International Journal of Computer Applications (0975 - 8887)
Volume 129 - No.12, December 2015

Categorization of 'Holy Quran-Tafseer' using K-Nearest Neighbor Algorithm

Geehan Sabah Hassan
Assistant Lecturer in Natural Language Processing
College education Ibn-Rushd
University of Baghdad (UOB)

Siti Khaotijah Mohammad
Senior Lecturer in Computational Linguistics
School of Computer Science
Universiti Sains Malaysia (USM)

Faris Mahdi Alwan
Operations Research/ Reliability and Maintenance
Statistics Department, College of Administration and Economics
University of Baghdad (UOB)

ABSTRACT

Text categorization, TC, is a process of labeling natural language texts with one or several categories from a predefined set. TC is a supervised learning where the set of categories and examples of documents belonging to those categories is given. The task of automatic TC is assigned an electronic document to several categories, based on a training set of labeled documents. The research objectives are, to formulate a K-Nearest Neighbor (KNN) algorithm for the automatic and suitable classification of any Holy Quran Tafseer segment; to identify relevant categories of Holy Quran Tafseer in the form of number classes; and to retrieve, Tafseer of verses of the Holy Quran in Malay language. Hence, this research aims to automatically categorize the Tafseer of verses of Holy Quran using the KNN algorithm as a technique to solve text categorization. This research has been designed to classify different verses in the Holy Quran. The first phase is to pre-process the Arabic text and then change the word in Arabic to Malay word. After that, categorize classes based on the cosine similarity between a test document and specific training documents. The majority of the same kind of nearest neighbors decides the category of the test sample and calculates precision and recall for a collection of documents. The result shows the outperform of TC using the KNN algorithm is one of the best algorithm for categorization Tafseer of Holy Quran. Furthermore, this study contributes in building a classifier to Tafseer Al-Quran in Malay language.

Keywords

Text categorization, K-Nearest Neighbor algorithm

1. INTRODUCTION

TC is a process of labeling natural language texts with one or several categories from a predefined set. TC is a supervised learning where the set of categories and examples of documents belonging to those categories is given. As a research area, TC appeared in the 1960s, but became a major field of information science only 15 years ago. The fast development is due to the increased interest in the diverse applications of TC which includes document indexing with controlled vocabulary, filtering of irrelevant information, web page categorization, email management, and detection of text genre among others. Text categorization techniques are necessary nowadays when most information is produced and stored digitally.

Business and personal correspondence, scientific and entertaining articles, conference proceedings, and patient data are a few examples of electronic text collections. The advent of the World Wide Web (WWW) a massive repository of text information which required automatic means of efficient and effective storage and retrieval system that can be provided through TC [16]. Such a system comprises different types of approaches that help solve many real world problems such as K-Nearest Neighbor, Rocchio?s algorithm, Decision Trees, Nave Bayes algorithm, Back propagation Network, and Support Vector Machines. Each approach implements text categorization techniques to achieve its goal [3].

2. BACKGROUND

Categorization is classifying data to improve effectiveness and efficiency. TC is one of the most widely used and significant methods of supervised learning in data mining.

Let$(d_j, c_j) \in D \times C$, where D set of documents and $C = \{c_1, c_2, \cdots, c_{|c|}\}$ is a predefined set of categories. The main mission of TC is to allocate a Boolean value to every pair in D [11]. Figure 1 shows that, various categories are found in the document domain D and set C. D includes three different types of documents, namely '#', '$' and '@,'after categorization. Every document is categorized in its own category. The following observations are worth considering:

(1) The categories are only symbolic labels. They are supposedly obtainable to assist in building the classifier. This observation implies that "text" forms the category label.

