Textual Document Categorization using Bigram Maximum Likelihood and KNN

Rounak Dhaneriya
Department of Computer Science & Engineering
Rajiv Gandhi Proudyogiki Vishwavidyalaya
Bhopal, India

Manish Ahirwar
Department of Computer Science & Engineering
Rajiv Gandhi Proudyogiki Vishwavidyalaya
Bhopal, India

Dr. Mahesh Motwani
Department of Computer Science & Engineering
Rajiv Gandhi Proudyogiki Vishwavidyalaya
Bhopal, India

***Abstract*—In recent year's text mining has evolved as a vast field of research in machine learning and artificial intelligence. Text Mining is a difficult task to conduct with an unstructured data format. This research work focuses on the classification of textual data of three different literature books. The collection of data is extracted from books entitled: Oliver Twist, Don Quixote, Pride and Prejudge. We used two different algorithms: KNN and Bigram based Maximum Likelihood for the mentioned purpose and the evaluation of accuracy is done using the confusion matrix. The results suggest that the text mining using bigram based maximum likelihood logic performs well.**

Keywords-Text Mining; KNN; Bigram; Maximum Likelihood

I. INTRODUCTION

The main objective of document categorization is to assign each document in a given data collection to a class or category, according to the nature of its textual content. In general, document categorization can be used to directly address different practical tasks, such as spam filtering, press clipping and document clustering, just to mention a few; or, alternatively, it can be used as a component of a larger system to tackle more complex tasks, such as, for example, opinion mining and plagiarism detection [1]. The document categorization is also called as an application of text mining. In this research we are using two different algorithms: KNN and bigram based maximum likelihood for the task of document categorization.

KNN is a sort of instance based learning, where the function is just approximated locally and all calculation is conceded until classification because of this KNN is also called lazy learning algorithm. The KNN algorithm is among the most straightforward of all machine learning algorithms. This method constitutes a very simple but robust classification algorithm that, similarly to k-means clustering, operates over a vector space model. The basic idea of the k nearest neighbour algorithm is to assign a new data sample to a category based on the categories of the closest samples for which the categories are known. In other words, given a new data sample x, its k closest samples are extracted from the train set, which are then referred to as the nearest neighbors of x. Finally, x is assigned to the most common category that is observed among its neighbors [1].

Bigram models can be considered as n-gram model of order two which follows the Markov property assumption. In the bigram case, the probability of a given word depends on the word immediately before. Let us consider, for instance, a unit of text w which consist of a sequence of words w1, w2,…wm. The bigram model for the above mentioned text unit can be defined as:

$$p(w) \approx p(w2|w1)\ p(w3|w2)\ p(w4|w3) \ldots p(wm|wm\text{-}1) \quad (1)$$

Maximum likelihood estimates can be easily computed for probability p(w) by using a training corpus. Indeed, when long word histories are involved, the model tends to become unreliable as most of the histories are not actually seen in the training dataset and the corresponding n-gram probability estimates are not reliable.

II. LITERATURE SURVEY

X. Ding et.al [2] in "A Holistic Lexicon-Based Approach to Opinion Mining" proposes a holistic lexicon-based approach by exploiting external evidences and linguistic conventions of natural language expressions which is able to handle sentiment words that are context dependent. A system called Opinion Observer is also implemented which aggregates multiple conflicting sentiment words in a phrase. The results show that the proposed technique is highly effective when using a benchmark of product review dataset.

M. Hu et.al [3] in "Mining and Summarizing Customer Reviews" used customer reviews of a product and summaries it. The author used feature based text summarization which works at the sentence level of the review. This approach mines features using several lexicon based methods and effectiveness is measured over online reviews of sold products.

M. Trupthi et.al [4] in "Improved Feature Extraction and Classification - Sentiment Analysis" explored different machine learning classification approaches for finding the best

possible approach with different feature selection schemes to obtain a SA model for the movie review domain.

V. Singh et.al [5] in "Sentiment Analysis of Movie Reviews: A new Feature-based Heuristic for Aspect-level Sentiment Classification" presents a domain specific feature-based heuristic for aspect-level sentiment analysis of movie reviews. The author had devised an aspect oriented scheme that analyses the textual reviews of a movie and assign it a sentiment label on each aspect. The author also used SentiWordNet scheme to compute document level sentiment and compared result using Alchemy API. The results obtained are more accurate and focused sentiment profile than simple document level analysis.

