KNN with TF-IDF based framework for text categorization

Conference Paper *in* Procedia Engineering · November 2013

DOI: 10.1016/j.proeng.2014.03.129

3 authors, including:

Trstenjak Bruno
Polytechnic of Medimurje in Cakovec
8 PUBLICATIONS **48** CITATIONS

SEE PROFILE

Dzenana Donko
University of Sarajevo
48 PUBLICATIONS **120** CITATIONS

SEE PROFILE

Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 69 (2014) 1356 – 1364

Procedia Engineering

www.elsevier.com/locate/procedia

24th DAAAM International Symposium on Intelligent Manufacturing and Automation, 2013

KNN with TF-IDF Based Framework for Text Categorization

Bruno Trstenjak[a]*,Sasa Mikac[b], Dzenana Donko[c]

[a]*Dept. of Computer Engineering, Medimurje University of Applied Sciences Cakovec, Croatia*
[b]*Dept. of Computer Science, Faculty of Electrical Engineering and Computer Science,Maribor, Slovenia*
[c]*Dept. of Computer Science, Faculty of Electrical Engineering, Sarajevo, Bosnia and Herzegovina*

Abstract

KNN is a very popular algorithm for text classification. This paper presents the possibility of using KNN algorithm with TF-IDF method and framework for text classification. Framework enables classification according to various parameters, measurement and analysis of results. Evaluation of framework was focused on the speed and quality of classification. The results of testing showed the good and bad features of algorithm, providing guidance for the further development of similar frameworks.

Keywords: text documents classification; K-Nearest Neighbor; TF-IDF; framework ; machine learning

1. Introduction

Online document classification or document categorization is a problem in today's library science, information and computer science. In the past 20 years, the number of text documents in digital form has grown exponentially [4]. As a consequence of exponential growth, great importance has been put on the classification of documents into groups that describe the content of the documents. The function of classifier is to merge text documents into one or more predefined categories based on their content [9]. Each document can belong to several categories or may present its own category. Very rapid growth in the amount of text data leads to expansion of different automatic methods aimed to improve the speed and efficiency of automated document classification with textual content. The documents to be classified can contain text, images, music, etc. Each content type requires special classification methods. This article is focused on the textual content documents with special emphasis on text classification. Text

* Corresponding author. Tel.: +385 40 328 246
*E-mail address:*btrstenjak@mev.hr

doi:10.1016/j.proeng.2014.03.129

classification is one of the problems of text mining. Text mining is an interdisciplinary field that draws on information retrieval, data mining, machine learning, statistics, and computational linguistics [5].

In this article the working principle of framework for text classification based on using KNN algorithm and TF-IDF method will be explained. A Framework allows measuring the document similarity on the basis of given textual pattern. Selecting documents on local or web-based environment, the framework performs the documents classification and statistical analysis of the results. In the classification of documents a modified original algorithm optimized at implementation level using Language-Integrated Query (LINQ) and C Sharp programming language are used.

The K-Nearest Neighbor (KNN) is one of the simplest lazy machine learning algorithms [13,14]. Algorithm objective is to classify objects into one of the predefined classes of a sample group that was created by machine learning. The algorithm does not require the use of training data to perform classification, training data can be used during the testing phase. KNN is based on finding the most similar objects (documents) from sample groups about the mutual Euclidean distance [1,11].

Term frequency-inverse document frequency (TF-IDF) is a numerical statistic method which allows the determination of weight for each term (or word) in each document. The method is often used in natural language processing (NLP) or in information retrieval and text mining [17,18]. The method determines the weight, measure which evaluates importance of terms (or words) in document collection. The importance of the text is increased proportionally to the number of appearing in the documents [2,3].

2. KNN & TF-IDF Framework

2.1. General information

The Framework is designed to enable the classification and measurement of similarity of documents based on the required text sample. It consists of several modules that take users through the process of classification. It consists with several modules who lead users through the process of classification. Implementation of the Framework is realized in an object-oriented development environment with the programming language C Sharp.

