Journal of Engineering and Applied Sciences 14 (7): 2292-2301, 2019
ISSN: 1816-949X

Feature Engineering for Arabic Text Classification

Ghassan Khazal and Alexander Zamyatin
Department of Computer Science, Tomsk State University, 36 Lenin Avenue, Tomsk, Russia

Abstract: Arabic is one of the most complex languages and it has a rich vocabulary also it has difficult and different structure when compared with the others languages. Arabic language has many challenges in text mining one these challenges are how to achieve highest classification accuracy. We proposed in this research a feature engineering of the best combination of preprocessing procedures with appropriate feature representation that has direct affected the classification accuracy of the Arabic text. Preprocessing and feature representation represent the main steps in any text classification framework. This phase is very important to design any text classifier that deals with this sophisticated language. In this study, we used four classification classifiers Support Vector Machine (SVM), Decision Tree (DT), Naive Bayes (NB) and K-Nearest Neighbor KNN. From analysis and experimental results on Arabic text data we reveal that preprocessing techniques and feature representation and weighting have an important influence on the classification accuracy. Also, its depend on choosing the suitable combinations of preprocessing tasks with the appropriate feature representation and classification techniques provides a good improvement in the accuracy of classification. This study shows that the SVM (82.6%) and KNN (78.33%) have better performance on average over the DT (57.49%) and NB (76.21%). The SVM achieved accuracy (88.67%) with the combination of tokenization, filtering, normalization and light stemming with TFIDF as feature representation and KNN classifier gives 88.00% using the combination of tokenization, filtering as preprocessing and TFIDF as feature representation with information gain as feature selection.

Key words: Feature engineering, Arabic text classification, text preprocessing, feature selection, stemming techniques, classification techniques

INTRODUCTION

Text classification can be found in many applications in the world, e.g., the news is typically organized by categories by topics, language and geographical also any scientific papers usually categorized by scientific domain and sub domain. Another widespread application is spam filtering where emails are classified into the spam and non spam. Then we can define text classification is an important natural language processing task in the current era of data mining and big data. A lot of researchers have been worked and studied text classification in English language and other languages. However, for the Arabic language the fifth language of the world in terms of numbers of native speakers (according to Wikipedia) Arabic text classification is still very limited. So, the goal of this study is to design and find an appropriate "feature engineering" that give best classification accuracy and also compare the influence of the different combination of preprocessing, feature representation and feature weighting on the classification accuracy of Arabic text using classical classifiers.

The nature of Arabic text is different than that of English text also there are many differences between Arabic text classification and English text classification (Mechti *et al*., 2016). Preprocessing of Arabic text is more challenging and need more work to clean the text than English. In tokenization process is usually sufficient in English but non-English language documents will certainly necessitate custom tokenization or segmentation. Also, in Arabic there are normalization rule can be used in to change some letters to reduce the dictionary size reduction while in English just change the character from upper to lower case or can be used to some dictionaries do this process to change some word with same synonym. Stemming or lemmatization in English this process just removes affixes from the word to ensure that morphologically different forms of a word are changed into the same stem. In Arabic there are not just affixes also there are infix and need complicated process to get the root of the word for this, we use different stemming algorithms light and root stemming (Ayedh *et al*, 2016; Pak and Gunal, 2017; Uysal and Gunal, 2014).

Corresponding Author: Ghassan Khazal, Department of Computer Science, Tomsk State University, 36 Lenin Avenue, Tomsk, Russia

To achieve our goal we must pass through a set of phases. The first phase is collecting deferent types of Arabic data set. The second phase is the preprocessing which consists of separate text to token and do some filtering operation like remove numbers, remove stop words, remove diacritics, etc., also can make some transformation in operation called normalization and also can extract root of a word or remove suffix and prefix using operation called stemming. The third phase of represent the text in different feature representation and feature weighting. The fourth phase is applied four types of classifiers and finally evaluates the classification accuracy where the text separated to training data and testing data. In this phase, training phase where we use one of the classification algorithms to get the trained model that will be evaluated with the testing data (Sallam *et al*., 2016; Axyonov *et al*., 2016; Mendez *et al*., 2005).

Literature review: There is previous work on feature engineering for text classification to enhance text the performance, the undertaken tasks on text classification is primarily conducted in English. Scott and Matwin (1999) examines a large set of text representations including lexical (bag-of-words), syntactic (noun phrases and key phrases) and semantic features (synonym and hyponyms relations and concepts from WordNet). Syntactic features have also been used in combination with other lexical or semantic ones. Os *et al*. evaluate the impact of different text engineering of biomedical texts for reproducing the Medical Subject Headings MeSH annotations of some of the most frequent (MeSH) headings with unigrams and bigrams for features that include noun phrases, citation meta-data, citation structure and semantic annotation of the citations. And conclude that specific combinations of learning algorithms and appropriate features could further increase the performance of an indexing system. Forman and Kirshenbaum (2008) describes a fast method for text feature engineering that folds together unicode conversion, forced lowercasing, word boundary detection and string hash computation And show empirically that our integer hash features result in classifiers with equivalent statistical performance to those built using string word features but require far less computation and less memory. Garla and Brandt (2012) presents a novel feature ranking method that utilizes the domain knowledge encoded in the taxonomical structure of the unified medical language system and we developed a novel context-dependent semantic similarity measure.

