2017 8th International Conference on Information Technology (ICIT)

Using KNN Algorithm for Classification of Textual Documents

Aiman Moldagulova
Universiti Tenaga Nasional
Malaysia
a.moldagulova@iitu.kz

Rosnafisah Bte. Sulaiman
Universiti Tenaga Nasional
Malaysia
Rosnafisah@uniten.edu.my

***Abstract*— Nowadays the exponential growth of generation of textual documents and the emergent need to structure them increase the attention to the automated classification of documents into predefined categories. There is wide range of supervised learning algorithms that deal with text classification. This paper deals with an approach for building a machine learning system in R that uses K-Nearest Neighbors (KNN) method for the classification of textual documents. The experimental part of the research was done on collected textual documents from two sources: http://egov.kz and http://www.government.kz. The experiment was devoted to challenging thing of the KNN algorithm that to find the proper value of k which represents the number of neighbors.**

***Keywords*— *Text classification; KNN algorithm*.**

I. INTRODUCTION

Fast development of the Internet led to the sharp growth of number of electronic documents. According to the experts, now about 70% of the digital information which is saved up and used by society is in an unstructured (text) form and only 30% make other types of data. The increase in number of unstructured data exponential eventually led in essence to collapse of traditional system of receiving and distribution of text information, turned routine operation of search and the analysis of necessary data into the labor-intensive and ineffective process causing information overload of users. In this situation special relevance is gained by works on creation of systems of processing of text information as even highly skilled experts experience difficulties on the organization of search of documents and distribution of the obtained text data on subjects. Documents today have an increasing variety of uses, and because of the involvement of computers in both production and analysis, may be encountered in many forms [9].

II. LITERATURE REVIEW

Nowadays the rapid increase in unstructured data, due to the expanding the Internet, has renewed and intensified the interest in document classification and text mining [14]. Researches on Text classification relate to many domain of human activity. Tran, Moon, Le, & Thoma described the process of classifying medical articles from on-line journals using constraint satisfaction method [20]. Amato et al., (2015) applied and compared different techniques, in particular explicit-rules, machine learning, and linear discriminant analysis based methods to classify a real time data collection of Web employment offers gathered from various heterogeneous sources with a standard job classification system [2].

Additional learning systems have been in exploration for managing text classification. Another interest in text classification lays in document representation. For instance, a semantic net for document representation in a five-dimensional space was proposed by M. Lipshutz and S. Liebowitz Taylor [9]. The main idea of their document classification system was based on decomposition of documents into physical, logical, functional components, topical organization, and document class. They developed the classifier which clusters documents by type such as newspapers, business letters, and technical journals.

Despite the many approaches to solve classification problems the dominant approach is machine learning technique. The advantages of machine learning systems are the efficiency, accuracy, performance, and usability to different domains [14]. The main aspects of machine learning paradigm issues such as document representation, classifier construction, and classifier evaluation were surveyed by Sebastiani [14]. Nowadays classifiers constructed using machine learning tools achieve impressive levels of efficiency and accuracy, bringing to automated classification high quality comparable to manual classification. Many researches of recent years prove that this tendency has been keeping since nowadays.

A. Review on Text Classification Methods

Among the existing text classification methods that are emphasized in the previous works are K-nearest neighbors (KNN), Naïve Bayes and Term Graph Model. The comparison of Naïve Bayes, Term Graph and KNN for Text and Document classification made a conclusion that KNN method shows the high accuracy as compared to the Naïve Bayes and Term-Graph algorithms [3]. Although, KNN has a disadvantage that its performance is low it is widely used in text classification due to fully dependence on every sample in the training set [6].

978-1-5090-6332-1/17/$31.00 ©2017 IEEE

Authorized licensed use limited to: Frankfurt University of Applied Sciences. Downloaded on January 22,2023 at 21:07:16 UTC from IEEE Xplore. Restrictions apply.

