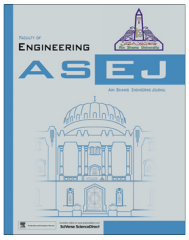
Ain Shams Engineering Journal (2014) 5, 411–420

Ain Shams University

Ain Shams Engineering Journal

www.elsevier.com/locate/asej
www.sciencedirect.com



ELECTRICAL ENGINEERING

Arabic summarization in Twitter social network

Nawal El-Fishawy [a], Alaa Hamouda [b], Gamal M. Attiya [a], Mohammed Atef [b,*]

[a] Faculty of Electronic Engineering, Menoufia University, Menoufia , Egypt
[b] Faculty of Computer Engineering, Al-Azhar University, Cairo, Egypt

Received 8 June 2013; revised 28 September 2013; accepted 22 November 2013
Available online 28 December 2013

KEYWORDS

Social networks;
Twitter;
Summarization;
Significance;
Similarity;
Feature selection

Abstract Twitter, an online micro blogs, enables its users to write and read text-based posts known as "tweets". It became one of the most commonly used social networks. However, an important problem arises is that the returned tweets, when searching for a topic phrase, are only sorted by recency not relevancy. This makes the user to manually read through the tweets in order to understand what are primarily saying about the particular topic. Some strategies were developed for summarizing English micro blogs but Arabic micro blogs summarization is still an active research area. This paper presents a machine learning based solution for summarizing Arabic micro blogging posts and more specifically Egyptian dialect summarization. The goal is to produce short summary for Arabic tweets related to a specific topic in less time and effort. The proposed strategy is evaluated and the results are compared with that obtained by the well-known multi-document summarization algorithms including; SumBasic, TF-IDF, PageRank, MEAD, and human summaries.

1. Introduction

Twitter, the micro blogging site, has become a social phenomenon. It started in 2006 and became one of the most commonly used social networks. Twitter reached half billion user [1]. To help people who read Twitter posts or tweets, Twitter provides an interesting API that allows users to search for tweets that contain a topic phrase. A user can search for a topic phrase and retrieve a list of the most recent tweets that contain the topic phrase.

An important problem arises with Twitter is that the returned tweets are only sorted by recency, not relevancy. This behavior makes some difficulties in interpreting the retrieved results. Therefore, the user is forced to manually read through the returned tweets in order to understand what users are primarily saying about the particular topic. This process requires more effort and time from the Twitter users. To overcome this problem, tweets summarization should be performed automatically for the purpose of saving users time and effort. So, a summarization system is required to automate this process.

Several strategies were developed for automatic summarizing micro blogs. However, most of the proposed strategies are developed for summarizing English tweets [2,3]. But, no algorithms are developed for summarizing Arabic micro blogging posts and more specifically Egyptian dialect summarization,

* Corresponding author. Tel.: +20 0111 161 2226.
E-mail addresses: nelfishawy@hotmail.com (N. El-Fishawy), dr.alaa.hamouda@gmail.com (A. Hamouda), gamal.attiya@yahoo.com (G. M. Attiya), atef_plc_mox@yahoo.com (M. Atef).
Peer review under responsibility of Ain Shams University.

Production and hosting by Elsevier

2090-4479 © 2013 Production and hosting by Elsevier B.V. on behalf of Ain Shams University.
http://dx.doi.org/10.1016/j.asej.2013.11.002

although the Arabic language has become the sixth most widely used language on the Twitter social network.

This paper presents a machine learning based summarization system for summarizing Arabic posts in Twitter social network. The purpose is to produce short summary for Arabic tweets related to a specific topic in less time and effort. In the proposed strategy, the problem is formulated as a regression problem, not a binary classification based problem. That is, instead of classifying the tweets to be important and not important, each tweet is given a score that determines whether this tweet may candidate in the summary or not. This makes the system to generate the summary according to the user predefined compression rate. The proposed strategy is evaluated and the results are compared with that obtained by the well-known multi-document summarization algorithms including; SumBasic [4], TF-IDF [2], PageRank [5], MEAD [6], and human summaries.

The rest of this paper is organized as follows. Section 2 presents an overview on related work. Section 3 states the summarization problem. The proposed system is described in Section 4 while the implementation of the proposed system is presented in Section 5. The evaluation and experimental results of the proposed system are discussed in Section 6. Finally, the conclusion is listed in Section 7.

2. Related work

Automatically summarizing micro blogging posts is a new area of research. A number of algorithms have been developed for English document summarization during recent years. MEAD [6] is a well-known flexible and extensible multi-document summarization system and was chosen to provide a comparison between the more structured document domains in which MEAD works fairly well. The platform implements multiple summarization algorithms such as position-based, centroid-based, largest common subsequence, and keywords. In [7], a LexRank algorithm is developed for computing the relative importance of sentences or other textual units in a document or a set of documents. It creates an adjacency matrix among the textual units and then computes the stationary distribution considering it to be a Markov chain. In [4], a SumBasic algorithm is proposed for document summarization. In this system, words that occur more frequently across documents have higher probability of being selected for human created multi-document summaries than words that occur less frequently. In [8], graph is applied for representing the structure of the text as well as the relationship between sentences of the document. Sentences in documents are presented as nodes. The edges between nodes illustrate connections between sentences. These connections are introduced by similarity relation between contents. The similarity between two sentences is calculated and each sentence is scored. All the scores for one sentence are combined to form a final score for each sentence. When the graph is processed, the sentences are categorized by their scores and sentences in higher orders are chosen for final summary. Other graph-based Phrase Reinforcement algorithm is developed in [3]. The algorithm first finds the most common phrase on one side of the search phrase, selects those posts that contain this phrase, and then finds posts with the most common phrase on the other side as well.