For Arabic researchers explore and compare different kinds of feature engineering and classification techniques in Arabic text classification. For example, Alahmadi *et al*. (2013) focus on combining a bag of word and bag of concepts as text representation. They described five different types of feature representation and experiment with different classifiers and the result was SVM has best performance over NB and C4.5, respectively. Mesleh (2007) also used a bag of word representation with Chi-square for feature selection. And he used SVM classifier and yield 88.11% accuracy when applied to in-house Arabic data set. Alabbas *et al*. (2016) summarize the main characteristics of the different text classification techniques and methods used in Arabic text classification. Wahbeh and Al-Kabi (2012) make a comparison between three classification methods using Arabic text which consist of four classes (sports, economics, politics, Al-Hadith Al-Shareef). The comparison is focused on two main aspects accuracy and time. The results showed that the NB gives better accuracy then SVM and classifier, respectively. But its shows the time is taken to build the SVM Model is better than NB and J48 Model classifiers. Abdelwad uses SVM the researchers focus on performance on Arabic text classification system. They used Chi-square as feature selection. They also use many steps of preprocessing to give a better evaluation. Also, the proposed system gives a good result when the researcher use many features to show the effectiveness of six feature selection methods (χ^2, NGL, GSS, IG, OR and MI) with SVM classifier. Also, Table 1 shows the differences between our research and previous works. In this table, we use some abbreviate, Normalization (NO),

Table 1: Comparison of existing Arabic text classification cases

Paper	SW	WL	NO	ST	SL	FR	FS	Classifier
Duwairi *et al*. (2009)	+	-	-	+	-	-	-	KNN
Al-Shargabi *et al*. (2011)	+	-	-	-	-	-	-	NB, SVM, J48
Khorsheed and Al-Thubaity (2013)	+	-	+	-	-	TFIDF	-	KNN, NB, SVM
Mesleh (2008)	+	-	+	-	-	TFIDF	Chi^2, NGL,GSS, OR, MI	SVM
Sharef *et al*. (2014)	+	+	+	-	+	TF	Chi^2, GSS, OR, MI	NB
Al-Walaie and Khan (2017)	--	-	-	-	-	-	-	NB, DT
Al-Thubaity and Al-Subaie (2015)	+	-	+	-	-	-	Chi^2	SVM
Al-Anzi and AbuZeina (2017)	+	-	+	-	-	TFIDF	NB, KNN, NN, SVM, DT	
Alahmadi *et al*. (2013)	+	-	+	-	-	BOW	-	SVM, NB and C4.5
Proposed study presenting	+	+	+	+	+	Bool, TF, TFIDF	Chi^2, IG, MI	SVM, KNN, NB, DT

Stop Word removal (SW), Word Length (WL), root Stemming (ST), Light Stemming (SL), Feature Representation (FR) and Feature Selection (FS).

Arabic language structure: The Arabic language represents the 5th languages in the world it is spoken and used by more than (450) million people http://www.mdpi.com/1999-4893/9/2/27/htm-B14-algorithms-09-00027 (Kanan and Fox, 2016). The Arabic is one of the semantic languages that have sophisticated morphology, Arabic is completely different from the most popular languages, like Spanish and English (Thabtah *et al*., 2012). Hence, Arabic grammar has different form and has a very complex morphology format when compared to the English language. Arabic words formed by connected three-root consonants with fixed vowel patterns and sometimes an affix in a cursive script. Arabic texts are read from right to left. There are no upper and lower case characters and the rules for punctuation is easier than in English. The Arabic language consists of 28 letters:

(آ ; ب ;ت ; ث ; ج ; ح ; د ; ذ ; ر ; س ; ش ; ض ; ص ; ط ; ظ ; ع ; غ ; ف ; ق ;ك ; ل; م ; ن ; ه ; و ; ي)

Arabic has three grammatical cases: nominative, accusative and genitive. And two genders: feminine and masculine. In general Arabic words can be classified into nouns, verbs and particles. A noun represents as nominative when it be a subject and represent as accusative when it be an object of a verb and as genitive when it be an object of a preposition. Also, verbs could be perfect, imperfect or imperative. Particle includes pronouns adjectives, adverbs, conjunctions, prepositions, interjections and interrogatives (Zrigui *et al*., 2012). Arabic also uses diacritics symbols below or above the letter to add grammatical formulation, distinct pronunciation or to make different meaning of the word. Diacritics include (ء, ّ, ً, ٌ, ٍ, ْ, ِ, ُ, َ, ٓ). So, because of this complex structure of Arabic, we can't apply the same preprocessing.

