Arabic Text Classification Using Maximum Weight Algorithm

*Zinah Abdulridha Abutiheen and **Ahmed H. Aliwy
**Kadhim S. Aljanabi
*Collage of Science, University of Kerbala, Iraq
**Collage of CS and Mathematic, University of Kufa, Iraq

Received Date: 5 / 2 / 2018
Accepted Date: 10 / 5 / 2018

الخلاصة

هناك العديد من خوارزميات التصنيف التي تطبق على قضايا تصنيف النصوص، الجزء الغالب هو تطبيقها على اللغة الانكليزية. على الجانب الاخر، قليل جدا من الباحثين طبق هذه الخوارزميات على اللغة العربية. اللغة العربية أصعب بكثير من اللغة الانكليزية وبالتالي معالجة اللغة العربية أكثر صعوبة وتحدي من اللغة الانكليزية. في هذا البحث خوارزمية Maximum Weight تطبق على النصوص العربية بعد المعالجة الاولية للنصوص حيث عدد النصوص (16757) ملفا عربيا نصيا يستخدم لأول مرة. بينت النتائج ان خوارزمية Maximum Weight يمكن ان تطبق على النصوص العربية حيث بلغت دقتها (83٪) بالمعدل، حيث تم استخدام 10-fold cross-validation لموثوقية نتائج هذا المصنف.

الكلمات المفتاحية

معالجة النصوص العربية، تصنيف النصوص، خوارزمية Maximum Weight

Abstract

Many algorithms of classification implemented to the issue of text categorization. A large portion of the work implemented in the English text. On the other hand, very few researchers implemented in the Arabic text. The nature of Arabic text is very different than English text, and the preprocessing of the Arabic text is extremely difficult and more challenging. In this paper, Maximum Weight (MW) algorithm applied after preprocessing on the Arabic dataset that consists of (16757) text files used for the first time. The results showed that MW is applicable to Arabic text, it reached about (0.83) on average. 10-fold cross-validation used to the reliability of the result.

Keywords

Exponentiated exponential Pareto distribution, reliability function, reversed hazard function.

1. Introduction

A huge amount of electronic texts is progressively accessible through the web and associations, making the process of retrieving data and information transforms into a genuine issue without great ordering and summarization of documents contents. Text or document classification is the solution for the issue. Numerous statistical learning technique implemented in the field of text classification. Text classification is the process of classifying text into predefined category belong to it depending on their contents. Text classification consists two phases: the first phase is preprocessing and the second phase is classification which cannot be applied without the first phase. In this paper, Maximum Weight applied on dataset after preprocessing that consists of Tokenization, Stop-words removal, Stemming, in addition to term weight[1].

Electronic text classification can significantly reduce human time and effort in arranging and organizing any type of electronic data.

Arabic corpus didn't as like as English corpus that commonly used such as Reuters dataset collection which has different versions. Arabic corpus lack of publicly available, for this reason,.a new corpus built from the Iraqi media consist of (16,757) documents manually classified into 5 different categories and can be used in future where it published on the internet.

Another difficulty is the nature of morphology Arabic that creates ambiguity in the text, the process of electronic text classification depends on the contents of documents. For this reason, Arabic documents classification is very difficult [2]. The Arabic language is very rich in its vocabulary where there is word has many of meanings, we can be extracted to one meaning using the stemming algorithm, and can be reduced of the dimensionality of the term space using feature selection.

There are many algorithms used for Arabic document classification as Decision Trees, K-Nearest Neighbor, Naïve Bayes, Maximum Entropy, and others which give good accuracy.

K Nearest Neighbor, Naïve Bayes, Multinomial Logistic Regression, and Maximum Weight on same data set. The results of these classifier for one test is (78.903%), (80.334%), (76.937%), (81.585%) respectively.

In this paper, Maximum Weight classifier implemented on the dataset collected after preprocessing phase.

The outline of this section as the following: in section 2 text classification, then section 2.1 preprocessing (first stage in text classification process) that consists four steps, classification using Maximum Weight in section 2.2, the data set that applied classifier on it in section 3, then implementation and result in section 4, and conclusion and future work in section 5.