In [9], multi-document summarization system is developed for the Web context. The system is useful in combining information from multiple sources. Information may have to be extracted from many different articles and pieced together to form a comprehensive and coherent summary. One major difference between single document summarization and multi-document summarization is the potential redundancy that comes from using many source texts. The solution presented in [9] is based on clustering the important sentences picked out from the various source texts and using only a representative sentence from each cluster.

In [2], a hybrid TF-IDF algorithm developed. The idea of the algorithm is to assign each sentence within a document a weight that reflects the sentence's saliency within the document. The sentences are ordered by their weights from which the top sentences with the most weight are chosen as the summary. In order to avoid redundancy, the algorithm selects the next top tweet and checks it to make sure that it does not have a similarity above a given threshold with any of the other previously selected tweets because the top most weighted tweets may be very similar. Another method in [2] collects a set of Twitter posts, clusters the tweets into a number of clusters based on a similarity measure and then summarizes each cluster by picking the most weighted post as determined by TF-IDF algorithm.

Comments summarization over a collection of YouTube videos are studied in [10]. The system starts by clustering the comments and selecting the most representative comments of each cluster. Then, a precedence-based ranking framework is used for automatically selecting informative user-contributed comments.

Recently, some summarization systems are developed for Arabic text [11]. Rhetorical Structure Theory (RST) is one of the leading theories in computational linguistics [11]. It improved the ability of extracting the semantic behind the processed text. The applicability of RST to process and understand texts has been studied in Arabic language to extract the text structure, and then extract the semantic from this structure. In [12], an automatic extractive Arabic summarization system called *Ikhtasir* is developed. It integrates the advantages of an RST-based system with a scoring scheme which is a variant of the FarsiSumscoring formula. In [13], the summarizer, *Lakhas*, was developed using extracting techniques to produce ten words summaries of a new articles. *Lakhas* first summarizes the original Arabic document and then applies Machine Translation (MT), translating the summary into English. These systems support the single document summarization. A multi-document summarization system for Arabic comments was developed in [14].

3. Problem statement

Summarizing micro blogs can be viewed as an instance of the more general problem of automated text summarization, which is the problem of automatically generating a condensed version of the most important content from one or more documents.

The summarization problem can be simply described as follows: given a set of tweets that are related to a common search phrase (e.g., a topic), the problem is how to generate a

summary that best describes what users are saying about that search phrase.

4. Proposed system

The proposed system depends on machine learning in summarizing Arabic tweets. In the proposed system, the problem is formulated as a regression problem, not a binary classification based problem. That is, instead of classifying the tweets to be important or not important and then include important in the summary and excluded not important from the summary, the proposed system gives each tweet a score that determines to which extent this tweet is candidate to be in the summary. This makes the system to generate the summary according to the user predefined compression rate.

The proposed system consists of several stages, as shown in Fig. 1.

4.1. Natural language processing

This stage is to make the required natural language processing on the text before the stage of features extraction. This includes the following steps:

4.1.1. Pre-processing

The challenge of text mining, in general, in Arabic arises from the complexity of the language in terms of both structure and morphology. Arabic is a highly inflectional and derivational language with many word forms and diacritics. The same three-letter root can give rise to different words with different meanings. Moreover, the same word can have several different forms with different suffixes, affixes, and prefixes. Special labels called diacritics are used instead of vowels and they differ according to the word form, additionally slang Arabic words. Pre-processing is very useful because it reduces the unusual words, increases the accuracy and unifies the words to compute a word frequency. Unfortunately, most of the Arabic contents on the social networks are in Arabic dialects, not in Modern Standard Arabic (MSA). This increases the challenge and complicates the pre-processing stage. In Egyptian dialect, there are many slang words as (معرفش=لأعرف), (معلش =آسف), (إزازة = زجاجه), (دبانة = ذبابة) and (مقشه = مكنسه). It is possible that the word comes in more than one form. So, the traditional Modern Standard Arabic stemmers will not work efficiently. Therefore we moved toward other cogitation.

Our pre-processing stage includes the following:

- Removing non-Arabic letters, digits, single Arabic letters, punctuations, special symbols ($, %, &, #, ...), diacritics.
- Word segmentation. Words are separated by spaces.
- Removing (ات ، ون ، ين ، ان ، ها ، وا) from the end of the word.
- Removing (ال ، تال ، كال ، وال ، وكال ، وتال ، ولل ، لل) from the beginning of the word except الله، اللهم ، إله لله [15].
- Normalization by replacing some variants of characters by a single one (أ ، إ ، آ are replaced by ا) ، (ى is replaced by ئ) (ي is replaced by يء) ، (ه is replaced by ة) [16].
- Normalization by replacing some of characters those appear more than one time (e.g., يااااااااااارب) by a single one يارب.
- Removing stop words. Stop words are common words that appear in the text but carry little meaning. They serve only a syntactic function but do not indicate subject matter. Eliminating the stop-words from the text helps in identifying the most important words. We used the list of Arabic stopwords supplied in [17].