2.2. Framework structure

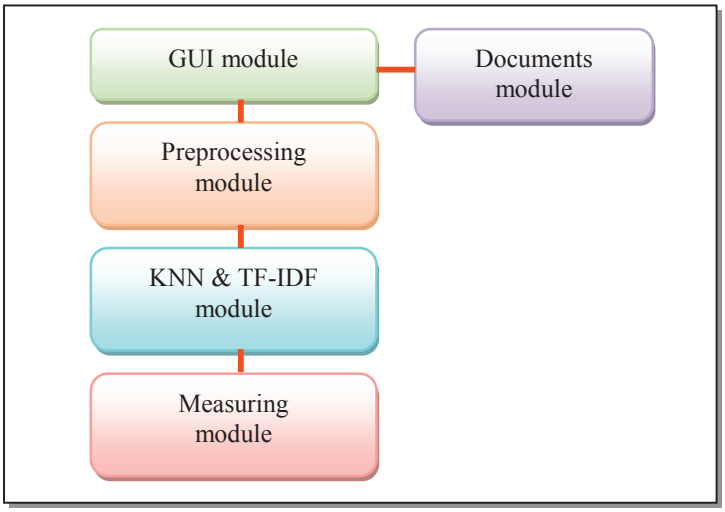

Fig. 1. Framework structure.

As previously noted, the framework consists of several separate modules. Fig. 1 shows the structure of the new concept of Framework.

The framework has five basic modules: GUI module, Documents module, Preprocessing module, KNN & TF_IDF module, Measuring module. GUI module allows user to control the application and the entire framework. The Documents module is designed for document management and selecting data resources. Document resources can be located on the local computer or on the Internet. In this module a user can define document categories, the feature which influences the final results of text classification. The Preprocessing module checks the document characteristics, prepares and adjusts them into a format suitable for classification. The module adjusts documents to classic text format, regardless of the type of font and character set. The documents with a specific format module automatically remove the control characters that might have a negative impact on the result of classification. The main module in the framework is the KNN & TF_IDF module. The module contains the main methods for classification and determination of the document weight value. The results of the classification forward to the last module for the measurement and presentation statistical indicators. The last module allows displaying the classification results and their simple statistical analysis.

2.3. KNN classifier

As previously mentioned the algorithm is based on the machine learning. Preprocessing and document preparation is followed by the learning phase. The algorithm determines the basic documents which will be comparing with each new document. Algorithm checks where a document is categorized by only looking at the training documents that are most similar to it.

The algorithm assumes that it is possible to classify documents in the Euclidean space as points. Euclidean distance is the distance between two points in Euclidean space. The distance between two points in the plane with coordinates p=(x, y) and q=(a, b) can be calculated[19]:

$$d(p,q) = d(q,p) = \sqrt{(x-a)^2 + (y-b)^2} \quad (1)$$

2.4. Learning process

Learning process starts with parsing the basic text which searches the words in documents and forms a vector [10]. Parsing process removes all control characters, spaces between words, dots, commas, and similar characters. The formed vector represents a fundamental object that will be used for classification of tested documents. Example 1 below illustrates the procedure for preprocessing search text and the formation of the main vector.

Example 1:
Search text: “Text classification, KNN method.”
Preprocessing: Text classification KNN method
Base vector: mainVec = [Text classification KNN method]
mainVec[0]= Text
mainVec[1]= classification
mainVec[2]= k-NN
mainVec[3]= method

Structure of the main vector object shown in Fig. 2.

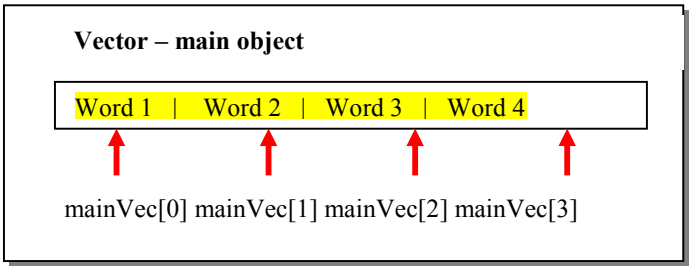

Fig. 2. Graphical presentation of vector.

2.5. Determination of the weight matrix

To provide text classification and searching the documents it is necessary to establish the weight matrix. The matrix contains the values of relations between each unique words and documents. It is the initial object in the algorithm to calculate the individual importance (weight) of each searched document. Each document is represented as a vector in n-dimensional vector space. We can imagine a matrix A with dimensions NxM, where N dimension is defined by a number of unique words in a sample of all documents. M represents the number of documents to be classified. Weight matrix can be characterized as a relational matrix of word - document. Dimension of the matrix is equal to product, the number of different unique words and the total number of documents. Each matrix element a_{ij} represents weight value of word i in the document j. Weight matrix is shown in Fig. 3.