In KNN algorithm, to measure a document relevancy to a given query is the Euclidean distance between the query vector and the document vector. This metrics is modestly successful. The ways of improving on it are described by Lars Elden that shows of replacement of the term document matrix by a low-rank approximation in an attempt to capture the important information and discard the irrelevant details [10]. It was found that KNN shows the best result with accuracy among Naïve Bayes, Term Graph and KNN algorithms for Text and Document classification tasks [3]. However, comparison of Naive Bayes, K-Nearest Neighbors and Support Vector Machine classification methods applied to predict user's personality based on texts written on Twitter shows that Naive Bayes method performs better than the other methods [13]. This is because Naive Bayes uses pure probability calculations on existing features. In certain cases the KNN algorithm shows a lower speed and applicability to text categorization because of high dimension of text vectors [23]. This indicate that KNN requires more time for classifying documents when a large number of training examples are given.

B. Naïve Bayes

Naïve Bayes Classifier (NBC) is one of examples using methods of vector analysis. It is based on concepts of conditional probability of relevance of document d to a class of c. NBC is one the most frequently used classifiers due to its relatively simplicity in implementation and testing. At the same time NBC demonstrates quite good results in comparison with other more complicated classifiers [16].

NBC is based on the Bayes theorem (formula):

$$P\ (A \mid B\ ) = \frac{P(B \mid A)P(A)}{P(B)}$$

For the given model a document is a vector d ={ w1, w2 ,…, wn}, where wi – weight of i-th term, and n – the size of dictionary of a sample set. Thus, according to the Bayes theorem the probability of a class c for a document d is:

$$P\ (c \mid d\ ) = \frac{P(c)P(d \mid c)}{P(d)}$$

Thus, it is computed the conditional probability for all classes.

C. *Term Graph*

The main idea of Term Graph method is to represent a text document as a relational tuple using the well known vector space model. Text classification process is preceded by preliminary preprocessing which involves parsing, cleaning and stemming of the text document. As a result a list of terms with corresponding frequencies is obtained. Then in order to represent the document a corresponding vector can be built. Thus, a collection of documents can be represented by a term document matrix, which can be subsequently interpreted as a relational table. However, the vector space model saves only key features of the document does not take into consideration the relationship among terms [21].

D. *The KNN algorithm*

K-nearest neighbors (KNN) is widely used classification technique. It is commonly used for its easy of interpretation and low calculation time. The choice of the parameter k is very crucial in this algorithm. The training error rate and the validation error rate are two parameters which are needed to be accessed on different k value. The domain of text classification using KNN algorithm is very wide. For instance, in order to involve citizens to the process of city development, a system that addressed to classify complaints of citizens was proposed to achieve good governance and democratic [19]. Another example is theoretical background of application of KNN algorithm which is used to forecast economic events such as stock market, currency exchange rate, bank bankruptcies, financial risk, trading futures, credit rating, loan management as well as bank customer profiling [7].

There are various modifications of KNN algorithm. It was shown that a flexible K-Nearest Neighbours algorithm with combination of K-variable algorithm and weighting algorithm enhances the efficiency of text classification [24]. Another modification of KNN algorithm is a combination of eager learning with KNN classification [4] which improved the efficiency and increased the accuracy of classification. A novel KNN classification algorithm combining model and evidence theory helps to overcome the shortage of lazy learning in traditional KNN method such as time-consuming [5].

Based on the discussion above, it shows that many researchers have tried various methods to combine different classification approaches to increase the classification accuracy and reduce time consumption. Despite the KNN algorithm is easy to use and effective in general, the performance of KNN algorithm is depends mostly on the allocation of the training set. Taking into account that the textual data distribution is uneven, it was proposed to use a modified KNN algorithm based in integration the density of the test sample and the density of its nearest neighbors [8, 15]. This is to decrease the effect of the uneven data distribution to the classification they amplify the distance between the test sample and samples in the sparse area and reduce the distance between the test sample and samples in the dense area. To solve the problem of the uneven distribution of training samples, it was presented an algorithm based on clustering the training samples making a relatively uniform distribution of training samples [25].

In order to increase the performance of KNN classifier TFKNN(Tree-Fast-K-Nearest-Neighbor) method based on similarity search tree was presented [22]. This approach shows how to search the exact k nearest neighbors and improve time consuming.

