DACS Dewey index-based Arabic Document Categorization System

A. F. Alajmi
Communication & Electronics Dept., Faculty of Engineering, Helwan University, Egypt

E. M Saad
Communication & Electronics Dept., Faculty of Engineering, Helwan University, Egypt

M H Awadalla
Electrical and computer engineering Department. SQU, Oman

ABSTRACT
This paper is devoted to the development of Arabic Text Categorization System. First, a stop-words list is generated using statistical approach which captures the inflation of different Arabic words. Second, a feature representation model based on Hidden Markov Model is developed to extract roots and morphological weights. Third, a semantic synonyms merge technique is presented for feature reduction. Finally a Dewey-Index Based Back-propagation Artificial Neural Network is developed for Arabic Document Categorization. The system was compared with other classifiers and the results reveal a promising architecture.

Keywords
Arabic Text Processing**,** Natural Language Processing, Classification, Feature Reduction, Feature representation, Morphological Analyzer

1. INTRODUCTION
Over the last decades, the volume of textual information in electronic format has increased enormously given the emerging of many new sources of information: WWW, emails, newsgroup messages, Internet news feeds, digital libraries, etc... In a so amount of available electronic text documents, the users started to feel the need of automatic system to profitably search and manage these huge repositories of information. Furthermore, the increase of the sources and of the production of textual information, new problems have arisen. The millions of pages available on the WWW, the hundreds of emails or updated news arriving daily at each user and all the other textual resources on Internet had to be categorized and easily organized in a way to allow a simple search and navigation. Thus the scientific community devoted many efforts in developing automatic ways to analyze and process all this information to help the user. Analysis of patterns in data is not new. The concept of average and grouping can be track back to the sixth century B.C. in china following the invention of the abacus. Furthermore, ancient china and Greece, statistics were gathered to help heads of states govern their countries [1]. Differentmethods were used for data analysis which varies such as artificial intelligence based methods, traditional statistics, and Machine Learning (ML).

Data mining is relatively new field which was developed during the 1990's. Data mining is the non-trivial extraction of implicit, previously unknown, and potentially useful information from data. It represents a group of several fields; traditional statistical analysis, artificial intelligence, machine learning, development of large database [1]. Whereas Knowledge discovery KDD is the process of data access, data exploration, data preparation, modeling, model deployment, and model monitoring. Data mining is a part of KDD. KDD process combines the mathematics used to discover interesting patterns in data with the entire process of extracting data and using resulting models to apply to other data sets to take advantage from the information for some purpose.

Data mining deals with structured data stored in databases and organized into records and fields. Despite the fact that large amount of structured data stored in databases; the vast majority of data are stored in documents that are virtually unstructured.

Text mining is the process of discovering new, previously unknown, potentially useful information from variety of unstructured data sources including documents, web pages, xml files, business reports, legal memorandum, email, research papers, manuscript, article, press release, news story, etc. It is the process of deriving novel information from a collection of texts, called corpus.Text mining deals with traditional data mining due to the patterns being extracted from natural language text rather than structured databases. Databases are designed for programs to process automatically, but text is written for people to read not for a program to read or understand [1].Texts mining on the basic level; numericalize an unstructured text document and then, using the data mining techniques, extracts patters from them [2].

Automatic Text Processing, a very wide area which includes many important disciplines. The problems deriving from need of managing large amount of electronic textual information, which is available in computers, have been studied since many years ago and in many areas the conclusions are rather definitive. There is no single data mining approach, but rather a collection of powerful techniques that can be used stand alone or in combination with each other. Automated Text Processing was deeply investigated and various different tasks were identified like, Automated Text Categorization, Automated question answering, Document Clustering, Automated Text Summarization, and Information Retrieval (IR).

Categorization is the task of assigning a text (document) to one or more predefined classes (categories). Starts with training set of objects each labeled with one or more classes which was coded via a data representation model. Each object in the training set is represented in the form ($\vec{x}$, c), where $\vec{X} \in R^n$a vector of measurements and c is the class label. In text categorization vector space model is frequently used as a data representation. Where each document is represented as a vector of (possibly weighted) word counts. Finally define a model class and a training procedure. The model class is parameterized family of classifier and the training procedure

selects on classifier from this family. Many classifiers exists and example of it K-Nearest neighborhood, Decision tree, naïve Bayes, neural networks, etc.

