVM also, had the highest accuracy value (98.1%) for bigram and tri-gram. These results show that n-grams improve the performance of machine learning algorithms.

References

Alberto, B., Chiara, B., Mirko, D., E. and Paolo, R. (2010), "Word length n-Grams for Text Re-use Detection", in A Gelbukh (Ed.), CICLing2010, LNCS 6008, pp. 687-699, Springer-Verlag Berlin Heidelberg.

Artur, S., Jean-Hugues, C., Bojana, D., B. and Annie, M. (2007), "N-Grams and Morphological Normalization in Text Classification: A Comparison on a Croatian-English Parallel Corpus", in J. Neves, M. Santos, and J. Machado (Eds.), EPIA 2007, LNAI 4878, pp. 671-682, Springer-Verlag, Berlin.

Asaf, S., Robert, M., Clint, F., Shlomi, D. and Yuval, E. (2012), "Detecting Unknown Malicious Code by Applying Classification Techniques on OpCode Patterns", in Shabtai et al. *Security Informatics.*

Bing, L. (2007), Web Data Mining: Exploring Hyperlinks, Contents and Usage Data, Springer-Verlag Berlin Heidelberg.

Canvar, W., B. and Trenkle, J., M. (1994), "N-Gram-based Text Categorization", In proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval (Las Vegas, US, 1994), pp. 161-175.

Christian, M., Sheila, M., Jorge, A., B. and Michael R. (2009), "Exploiting N-gram Analysis to predict Operator Sequences", *Association for the Advancement of Artificial Intelligence.*

Daniel, J. and James, H., M. (2014), Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition, Prentice Hall, New Jersey.

Durga, B., D. and Venu, G., R., K. (2012), "Text Categorization and Machine Learning Methods: Current State of the Art", *Global Journal of Computer Science & Technology Software and Data Engineering*, 12.

Fabrizio Sebastianin (2002), "Machine Learning in Text Categorization". *ACM Computing Surveys*, 34, pp. 1-47.

Ikonomakis, M., Kotsiantis, S. and Tampakas V. (2005), "Text Classification Using Machine Learning Techniques", *WSEAS Transactions on Computers*, 4, pp. 966-974.

Karl-Michael, S. (2005), "Techniques for Improving the Performance of Naïve Bayes for Text Classification", in A. Gelbukh (Ed.): CICLing, 3406, pp. 682-693.

Krzysztof, J., C., Witold, P., Roman, W., S. and Lukasz, K. (2007), Data Mining: A Knowledge Discovery Approach, Springer Science, New York.

Liangyou, L. and Zhengxian, G. (2013), "Fuzzy Matching for N-Gram-based Machine Translation Evaluation", in D.H.Ji and G.Z. Xiao (Eds.), CLSW 2012, pp. 40-48, Springer-Verlag, Berlin.

Marcello, F. and Mauro, C. (2007), "Efficient Handling of n-gram Language models for Statistical Machine Translation", *in Proceedings of the second workshop on Statistical Machine Translation, 88-95.*

Mladenic, D. and Grobelnik, M. (1998), "Word Sequences as features in text-learning", *in proceedings of the 7th Electrotechnical and Computer Science Conference (Ljubljana, SL, 1998), pp.343-354, Springer Verlag.*

Myunggwon, H., Dongjin, C., Hyogap, L. and Pankoo, K. (2011), "Domain N-Gram Construction and its Applications to Text Editor", in N.T. Nguyen, C. –G. Kim, and A. Janiak (Eds.), ACIIDS, 6891, pp. 268-277.

National Cohesion and Integration Commission, (2010), Guidelines for Monitoring Hate Speech in the Electronic Media in Kenya. Nairobi: Kenya.

Preety, S., Vijay, L., Deepika, G. and Gaur, M., S. (2010), "Lipreading Using n-gram Feature Vector", in A. Herrero et al. (Eds.): CISIS 2010, AISC 85, pp. 81-88, Springer-Verlag, Berlin.

Ramasundaram, S. and Victor, S., P. (2013), "Algorithms for Text Categorization: A Comparative Study", *World Applied Sciences Journal,* 22, pp. 1232-1240.

Ronen, F. and James, S. (2006), The Text Mining Handbook: Advanced approaches in Analyzing Unstructured Data, Cambridge University Press, New York.

Shereen, A., Bernard, E. and Sebastien, F. (2013), "Semantic Enrichment in Text Supervised Classification: An Application to Medical Domain", Association for the Advancement of Artificial Intelligence.

Thomas, O., Sandara, B., Jerome, A. and Cyrille, J. (2015), Enriching Online Breast Cancer Patient narratives with clinically relevant Information. Downloaded from www.researchgate.net

Yuqian, J., Huaizhong, L., Xuesong, W. and Dongming, L. (2011), "A Technique for Improving the Performance of Naïve Bayes Text Classification", in Z. Gong et al. (Eds.): WISM 2011, 6988, pp. 196-203.

Zdravko, M. & Daniel, T., L. (2007), "Data Mining the Web: Uncovering Patterns in Web Content, Structure and Usage", John Wiley & Sons, New Jersey.

The IISTE is a pioneer in the Open-Access hosting service and academic event management. The aim of the firm is Accelerating Global Knowledge Sharing.

More information about the firm can be found on the homepage:
http://www.iiste.org

CALL FOR JOURNAL PAPERS

There are more than 30 peer-reviewed academic journals hosted under the hosting platform.

Prospective authors of journals can find the submission instruction on the following page: http://www.iiste.org/journals/ All the journals articles are available online to the readers all over the world without financial, legal, or technical barriers other than those inseparable from gaining access to the internet itself. Paper version of the journals is also available upon request of readers and authors.

MORE RESOURCES

Book publication information: http://www.iiste.org/book/

Academic conference: http://www.iiste.org/conference/upcoming-conferences-call-for-paper/

IISTE Knowledge Sharing Partners

EBSCO, Index Copernicus, Ulrich's Periodicals Directory, JournalTOCS, PKP Open Archives Harvester, Bielefeld Academic Search Engine, Elektronische Zeitschriftenbibliothek EZB, Open J-Gate, OCLC WorldCat, Universe Digtial Library , NewJour, Google Scholar