MATERIALS AND METHODS

Our proposed methodology aims to develop an Arabic text classification model based four main tasks: pre-processing (where the corpus is prepared, text is tokenized, filtered according to words characters number, stop-words are removed, morph syntactic or semantic information is added or removed, etc., so that, at the end, we have not words but "features"), feature representation where we select the most relevant features in order to increase pertinence and decrease memory and CPU cost like Booleans (Bool), Term Frequency (TF) and Term Frequency Inverse Document Frequency (TFIDF). Feature weighting schemes also called "feature ranking" and finally classification. Feature selection and feature weighting are sometimes merged into a single operation, for example, one may use TFIDF which is strictly speaking, a feature weighting method for feature selection by sorting features in decreasing order and taking the most highly ranked. Nevertheless, feature selection and weighting are clearly independent of the classification operation and one can combine feature selection/weighting and classification methods freely in search of optimal solutions. Figure 1 represents classical Arabic text classification and our scheme for the proposed method.

Pre-processing task

Tokenizing (TOK): The process of splitting written string into tokens called tokenization which it represent the first step in many natural language processing tasks. For most languages, tokenization involves splitting punctuation and some affixes off of the words. In rich morphological languages, like Arabic, it's requiring a more extensive process to separate different types of facilities and particles from the word (Green and DeNero, 2012).

Filtering (FL): The purpose of this task to remove insignificant terms and the most common words in text documents to minimize the dimensionality of term space. This step may be including:

- Removing non-Arabic characters
- Removing stop words (articles, prepositions and pronouns
- Removing numbers
- Removing diacritics
- Word length <3 characters

Normalization (NOR): Normalization in Arabic word means replacing specific letters within the word with other letters according to a predefined set of rules as shown in the Table 2 (Sallam *et al*., 2016).

Sometimes these replacements can produce misspelled words, these misspellings are common in Arabic words and the normalization may help to avoid the side effects of such misspellings on the performance of classification. For example, the normalization of the word "مكتبة" (library) and its misspelled version "مكتبه" (his office) results in the same normalized word "مكتبه". Hence, the two will be the same it is worth such misspellings occurring in official documents or newspapers.

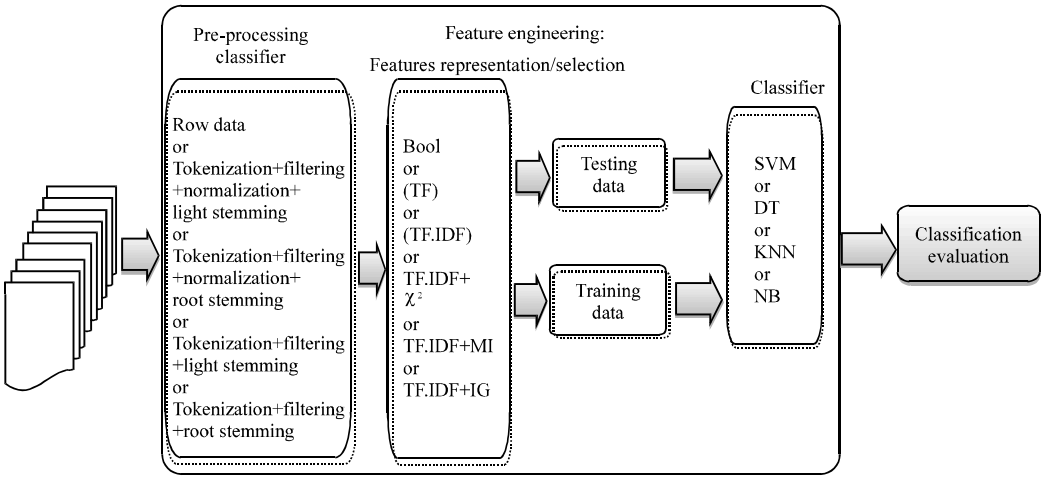

Fig. 1: Proposed a methodology Arabic text classification

Table 2: Normalization rules table

Rules	Orignal character	Normalize character	-----Examples----	
Alef	ء	ا	غناء	غنااا
Alef	أ	ا	احمد	احمد
Alef	إ	ا	إحسان	احسان
Waw	ؤ	و	مؤيد	موید
Yaa	ئ	ي	موائئ	موائي
Alef	ا	أ	أل	ال
Yaa	ى	ي	روى	روي
Haa	ة	ه	الة	اله
Tshkeel	ـ	-	مكتب	مكتب

Stemming Arabic words: To reduce the high dimension feature space of the text, we can use stemming, ant its mean extract the words from the different forms. Stemming can improve the accuracy of classification model by this reduction of representation vector. In Arabic we can use two types of stemmers Root base Stemmer (RST) and Light Stemmer (LST). RST used to remove all the suffix, prefix and infix attached to the word to extract the root of the given word. LST use predefined rules to remove the suffix and prefix from the word. These two kinds of stemmers are developed by Khoja (2001) and Larkey *et al.* (2002), respectively.

For each word in Arabic there are basic form represent the root. So, we can get different words types such nouns, verbs or adjectives by attach letters at beginning inside/ending of any root or just adding diacritics over any letter as shown in an example of different Arabic words for the root "كتب".