4.1.2. Calculating word matching score

The very basic process of making a matching score between every two words in the document body is to give a score of 1 if the two words exactly match or 0 if there is even one mismatch character. This basic step is called the Exact Word Matching (EWM). Unfortunately, Arabic language is morphologically rich. This means the frequency computation for each word may not be calculated properly. The same word could

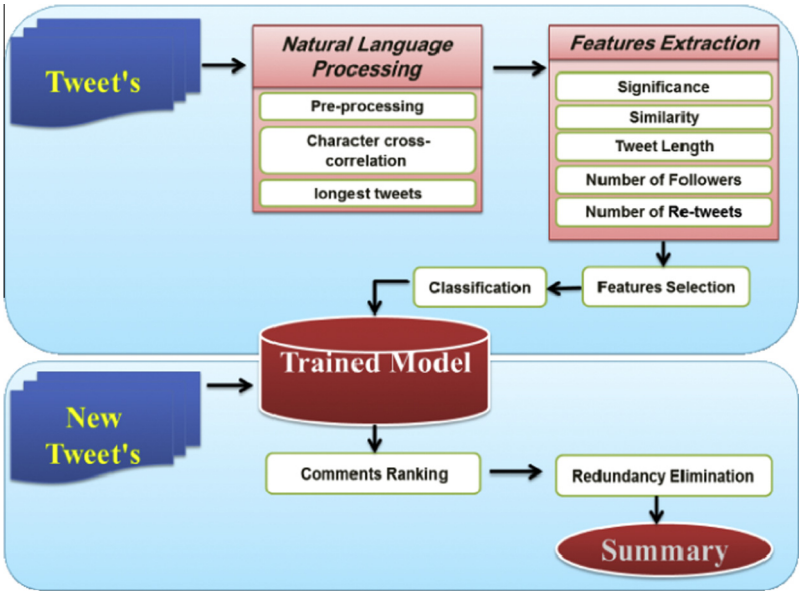


Figure 1 The overall tweets summarization approach.

appear with a more shape as a different word in the EWM method. Therefore, the idea of using Character Cross-Correlation (CCC) method emerged, in which a variable score in the range of 0–1 is calculated depending on how many characters match with each other. For example, if the word وكتبها "he wrote it" is compared with the word كتب "wrote" using the EWM method, the resulting score will be 0. But when using the CCC method after pre-processing, it will be 0.857. The CCC method comes from signals cross-correlation which measures the similarity of two waveforms. In the CCC method, the score is calculated according to the following equation:

$$CCC(w_i, w_j) = \frac{2(\#UniqueMatchedCharachter1)}{M+N} \quad (1)$$

where the w_i is the first word containing M characters. w_j is the second word containing N characters. The result is 1 if the two corresponding characters match each other and 0 otherwise [18].

4.1.3. Selecting the longest tweets with specific threshold

Some tweets may have only 1–2 words of topic phrases. Such tweets are useless. So, they are excluded in this stage.

4.2. Features extraction

In this stage, we extract the features of tweets. They include the following features:

4.2.1. Significance

Significant (important) tweets are determined by counting the number of important words in sentences. To calculate word importance, we consider several approaches: Term Frequency (TF) [10], Term Frequency Inverse Document Frequency (TFIDF) [2] and SumBasic [4].

(a) Term Frequency (TF)

Term Frequency is defined as the number of times a term i appears in the tweets. We use Eq. (2) to determine the weight of each term w_s:

$$w_s = \frac{(w_{d*}w_c)}{n_{max}} \quad (2)$$

where w_d is the number of times a term appears in tweets. w_c is the number of times a term appears in this tweet. n_{max} is the maximum number of times a term appears in all tweets. Finally the score for each tweet c_s is calculated using Eq. (3).

$$c_s = \sum_{n=0}^{\#allwords} \left(\frac{(w_{d*}w_c)/n_{max}}{n} \right) \quad (3)$$

where n is the number of important terms in the tweet. We use specific threshold to determine the important terms.

(b) Term Frequency Inverse Document Frequency (TF-IDF)

Term Frequency-Inverse Document Frequency, is a statistical weighting technique that has been applied to many information retrieval problems. It has been used for automatic indexing, query matching of documents, and automated summarization. Generally TF-IDF is not known as leading algorithms in automated summarization. In this technique, the weight of a term is determined by the following formula:

$$tf - idf = tf_{ij} * log_2 \frac{N}{df_j} \quad (4)$$

where tf_{ij} is the frequency of the term T_j within the document D_j, N is the total number of documents, and df is the number of documents within the set that contains the term T_j. This equation defines the weight of a term in the context of a document. But we have a set of Tweets that are related to a topic. We define each Tweet as a document making the IDF component's definition clear. But, the TF component now has a problem: Because each Tweet contains only a handful of words, most term frequencies will be a small constant for a given Tweet. To handle this situation [2], we redefine TF-IDF in terms of a hybrid document. We primarily define a document as a single sentence. However, when computing the term frequencies, we assume the document is the entire collection of Tweets. This way, we have differentiated term frequencies but also do not lose the IDF component. A term is a single word in a sentence. We next choose a normalization method since otherwise the TF-IDF algorithm always biases toward longer sentences. We normalize the weight of a sentence by dividing it by a normalization factor. Our definition of the TF-IDF summarization feature is now complete for micro blogs. We summarize its equation's below [2].

$$W(s) = \frac{\sum_{i=0}^{\#WordsInPost} W(w_i)}{nf(s)} \quad (5)$$

$$W(w_i) = tf(w_i) * log_2(idf(w_i)) \quad (6)$$

$$tf(w_i) = \frac{\#OccurrencesOfWordInAllPosts}{\#WordsInAllPosts} \quad (7)$$

$$idf(w_i) = \frac{\#Posts}{\#PostsInWhichWordOccurs} \quad (8)$$

$$nf(s) = \max[MinimumThreshold, \#WordsInPost] \quad (9)$$

where W is the weight assigned to a tweet or a word, nf is a normalization factor, w_i is the ith word, and s is a tweet.

(c) SumBasic

SumBasic [4] is a system that produces generic multi-document summaries. Its design is motivated by the observation that words occurring frequently in the document cluster occur with higher probability in the human summaries than words occurring less frequently. Specifically, SumBasic uses the following algorithm:

Step 1: Compute the probability distribution over the words w_i appearing in the input, $P(w_i)$ for every i,

$$P(w_i) = n/N \quad (10)$$

where n is the number of times the word appeared in the input and N is the total number of content word tokens in the input.