(2) The predication of documents to categories, is generally, achieved on the basis of the meaning of words in documents, and not on metadata. Document categorization should underlie

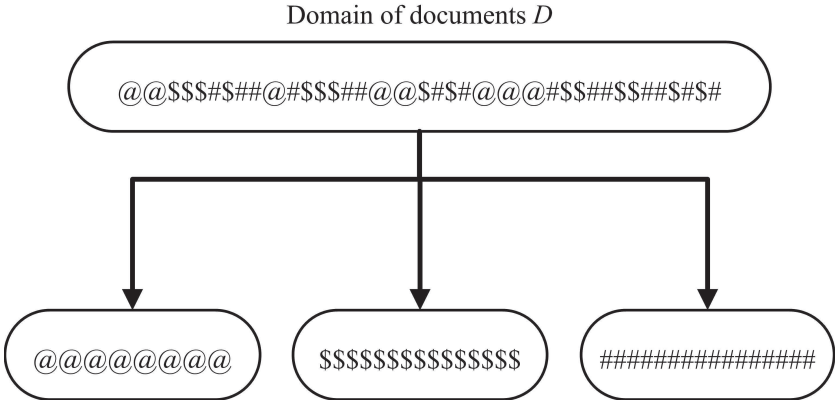

Fig. 1. Pictorial representation of categorization.

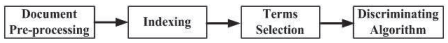

Fig. 2. This is example of the image in a column.

only on subjective knowledge and not on exogenous knowledge [3].

Figure 2 shows the general structure of a categorization system. The first stage is preprocessing and provides the document for the classification process. Text are normalized the text with tags, and stop words are removed. Words in the documents are then listed and indexed. The vocabulary is extracted, and phrases and terms may be recognized. Word stemming is optionally performed. Finally, a discriminating algorithm is invoked to distinguish between categories [9].

3. METHODOLOGY

The schema of the methodology for this study comprises three phases show in figure 3. The first phase is pre-processing and modeling involves normalization, term extraction, stemming, and using the dictionary to convert Arabic words to Malay words after using inverted index, dimensionality reduction and weighing terms (TF*IDF). The second phase addresses how to K-Nearest Neighbor algorithm to derive the solution applied to the Tafseer Holy Quran. The third phase evaluates the performance of the categorization model by using the precision and recall values.

3.1 Classifier Learning Algorithm

A text classifier for $c_i \in C$ is generated automatically by a general inductive process. This process is conducted by monitoring the properties of the document sets before classification according to c_i or $\acute{c}_i$ which obtain the properties of a new hidden document wherein c_i must belong. With the goal of building classifiers for C, a document set S is needed such that the value of $\Phi(d_i, c_i)$ is well known for every $(d_i, c_i) \in \Omega \times C$ the experimental results of the TC show that S is generally divided into two separate sets, namely, T_r (training set) and T_e (test set) [17]. The training data set (T_r) indicates the record collection already known in class labels. (T_r) is used for building the categorization model and is applied to the testing data set.

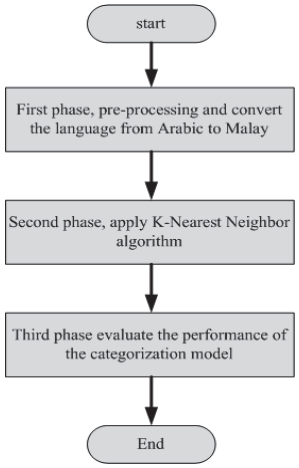

Fig. 3. The schema of the proposed work.

The test data set (T_e) indicates the record collection known in class labels. However, when given as an input to build categorization models, accurate class labels should be returned to the records. T_e would help identify the accuracy of the model [6]. Figure 4 explains how to build a categorization model to solve categorization problems. Different learner methods are implemented in TC. Several of these methods create binary-valued classifiers in the form of $\hat{\Phi} : D \times C \longrightarrow \{T, F\}$ By contrast, other learner methods create real-valued functions in the form $CSV : D \times C \longrightarrow [0, 1]$, where CSV is stands for categorization status value. When dealing with categorization problems, a set of thresholds τ_i need to be determined to allow the conversion of real- valued CSVs into final binary decisions [17]. In different applications, the methods noticeably perform a real-valued function that can be useful when thresholds are not required. The training efficiency (viz, the average time required to build a classifier $\hat{\Phi}_i$ from a corpus Ω), classification efficiency (viz, average time required to classify a document through $\hat{\Phi}_i$) and effectiveness (viz, the correctness of the average of $\hat{\Phi}_i$ s behavior of classification) are all legitimate procedures of learner success.