2. text classification

Text classification (TC) (also known as text categorization, topic spotting, document categorization, or document classification) is an important part of text mining (TM) and natural

language processing (NLP) that tries to replace and save human effort required in performing manual classification. It consists of assigning and labelling documents using a set of pre-defined categories based on their content. Categories selected from a previously established taxonomy[1].

There are two approaches to text categorization: rule-based, and machine learning-based. The rule-based approach means that the classification rules defined manually and documents classified based on these rules. Machine learning approach means that classification rules or equations defined automatically using sample labelled documents[3].

Text classification consists of two phases: the first phase is preprocessing that contains tokenization, stop-words removal, stemming, in addition to term weight. The second phase is classification by classifier. It explained in Fig. (1) below:

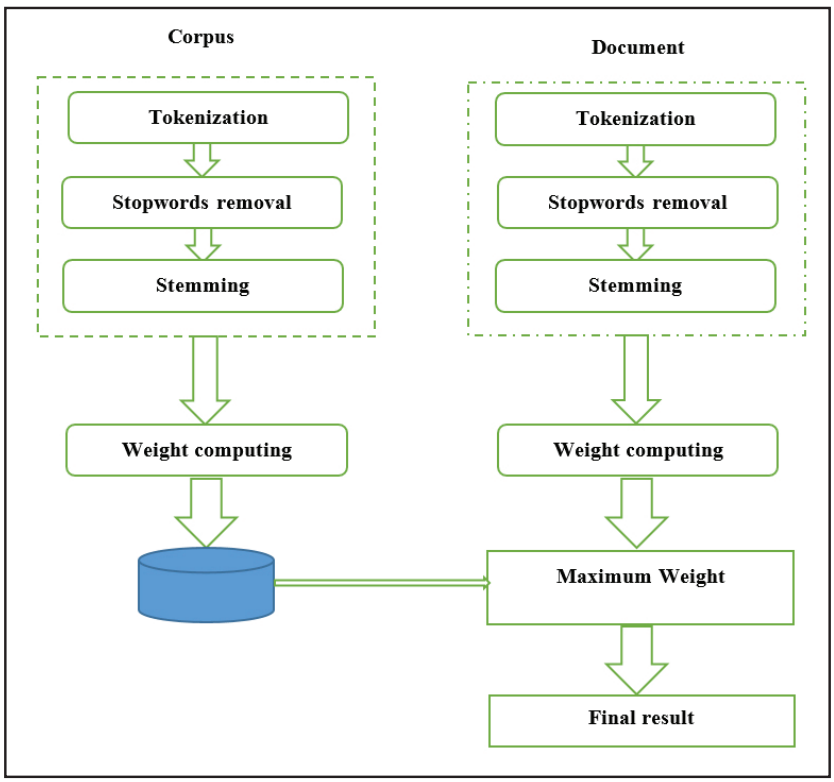

Fig (1): Methodology of Maximum Weight

2.1. preprocessing

Preprocessing represents the first step in TC. It is process prepared and format documents to be an input to machine learning techniques with high efficiency and accuracy. Documents should, firstly, be preprocessed which is an important stage in TC.

The aim of preprocessing is to reduce dimensions and to distinguish between the most important documents by selecting the significant or relative words and ignore the irrelevant words. This will convert the docu-

ments into the more suitable format can manipulate by computer.

The general idea of preprocessing explained in algorithm (1) below.

Algorithm 1: Preprocessing

Input: D: collection of documents

Output: V: collection of vectors

Step 1: for each document in D do:

- Tokenization
- Stop-Word Removal
- Stemming
- Calculate the weights of each word
- Add the vector of weights to V.
- End.

Algorithm (1): The general idea of preprocessing

Step 1: Tokenization

Tokenization is the first phase of preprocessing. It is the act of breaking up a sequence of strings into pieces such as words, keywords, phrases, symbols and other elements called tokens. In the process of tokenization, some characters like punctuation mark discarded. The tokens are the input for another process like parsing and text mining. Usually, tokens or words separated by white space, punctuation marks or line breaks that represent the boundary of words. All characters within contiguous strings are part of the token[4].