Another issues in text classification are vague and uncertain nature of the boundaries of the categories which results high time complexity and low efficiency of KNN algorithm. One of the tool used avoiding the uncertainty is rough set method. By dividing text vector spaces into certain and uncertain areas and then applying correlation analysis for uncertain areas it is possible to enhance the efficiency and accuracy of text classification [1]. The classification accuracy in KNN classifier can be significantly improved if to identify neighbors

Authorized licensed use limited to: Frankfurt University of Applied Sciences. Downloaded on January 22,2023 at 21:07:16 UTC from IEEE Xplore. Restrictions apply.

with trustable labels 4. Combination of KNN algorithm with other classifiers such as C4.5 algorithm, Naive Bayes classifier and SVM demonstrated better utility and feasibility [23]. The KNN algorithm can be applied to email classification [12]. Applicability of KNN algorithm in contexts (e.g. with the World Wide Web) where the volume of documents as well as the size of the vocabulary are high, was proven [17].

In this paper we considered some researches devoted to the KNN algorithm in order to understand that the method does not still lose its actuality. No doubt that it has drawbacks in comparison with other classifiers, but it is needed to take into account that it is easy in implementation. Despite numerous attempts to modify it and achieve the reduction of time consuming the conventional version of the algorithm remains indispensable today. The proof of this fact is its implementation in scripting languages such as Python , R and etc. The R language is the most popular tool for researchers in the field of data mining.

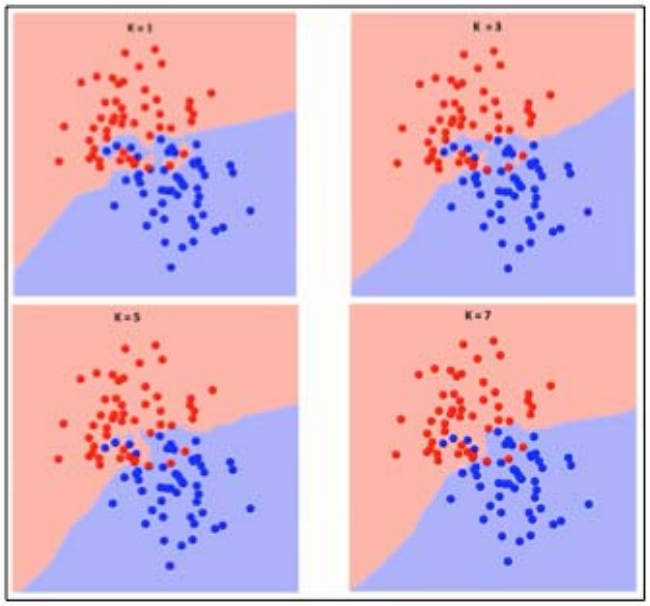

Figure 1. Boundaries separating the two classes with different values of k. (source: http://www.analyticsvidhya.com/blog/wp-content/uploads/2014/10/K-judgement2.png)

Figure 1 shows that the boundary becomes smoother with increasing value of k. The challenging thing of the KNN algorithm is to find the proper value of k. For example, if k is equal to 1, then the algorithm is simply called the nearest neighbor algorithm. The KNN algorithm is very easy to implement and need only two parameters. Description of the KNN algorithm was explored as well studied and lazy learning method [11]. The performance of the classifier depends only on two parameters, k and similarity or dissimilarity. The KNN classification is also used for test set from training set. For every row of the test set given the k nearest (using Euclidean metric) training set vectors are calculated. It is classified by a majority vote, with ties broken arbitrarily. If there are ties for the k-the nearest vector, all candidates are included in the vote.

III. Methodology

All these methods are good enough and investigated in sufficient manner. The choice of KNN classifier was made due to its implementation in R environment. We are going to continue with integrating our classifier with Hadoop in the future work.

R is a programming language for statistical data processing and free software environment [18].

Methods of classification of text documents lie on a joint of two areas - information search and machine learning. The general parts of two of these approaches - ways of submission of documents and ways of an assessment of quality of classification of texts, and distinctions consist only in ways actually of search.

In spite of the fact that problems of classification of text documents are in the center of attention of a number of research teams, on many questions is still not found satisfactory answers. Accuracy of various methods significantly depends on implementation of aprioristic assumptions and assumptions, and also structure of text data. Figure 2 depicts a generic process model for a text classification application using machine learning algorithms.