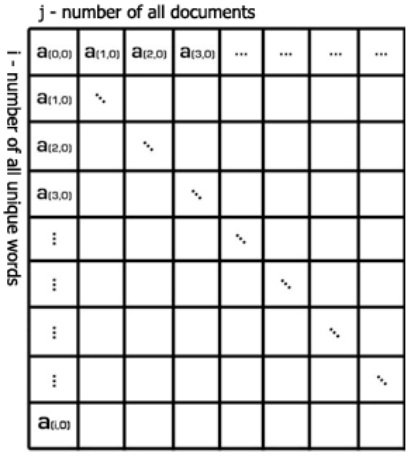

Fig. 3. Weight matrix.

In determining the weight values in the matrix, we can use different metrics and methods of calculation.

2.6. Determination of the weight matrix

Term binary
Binary method is extremely simple and easy to implement. The method checks if a particular word (term) appears in the document. The values in the matrix can be 0 or 1. If the word appears in the document, then the weight value a_{ij} is set to 1, otherwise is set to 0.

Term frequency
The method computes the number of repetitions of a word (term) in the document j [4].

$$a_{ij} = f_{ij} = frequency\ of\ term\ i\ in\ document\ j \quad (2)$$

Some text classification algorithms performed normalize term frequency by dividing with the frequency of the most common term in the document.

$$a_{ij} = f_{ij} \,/\, max_i\{f_{ij}\} \quad (3)$$

Term frequency – inverse document frequency (TF-IDF)
A very popular research method in the field of natural language processing (NLP) which is used in the implementation of the algorithm described in this article. TF-IDF method determines the relative frequency of words in a specific document through an inverse proportion of the word over the entire document corpus. In determining the value, the method uses two elements: **TF** - term frequency of term i in document j and **IDF** - inverse document frequency of term i. In our research and testing the algorithm of framework this method showed good results. TF-IDF can be calculated as [16]:

$$a_{ij} = tf_{ij}idf_i = tf_{ij} \times log_2\left(\frac{N}{df_i}\right) \quad (4)$$

where a_{ij} is the weight of term i in document j, N is the number of documents in the collection, tf_{ij} is the term frequency of term i in document j and df_i is the document frequency of term i in the collection. This formula implemented in the framework, during the testing has shown good results when we used the documents with equal length. In order to obtain better results with documents of different length, we used a modified equation as shown [7]:

$$a_{ij} = tf_{ij}idf_i = \frac{f_{ij}}{\sqrt{\sum_{s=1}^{N}\left(tfidf(a_{sj})\right)^2}} \times log_2\left(\frac{N}{df_i}\right) \quad (5)$$

Shown below is the experimental pseudo code for the method implemented in the framework for determining the weight matrix.

```
Pseudo code 1:
for(i=0   i<numnberOfUniqueWordsi++)
  for(j=0   j<numberOfDocuments j++)
      tfidf = fij _ log(numberOfDocuments = ni)
      for(s=0 s<numnberOfUniqueWords s++)
         fijTemp = number_of_occurrences_ofword_S_in_the_document_J
         tfidfTemp = fijTemp * log(numberOfDocuments / dfi)
         summTfidf += (tfidfTemp)^2
```

```
      end
      A[i,j] = tfidf/summTfidf
    end
  end
```

In order to limit matrix dimension and optimize the framework work, it is possible to include preprocessor and remove useless words, which often repeat in documents and do not contain any useful information.

2.7. Text document KNN classification process

At the beginning of the classification it is necessary to select the document that will be carried out for classification and include it in the belong category [9]. For the selected document its weight value also must be determined by TF-IDF method, as well as for all other documents. Classification process writes data of the selected document in a weight matrix to its end, as shown in Fig. 4. Writing down all data in the same weight matrix has resulted with optimization and reducing the total time of calculation.

In the next step of the process it is necessary to determine K value. K value of the KNN algorithm is a factor which indicates a required number of documents from the collection which is closest to the selected document. The classification process determinates the vectors distance between the documents by using the following equation [12]:

$$d(x,y) = \sqrt{\sum_{r=1}^{N}(a_{rx} - a_{ry})^2} \tag{6}$$

where d(x,y) is the distance between two documents, N is the number unique words in the documents collection, a_{rx} is a weight of the term r in document x, a_{ry} is a weight of the term r in document y. Pseudo code 2 shows implementation Euclidean distance calculation, and Fig. 4 shows documents in Euclidian space for factor K=3. Smaller Euclidean distance between the documents indicates their higher similarity. Distance 0 means that the documents are complete equal.