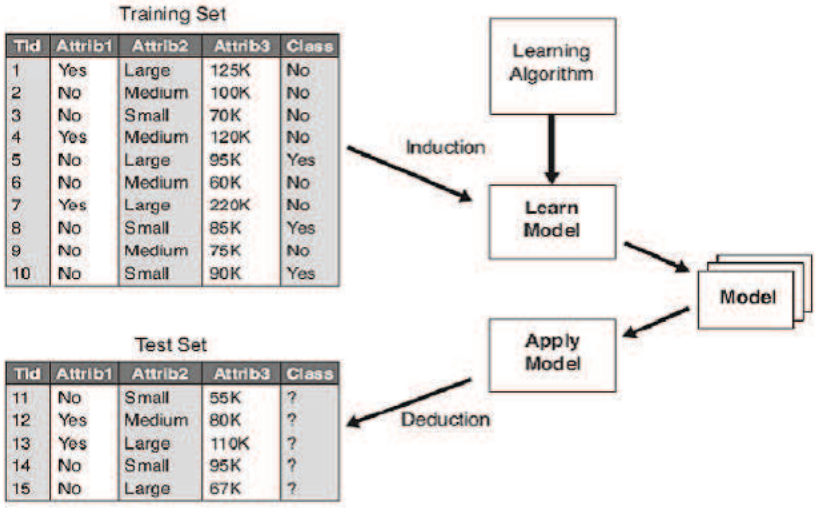

Fig. 4. General approach for building a categorization model.

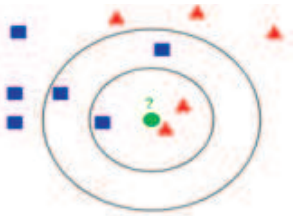

Fig. 5. The method of the KNN.

3.2 K-Nearest Neighbor (KNN) Algorithm

The KNN algorithm [1], is the simplest and most user-friendly technique in the area of statistical discrimination. This method is a nonparametric process, wherein a new observation is developed in the class of observations from the learning group closest to the new observation (figure 5) with regard to using the covariates. The similarity is determined by using distance measures. Figure 5 Shows the test sample (green circle) should be classified either to the first class of blue squares or to the second class of red triangles. If $k = 3$, it is assigned to the second class because there are 2 triangles and only 1 square inside the inner circle. If $k = 5$, it is assigned to the first class (3 squares vs.2 triangles inside the outer circle). Thereafter, a new observation (x, y) to the nearest neighbor (x(1), y(1)) in the learning set is selected by $d(x, x(1)) = min(d(x, x_i))$ and y=y(1). The class of the nearest neighbor, can be determined while y is being predicted. One can find similarities between the two documents by measuring the cosine value between samples [2] defined as

$$cos(\vec{x}, \vec{y}) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} {x_i}^2 \sum_{i=1}^{n} {y_i}^2}} \quad (1)$$

This distance is an instance-based learning or memory-based learning and is often known as the lazy learning algorithm because hypotheses are created locally. Moreover, computation is postponed until the test dataset is obtained [12]. There are many researches have reviewed the use of KNN with regard to classification tasks. Jiang et al. [10] summarized the various enhancements of KNN algorithm classifier. Bhatia [5] compared the nearest neighbor techniques. Some of them are structure less and some are structured base. [13] refined weighted KNN algorithm needs a little more running time than traditional KNN. Duwairi [7] also has a number of publications in this topic. She debated a survey of numerous papers published in text categorization and evaluated the use of classifications wherein distance depends on KNN. Among all supervised learning techniques, this algorithm is the easiest and classifies the objects based on closest training examples in the feature space. The KNN classifier uses the cosine value distance between a test sample and specified training samples. When an unknown document $\vec{x}$ comes, rank the training documents by the similarity of each one with the document $\vec{x}$. Then we get the k most similarity documents of $\vec{x}$. The similarity is calculated with cosine distance as in Equation (1) [13]. After loading the corpus, all the preprocessing techniques are applied. TF*IDF, calculations helps us create document vectors in the feature space. This observation concludes the training phase for KNN algorithm.