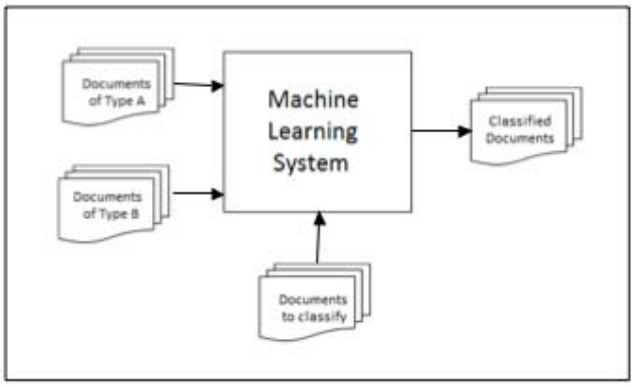

Figure 2. An example of text classification (Adopted from R. Feldman, J. Sanger(2007)).

In R language the KNN algorithm is used as follows:

```
55  # knn(train, test, cl, k = 1, l = 0, prob = FALSE, use.all = TRUE)
56  # where,
57  # train is a matrix or data frame of training set cases;
58  # test is a matrix or data frame of test set cases.
59  # cl is a factor of true classifications of training set;
60  # k is a number of neighbors considered;
61  # l is a minimum vote for definite decision, otherwise doubt.
62  # prob is a parameter which  can have true or false value.
63  # use.all controls handling of ties.
```

Figure 3. Syntax of KNN function in R language.

IV. Results

In this research the experimental evaluation is conducted as a document classifier for the following sources of information: i) Governmental news stream and ii) e-Government news stream. We collected some news articles aggregated in the

Authorized licensed use limited to: Frankfurt University of Applied Sciences. Downloaded on January 22,2023 at 21:07:16 UTC from IEEE Xplore. Restrictions apply.

English sites of government.kz and egov.kz for a period of time in 2015-2016.

For this experiment, a few steps required:

Step 1. Conduct text analysis of two articles collected from two sites:

http://egov.kz/cms/en/news/damu_new

http://www.government.kz/en/novosti/1001266-b-sagintayev-summarize-results-of-visiting-session-of-land-reform-commission-in-akmola-region.html

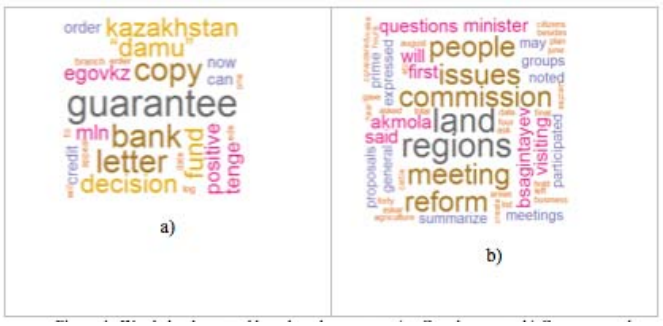

Figure 4. Word clouds created based on documents: a) e-Gov document; b) Government document.

Word cloud technique is used to select the most frequently used terms in a text. A text mining package (tm) and word cloud generator package (wordcloud) are made available in R environment for analysis of texts and visualization of the frequent terms as a word cloud.

Step 2. Analyze frequencies of terms (refer to Figure 5).

During this step, the documents are modified into a more easily managed representation. Commonly, a vector of terms and their frequencies represents a document.

Consider the following selection of five documents. Key words, which we call terms, are marked in boldface.