Pseudo code 2:

```
for(i=0  i<numberOfDocumentsi++)
  for(r=0  r<numnberOfUniqueWordsi++)
    d[i] += (A[r,i] - A[r,( numberOfDocument-1)])^2
  end
  d[i] = Sqrt( d[i] )
end
```

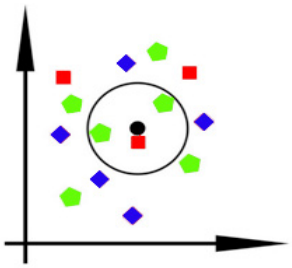

Fig. 4. Euclidian space for K=3.

3. Framework evaluation

For the purposes of this paper, this section presents the results of framework testing over the sets of 500 online documents from different categories in the learning phase and classification. Evaluation of Framework was performed in several test phases:

- test speed calculation of TF-IDF methods with optimization
- test the quality classification
- test speed text classification
- test classification sensitivity according to categories of documents

Calculation of TF-IDF values in weight matrix has shown as the most demanding part of the implemented algorithm. For this reason it was necessary to optimize exactly that part of the framework code. Optimization TF-IDF was performed using LINQ class in C # language, in order to calculate the frequency of appearance of a particular word in the document. The optimized method showed six times better results than the classic method. Thus, the optimized code is used in other test phases.

The results of testing text classification quality are shown in Fig. 5. The tests used the documents of different lengths, different categories from online sources with 500 documents [8].

Testing frameworks was performed in online environment, using different combinations and different amounts of documents [15]. The results indicate the successful implementation of algorithm in the framework whose quality of classification slightly decreases with the increasing number of documents. Additional testing and analysis showed much better obtained results if working with documents which belong to the same category, as we predicted. Fig. 6 shows the results of classification depending on the selected documents category. The best results were achieved with the documents from "Sport" category. The main reason for good results of classification in this category is that the documents have not been textually demanding. The documents did not contain a large number of different words which reduced the impact of unusable words and character.

Table 1 shows the speed measurement of classification based on the amount of data. These results directly indicate performance of the implemented algorithm. The developed framework has certain limitations on his work. The time required for data processing exponentially increased with the increasing amounts of data. One of the solutions can be implementation of parallel processing in the framework. It would allow speed up calculation of weight matrix which is the most demanding part in algorithm.

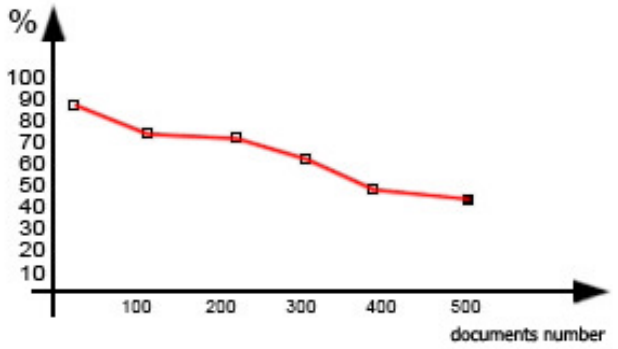

Fig. 5. The results of testing text classification quality.

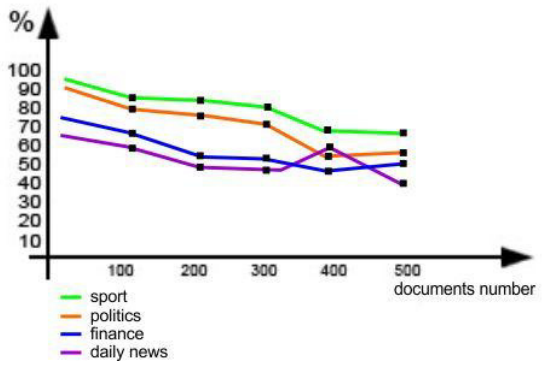

Fig. 6. The results of testing text classification quality by categories.

The last test was focused on measuring the quality of classification depending on the documents category. It was previously indicated that the quality of classification depends on the preprocessing documents, removing undesired characters and words that have no significant information. Table 2 shows the results of successful classification over the documents from various categories. The results show that the worst classification was performed in the category Daily News. The analysis of document contents in this category showed that documents contained a lot of "unusable words", the words that are often repeated and do not have important weight but have adverse impact on KNN classification.