Step 2: Stop-Words Removal

Stop word removal is one of the most commonly used preprocessing steps across different NLP applications. The idea is simply removing the words that occur commonly across all the documents in the corpus and unimportant words that appear in a text such as an article or web page (i.e. pronouns, prepositions, conjunctions, etc.). Stop-word will be the word that sifted through earlier or after the handling of natural language processing (NLP). They are generally thought to be a "single set of words". It really can mean different things to different applications. For example, in some applications removing all stop words right from determiners (e.g. the, a, an) to prepositions (e.g. above, across, before) to some adjectives (e.g. good, nice) can be an appropriate stop word list. To some applications. However, this can be detrimental. For instance, in sentiment analysis removing adjective terms such as 'good' and 'nice' as well as negations such as 'not' can throw algorithms off their tracks. In such cases, one can choose to use a minimal stop list consisting of just determiners or determiners with prepositions or just coordinating conjunctions depending on the needs of the application. Finally, this phase consists of the following steps[5]:

i. Converting the documents into individual words stored in an array (Tokenization)

ii. Stop word removal. A comparison made

between the arrays received from step i and the array representing the stop word list, if a match is received, the word deleted from the array. Some fast search technique used by scanning the words array.

iii. Repeating step ii until all stop-words removed from the array.

This step can be speeded up by the used representation of stop list as a suffix tree.

According to this paper, the list of stop words is containing (16178) words that deleted from the text file, in addition to a list containing 169) words including correlatives, signal pronouns, prepositions, etc.

Step 3: Stemming

Different forms of a word often communicate essentially the same meaning. Stemming is the process for reducing inflected words to their word stem (base form) which can be removed any affixes (prefixes that added to the beginning of the word, infixes that added to the middle of the word, and suffixes that added to the ending of the word) from the words to reduce these words to their stems or roots under the assumption that words sharing the same stem. Thus, stemming process is merging those forms to the same stem. In other words, it is the process of reducing modulated words to their word stem, root or base. It is a hard stage in light of the fact that the word could have numerous determinations that change the stem itself, along these lines, it is difficult to recognize root letters and fasten letters[6].

There are two major approaches are very common in Arabic stemming: Light stemming (also called stem-based stemming) that removes prefixes and suffixes of words such as "Khoja stemmer" works well with this approach. And Root-based stemming (also called aggressive stemming) which reduces a word to its root [6].

In this, ISRI [7] has been used. It belongs to the Information Science Research Institute (ISRI) and shares many characteristics with Khoja stemmer, but it does not use a dictionary for word roots.

Step 4: Term Weight

There are many of methods for compute weight of feature subset selection are TF-IDF, Information Gain, Chi-Square, mutual information, and Gini index. In this paper, the method to select feature is TF-IDF where compute term frequency and inverse document frequency according to equation (1).

tf: Term Frequency $tf_{(t,d)}$ of term t in document d is defined as the number of times that t occurs in d.

idf: Inverse Document Frequency estimates the rarity of a term in the whole document collection. (If a term occurs in all the documents of the collection, its IDF is zero)

$$idf_i = log(\frac{\text{Number of documents}}{Number\ of\ documents\ that\ contain\ word\ i}) \quad \ldots(1)$$

TF-IDF combine the definitions of term frequency and inverse document frequency, to produce a composite weight for each term (t)

in each document that is calculated as follows:

$$\text{Weight}(t) = tf * idf \quad \ldots(2)$$

Each word with idf equal to zero is deleted when computing weights, thus reduces the size of the text file.

2.2. classification

After the preprocessing stage that mentioned in detail in the previous section, the data set is ready to input to the second stage from phases of text classification, which is techniques of classification that applying on the text. The classifier applies rules (may be learned from labelled data) on the input.