Step 2: For each sentence S_j in the input, assign a weight equal to the average probability of the words in the sentence, i.e.,

$$Weight(S_i) = \sum_{W_i \in S_j} \frac{P(W_i)}{|\{W_i | W_i \in S_j\}|} \quad (11)$$

Step 3: Pick the best scoring sentence that contains the highest probability word.
Step 4: For each word w_i in the sentence chosen at step 3, update their probability:

$$pnew(wi) = pold(wi) * pold(wi) \quad (12)$$

Step 5: If the desired summary length has not been reached, go back to step 2.

4.2.2. Similarity

Sentence similarity plays an important role in many applications such as information retrieval, text mining, natural language processing, and Machine Translation. Calculating similarity between sentences is the basis of measuring the similarity between texts which is the key of document classification and clustering [19]. If a tweet is similar to other tweets, then this tweet is more important. So, many of tweets are voted to it. Similarity between tweets is used to determine the more similar tweets using Eq. (13).

$$sim(C) = \sum_{0}^{n} \frac{sim(C, Cn)}{n} \quad (13)$$

n is the number of tweets that have similarity with each other's. We use three methods to calculate similarity between tweets with specific threshold.

(a) Similarity based on word vector

To calculate tweets similarity based on word vectors, the word vectors of sentences should be constructed first. If the words in $w(Sa)$ and $w(Sb)$ are assigned with weights, Sa and Sb can be represented by the bags of words.

$$\left\{ \begin{array}{l} v(Sa) = \{(w_1, w_{a1}), (w_2, w_{a2}), \ldots, (w_{i+j}, w_{a(i+j)})\} \\ v(Sb) = \{(w_1, w_{b1}), (w_2, w_{b2}), \ldots, (w_{i+j}, w_{b(i+j)})\} \end{array} \right\}$$

Then cosine similarity between sentences can be calculated using Eq. (14).

$$Cosine(Sa, Sb) = \frac{\sum_{k=1}^{i+k} wakwbk}{\sqrt{\sum_{k=1}^{i+k} w^2 ak} \sqrt{\sum_{k=1}^{i+k} w^2 bk}} \quad (14)$$

Sentence similarity based on word vector considers words and their weights in two sentences. Vector based similarity is popular in information retrieval. However, it does not consider the orders and distances between words. It is also a symbolic similarity, and different word weights distinguish the importance of words [19].

(b) Similarity based on word order

Similarity is calculated according to positions of words in sentence; the orders between word pairs such as before after could be established [19]. The sequential relation between words formulates a sequential network of words, the sequential network could be used to discover frequent patterns, and we segment each tweet into pairs of sequential words and calculate the similarity from this formula.

Table 1 Sequential relations between words in two example sentences.

(that,is), (that,the), (that,old), (that,file), (that,.), (is,the), (is,old), (is,file), (is,.), (the,old), (the,file), (the,.), (old,file), (old,.), (file,.)	$L(s_a)$
(this,is), (this,the), (this,new), (this,file), (this,.), (is,the), (is,new), (is,file), (is,.), (the,new), (the,file), (the,.), (new,file), (new,.), (file,.)	$L(s_b)$

$$Set_{sim(Sa,Sb)} = \frac{|L(s_a) \cap L(s_b)|}{|L(s_a) \cup L(s_b)|} \quad (15)$$

For example, if we have two sentences (That is the old file.) and (This is the new file.), sequential relation between words is listed in Table 1.

(c) Similarity based on Dice's coefficient

To calculate sentence similarity based on Dice's coefficient [19], we segment each tweet into single of words and calculate the similarity from this formula.

$$Set_{sim(Sa,Sb)} = \frac{2|L(s_a) \cap L(s_b)|}{|L(s_a) \cup L(s_b)|} \quad (16)$$

where words in $w(s_a)$ intersection $w(s_b)$ are assigned with the similar words in sentences s_a and s_b divided by summation of words.

(d) PageRank

We use PageRank algorithm to calculate the similarity, PageRank algorithm is used by the famous search engine, Google. They applied the citation analysis in Web search by treating the incoming links as citations to the Web pages. PageRank provides a more advanced way to compute the importance or relevance of a Web page than simply counting the number of pages that are linking to it (called as "backlinks") [5]. If a backlink comes from an "important" page, then that backlink is given a higher weighting than those backlinks come from non-important pages. In a simple way, link from one page to another page may be considered as a vote. However, not only the number of votes a page receives is considered important, but the "importance" or the "relevance" of the ones that cast these votes as well. To calculate the score of a tweet $S(C_i)$ we add the score of all the neighbors pointing to it divided by the number of output links of each of these neighbors, where $S(C_j)$ is the number of neighbor's tweets. We used 0.85 as our damping factor d.

$$S(C_i) = d \sum \frac{S(C_j)}{Count_{C_j}} + (1 - d) \quad (17)$$

The random walk continues iteratively until the scores of the nodes converge.

4.2.3. Number of re-tweet

Micro bloggers often repeat the most relevant tweets for a trending topic by quoting others. Quoting uses the following form: RT @ [Twitter Account-Name]: Quoted Message. RT refers to Re-Tweet and indicates one is copying a tweet from the indicated Twitter account. While users writing their own tweets occasionally use the same or similar words, re-tweeting causes entire sentences to perfectly overlap with one another. This, in turn, greatly increases the average length of an overlapping phrase for a given topic. The main idea of the algorithm is to determine the most heavily overlapping phrase centered about the topic phrase. The justification is that repeated information is often a good indicator of its relative importance [20].

4.2.4. Number of followers

Most people are interested in what is said by public persons. These persons have a large number of followers. Followers to a certain person are the people who have agreed to receive tweets from him. This means that the more followers the more important micro bloggers. Oftentimes users who have great followers produce important tweets. The main idea is to use number of followers for each micro blogger as a feature for the tweet.