B. Trstenjak et.al [6] in "KNN with TF-IDF Based Framework for Text Categorization" presents KNN algorithm with TF-IDF method and framework for text classification which classifies according to various parameters. Evaluation of the framework is focused on the speed and quality of classification and the results shows good and bad features of the algorithm.

B. Pang et.al [7] in "Thumbs up? Sentiment Classification using Machine Learning Techniques" introduces the grouping of documents not by topic but by overall opinion score of the record. The author uses motion picture surveys from IMDB (Internet Movie database) and employed Naïve Bayes, Maximum Entropy Classification and Support Vector Machines for document level SA. In this paper, author examines the effectiveness of applying machine learning techniques to the opinion mining problem. A challenging aspect of this issue appears to distinguish it from traditional topic-based categorization is that while topics are often identifiable by keywords alone, opinion can be expressed in a subtler way. Thus, sentiment seems to require more understanding than the usual topic-based categorization. So, apart from presenting their results obtained via machine learning techniques, author also analyze the problem to gain a better understanding of how difficult it is. The results represent that the above mentioned algorithms so not perform as well on sentiment classification as on traditional topic-based categorization.

III. Data Collection And Preparation

This work is implemented on MATLAB computing environment. The data collection used here has been extracted from three different books:

- Oliver Twist. A novel by English author Charles Dickens, which was published in 1838. It tells the story of an orphan boy, Oliver Twist, who escapes from his guardian and goes to London, where he meets a leader of a gang and unknowingly gets involved in their criminal activities.
- Don Quixote (English translation). A novel by Spanish writer Miguel de Cervantes, which was published in two volumes, in 1605 and 1615. It tells the adventures of a country gentleman that gets obsessed after excessively reading chivalry books and ends up thinking he is also a chevalier. Don Quixote is considered the most influential work of the Spanish literature.
- Pride and Prejudice. A novel by English novelist Jane Austen, which was published in 1813. It tells the story of Elizabeth Bennet, the second of five sisters in a landed gentry family, who deals with matters of morality, education and manners in her 19th century England context.

The complete books are publicly available in digital format from the Project Gutenberg website (http://www.gutenberg.org/). The data collection to be used here was prepared by extracting sample paragraphs from the three books, where the only restriction imposed was for each paragraph to be between 60 and 300 words in length. According to this, the documents in our data collection correspond to paragraphs of the original books. The dataset has been formatted into a structure array and saved into three separated files (one for each book derived subset): OliverTwist.mat, DonQuixote.mat and Pride&Prejudice.mat. Each of the elements in the data structures contains the following fields: book, a string containing the name of the corresponding book; chap, an integer identifying the number of the corresponding chapter; text, a string containing the original raw text of the document (paragraph) represented by such element; token, a cell array of strings containing the individual tokens within the document; vocab, a cell array of strings containing the unique tokens within the document, i.e. the document's vocabulary; and count, an integer array containing the term-frequency counts for the corresponding vocabulary terms. We generated random indexes to randomize the resulting test, development and train datasets. We will consider 100 documents from each of the three sub collections for constructing the test set, so the total size of the resulting test set will be 300 documents. Similarly, we will consider 100 documents from each sub collection for constructing the development set, so the total size of the resulting development set will be 300 documents, these sets are generated and saved in file randomvars.mat. The remaining documents will be used for the train set.

TABLE I. BASIC STATISTICS FOR THE PREPARED DATA COLLECTION [1]

Sub collection	**Dataset 1**	**Dataset 2**	**Dataset 3**
Book title	Oliver twist	Don Quixote	Pride & Prejudice
Documents (paragraphs)	840	843	666
Running words	85,419	111,507	76,203
Vocabulary	8,200	8,338	5,244
Minimum document size	60	60	60
Maximum document size	295	300	300
Average document size	101.69	132.27	114.42

IV. KNN DOCUMENT CLASSIFICATION

In this section, more specifically, we will be considering a very common supervised approach known as k nearest neighbors or knn. This method constitutes a very simple but robust classification algorithm that, operates over a vector space model. The basic idea of the k nearest neighbor algorithm is to assign a new data sample to a category based on the categories of the closest samples for which the categories are known. In other words, given a new data sample x, its k closest samples are extracted from the train set, which are then referred to as the nearest neighbors of x. Finally, x is assigned to the most common category that is observed among its neighbors. Before applying knn to the test set under consideration, we should select an optimal value for k. We do this by exploring the resulting classification accuracy for different values of k over the development set. After having an optimal value for the parameter k, we applied the knn algorithm to the test set by exactly following the same procedures used for development dataset.