Table 1. Speed classification.

The amount of data (kB)	Speed (*ms*)
4.00	300.00
400.00	3000.00
4000.00	45,000.00

Table 2. Category classification.

Category of documents	KNN Classification
Sport	0,92%
Politics	0,90%
Finance	0,78%
Daily News	0,65%

4. Conclusion

In this paper we present a framework for text classification based on KNN algorithm and the TF-IDF method. The main motivation for the research was to develop concept of frameworks with emphasis on KNN & TF-IDF module.

The framework with embedded methods gave good results, confirmed our concept and initial expectations. Evaluation of framework was performed on several categories of documents in online environment. Tests are supposed to provide answers about the quality of classification and to determine which factors have an impact on performance of classification. The framework work was very stable and reliable. During testing the quality of classification we have achieved good results regardless of the K factor value in the KNN algorithm.

Performed tests have detected a sensitivity of the implemented algorithm. Tests have shown that the embedded algorithm is sensitive to the type of documents. The analysis of documents contents showed that the amount of unusable words in documents has a significant impact on the final quality of classification. Because of this, it is necessary improve the preprocessing of data for achieving better results.

The combination of KNN algorithm and TF-IDF method has been shown as a good choice with minor modifications in their implementation. The framework provides the ability to upgrade and improve the present embedded classification algorithm.

References

[1] S.Tan, Neighbor-weighted K-nearest neighbor for unbalanced text corpus, Expert Systems with Applications 28 (2005) 667–671.
[2] G.Salton, C.S.Yang, On the specification of term values in automatic indexing, Journal of Documentation, 29 (1973) 351-372.
[3] W.Zhang, T.Yoshida, A comparative study of TF-IDF, LSI and multi-words for text classification, Expert Systems with Applications 38 (2011) 2758–2765.
[4] H.Han, G.Karypis, V.Kumar, Text Categorization Using Weight Adjusted k-Nearest Neighbor Classification, PAKDD (2001) 53-65.
[5] F.Sebastiani, Machine Learning in Automated Text Categorization, Consiglio Nazionale delle Ricerche, 2002.
[6] H.Jiang, P.Li, X.Hu, S.Wang, An improved method of term weighting for text classification, Intelligent Computing and Intelligent Systems, 2009.
[7] J. T.-Y. Kwok, Automatic Text Categorization Using Support Vector Machine, Proceedings of International Conference on Neural Information Processing, (1998) 347-351.
[8] M.Miah, Improved k-NN Algorithm for Text Classification, DMIN (2009) 434-440.
[9] Y.Liao, V. Rao Vemuri, Using K-Nearest Neighbor Classifier for Intrusion Detection, Department of Computer Science, University of California, Davis One Shields Avenue, CA 95616.
[10] L.Wang , X. Zhao, Improved KNN classification algorithms research intext categorization, IEEE, 2012.
[11] M.Lan, C.L.Tan, J.Su, Y.Lu, Supervised and Traditional Term Weighting Methods for Automatic Text Categorization, IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL. 31, NO. 4, 2009.
[12] K. Mikawa, T. Ishidat, M.Goto, A Proposal of Extended Cosine Measure for Distance Metric Learning in Text Classification, 2011.
[13] l.Wang , X. Li, An improved KNN algorithm for text classification, 2010.
[14] G. Guo, H.Wang, D.Bell, Y. Bi, K. Greer, KNN Model-Based Approach in Classification, (2003) 986 – 996.
[15] G. Forman, An Extensive Empirical Study of Feature Selection Metrics for Text Classification, Journal of Machine Learning Research 3 (2003) 1289-1305.
[16] H. Uguz, A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm, Knowledge-Based Systems 24 (2011) 1024–1032.
[17] K. Masudaa, T.Matsuzakib, J.Tsujiic, Semantic Search based on the Online Integration of NLP Techniques, Procedia - Social and Behavioral Sciences 27 (2011) 281 – 290.
[18] C.Friedman, T.C.Rindflesch, M. Corn, Natural language processing: State of the art and prospects for significant progress, a workshop sponsored by the National Library of Medicine, Journal of Biomedical Informatics 46 (2013) 765–773.
[19] Ming-Yang Su, Using clustering to improve the KNN-based classifiers for online anomaly network traffic identification, Journal of Networkand Computer Applications 34 (2011) 722–730.