4.2.5. Tweet length

Any tweet has 140 characters or less. Oftentimes short tweets carry little meaning because they do not have any important words except topic phrase. So, they are not expressive. Therefore we observed that long tweets are more important than others. The idea is to use the tweets length as a tweet feature.

4.3. Features selection

When calculating the Significance and similarity for each feature at different thresholds are obtained and compute number of Re-tweet, number of Followers and Tweet length, tens of different independent features. However, the important features are not known. The purpose of feature selection stage is to decide which of the initial (possibly large number) of features to include in the final subset and which to ignore. In the proposed system, the Correlation Feature Selection (CFS) Attribute Evaluators [21] is used. The CFS is a correlation-based filter method. It gives high scores to subsets that include features that are highly correlated to the class attribute but have low correlation to each other. The results are validated using cross validation.

4.4. Classification

After features selection stage, we have the important features, but so far, do not know the weight of the features. So we use model tree to calculate the weight of each feature. We considered it a regression problem. Each tweet is given a value expressing the final weight of it. Model tree algorithm is used to induce trees of regression models. It combines a conventional decision tree with the possibility of linear regression functions at the nodes. First, a decision-tree induction algorithm is used to build a tree, but instead of maximizing the information gain at each inner node, a splitting criterion is used that minimizes the intra-subset variation in the class values down each branch. The splitting procedure in model tree classifier stops if the class values of all instances that reach a node vary very slightly, or only a few instances remain. Second, the tree is pruned back from each leaf. When pruning an inner node it is turned into a leaf with a regression plane. Third, to avoid sharp discontinuities between the sub trees a smoothing procedure is applied that combines the leaf model prediction with each node along the path back to the root, smoothing it at each of these nodes by combining it with the value predicted by the linear model for that node [22].

4.5. Redundancy elimination

In online micro blogs, redundant tweets can be a challenge. Redundancy must be eliminated to avoid repeated information, and improve readability. In order to avoid redundancy, the tweets are reordered according to their scores determined by the model tree. For each tweet in the list, the next top tweet is selected and checked it to make sure that it does not have a similarity with any of the other previously selected tweets above a given threshold t. If not, the similar (redundant) tweet is removed. In our work we rely on three previous similarity models for redundancy removal.

5. Implementation

To implement the proposed algorithm, 15 trending Arabic topics are first collected from Twitter. For each topic, about 1500–3000 tweets are downloaded. Then, three human experts are asked for mark each of the retrieved tweets with a value of 3, 2, 1, or 0. A score of 3 means that the tweet is informative and interesting. A score of 0 means that the tweet is neither informative nor interesting. A score of 2 and 1 are in between. Finally, the average score for all experts are calculated. The resulting data set is used to train the model tree classifier using the proposed features. The corpus is divided into training and testing sets using 5-folds cross validation.

Several trails were done to adapt several parameters to reach the best performance. After pre-processing stage on all tweets, Word Matching Score is calculated using Character Cross-Correlation (CCC) for each two terms at different threshold t from 0 to 0.99. At the end, human said these two terms similar or not as Table 2. The proportion of agreement between human and different threshold is calculated, as shown in Fig. 2. After several trails of hundreds of words, the best threshold for calculating Word Matching Score using Character Cross-Correlation (CCC) is found 0.8.

We also calculated all proposed features; TF, TF-IDF, SumBasic, similarity based on word vector, similarity based on word order, similarity based on Dice's coefficient, similarity based on page rank at different threshold t from 0 to 0.99, number of re-tweets, number of followers and tweet length. We used the training data as input for Correlation Feature Selection evaluator to select the most appropriate features. Using CFS Attribute Evaluators, we found that the most important features are Term Frequency (TF) at threshold 0.6, PageRank using Similarity based on Dice's coefficient at threshold of 0.3, tweet length, number of followers and number of re-tweets. One of the biggest problems we have observed with Tweets is redundancy. In other words, quite frequently, a

Table 2 Contrast ratio between words using different thresholds from 0 to 0.99.

Term 1	Term 2	*T* 0.01	*T* 0.22	*T* 0.33	*T* 0.57	*T* 0.6	*T* 0.75	*T* 0.8	*T* 0.85	*T* 0.91	*T* 0.99	Human decision
عسكري	عسكرية	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	–	Yes
يدرس	مدرس	Yes	Yes	Yes	Yes	Yes	Yes	–	–	–	–	Yes
يرشي	استرشادي	Yes	Yes	Yes	–	–	–	–	–	–	–	No
استنفار	استفتاء	Yes	Yes	Yes	Yes	–	–	–	–	–	–	No
معركة	معارك	Yes	Yes	Yes	Yes	Yes	Yes	Yes	–	–	–	Yes
متعرفش	معرفش	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	–	Yes
كويسة	كويس	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	–	–	Yes
يهوي	اهانة	Yes	Yes	–	–	–	–	–	–	–	–	No
معترك	معذرة	Yes	Yes	Yes	Yes	Yes	–	–	–	–	–	No

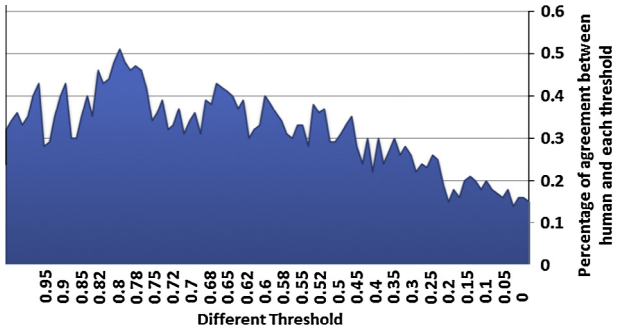


Figure 2 Percentage of agreement between human and each threshold for CCC.