Algorithm 1: knn document classification for development set [1]

```
1: INPUT: devdata
2: OUTPUT: assignedcat
-------------------------------------------------------------------------
3: for all k = 1 to 50 do
4:      for all n = 1 to size(devmtx,2) do
5:           cossim ← devmtx(:,n)'*trnmtx
6:           [void,order] ← sort(cossim,'descend')
7:           vals ← trncat(order(1:k))
8:           hcat ← hist(vals,1 to length(unique(trncat)))
9:           [void,thecat] ← max(hcat)
10:          assignedcat(n,1) ← thecat
11:     end for
12:     accuracy(k)←sum(devcat==assignedcat)
                                   /length(devcat)*100
13: end for
```

Note: Repeat steps 4 to 11 for the knn classification for the test set using optimal value for k.

V. BIGRAM MAXIMUM LIKELIHOOD

In this section, we continue the supervised approach to document categorization. But, different from the methods discussed in the previous section, here we focus our attention on statistical methods. More specifically, we will consider the likelihood ratio approach, in which the probability of a given document is estimated by means of different category-dependent statistical models, and the ratio between such probabilities is used to finally determine the most probable category for the given document.

The likelihood ratio framework is actually a very general one, as it does not impose any restriction on the class of statistical model to be used. Indeed, it can be implemented by means of any model as far as good model parameters can be estimated from the available train data and good likelihood estimates can be derived from the model. In the experimental work we used bigram statistical model (calculated and stored in file named bigram_model.mat) for the task of document categorization. In our experimental setting we are dealing with three categories: Oliver Twist, Don Quixote and Pride and Prejudice. A simple approach is to consider three binary classification problems: Oliver Twist versus non Oliver Twist, Don Quixote versus non Don Quixote, and Pride and Prejudice versus non Pride and Prejudice; which requires training six different models, one for each category and one for each category complement. As from the bigram model. we only have available three bigram models, one for each category, but we still will be able to compute probability estimates for the three category complements by using a simple linear combination of models. Regarding category assignment decisions, instead of making independent binary decisions for each category, we implemented a simple multi-category decision process. Now we had applied likelihood ratio classification to our data collection. Before applying the procedure to the test data, we apply it to the development data, this have helped us to select appropriate values for $\zeta 1$, $\zeta 2$ and $\zeta 3$ (logarithm of the inverted prior ratio for each category).

First, we compute bigram probability estimates for each document in the development set by using each of the three category bigram models but instead of probabilities, we computed log-probabilities and logarithm of the likelihood ratios for each document with respect to each category (as log-probability estimates for each complement, we just average the log-probabilities for the other two categories).

Algorithm 2: Bigram log-probability and log-likelihood ratio estimates for each category in development set [1]

```
1: INPUT: devdata
2: OUTPUT: logirat
-------------------------------------------------------------------------
3: Initialise: logprobs = zeros(length(devdata),3)
4: for all k = 1 to length(devdata) do
5:      for all n = 2 to length(devdata(k).token) do
6:           token1 ← devdata(k).token{n-1}
7:           token2 ← devdata(k).token{n}
8:           for all model = 1 to 3 do
9:                idx1←find(strcmp(bigram
                        (model).vocab,token1))
10:               if isempty(idx1) then
11:               c1 ← bigram(model).un_unk
12:               else c1←bigram
                        (model).un_cnt(idx1)
                        end
13:               w1←strcmp(bigram
                        (model).word1,token1)
14:               w2←strcmp(bigram
                        (model).word2,token2)
15:               idx2 ← find(w1&w2)
16:               if isempty(idx2) then
```

```
17:                     c2 ← bigram(model).bi_unk
18:                     else c2←bigram
                                (model).bi_cnt(idx2)
                                end
19:                     logprobs(k,model)←logprobs
                                (k,model) + log2(c2/c1)
20:             end for
21:     end for
22: end for
23: for all k=1 to size(logprobs,1) do
24:     loglirat(k,1)←logprobs(k,1)-
        (logprobs(k,2)+logprobs(k,3))/2
25:     loglirat(k,2)←logprobs(k,2)-
        (logprobs(k,1)+logprobs(k,3))/2
26:     loglirat(k,3)←logprobs(k,3)-
        (logprobs(k,1)+logprobs(k,2))/2
27: end for
```

We searched for optimal values for ζ_1, ζ_2 and ζ_3, we conducted an ad hoc exploration of the solution space. For this, we generate accuracy curves by varying one ζ value at a time (from -100 to 100), while maintaining the other two values equal to zero. The highest accuracies are observed for the intervals $40 < \zeta_1 < 60$, $-20 < \zeta_2 < 0$, and $-60 < \zeta_3 < -40$ when each of these three parameters is varied independently from the others.