Document 1: The guarantee of the "Damu" fund can be acquired through eGov.kz

Document 2: The competition for development of open- data-based mobile applications announced

Document 3: New information is published on the Open Data portal

Document 4: The project "Data Centers for government agencies" of the Republic of Kazakhstan was recognized as the best in ensuring the information infrastructure

Document 5: Today, April 12, 2016, is 10 years since the launch of e-Government Portal of Kazakhstan

Counting the frequency of terms in every document we get the accompanying output:

Table1. Conversion of documents into term-document matrix

Term	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5
competition	0	1	0	0	0
data	0	1	1	1	0
eGov	1	0	0	0	0
fund	1	0	0	0	0
government	0	0	0	1	1
guarantee	1	0	0	0	0
information	0	0	1	1	0
infrasructure	0	0	0	1	0
portal	0	0	1	0	1
Kazakhstan	0	0	0	1	1

In this manner every document is represented by a vector, or a point, in R10, and we can sort out all documents into a term-document matrix:

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Now assume that we need to discover all documents that are pertinent to the request "The portal for open discussion of draft legislations launched in Kazakhstan". This is represented by a request vector, constructed in a way analogous to the term-document matrix:

$$q = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} \in R^{10}.$$

Thus the query itself is considered as a document. The text classification task can now formulated as a mathematical problem: find the columns of A that are close to the vector q. To solve this problem we must use some distance measure in R10.

Authorized licensed use limited to: Frankfurt University of Applied Sciences. Downloaded on January 22,2023 at 21:07:16 UTC from IEEE Xplore. Restrictions apply.

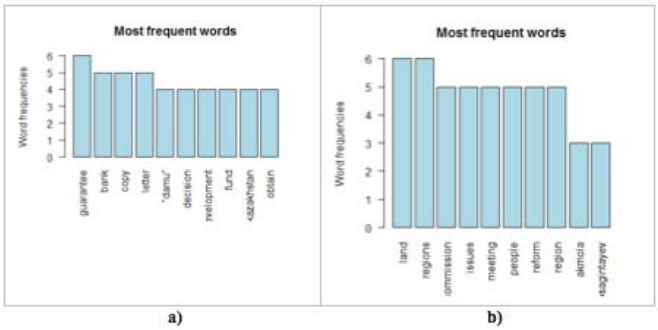

Figure 5. Term frequencies: a) for an e-Gov document; b) for a Government document.

Step 3. Deploy the KNN algorithm by training the model with "known" data (i.e. previously classified data) and test it on "unknown" data (i.e. data where the class should be determined by the classifier).

The important stage in text analysis is preprocessing texts that will be done after loading all documents. This stage will remove numbers, capitalization, stop words, and punctuation in order to prepare texts for analysis.

```
#init
libs <- c("tm", "plyr", "class")
lapply(libs, require, character.only = TRUE)

#set options
options(stringsAsFactors = FALSE)

#set parameters
categories <- c("GovernmentKZ", "eGovKZ")
pathname <- "C:/ClassifyTextwithR"

#clean text
cleanCorpus <- function(corpus){
  corpus.tmp <- tm_map(corpus, PlainTextDocument)
  corpus.tmp <- tm_map(corpus.tmp, removePunctuation)
  corpus.tmp <- tm_map(corpus.tmp, stripWhitespace)
  corpus.tmp <- tm_map(corpus.tmp, content_transformer(tolower))
  corpus.tmp <- tm_map(corpus.tmp, removeWords, stopwords("english"))
  return(corpus.tmp)
}
```

Figure 6 shows the fragment of the code for cleaning corpus.

Figure 6. The code of the function for cleaning corpus.

The next stage of the text classification is creation of the term-document matrix. A term-document matrix (TDM) is constructed with one row for each term and one column for each document. The (i, j) element of the TDM is positive if and only if the i-th term appears in the j-th document. The value of this element can be weighted by the significance of the term and/or the document. Thus each and every document is represented by a vector, a column of TDM.

```
#build TDM
generateTDM <- function(cat, path) {
  s.dir <- sprintf("%s/%s", path, cat)
  s.cor <- Corpus(DirSource(directory = s.dir, encoding ="UTF-8"))
  s.cor.cl <- cleanCorpus(s.cor)
  s.tdm <- TermDocumentMatrix(s.cor.cl)
  s.tdm <- removeSparseTerms(s.tdm, 0.7)
  result <- list(name = cat, tdm = s.tdm)
}

tdm <- lapply(categories, generateTDM, path = pathname)

#attach name
bindCategoryToTDM <- function(tdm) {
  s.mat <- t(data.matrix(tdm[["tdm"]]))
  s.df <- as.data.frame(s.mat, stringsAsFactors = FALSE)

  s.df<- cbind(s.df, rep(tdm[["name"]], nrow(s.df)))
  colnames(s.df)[ncol(s.df)] <- "targetcategory"
  return(s.df)
}
 catTDM <- lapply(tdm, bindCategoryToTDM)
```

Figure 7. The code of the function for generating the term-document matrix.