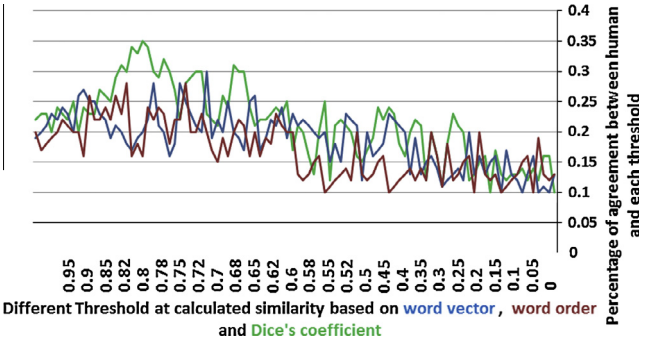


Figure 3 Percentage of agreement between human and each threshold for redundancy elimination.

large number of Tweets are similar to each other's. We use three previous similarity models at different thresholds from 0 to 0.99 to apply the same method employed in selecting the best threshold for calculating Word Matching Score using Character Cross-Correlation (CCC). At the end human expert judges that this tow Tweets are similar or not, after several trails of hundreds of Tweets, we found the best threshold for redundancy elimination was 0.8 using similarity based on Dice's coefficient as Fig. 3.

6. Evaluation

In this section, the proposed system is evaluated by using two methods namely; the Precision and Recall method and the Normalized Discounted Cumulative Gain (NDCG). The results of summarization of the proposed system are compared with that obtained by the well-known multi-document summarization algorithms including; SumBasic, TF-IDF, PageRank, and MEAD summarizer. Also, three experts are asked to make summarization of several groups of tweets and the results obtained by the proposed system are compared with that obtained by the manual summarization experiments.

6.1. Precision and recall

This is the first method used to evaluate the proposed system. Three important measures are commonly used, precision, recall and *F*-measure [23,24]. Precision is a measure of how much of information that the system returned is correct. Recall is a measure of the coverage of the system. They are calculated as follows:

$$\text{Precession} = \textit{Number of system correct summary sentences}/\textit{Total number of system summary sentences}$$

$$\text{Recall} = \textit{Number of system correct summary sentences}/\textit{Total number of human summary sentences}$$

System strives for coverage will get lower precision and a system strives for precision will get lower recall. Recall and precision are antagonistic to one another. *F*-measure balances recall and precision using a parameter β (used $\beta = 1$). The *F*-measure is defined as follows:

$$F = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \tag{18}$$

Fig. 4 shows the *F*-measure score of the proposed system and other algorithms. It is clear that the proposed system has the best performance compared with other systems or algorithms.

The main human summarizer is called the "reference human summarizer". In order to understand how humans may generate different extractive summaries for the same group of tweets, two additional independent human summarizers asked to extract sentences from the same testing set with different compression ratios. Then, the common selected sentences between each pair are computed and the average precision/recall/*F*-measures are calculated between them. It is noted that, the total average *F*-measure for the manual summarization between each pair of the three experts is between 46% and 60%. This can be considered the reference value with which we can compare our proposed system. It is evident that our system's *F*-measure is between 44% and 56% at different compression rate of 5%, 10%, 15%, 20%, 25% and 30%.

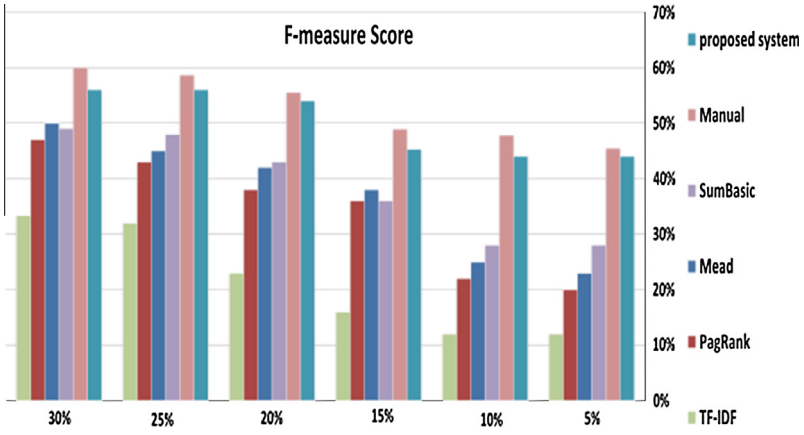


Figure 4 Comparison between our proposed system and other algorithms using *F*-measure at different compression rates.

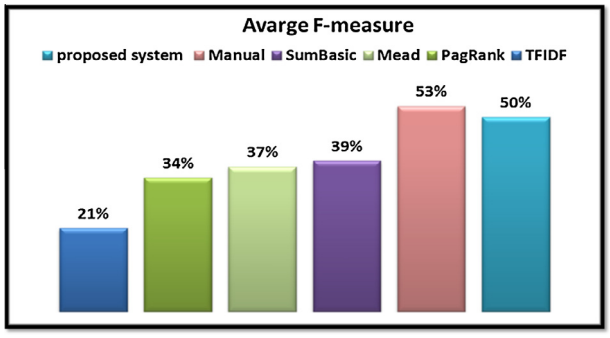


Figure 5 Comparison between our proposed system and other algorithms using average *F*-measure.

Fig. 5 shows that the average *F*-measure of our system is 50% while the total average *F*-measure of the manual summarization is 53%. Hence, the proposed system provides 50%/53% = 94% of the performance of the manual summarization.