In this paper, Maximum Weight classifier has been used to classify Arabic text documents. The input for this classifier is the out-

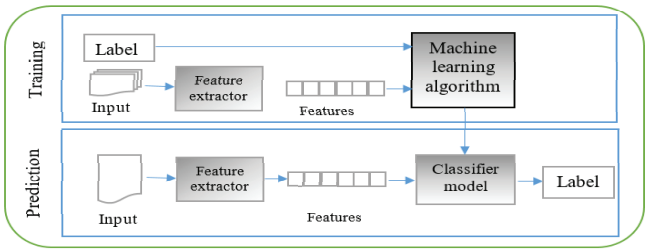

Fig (2): supervised classification

- **Maximum Weight**

Maximum Weight can be implemented simply where each feature will be related to one class not more, this help us where feature reduction is not matter for the same mentioned reason because each class has different subset of all features set and then feature reduction inclusively used. The algorithm (2) of this classifier is below:

algorithm 2: Maximum Weight

Input: set of classes C ={c_1,c_2 … c_n}; txtfwtr= dictionary of words and weight of it for traning data; txtfwts =test document

.**Output**: name of class

Step1: Let W_{imax}= maximum weight of the term i .among all these classes

Step2: Let C_{ij}= is a binary value which indicate the class j contain the maximum weight of the term i

Step3: **for** each word (w) in txtfwts **do**

for each class in C **do**

If (class contain w) **then**

Result=C_{ij} *W_{imax}

End for

Add result to dictionary

End for

.Name of class=argmax dictionary

Return name of class

.End

Algorithm (2): The algorithm of this classifier

3. corpus

Corpus used in this paper collected from Al Sabah newspaper. It is written in Arabic Language. It used for first time in machine learning. The corpus took a long of time to clean and arrangement and classified it manually. It consists of (16757) Arabic documents belong to 5 different categories (see Table (1)). Preprocessing implemented on it then MW classifier that is supervised learning; the

reform the corpus used for learning and evaluating the performance which is divided into two parts training and test data. (15080) documents (90%) is training data and (1677) documents is used for testing (10%).

Table (1): Dataset

Category	Total documents#	training documents#	testing documents #
Literature and Arts	2175	1957	218
Family and community	1017	915	102
Economy	3411	3070	341
Sport	8546	7691	855
Science and Technology	1608	1447	161
Total	16757	15080	1677

غياب أوين عن مانشستر يونايتد ستة أسابيع

16 /11 /2011 12:00 صباحا

بغداد –وكالات

أعلن نادي مانشستر يونايتد بطل الدوري الإنكليزي لكرة القدم في الموسم الماضي أن المهاجم مايكل أوين سيغيب عن الملاعب نحو ٦ أسابيع بداعي الإصابة.

وأوضح النادي أن أوين يaعاني من إصابة في ظهره تعرض لها خلال المباراة التي فاز فيها الفريق على غالاتي الروماني (٢-٠) قبل نحو أسبوعين ضمن مسابقة دوري أبطال أوروبا.وساهم أوين في صنع الهدف الأول بتمريرة جميلة إلى زميله الإكوادوري لويس انطونيو فالنسيا الذي وضعها في الشباك (٨).

من جانبه، قال المدرب الاسكتلندي أليكس فيرغوسون أن أوين الذي أصيب لدى قيامه بحركة صعبة للسيطرة على الكرة، «يعاني من تمزق عضلي وسيغيب ٦ أسابيع".

Fig. (3) Sample on Data Set

4. Implementation and Result

MW text classification algorithm on the new Arabic corpus after many stages from pre-processing (Tokenization, Stop-Words Removal, Stemming, Stop-Words Removal second time to reduce text file). In addition to compute weight of each term is calculated using.

The result of implement MW is (82.66%) in average where 10-fold cross validation to reliability of this classifiers. The result shown in Table (2). Below.