Text mining is a knowledge intensive process in which a user seeks to extract useful information from data sources through the identification and exploration of interesting patterns [2]. However the data sources are document collections and interesting patterns are found not in structured database records but in unstructured text data in the document of those collections. Text processing undergoes different phases for analyzing and understanding data to extract useful information (knowledge). Those phases are preprocessing, feature extraction, feature reduction, text processing tasks.

Preprocessing; Preprocessing are the processes of preparing data for the core text mining task. These processes convert the documents from original data source into a format which suitable for applying various types of feature extraction methods against these documents to create a new collection of documents fully represented by concepts [2]. It includes all those routines, processes and methods required to prepare data for a text mining system which is the core of knowledge discovery operations.Text mining preprocessing operations are centered on the identification and extraction of representative features for natural language documents. The preprocessing operations are responsible for transforming unstructured data stored in document collection into a more explicitly; structured intermediate format. It includes a variety of different types of techniques adapted from information retrieval, information extraction, and computational linguistics research that transform raw, unstructured, original-format content into structured data format. There are two goals of preprocessing phase. One is to identify features in a way that is most computationally efficient and practical for pattern discovery. Second is to capture the meaning of a document accurately; on the semantic level [2].

In order to transform from irregular form in to structured representation; features must be identified. There are a vast number of words, phrases, sentences, typographical elements and layout artifacts that a short document may have. Furthermore, it is necessary to filter out noise from important text; noise is an extraneous text that is not relevant to the task at hand [3]. An example of noise is stop-words. Another important technique in preprocessing phase is splitting an input into individual tokens (as well as other units, such as punctuation marks). This is known as ***Tokenization***. Tokens are the features which represent a document; the representational model. Examples of such representations are characters, words, terms, and concepts. Documents are represented then by feature vectors. A feature is simply an entity without internal structure. A document is represented as a vector in this space –sequence of features and their weights. An example of most common model is ***bag of words*** which uses all words in a document as the features, thus dimension is equal to the number of different words in all of the documents. There are different methods of giving weights to the features. One example is TF-IDF that is the most common used as a weighting

Feature Selection & Reduction; An essential task is identification of a simplified subset of document features that can be used to represent a particular document as a whole. This process includes identifying which features to keep, and weighting how relevant those feature to the document. So the goal is to select relevant features to the categorization task and drop irrelevant with no harm to the classifier performance. After developing efficient representational model, each document in a collection is made up of a large number of features. High feature dimensionality is one of the characteristic of text mining. Another characteristic is feature sparsity that is only small percentage of all possible features for a document collection as a whole appears in any single document. One of the most common techniques for decreasing the dimensionality of the features space are latent semantic indexing (LSI). Furthermore, the classifier performance is affected by the noise introduced by the irrelevant features. Many systems perform aggressive filtering to remove 90->99% of all features and experiments show that using top 10% of most frequent words does not reduce the performance of the classifier [2]. Meanwhile terms with low to medium document frequency are the most informative. The filter a measure of a relevance of each feature needs to be defined. Simplest are DocFreq(w) and other are IG information gain, and X^2 statistics.

***Text processing tasks*:** The main goal of this research is to develop a categorization system that start with row text data and ends up with a tagged document with relevant classes. Figure 1.1 depicts the proposed system that comprised three stages. The preprocessing stage, feature reduction phase, and the categorization stage which are the core mining operation.

This paper is organized as follows; section 2 presents a review on different preprocessing techniques; followed by suggested approach in section 3.Experimental results are presented in Section4. We conclude our work in section 5.