Requests are also documents and can be likewise represented as vectors. To figure the relevance of a document to a given request it is used the Euclidean distance between the request vector and the document vector is used. This metric is then implemented in the KNN method.

```
#stack
tdm.stack <- do.call(rbind.fill, catTDM)
tdm.stack[is.na(tdm.stack)] <- 0


#hold-out
train.idx <- sample(nrow(tdm.stack), ceiling(nrow(tdm.stack) * 0.7))
test.idx <- (1:nrow(tdm.stack))[-train.idx]

#model - KNN
tdm.cat <- tdm.stack[,"targetcategory"]
tdm.stack.nl <- tdm.stack[, !colnames(tdm.stack) %in% "targetcategory"]
knn.pred <- knn(tdm.stack.nl[train.idx, ], tdm.stack.nl[test.idx, ],
                tdm.cat[train.idx], k=50)

#accuracy

conf.mat <- table("Predictions" = knn.pred, Actual = tdm.cat[test.idx])
(accuracy <- sum(diag(conf.mat)) / length(test.idx) * 100)
```

Figure 8. Invocation of the function knn to classify text documents.

We have experimented with the value of k. In the figure 9 it is given the results of accuracy of classification obtained at values of k from 1 to 50. The worst case of the accuracy is about 88. We computed classification accuracy when the value of k is more than 50. In that cases the accuracy is under 50 % which indicate the impact of the value of k to the accuracy of classification.

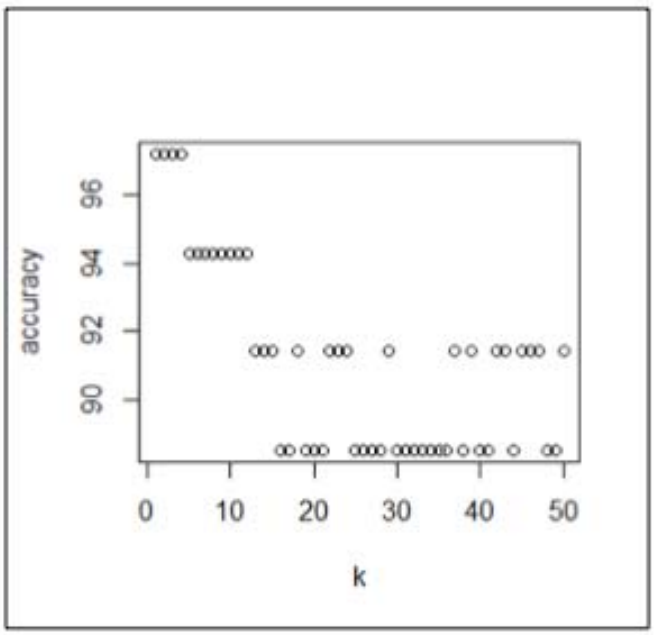

Figure 9. The tendency of accuracy of classification at k from 1 to 50.

Authorized licensed use limited to: Frankfurt University of Applied Sciences. Downloaded on January 22,2023 at 21:07:16 UTC from IEEE Xplore. Restrictions apply.