(كتب، كتبت، كتبا، كتبوا، كتبن، يكتب، يكتبان، تكتبان، يكتبون، يكتبن، سيكتب، ستكتب، سيكتبان، ستكتبان، سيكتبون، ستكتبن، كاتب، كاتبة، كاتبان، كاتبتان، كتاب، كاتبات، كتاب، كتابان، كتب، مكتب، مكتبه، مكتبة)

Feature representation: To apply classification algorithm we need an appropriate form to represent the feature space usually represented in matrix form. These elements represent feature and weight of the text in the dataset. Three schemes used in this resaerch to represent the feature as Bool, TTF and TFIDF (Al-Thubaity *et al.*, 2013). Equation 1-3 represent the mathematical form of these schemes, respectively:

$$\alpha_{ti}\begin{cases}1, \text{ if the word exists in the document}\\0, \text{ if the word does not exists in the document}\end{cases} \quad (1)$$

$$a_{tj} = C(w_i, t)\log\left(\frac{T}{f(t, w_i)}\right) \quad (2)$$

$$a_{tj} = \frac{\log\left(f(t, w_i)+1\right)\left(\frac{T}{f(t, w_i)}\right)}{\sqrt{\sum_{i=1}^{i=m}\log\left(f(t, w_i)+1\right)\left(\frac{T}{f(t, w_i)}\right)^2}} \quad (3)$$

Where:

T = Total No. of the words in text data
$f(w_i, t)$ = Word frequency in the text
$f(w_i, t)$ = Represent how many times word occur in text

Feature selection: The number of the words if large even the text is small and the text is big the number of words will huge. So, the dimension of feature space for the collected text data can be reached to thousands or tens of thousands, moreover, when the text representation as a vector also it will have thousands or tens of thousands of non zero component (Kobyz and Zamyatin, 2015). These

words may be unusual in the classification process and can be removed without any disadvantage in the classification accuracy and also it may improve the accuracy.

Chi-square (χ^2): Use statistical test to determine the divergence from the distribution expected feature that obvious independent from class value. χ^2 measures the dependence of the maximum strength between the feature and the category (Bahassine *et al*., 2016; Feldman and Sanger, 2007):

$$X^2(c, t) = \frac{N*(A.D-B.C)}{(A+C).(B+C).(A+B).(C+D)} \quad (4)$$

Where:

A = Frequency of word
t = Occur in class
c = Occurrences
B = Frequency of word
t = Occur without Class C
C = Frequency of Class C without t
D = Frequency of non occurrence of both Class C and word t
N = The quantity of document

Information Gain (IG): Measures the bits number of the obtained information of prediction of classes by know the absence or presence in tern in the document. The probabilities can calculated as ratios of frequencies of the trained data:

$$IG(t) = -\sum_{i=1}^{|c|} P(C_i).\log P(C_i) + P(t).\sum_{i=1}^{|c|} P(C_i|t).\log P(C_i|t) + P(\bar{t})\sum_{t=1}^{|c|} P(C_i|\bar{t}).\log P(C_i|\bar{t}) \quad (5)$$

Where:

$P(c_i)$ = Likelihood of c_i class
P(t) = Likelihood occurrence of t

Mutual Information (MI): It's statistical language modeling method that determine the mutual dependency between a term t and a class c by using two way contingency table, we suppose that a term t and a document class c, A represent how many times that t appears in c, B represent how many times that t happens in all classes except c, C represent how many times c happens without t and N is the total number of text documents, then MI can calculated in Eq. 6 and 7 (Sharef *et al*., 2014):

$$MI(t, c) = \log \frac{P(t, c)}{2*P(t)*P(c)} \quad (6)$$

$$MI(t, c) = \log \frac{A*N}{(A+C)*(A+B)} \quad (7)$$

Classifiers

K-Nearest Neighbors (KNN): This classifier based on the nearest training sample located in the feature space. KNN is lazy learning where the function is only approached locally and all the computations are deferred until classification complete. Also, KNN is the easiest machine learning algorithms where the object categorized by the majority vote of its neighbors with the object that assigned to the class most common amongst its k nearest neighbors (Saad, 2010).

Naive Bays (NB): This classifier is probabilistic based on Bayes theorem with strong assumptions of independence. i.e., a this classifier suppose the presence or absence of a particular feature of the class and it is not related to the presence or absence of any others classes features (Al-Anzi and AbuZeina, 2015, 2017).

Decision Tree (DT): This classifier take tree form was the training cases collected to construct the classification tree. C4.5 represents the most known DT algorithm and it's an extension of the earlier version of DT algorithm. The nods in the tree mean the label of class or the target of the classification. DT mechanism is how classifying unseen instances by testing at each node some features values and determine is this class of given unseen instance. The test begins form the root node and goes down until a reached to the node that indicates the class of the unseen instance (Stamate *et al*., 2018).

Support Vector Machine (SVM): This classifier related to supervised learning algorithm that widely used for classification and regression, i.e., it mark set of given training classes to one or more classes and then the training algorithm builds the model that will predics the new test classes into one or to the others. SVM represents the examples classes as points in space map, so, these examples of the different classes are divided by a clear gap as possible. Then the new examples mapped into that same space and predicted to which category belongs based on which side of the gap they fall on (Lin *et al*., 2017; Saad, 2010).