2. PREVIOUS WORK

A review of related research is presented as follows; Text classification is an important task of text processing. A typical text classification process consists of the following steps: preprocessing, indexing, dimensionality reduction, and classification [4]. A number of statistical classification and machine learning techniques has been applied to text classification, including regression models like linear least square fit mapping LLSF [45], K-Nearest Neighbor classifiers [45-48-50], Decision Tress, Bayesian classifiers, Support Vector Machines [46-61-42-50-51-52-63], Neural Networks [14] and AdaBoost [52]. SVM has been applied to text classification [51] and achieved remarkable success [5] proposed a hybrid method used transductive support vector machine (TSVM) and simulated annealing (SA), which selected top 2 thousands high CHI-square value features to form dataset and gained better classification results compared to standard SVM and TSVM. F. Thabtah et al. [6] implemented an Arabic categorization system using Naïve Bayesian classifier based on chi-square feature weighting to classify single label data sets. Al-Shalabi and Obeidat[7] developed K nearest neighborhood is used to classify arabic documents. they extract feature as unigram and bigrams keywords, then TFIDF is applied as a feature selection method. Mesleh and Kanaan [67] applied ant colony optimization ACO as a feature reduction mechanism with chi-square statistics as a score function, then classifies Arabic documents with support vector machine (SVM) classifier. Alsaleem[8] investigated naïve Bayesian (NB) and support vector machine (SVM) on different Arabic data sets, the result shows SVM outperforms NB.Kanaan et al. [9] classified Arabic documents with Expectation maximization (EM) algorithm. TFIDF is applied as a feature weighting method were the naïve Bayesian is used to calculate initial document labels then final classifier is built using the EM algorithm.El-Kourdi et al. [10] presented a naïve Bayesian classifier for Arabic documents with accuracy 92%.Al-Shalabi et al. [11]

applied K-nearest neighborhood algorithm, and extract keywords based on the document frequency TFIDF method with a micro-average precision 0.95.Noaman et al. [12] extracted roots as features for the classifier, then classify Arabic documents with Naïve Bayesian (NB) classifier that results an average accuracy of 62%.Al-Harbi et al. [13] conducted a test on two classifier support vector machine (SVM) and C5.0 document classifiers on 7 different Arabic corpuses. Features are weighted with chi-square. SVM performance is 86% and C5.0 performance 92%.Harrag et al. [14] conducted a comparative study between three text reprocessing techniques light stemming, root-based stemming and, dictionary lookup stemming to reduce feature space. Two classifiers were tested artificial neural network (ANN) and support vector machine (SVM). SVM results on a higher performance for ANN with the light stemmer.Zubi[15] employs a comparison between two classifiers KNN and NB for 1562 documents classified into 6 categories and, weighted using TFIDF method. The experiment showed that KNN performs better.Duwairi[16] implemented three classifiers for Arabic document classification using stems as features. Naïve Bayesian (NB), K nearest neiborhood (KNN) and, distance-based classifier. Their result shows NB outperforms the other classifiers.El-Khoribi and Ismael [17] applied stemming as feature representation method. The features then presented as vectors that consists of a number of components equal to number of topics that probability of stem happens in a topic. Then a stem lookup table is constructed from a stem and label of the class in which it belongs to. A HMM is used to evaluate if a new document belong to a topic. Bawaneh et al. [18] compared between to classifiers KNN and, NB. lighter stemmer was used as a feature and, TFIDF as feature weighting. The KNN classifier performs better.Mesleh[19] investigated six feature selection techniques with SVM classifiers. Their experiments show that Chi-square statistics outperforms better.Gharib et al. [20] applied a four classifier to Arabic documents. Support vector machine (SVM), naïve bayesian (NB), K nearest neighborhood (KNN) and, Rocchio classifiers using stemming as a feature extraction and TFIDF as a weighting schema. Rocchio classifier performs better when features space is small but SVM outperforms for higher space.Raheel et al. [21] combined Boosting with Decision tree as classifier. It uses stemming as a feature extraction and TFIDF as a weighting schema. A comparison was conducted of the method with two classifiers Naïve Bayesian (NB) and, SVM. The result shows SVM and NB outperforms the presented approach.Mohamed and Watada[22] used (LSA) latent semantic analysis to evaluate each term in a document, then it uses Evidential reasoning (ER) to assign the category to new document according to the data set. The experiment on ER classifier with LSA and ER classifier with TFIDF shows that ER-LSA performs better.Al-Shargabi et al. [23] conducted a comparative study between SVM, Naïve Bayesian and, sequential minimal optimization (SMO). The results show SMO have higher accuracy.Musa et al. [24] presented a comparison between NB and, SVM which outperforms the NB.Alwedyan et al. [25] developed a multiclass association rule classifier which performs better than NB and SVM.Khreisat[26] constructed a classifier for Arabic text documents using the N-gram frequency statistics technique employing a dissimilarity measure called the "Manhattan distance", and Dice's measure of similarity. The Dice measure was used for comparison purposes. Results showed that N-gram text classification using the Dice measure outperforms classification using the Manhattan measure.Al-Salemi et al. [27] Investigated Bayesian learning models by implementing three classifiers based on Bayesian theorem, which are Simple Naïve Bayes (NB), Multi-variant Bernoulli Naïve Bayes (MBNB) and Multinomial Naïve Bayes (MNB) models. MBNB classifier outperforms both of NB and MNB classifiers. Raheel et al [28] offered a comparative study between four feature types: words in their original form, lemmas, roots, and character level n-grams and shows how each affects the performance of the classifier. Two classifiers Support Vector Machines and Naïve Bayesian Networks algorithms were used. Support Vector Machines based on 3-grams gave the best classification results with an accuracy exceeding 92% and an F1 measure exceeding 0.92. Al-Kabi and Al- Sinjilawi[29] conducted a comparative study on six distance measures (inner product, cosine, Jaccard, Dice, Naïve Bayesian, and Euclidean) in order to find the optimal one that can be used to classify Arabic text. The results showed that the cosine measure outperformed the other three associative coefficients of the VSM. Finally they compared the efficiencies of the cosine measure, Naïve Bayesian, and Euclidean Measure to classify Arabic text and, results show that Naïve Bayesian slightly outperforms the other methods.Chantar and Corne [89] proposed a feature selection method called Binary Particle Swarm Optimization And KNN (BPSO-KNN) which are used with three machine learning algorithms – SVM , Naïve Bayes and C4.5 decision tree learning to classify Arabic documents. The results achieved by SVM, as well as Naïve Bayes, overall suggest that BPSO/KNN performs well as a feature selection technique for this task.