Table (2): Result of MW using 10-fold cross validation

Fold#	training files #	testing files #	MW
1	15080	1677	81.585%
2	15080	1677	83.13%
3	15080	1677	82.11%
4	15080	1677	74.84%
5	15080	1677	85.39%
6	15082	1675	83.363%
7	15082	1675	85.868%
8	15082	1675	83.304%
9	15082	1675	84.496%
10	15082	1675	82.469%

The accuracy of MW for 10 tests are explained in Fig. (4) below:

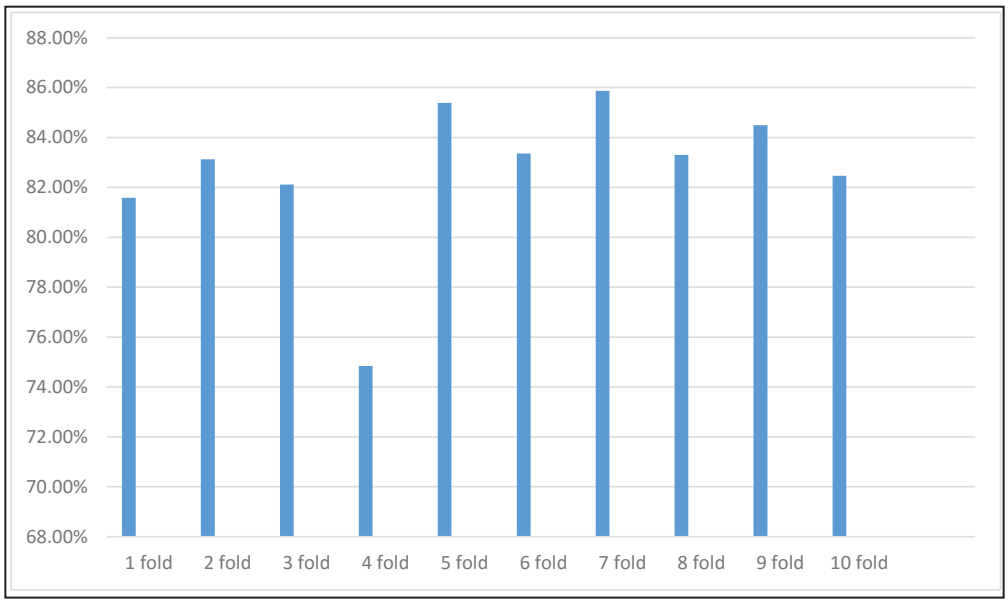

Fig. (4): Accuracy of Maximum Weight

5. Conclusion And Future Work

As we can see, the work focus on: (i) pre-processing Arabic text documents (ii) constructing a new Arabic corpus taken from Iraqi media which can be used by the researcher in the future, (iii) testing the corpus with a new classifier called Maximum Weight that proves efficiency and speedily.

In MW where each feature is relative to one class not more result in each class has its own features. In other words, each class has the different subset of all features set, and then feature reduction inclusively used. This algorithm tested with a new Arabic corpus. It can't be used for multi-class classification problems without modification.

It can be applied on anther dataset such as handwritten, social web, image, video, etc.

References

[1] A. Abu-Errub, "Arabic Text Classification Algorithm using TFIDF and Chi Square Measurements," International Journal of Computer Applications, vol. 93, (2014).

[2] A. Al-Badarenah, E. Al-Shawakfa, K. Al-Rababah, S. Shatnawi, and B. Bani-Ismail, "Classifying Arabic text using KNN classifier", (2016).

[3] P. Y. Pawar and S. Gawande, "A comparative study on different types of approaches to text categorization," International Journal of Machine Learning and Computing, vol. 2, p. 423, (2012).

[4] H. M. Dawoud, "Combining different approaches to improve Arabic text documents classification," MSc Thesis, Islamic University, (2013).

[5] J. K. Raulji and J. R. Saini, "Stop-Word Removal Algorithm and its Implementation for Sanskrit Language."

[6] A. S. A. Babiker, "Improving Stemming Algorithm for Arabic Text Search," Sudan University of Science and Technology, (2014).

[7] K. Taghva, R. Elkhoury, and J. Coombs, "Arabic stemming without a root dictionary," in Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on, pp. 152-157, (2005).

[8] S. Bird, E. Klein, and E. Loper, Natural language processing with Python: analyzing text with the natural language toolkit: " O'Reilly Media, Inc.", (2009).