6.2. Normalized Discounted Cumulative Gain (NDCG)

This is the second method to evaluate our proposed system. We used the well-known method; Normalized Discounted Cumulative Gain (NDCG) that values highly relevant tweets that have appeared earlier in the ranking result. NDCG uses a graded relevance scale of tweets in a search engine result set. It measures the usefulness, or gain, of a tweet based on its position in the result list. The gain is accumulated from the top of the result list to the bottom with the gain of each result discounted at lower ranks [25]. The discounted CG accumulated at a particular rank position p is defined as:

$$DCG_p = rel1 + \sum_{i=2}^{p} \frac{rel_i}{log_2(i)} \tag{19}$$

Comparing a summarization systems performance from a set of tweets to the next cannot be consistently achieved using DCG alone, so the cumulative gain at each position for a chosen value of p should be normalized across sets of tweets. This is done by sorting tweets of a result list by relevance, producing the maximum possible DCG till position p also called Ideal DCG (IDCG) till that position. For a query, the Normalized Discounted Cumulative Gain, or NDCG, is computed as:

$$\mathrm{NDCG}_p = \frac{DCG_p}{IDCG_p} \tag{20}$$

The NDCG values for all sets of tweets can be averaged to obtain a measure of the average performance of a search engine's ranking algorithm. In a perfect ranking algorithm, the DCG will be the same as the IDCG producing an NDCG of 1.0. All NDCG calculations are then relative values on the interval 0.0–1.0 and so are cross-sets of tweets comparable.

To calculate the NDCG, we used the average of the three experts' scores to re-order the tweets and compare the results

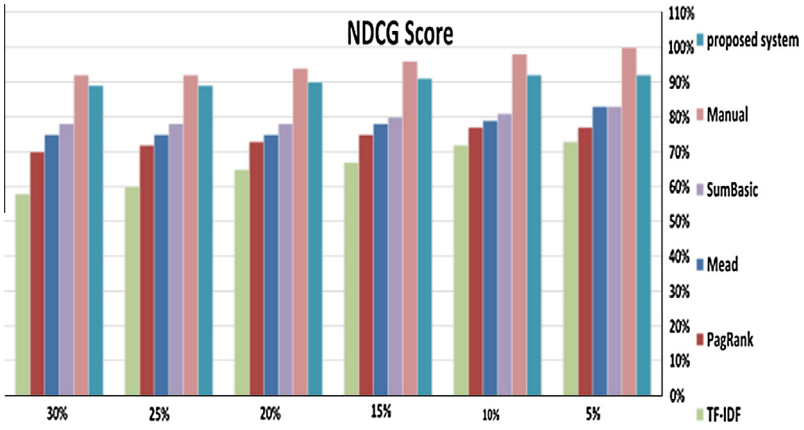


Figure 6 Comparison between our proposed system and others using NDCG at different compression rates.

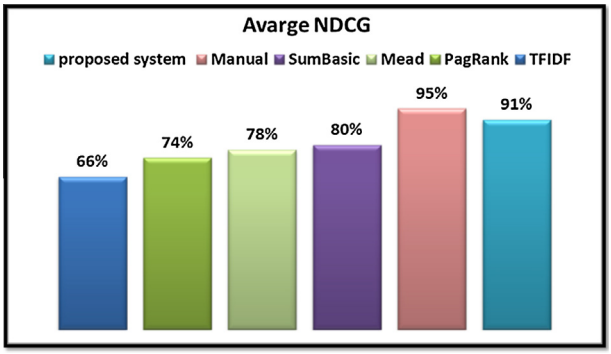


Figure 7 Comparison between our proposed system and others using average NDCG.

with the order generated from the scores of our system and other algorithms.

Fig. 6 shows the NDCG score of the proposed system and other algorithms. The figure shows that the NDCG of the proposed system NDCG is ranged from 89% and 92% at different compression rates of 5%, 10%, 15%, 20%, 25% and 30%. This emphasizes that the proposed system performance outperforms all other algorithms. Also, the orders of tweets generated by each expert are compared with that order generated by the average of the three experts' scores. The resulting NDCG is ranged between 92% and 100%.

We also compared between the orders of tweets generated by each expert with that order generated by the average of the three experts' scores. The resulting NDCG was between 92% and 100%. Again, it is considered the reference NDCG for our proposed system.

Fig. 7 shows the average NDCG of the proposed system and other algorithms. From the figure, the average NDCG of the proposed system is 91% while the average NDCG of manual summarization is 95%. So, we can conclude that the proposed system performance is 91%/95% = 95.7% of the performance of the manual system.

Finally both *F*-measure and NDCG indicate that our proposed system produced good level of summarization performance compared to the other systems and compared to the manual summarization.

7. Conclusion

In this paper, a machine learning based summarization system for summarizing Arabic posts in Twitter social network has been introduced. In the system, the problem was formulated as regression problem, using model tree, instead of a binary classification based problem. The proposed strategy is evaluated and the results are compared with that obtained by the well-known multi-document summarization algorithms including; SumBasic, TF-IDF, PageRank, MEAD, and human summaries. Using the *F*-measure evaluation method, the proposed system provides best results of 50% compared to the manual summarization of 53%. Using NDCG evaluation method, the proposed system produced best results of 91% compared to the manual summarization of 95%. Generally, both *F*-measure and NDCG indicate that the proposed system achieves good level of summarization performance compared to other algorithms and the manual summarization.