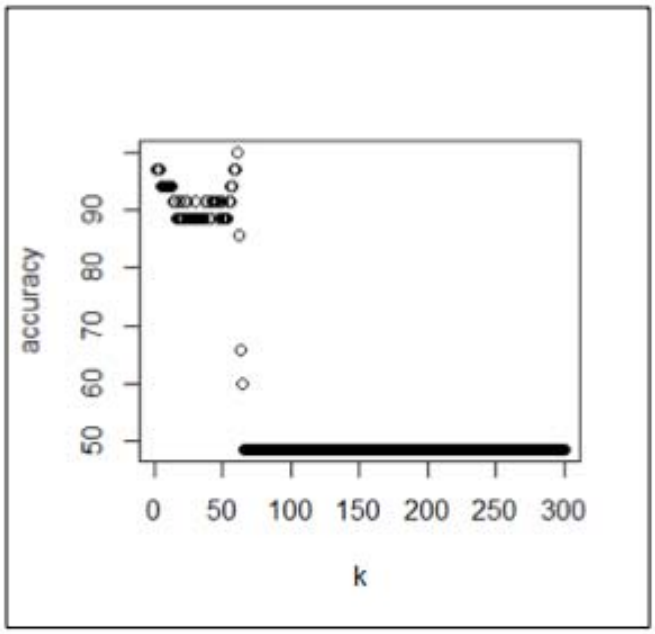

Figure 10. The tendency of accuracy of classification at k from 1 to 300.

V. Discussion

A wide range of tasks, which may use the classification techniques, and the availability of large collections of documents in electronic form that can be used as training corpus, caused a lively interest to text classification area from the part of many applied disciplines. Text classification is used in many domains, from automatic indexing of documents to document filtering, automatic metadata generation, expanding the hierarchical directory of web resources, and structuring documents. For example, text classification can be used for classifying articles from on-line journals, Web employment offers and so on. Office workers spend their working hours on routine, non-automated work related to the necessity of processing unstructured data: emails, memos, news, chat, reports, marketing materials, presentations, and other documents that don't fit properly in relational databases, they can be stored as text files in various formats, and these kinds of files may have an internal structure. Moreover, text classification methods can be addressed to issues on analysis of text documents for the subject to containing certain criteria with further identifying duplicate documents, documents delaying the implementation of governmental assignments, documents containing requests for background information and reports which require a large amount of working hours thereby diverting employees from the primary work on processes of responding letters with minor contents.

In this research, we started with experiments on classification of documents from two sources: http://egov.kz and http://www.government.kz using KNN algorithm in R language. The KNN classifier was chosen due to its easy implementation in R environment and comparatively good efficiency. In its own turn the R language was selected in virtue of its integration with Hadoop for the future work. In order to research how the classifier based on KNN algorithm works on collected documents we experimented with the value of k. In the Table 2 it is given the results of accuracy of classification obtained at different values of k. At $k \leq 50$ the accuracy is above 88 which means the documents are trusted. When the value of $k \geq 50$ the accuracy is under 50 % which indicate the impact of the value of k to the accuracy of classification is lesser.

Table2. Accuracy of classification

S. No	k	Accuracy
1	1	100
2	5	97,14
3	25	94,28
4	50	88,87
5	51	48,57
6	75	48,57
7	300	48,57

VI. Conclusions

This paper presented an approach for building a machine learning system in R that uses K-Nearest Neighbors (KNN) algorithm for the classification of textual documents. KNN is one the most popular classifiers, easy to use and efficient enough. The challenging thing of the KNN algorithm is to find the proper value of k which represents the number of neighbors. The experimental results shows the best accuracy percentage when the value of k varies from 1 to 50. Above the 50 the accuracy falls sharply. As the R environment is integrated with Hadoop the future work will be devoted to the large amount of documents. High accuracy is importance in government documents as they are improving the quality of provision of public services to the population.

Acknowledgment

I would like to thank my co-author and research supervisor Dr. Rosnafisah Bte. Sulaiman for guiding me during the writing this paper and giving me her valuable comments. I am also deeply grateful to my university where I work and in particular to my second supervisor Professor Raissa Uskenbayeva for her involving me in research in Big Data area.

References

[1] Aizhang, G. (2015). Based on rough sets and the associated analysis of KNN text classification research, (3). doi:10.1109/DCABES.2015.127

[2] Amato, F., Boselli, R., Cesarini, M., Mercorio, F., Mezzanzanica, M., Moscato, V., … Picariello, A. (2015). Challenge: Processing web texts for classifying job offers. Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing, IEEE ICSC 2015, 460–463. doi:10.1109/ICOSC.2015.7050852

Authorized licensed use limited to: Frankfurt University of Applied Sciences. Downloaded on January 22,2023 at 21:07:16 UTC from IEEE Xplore. Restrictions apply.