TC evaluation measure: One of the standard measures of text classification accuracy is confusion matrix. It used to

(a)

Accuarcy: 68.00% +/-3.48%	True cards issue	True customer compliance	True bank protection	True client protection	True bank clients	True insurance clients	Class precision
Pred. cards issue	48	34	6	30	10	10	34.78%
Pred. customer compliance	0	13	0	0	0	0	100.00%
Pred. bank. protection	2	1	43	0	0	0	93.48%
Pred. client protection	0	2	0	20	0	0	90.91%
Pred. bank clients	0	0	0	0	40	0	100.00%
Pred. insurance clients	0	0	1	0	0	40	97.56%
Class recall	96.00%	26.00%	86.00%	40.00%	80.00%	80.00%	

(b)

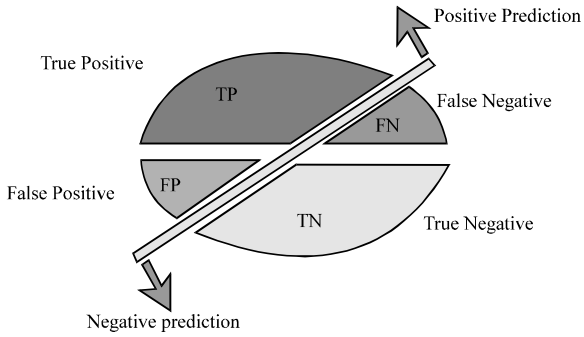

Fig. 2: a, b) Snapshot of classified text in confusion matrix

evaluate and provide straightforward way to clearly understand the definition of True Positive or Negative (TP or TN) and False Positive or False Negative (TF or FN) with respect to the category (Alayba *et al*., 2017) as shown in Fig. 2. The accuracy and error rate of the classifier can be calculated using following Eq. 8:

$$\text{Accuarcy} = \frac{TP+TN}{P+N} \quad (8)$$

$$\text{Error rate} = \frac{FP+FN}{P+N} \quad (9)$$

RESULTS AND DISCUSSION

The goal of this research is to design "feature engineering" to show the effect of the different combination of preprocessing, feature representation and feature weighting on the classification accuracy of Arabic text using classical classifiers. As described in section 1, keyword-based Arabic text classification is constructed of four steps:

Alghorithm 1; Keyword-based Arabic text classification:
Step 1: gathering Arabic text data to apply our model
Step 2: apply a different combination of text data preprocessing to clean the text from the unwanted text
Step 3: apply different feature representation/selection
Step 4: apply different classification algorithms and evaluate the results

Arabic corpus collection: We created an Arabic corpus dataset that contains 300 documents belonging to 6 different topics (50 documents for each category) in banking client's questions and queries. The corpus contains 3150 unique words. Table 3 shows the statistics of the created corpus dataset. Also, we apply our model on different corpora to perform our tasks, the corpora include small or large size corpus with many categories. Figure 3 shows different Arabic corpora size and number of classes in each corpus.

Experimental results and analysis: In this part apply a combination of all possible preprocessing tasks and different kinds of feature representation on the gathered dataset to show how it's effect on classification accuracy of Arabic text classification. We apply thirty experiments

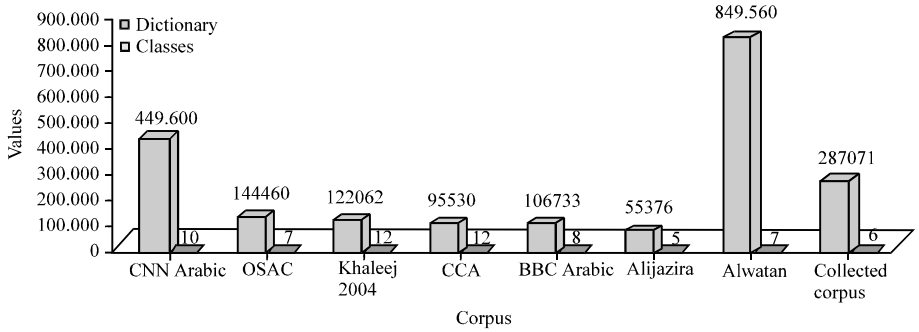

Fig. 3: Dictionary size and number of classes for each corpus

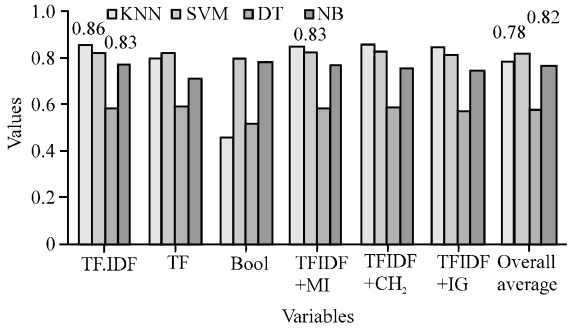

Fig. 4: The distribution of average acuuracy achieved with different classifiers

Table 3: Data collection statistics

Category name in Arabic	Category name in English	Doc. No.
إصدار وتشغيل بطاقات الإئتمان	Issuing and operating credit cards	50
المنازعات والمخالفات التامينية	Disputes and insurance violations	50
حماية عملاء البنوك	Protecting bank customers	50
حماية عملاء شركات التمويل	Protecting clients of finance companies	50
عملاء البنوك	Bank customers	50
حماية عملاء شركات التامين	Clients of insurance companies	50