Let we recapitulate on how the accuracies have been computed. First, we added a matrix of ζ values (one different value per column) to the matrix of log-likelihood ratios loglirat. Then, we extracted the indexes of maximum values along each row. This is actually a multi-category selection criterion, as we are forcing the selection of one category from the three possible ones. This selection is based on the highest binary classification score, from the three computed scores. Finally, the accuracy was computed by counting and normalizing the total amount of correct category assignments. We computed the accuracy over the development set when all three ζ parameters are set to zero.

Next, we computed accuracy values for all integer combinations of ζ_1, ζ_2 and ζ_3 within the region of interest defined above. The values of ζ_1, ζ_2 and ζ_3, the ones producing the highest accuracy over the development set are considered as the optimal values of ζ within the consider region of the ζ space. A noticed improvement of the accuracy is reported, that has been achieved by adjusting the ζ parameters. However, it can be verified that this optimal set of parameters is not unique! Indeed, there are many more value combinations for these parameters that produce the same maximum accuracy. Next, we applied the likelihood ratio classification algorithm to the test dataset by exactly following the same procedures used for the development set for estimating bigram probabilities and likelihood ratios and computed accuracy and confusion matrix with the help of optimal ζ matrix.

VI. RESULTS

The accuracy values obtained from the evaluation of confusion matrix of both KNN and Likelihood ratio can be directly compared. Accuracies obtained is derived from:

Accuracy=sum(diag(confusion_mtx))/sum(sum(confusion_mtx))*100 (2)

In the case of KNN classification we noticed the accuracy of 92.33% and in the case of Likelihood ratio classification we noticed the accuracy of 95%. The measure shows that the likelihood approach performs 2.67% better than the KNN approach.

TABLE II. CONFUSION MATRIX FOR BIGRAM CLASSIFICATION

	Cat 1	Cat 2	Cat 3
Assigned to Cat 1:	95	2	4
Assigned to Cat 2:	3	94	0
Assigned to Cat 3:	2	4	96

TABLE III. CONFUSION MATRIX FOR KNN CLASSIFICATION

	Cat 1	Cat 2	Cat 3
Assigned to Cat 1:	86	1	0
Assigned to Cat 2:	4	92	1
Assigned to Cat 3:	10	7	99

VII. CONCLUSION

This research work focuses on classifying textual documents of different categories of books. For completing these tasks, we used different machine learning supervised classification approaches: Supervised classification in vector space using KNN, Supervised classification in probability space using bigram based maximum likelihood approach. The inputs are in the form of textual data with their respective characteristics such as, vocabulary, index, tokens, text and categories etc. and the output is the classified documents in different categories. As seen from result bigram classification preforms well on this data domain.

VIII. REFERENCES

[1] R. E. Banchs, Text Mining with MATLAB, Barcelona: Springer Science+Business Media , 2013.

[2] X. Ding, B. Liu and P. S. Yu, "A Holistic Lexicon-Based Approach to Opinion Mining," in *WSDM, (ACM) Association for Computing Machinery*, 2008.

[3] M. Hu and B. Liu, "Mining and Summarizing Customer Reviews," in *KDD*, 2004.

[4] M. Trupthi, S. Pabboju and G. Narasimha, "Improved Feature Extraction and Classification - Sentiment

Analysis," *IEEE,* 2016.

[5] V. Singh, R. Piryani, A. Uddin and P. Waila, "Sentiment Analysis of Movie Reviews: A new Feature-based Heuristic for Aspect-level Sentiment Classification," *IEEE,* 2013.

[6] B. Trstenjak, S. Mikac and D. Donko, "KNN with TF-IDF Based Framework for Text Categorization," in *24th DAAAM International Symposium on Intelligent Manufacturing and Automation*, 2013.

[7] B. Pang, L. Lee and S. Vaithyanathan, "Thumbs up? Sentiment Classification using Machine Learning Techniques," in *Proceedings of the Associaltion for Computational Linguistics (ACL)-02 conference on Empirical Methods in Natural Language Processing (EMNLP)*, Philadelphia, 2002.