[3] Bijalwan, V., Kumar, V., Kumari, P., & Pascual, J. (2014). KNN based Machine Learning Approach for Text and Document Mining, 7(1), 61–70.

[4] Dong, T., & Cheng, W. (2012). The Research of kNN Text Categorization Algorithm Based On Eager Learning, (d), 1120–1123. doi:10.1109/ICICEE.2012.297

[5] Guo, G., Ping, X., & Chen, G. (2006). A Fast Document Classification algorithm based on improved KNN, 3–6.

[6] Hassanat, A. B., Abbadi, M. A., & Alhasanat, A. A. (2014). Solving the Problem of the K Parameter in the KNN Classifier Using an Ensemble Learning Approach. International Journal of Computer Science and Information Security (IJCSIS), 12(8), 33–39. doi:10.1007/s00500-005-0503-y

[7] Imandoust, S. B., & Bolandraftar, M. (2013). Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events : Theoretical Background, 3(5), 605–610.

[8] Li, L., & Che, Y. (2011). KNN Text Categorization Algorithm Based On LSA Reduce Dimensionality. Proceedings of IEEE CCIS2011.

[9] Liebowitz, S., & Problem, T. H. E. (1997). Document Representation, 25(4), 85–93.

[10] Lars Elden. (2007). Matrix Methods in Data Mining and Pattern Recognition. SIAM, Philadelphia, PA, 224 pp., ISBN 978-0-898716-26-9.

[11] Nidhi, & Gupta, V. (2011). Recent Trends in Text Classification Techniques. International Journal of Computer Applications, 35(6), 45–51.

[12] Nikhath, A. K., Subrahmanyam, K., & Vasavi, R. (2016). Building a K-Nearest Neighbor Classifier for Text Categorization, 7(1), 254–256.

[13] Pratama, B. Y., & Sarno, R. (2015). Personality classification based on Twitter text using Naive Bayes, KNN and SVM. 2015 International Conference on Data and Software Engineering (ICoDSE), 170–174. doi:10.1109/ICODSE.2015.7436992

[14] Sebastiani, F. (2002). Machine Learning in Automated Text Categorization, 34(1), 1–47.

[15] Shi, K., Li, L., Liu, H., He, J., Zhang, N., & Song, W. (2011). An improved KNN Text classification algorithm based on density. Proceedings of IEEE CCIS2011.

[16] Shimodaira, H. (2015). Text Classification using Naive Bayes, (4).

[17] Soucy, P., & Mineau, G. W. (2001). A Simple K" Algorithm for Text Categorization, 647–648.

[18] Theußl, S., & Hornik, K. (2012). A tm Plug-In for Distributed Text Mining in R, 51(5).

[19] Tjandra, S., Alexandra, A., & Warsito, P. (2015). Determining Citizen Complaints to The Appropriate Government Departments using KNN Algorithm, 2–5.

[20] Tran, L. Q., Moon, C. W., Le, D. X., & Thoma, G. R. (2001). Web Page Downloading and Classification. Proceedings 14th IEEE Symposium on Computer-Based Medical Systems. CBMS 2001, 321–326. doi:10.1109/CBMS.2001.941739

[21] Wang, L., & Zhao, X. (2012). Improved KNN classification algorithms research in text categorization, i, 1848–1852.

[22] Wang, Y. U., & Wang, Z. (2007). A fast KNN algorithm for text categorization, (August), 19–22.

[23] Yan, Z. (2010). Combining KNN Algorithm and Other Classifiers, (1), 1–6.

[24] Yunliang, Z., Lijun, Z., Xiaodong, Q., & Quan, Z. (2009). Flexible KNN Algorithm for Text Categorization by Authorship based on Features of Lingual Conceptual Expression, 601–605. doi:10.1109/CSIE.2009.363

[25] Zhou, L., & Wang, L. (2010). A Clustering-Based KNN Improved Algorithm CLKNN for Text Classification, 4–7.

Authorized licensed use limited to: Frankfurt University of Applied Sciences. Downloaded on January 22,2023 at 21:07:16 UTC from IEEE Xplore. Restrictions apply.