Table 4: Experiment parameters

Parameters	Descriptions
Dataset	In-house collected dataset "banking client's questions and queries"
Training size	70%
Testing Size	30%
Pre-processing	Tokenizing, filtering (stop word removal, word length <3 characters, non-Arabic word removal, numbers removal, districts removal), Normalization, steaming and light stemming
Feature selection	IG, CHI, MI
Feature representation	Boolean, term frequency and TFIDF
Classifiers	NB, KNN, DT, SVM

to show the effect of each of the preprocessing and feature tasks (tokenization, filtering (word length <3 characters, non-Arabic character removal, stop word removal, etc.,), normalization, stemming) and feature representation (Boolean, TF, TFIDF) and feature selection (χ^2, IG and MI). The results showed that the SVM accuracy in average better than the other classifiers for all experiments. The results of all applied experiments on the four classification algorithms are illustrated in Table 4 and Fig. 4.

According to the proposed method, TFIDF in general has a positive influence on classification accuracy in general and with Boolean, the accuracy downgraded except SVM working well with steaming. DT is not scalable in the high dimensional dataset and it requires very long training time also gives accuracy less more than other classifiers. NB the accuracy decrease when we using stemming, additionally, term weighting schemes have a sometimes not affect or make some enhancement on the classifying accuracy. KNN, SVM and NB variant have superior performance and achieved the best classification accuracy than DT. Where NB gives the best result (0.87) when we use al filtering techniques, normalization and root stemming with TF as feature representation without any feature selection method. KNN gives the best result (0.87) when we use al filtering techniques with light stemming from TFIDF as feature representation without any feature selection method. SVM gives the best result (0.87) when we use al filtering techniques with light stemming from TFIDF as feature representation with MI feature selection method. DT gives the best result (0.6100) when we no use any preprocessing with TFIDF as feature representation with any feature selection method. In these results, we observe that the normalization improves the accuracy. As long as normalization should be used without depending on the representation of feature or classifier type. However, filtering techniques and light stemming and root stemming can influence classification accuracy also affected by the feature size and the classifier type. Finally, feature selection that used in our model

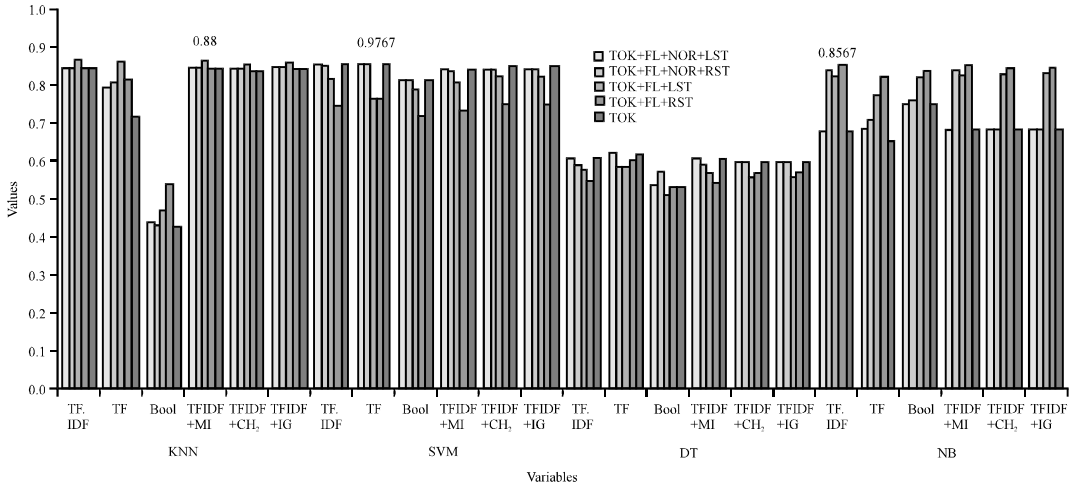

Fig. 5: Experimental results proposed tasks using the four classifiers with a different combination of features representation

approves that, we reduce the dimensionality of the feature size not harm or influence the classifying accuracy (Fig. 5).

CONCLUSION

The goal of this research is to design appropriate feature engineering for Arabic text classification. We construct our model with four classical algorithms (SVM, KNN, NB and DT) and applied on Arabic text dataset. The result shows that the SVM and KNN have better performance on average over the DT and NB. The SVM achieved 88.67% using these combinations of feature engineering in preprocessing tokenization, filtering, normalization and root stemmer and TFIDF as feature representation with/without any feature selection. KNN achieved 88.00% using the combination of tokenization, filtering as preprocessing and TFIDF as feature representation with information gain as feature selection.