The KNN algorithm finds the document's k-nearest neighbors from the training documents. By using the class labels of these k-nearest neighbors, the document class can be predicted. The formal KNN algorithm is described in the following: All training documents are stored in $\bar{d}_i = (\bar{d}_1, \bar{d}_2, \cdots, \bar{d}_m)$, where $\bar{d}_i$ indicates one training document.
Whenever an unknown document $\vec{x}$ appears, the training documents are ranked on the basis of the similarity of each document to document. Thereafter the k most similar documents of $\vec{x}$ is obtained. The similarity is calculated in Equation ([**?**]) by using the cosine value distance, where N is the total number of feature items. The k most be similar documents of $\vec{x}$ the class weight of $\vec{x}$ for each class c_j can be calculated as follows:

$$p(\vec{x}, C_j) = \sum_{\bar{d}_j \in KNN(\vec{x})} sim(\vec{x}, \bar{d}_i) y(\bar{d}_i, C_j) \quad (2)$$

```
TRAIN-KNN(C, D)
1  D' ← PREPROCESS(D)
2  k ← SELECT-K(C, D')
3  return D', k

APPLY-KNN(C, D', k, d)
1  S_k ← COMPUTENEARESTNEIGHBORS(D', k, d)
2  for each c_j ∈ C
3  do p_j ← |S_k ∩ c_j|/k
4  return arg max_j p_j
```

Fig. 6. The pseudo code of the KNN algorithm.

KNN($\vec{x}$) denotes the set of K most similar documents of $\vec{x}$, whereas $y(\bar{d}_i, C_j)$ indicates the classification of document $\bar{d}_i$ for class C_j.

$$y(\bar{d}_i, C_j) = \begin{cases} 1, & \bar{d}_i \in C_j \\ 0, & \text{elsewhere} \end{cases} \quad (3)$$

Lastly, the class weight of $\vec{x}$ is compared for all classes, and $\vec{x}$ is classified in the class with the maximum class weight $p(\vec{x}, C_j)$ [13].

$$C = \arg\max_{cj} \left(p(\vec{x}, C_j) \right) \quad (4)$$

Figure 6 shows the pseudo code interpretation of the KNN algorithm. KNN is used in this paper because of the following reasons:

(1) In Rocchio classification, parameter estimation (centroids) is performed. Naive Bayes are employed (prior probabilities). However, KNN simply memorizes all objects in the training documents. Moreover, KNN compares the testing document and training document. Therefore, KNN is generally known as a memory-based learning or instance-based learning method [14].

(2) Training data in the KNN algorithm is extremely fast and effective if the training data size is large [5]. However, in Naive Bayes, the classifier requires a small quantity of training data to assess the necessary parameters for classification [4].

(3) KNN also possible to compute the decision boundary itself explicitly, and to do so in an efficient manner so that the computational complexity is a function of the boundary complexity [15].

(4) The K parameter is very important and the optimal choice for its value depends upon many factors, and the most influencing one is the nature of the data. Although larger values make boundaries between labels less distinct, they reduce the effect of noise on the classification process [8].

(5) The k-nearest neighbor algorithm is sensitive to the local structure of the data. Nearest neighbor rules in effect compute the decision boundary in an implicit manner.

3.3 Implementation

The implementation detail of the proposed method using the KNN algorithm containing three phases. First phase: Database that included Document Pre-processing containing several methods; Second phase: KNN algorithm implementation; and Third phase: Precision and Recall values evaluate the performance of the categorization model.

3.3.0.1 Database:. The database of this research consisting of 1000 Ayat (verses) of the Holy Quran and data was divided into two text document sets (training and testing). The training set is (800) Ayat (verses) and for testing set (200) Ayat (verses). We choose only seven categories of Tafseer texts shown in the table 1.