References

[1] <http://www.semiocast.com/en/publications/2012_07_30_Twitter_reaches_half_a_billion_accounts_140m_in_the_US>.
[2] David Inouye, Kalita Jugal K. Comparing Twitter summarization algorithms for multiple post summaries. In: IEEE international conference on privacy, security, risk, and trust, and IEEE international conference on social, computing; 2011 IEEE.
[3] Beaux Sharifi, Mark-Anthony Hutton, Kalita Jugal K. Experiments in microblog summarization. In: IEEE international conference on social, computing; 2010 IEEE.
[4] Vanderwende L, Suzuki H, Brockett C, Nenkova A. Beyond SumBasic: task-focused summarization with sentence simplification and lexical expansion. Inf Process Manage 2007;43(6):1606–18.
[5] Kumar Singh Ashutosh, Ravi Kumar P. "A comparative study of page ranking algorithms for information. Int J Electr Comput Eng 2009;4:7.
[6] Radev D, Allison T, Blair-Goldensohn S, Blitzer J, Elebi AC, Dimitrov S, et al. Mead – a platform for multi-document multilingual text summarization. In: LREC 2004, Lisbon, Portugal; May 2004.
[7] Erkan G, Radev D. LexRank: graph-based centrality as salience in text summarization. J Artif Intell Res 2004;22:457–80.
[8] Jezek Karel, Steinberger Josef. Automatic text summarization: the state of the art 2007 and new challenges. Znalosti 2008:1–12.
[9] Xiao-Chen Ma, Gui-Bin Yu, Liang Ma. Multi-document summarization using clustering algorithm; 2009.
[10] Khabiri Elham, Caverlee James, Hsu Chiao-Fang. Summarizing user-contributed comments. Assoc Adv Artif Intell 2011.
[11] Waleed Al-Sanie, Amir Touir, Hassan Mathkour. Towards a rhetorical parsing of Arabic text. In: Proceedings of the 2005 international conference on computational intelligence for modeling, control and automation, and international conference, intelligent agents, Web technologies and internet commerce (CIMCA-IAWTIC'05); 2005.
[12] Aqil Azmi, Suha Al-Thanyyan. Ikhtasir – a user selected compression ratio Arabic, text summarization system; 2009.
[13] Douzidia FS, Lakhas Lapalme G. An Arabic summarising system. In: Proceedings of the document understanding conferences (DUC) workshop, DUC; 2004. p. 128–35.
[14] Mahmoud El-Haj, Udo Kruschwitz, Chris Fox. Multi-document Arabic, text summarisation; 2011.
[15] Riyad Al-Shalabi, Rasha Obeidat. Improving KNN Arabic text classification with N-grams based document indexing. In: INFOS2008, March 27–29, 2008, Cairo-Egypt.
[16] Riyad Al-Shalabi, Al-sarrayrih Haytham S. Clustering Arabic documents using frequent itemset-based hierarchical clustering with an N-grams, 2009.
[17] El-Khair Ibrahim Abu. Effects of stop words elimination for Arabic information retrieval: a comparative study. Int J Comput Inf Sci 2006;4(3), December, On-Line.
[18] Alotaiby Fahad A. Automatic headline generation using character cross-correlation. In: Proceedings of the ACL-HLT June 2011, Association for Computational Linguistics.
[19] Zhang Junsheng, Sun Yunchuan, Wang Huilin, He Yanqing. Calculating statistical similarity between sentences. J Convergence Inf Technol 2011;6(2).
[20] Luhn HP. The automatic creation of literature abstracts. IBM J Res Dev 1958;2(2):159–65.
[21] Selvakuberan K, Indradevi M, Rajaram R. Combined feature Selection and classification – a novel approach for the categorization of Web pages (received 10.03.08, accepted 22.04.08).
[22] Frank Eibe et al.. Using model trees for classification. Machine learning, vol. 32. Klewer Academic Publisher; 1998, p. 63–76.
[23] Gong Y, Liu X. Generic text summarization using relevance measure and latent semantic analysis. In: Proceedings of special interest group on information retrieval, SIGIR, ACM; 2001. p. 19–25.

[24] Steve J, Stephen L, Gordon W. Interactive document summarization using automatically extracted key phrases. In: Proceedings of the 35th annual Hawaii international conference on system sciences, HICSS-35; 2002.
[25] Jarvelin Kalervo, Kekalainen Jaana. Cumulated gain-based evaluation of IR techniques. ACM Trans Inf Syst 2002;20(4):422–46.

Nawal El-Fishawy received the Ph.D degree in mobile communications the faculty of Electronic Eng., Menoufia University, Menouf, Egypt, in collaboration with Southampton University in 1991. Now she is the head of Computer Science and Engineering Dept., Faculty of Electronic Eng. Her re-search interest includes computer communication networks with emphasis on protocol design, traffic modeling and performance evaluation of broadband networks and multiple access control protocols for wireless communications systems and networks. Now she directed her research interests to the developments of security over wireless communications networks (mobile communications, WLAN, Bluetooth), VOIP, and encryption algorithms. She has served as a reviewer for many national and international journals and conferences.

Alaa Hamouda Lecturer, Faculty of Engineering, Al-Azhar University, Department of Systems Engineering and Computing. Earned a master's degree in the field of intelligent system design for agents in the field of e-commerce. And then a doctorate in the field of intelligent system is designed to prevent the spread of the worm-mail Worms . His research interests revolve around artificial intelligence and concentrated in data mining, text, abbreviation of Arabic texts, exploration in the opinions and analysis of trends, Intelligent agents, systems design, software engineering and management in accordance with the quality system CMMI and PMI. He has led several research projects in the field of a shortcut and analysis of Arabic texts.

Gamal M. Attiya graduated in 1993 and obtained his M.Sc. degree in computer science and engineering from the Menoufia University, Egypt, in 1999. He received his PhD degree in computer engineering from the University of Marne-La-Vallée, Paris-France, in 2004. He is currently Lecturer at the department of Computer Science and Engineering, Faculty of Electronic Engineering, Menoufia University, Egypt. His main research interests include distributed computing, task allocation and scheduling, computer networks and protocols, congestion control, QoS, multimedia networking and image processing.

Mohmmed Atef earned a Bachelor of Engineering in Computer Science in 2005, and is currently a design engineer and software development ground stations Egyptian space program.