We conclude that light stemming is the best feature reduction technique because light stemming is more proper than stemming from linguistics and semantic viewpoint and it has the least preprocessing time it also has superior average classification accuracy.

RECOMMENDATIONS

From the experimental analysis, feature engineering very important to enhance the classification accuracy and depend on language studied. Finally, Arabic text classification and feature engineering is promising research field due to the complexity and problems in different aspects.

REFERENCES

Al-Anzi, F.S. and D. AbuZeina, 2017. Toward an enhanced Arabic text classification using cosine similarity and Latent Semantic Indexing. J. King Saud Univ. Comput. Inf. Sci., 29: 189-195.

Al-Anzi, FS. and D. AbuZeina, 2015. Stemming impact on Arabic text categorization performance: A survey. Proceedings of the 5th International Conference on Information and Communication Technology and Accessibility (ICTA), December 21-23, 2015, IEEE, Marrakech, Morocco, ISBN: 978-1-4673-8749-1, pp: 1-7.

Al-Shargabi, B., F. Olayah and W.A. Romimah, 2011. An experimental study for the effect of stop words elimination for Arabic text classification algorithms. Intl. J. Inf. Technol. Web Eng., 6: 68-75.

Al-Thubaity, A. and A. Al-Subaie, 2015. Effect of word segmentation on Arabic text classification. Proceedings of the 2015 International Conference on Asian Language Processing (IALP), October 24-25, 2015, IEEE, Suzhou, China, ISBN:978-1-4673-9595-3, pp: 127-131.

Al-Thubaity, A., N. Abanumay, S. Al-Jerayyed, A. Alrukban and Z. Mannaa, 2013. The effect of combining different feature selection methods on Arabic text classification. Proceedings of the 14th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, July 1-3, 2013, Honolulu, HI., pp: 211-216.

Al-Walaie, M.A. and M.B. Khan, 2017. Arabic dialects classification using text mining techniques. Proceedings of the 2017 International Conference on Computer and Applications (ICCA), September 6-7, 2017, IEEE, Doha, United Arab Emirates, ISBN: 978-1-5386-2752-5, pp: 325-329.

Alabbas, W., H.M. Al-Khateeb and A. Mansour, 2016. Arabic text classification methods: Systematic literature review of primary studies. Proceedings of the 4th IEEE International Colloquium on Information Science and Technology (CIST), October 24-26, 2016, IEEE, Tangier, Morocco, ISBN:978-1-5090-0751-6, pp: 361-367.

Alahmadi, A., A. Joorabchi and A.E. Mahdi, 2013. Combining bag-of-words and bag-of-concepts representations for Arabic text classification. Proceedings of the 25th IET Irish Signals and Systems Conference on China-Ireland International Conference on Information and Communications Technologies, June 26-27, 2013, Limerick, Ireland, pp: 343-348.

Alayba, A.M., V. Palade, M. England and R. Iqbal, 2017. Arabic language sentiment analysis on health services. Proceedings of the 1st International Workshop on Arabic Script Analysis and Recognition (ASAR), April 3-5, 2017, IEEE, Nancy, France, ISBN:978-1-5090-6628-5, pp: 114-118.

Axyonov, S., A.V. Zamyatin, J. Liang and K. Kostin, 2016. [Advanced pattern recognition and deep learning for colon polyp detection]. Proceedings of the 19th International Scientific Conference on Distributed Computer and Telecommunication Networks: Management, Computing, Communication (DCCN'16) Vol. 2, November 21-25, 2016, Tomsk State University, Tomsk, Russia, pp: 27-34 (In Russian).

Ayedh, A., G. Tan, K. Alwesabi and H. Rajeh, 2016. The effect of preprocessing on Arabic document categorization. Algorithms, Vol. 9, 10.3390/a9020027

Bahassine, S., A. Madani and M. Kissi, 2016. An improved Chi-sqaure feature selection for Arabic text classification using decision tree. Proceedings of the 11th International Conference on Intelligent Systems: Theories and Applications (SITA), October 19-20, 2016, IEEE, Mohammedia, Morocco, ISBN :978-1-5090-5782-5, pp: 1-5.

Duwairi, R., M.N. Al-Refai and N. Khasawneh, 2009. Feature reduction techniques for Arabic text categorization. J. Am. Soc. Inform. Sci. Technol., 60: 2347-2352.

Feldman, R. and J. Sanger, 2007. The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data. Cambridge University Press, New York, ISBN: 9780521836579, Pages: 410.

Forman, G. and E. Kirshenbaum, 2008. Extremely fast text feature extraction for classification and indexing. Proceedings of the 17th ACM International Conference on Information and Knowledge Management, October 26-30, 2008, ACM, Napa Valley, California, USA., ISBN:978-1-59593-991-3, pp: 1221-1230.

Garla, V.N. and C. Brandt, 2012. Ontology-guided feature engineering for clinical text classification. J. Biomed. Inf., 45: 992-998.

Green, S. and J. DeNero, 2012. A class-based agreement model for generating accurately inflected translations. Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers Vol. 1, July 08-14, 2012, Association for Computational Linguistics, Stroudsburg, Pennsylvania, USA., pp: 146-155.