Table 1. Categories in this research for Holy Quran Tafseer

Arabic Main Categories	Malay Translation	English Translation
الزواج	Perkahwinan	Marriage
الارث	Warisan	Inheritance
الصلاة	Sembahyang	Pray
الزكاة	akat	Almsgiving
بر الوالدين	Menghormati ibu bapa	Honouring parents
الحلال	Halal	Halal
الجهاد	Jihad	Jihad

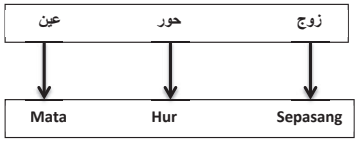

Fig. 7. The example for stemming Ayat (verses) and dictionary.

Table 2. Training data set collection

Category	Total Documents
Perkahwinan	139
Warisan	100
Sembahyang	129
Zakat	114
Menghormati ibu bapa	90
Halal	110
Jihad	118

3.3.0.2 Training Data Phase:. The all training data in this phase in Arabic language and containing the label (Ayat (verses) text, label). The process of preprocessing data includes tokenization, stop word list, normalization, stemming and dictionary. After the pre-processing phase all stemming words will be matched with the dictionary and convert words from the Arabic language to the Malay language. Figure 7 shows the example for stemming algorithm and dictionary. The next step is to generate the weighting terms by using (TF*IDF). (TF*IDF) method put document vector in the feature space this end step in the training phase of KNN algorithm. Table 2 shows the training data set collected in this study. There is a certain number of documents, which located in this class.

3.3.0.3 Testing Data Phase:. A new document given the categorization model must predict the correct category label based on previous training. In this paper, counting on the cosine similarity for a given document to its 30 nearest neighbors. Thus, k=30. Later all the pre-processing techniques are used, significant terms are obtained. TF*IDF calculations helps to create document vectors in feature space. Then the cosine similarity between the test document and specific training documents are calculated. The majority of the same kind of nearest neighbors decides the category of the test sample.

Table 3. Precision and recall value of KNN

Category	Precision	Recall	Fallout	Error rate
Perkahwinan	0.83	0.9	0	0.02
Warisan	0.79	0.74	0.02	0.05
Sembahyang	0.74	0.8	0.14	0.15
Zakat	0.84	0.78	0.01	0.05
Menghormati ibu bapa	0.8	0.75	0.04	0.07
Halal	0.83	0.83	0	0.08
Jihad	0.87	0.82	0.16	0.15

4. RESULTS

The effectiveness of the categorization depends on the KNN algorithm or the accuracy of the results obtained after the implementation of KNN algorithm and calculating the precision and recall are computed based on the following equation:

$$precision = \frac{a}{a+b} \tag{5}$$

$$Recall = \frac{a}{a+c} \tag{6}$$

$$Fallout = \frac{b}{b+d} \tag{7}$$

$$Error\ rate = \frac{b+c}{a+b+c+d} \tag{8}$$

where a is number of documents that both the human and the computer classify as positive examples, b is number of documents that the human classifies as negative examples but the computer classifies as positive examples, c is number of documents that the human classifies as positive example but the computer classifies as negative examples, d is number of documents that both the human and the computer classify as negative documents, and (a+b+c+d) represent total number of test documents (n). Table 3 shows precision, recall, fallout, and error rate for every category. As can be seen from Table 3, recall reaches its highest value (0.90) from Perkahwinan (Marriage) category, and the lowest value (0.74) for the Warisan (Inheritance) category. The second lowest value (0.75) for recall was for the Menghormati ibu bapa(Honouring parents) category. When the classifier's output was re-examined, a large percentage of the misclassified documents in the Inheritance categorye were categorized under the Honouring parents category, and vice versa. The reason for this misclassification is that documents that belong to the Inheritance category and those that belong to the Honoring parents category share many common words. Figure 8, depict the recall, precision, fallout, and error rate over the 7 categories.