Kanan, T. and E.A. Fox, 2016. Automated Arabic text classification with P-Stemmer, machine learning and a tailored news article taxonomy. J. Assoc. Inf. Sci. Technol., 67: 2667-2683.

Khoja, S., 2001. APT: Arabic part-of-speech tagger. J. Comput., 2001: 20-25.

Khorsheed, M.S. and A.O. Al-Thubaity, 2013. Comparative evaluation of text classification techniques using a large diverse Arabic dataset. Language Resour. Eval., 47: 513-538.

Kobyz, G.V. and A.V. Zamyatin, 2015. Conditional probability density estimation using artificial neural network. Proceedings of the 2015 9th International Conference on Application of Information and Communication Technologies (AICT), October 14-16, 2015, IEEE, Rostov on Don, Russia, ISBN:978-1-4673-6855-1, pp: 441-445.

Larkey, L.S., L. Ballesteros and M.E. Connell, 2002. Improving stemming for Arabic information retrieval: Light stemming and co-occurrence analysis. Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 11-15, 2002, ACM, New York, USA., ISBN:1-58113-561-0, pp: 275-282.

Lin, Y., H. Yu, F. Wan and T. Xu, 2017. Research on classification of Chinese text data based on SVM. IOP. Conf. Ser. Mater. Sci. Eng., 231: 1-5.

Mechti, S., A. Abbassi, L.H. Belguith and R. Faiz, 2016. An empirical method using features combination for Arabic native language identification. Proceedings of the 2016 IEEE/ACS 13th International Conference on Computer Systems and Applications (AICCSA), November 29-December 2, 2016, IEEE, Agadir, Morocco, ISBN: 978-1-5090-4320-0, pp: 1-5.

Mendez, J.R., E.L. Iglesias, F. Fdez-Riverola, F. Diaz and J.M. Corchado, 2005. Tokenising, stemming and stopword removal on anti-spam filtering domain. Proceedings of the 11th Spanish Association for Artificial Intelligence Conference on Current Topics in Artificial Intelligence, November 16-18, 2005, Springer, Santiago de Compostela, Spain, ISBN: 978-3-540-45914-9, pp: 449-458.

Mesleh, A.M.A., 2007. Chi square feature extraction based svms Arabic language text categorization system. J. Comput. Sci., 3: 430-435.

Mesleh, M.A., 2008. Support Vector Machines Based Arabic Language Text Classification System: Feature Selection Comparative Study. In: Advances in Computer and Information Sciences and Engineering, Sobh, T. (Ed.). Springer, Dordrecht, Netherlands, ISBN:978-1-4020-8740-0, pp: 11-16.

Pak, M.Y. and S. Gunal, 2017. The impact of text representation and preprocessing on author identification. Anadolu Univ. J. Technol. A Appl. Sci. Eng., 18: 218-224.

Saad, M.K., 2010. The impact of text preprocessing and term weighting on Arabic text classification. M.Sc. Thesis, The Islamic University, Gaza, Egypt.

Sallam, R.M., H. Mousa and M. Hussein, 2016. Improving Arabic text categorization using normalization and stemming techniques. Intl. J. Comput. Appl., 135: 38-43.

Scott, S. and S. Matwin, 1999. Feature engineering for text classification. Proceedings of the 16th International Conference on Machine Learning (ICML), June 27-30, 1999, ACM, San Francisco, California, USA., pp: 379-388.

Sharef, B.T., N. Omar and Z.T. Sharef, 2014. An automated Arabic text categorization based on the frequency ratio accumulation. Int. Arab J. Inform. Technol., 11: 213-221.

Stamate, D., W. Alghamdi, D. Stahl, D. Logofatu and A. Zamyatin, 2018. PIDT: A novel decision tree algorithm based on parameterised impurities and statistical pruning approaches. Proceedings of the IFIP International Conference on Artificial Intelligence Applications and Innovations, May 25-27, 2018, Springer, Cham, Switzerland, ISBN:978-3-319-92006-1, pp: 273-284.

Thabtah, F., O. Gharaibeh and R. Al-Zubaidy, 2012. Arabic text mining using rule based classification. J. Inform. Knowl. Manage., Vol. 11. 10.1142/S0219649212500062

Uysal, A.K. and S. Gunal, 2014. The impact of preprocessing on text classification. Inf. Process. Manage., 50: 104-112.

Wahbeh, A.H. and M. Al-Kabi, 2012. Comparative assessment of the performance of three WEKA text classifiers applied to arabic text. Abhath Al-Yarmouk: Basic Sci. Eng., 21: 15-28.

Yepes, A.J.J., L. Plaza, J. Carrillo-de-Albornoz, J.G. Mork and A.R. Aronson, 2015. Feature engineering for MEDLINE citation categorization with MeSH. BMC. Bioinf., 16: 113-124.

Zrigui, M., R. Ayadi, M. Mars and M. Maraoui, 2012. Arabic text classification framework based on latent dirichlet allocation. J. Comput. Inform. Technol., 20: 125-140.