5. CONCLUSIONS AND FUTURE WORK

In this study, we described the design and successful implementation of a new text classification suitable for classifying different Ayah (verses) of the Holy Quran using the KNN algorithm has been implemented using Matlab. This research deals a limited number of text files in the training set and test set. Increase the number of these files in both sets to extend conclusions it has schemes in the future. On the other hand, the number of classes (characters) it will be increased. The training data set used, and the value of k can enormously affect the accuracy of classification.

6. REFERENCES

[1] Charu C Aggarwal and ChengXiang Zhai. *Mining text data*. Springer Science & Business Media, 2012.

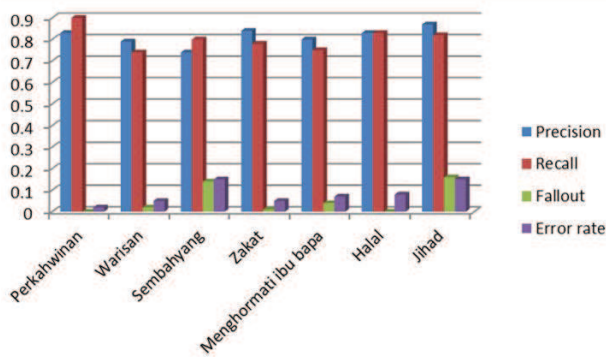

Fig. 8. Describe the recall, precision, fallout, and error rate over the 7 categories.

[2] Hamood Alshalabi, Sabrina Tiun, Nazlia Omar, and Mohammed Albared. Experiments on the use of feature selection and machine learning methods in automatic malay text categorization. *Procedia Technology*, 11:748–754, 2013.

[3] Baharum Baharudin, Lam Hong Lee, and Khairullah Khan. A review of machine learning algorithms for text-documents classification. *Journal of advances in information technology*, 1(1):4–20, 2010.

[4] P Bhargavi and S Jyothi. Applying naive bayes data mining technique for classification of agricultural land soils. *International journal of computer science and network security*, 9(8):117–122, 2009.

[5] Nitin Bhatia. Survey of nearest neighbor techniques. *arXiv preprint arXiv:1007.0085*, 2010.

[6] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[7] Rehab M Duwairi. Arabic text categorization. *Int. Arab J. Inf. Technol.*, 4(2):125–132, 2007.

[8] Rehab M Duwairi and Rania Al-Zubaidi. A hierarchical k-nn classifier for textual data. *Int. Arab J. Inf. Technol.*, 8(3):251–259, 2011.

[9] Caspar J Fall and Karim Benzineb. Literature survey: Issues to be considered in the automatic classification of patents. *World Intellectual Property Organization*, 29, 2002.

[10] Liangxiao Jiang, Zhihua Cai, Dianhong Wang, and Siwei Jiang. Survey of improving k-nearest-neighbor for classification. In *fskd*, pages 679–683. IEEE, 2007.

[11] Shengyi Jiang, Guansong Pang, Meiling Wu, and Limin Kuang. An improved k-nearest-neighbor algorithm for text categorization. *Expert Systems with Applications*, 39(1):1503–1509, 2012.

[12] Asha Gowda Karegowda, MA Jayaram, and AS Manjunath. Cascading k-means clustering and k-nearest neighbor classifier for categorization of diabetic patients. *International Journal of Engineering and Advanced Technonlogy*, 1:147–151, 2012.

[13] Fang Lu and Qingyuan Bai. A refined weighted k-nearest neighbors algorithm for text categorization. In *Intelligent Systems and Knowledge Engineering (ISKE), 2010 International Conference on*, pages 326–330. IEEE, 2010.

[14] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[15] Asha Rajkumar and G Sophia Reena. Diagnosis of heart disease using datamining algorithm. *Global journal of computer science and technology*, 10(10):38–43, 2010.

[16] Dr Sadiq, T Ahmed, and Sura Mahmood Abdullah. Hybrid intelligent techniques for text categorization. *International Journal of Advanced Computer Science and Information Technology (IJACSIT) Vol*, 2:23–40, 2014.

[17] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image processing, analysis, and machine vision*. Cengage Learning, 2014.