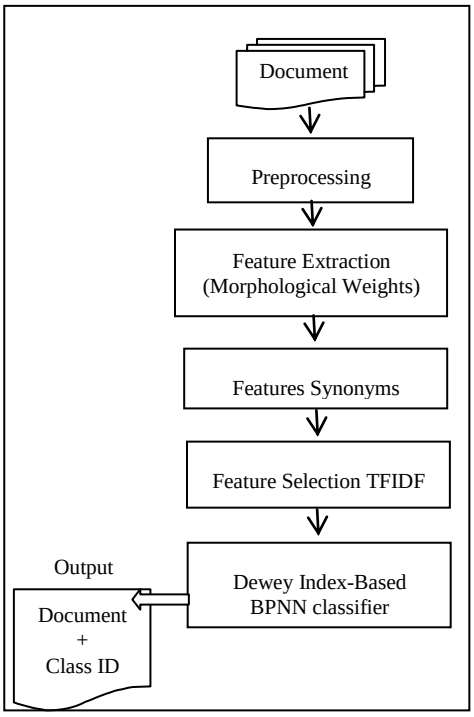

Fig.1 Proposed Architecture

3. CATEGORIZATION SYSTEM

A document categorization system is presented in Fig.1. the system is based on Back-Propagation Neural Network which was trained with Dewey indexes and categories lists. The categorization system is divided into four phases.

Preprocessing: concerns with removing the digits and punctuation marks;the normalization of some of the Arabic letters such as normalization and, stop words (common words) removal.

Feature extraction: convert words to unified form, in our case morphological weights which are then merged.
Feature Reduction: a synonyms merge technique is used to reduce feature space semantically, followed with TFIDF weighting terms.
***Categorization*:** a BPNN was trained with Dewey indexes to assign a document to a labeled category.

3.1 Preprocessing

Text is represented in a compact and applicable form for further processing. The preprocessing includes removing the digits and punctuation marks. The normalization of some of the Arabic letters. Furthermore stop-words are removed, which are common words that do not add any value to the text classification processes. The end result is bag of words "Tokens".

أظهر تأرقامنشر تالجمعةأنمالايقلعن 40
ألفبريطانيفقدوامنازلهمفيالربعالأخيرمنالسنةالماضيةلعجز همعنسدادأقساطق
روضعقارية،فيمؤشر آخر علىالركودالذييعانيمنهالاقتصادالبريطاني.
ويعيدفقدانأعدادكبير تمنالبريطانييننازلهمإلىالأذهانأز مةالر هنالعقاريبالو لاي
اتالمتحدةالتيفجر تالأز مةالمالية.
وقالمجلسالمقرضينالعقاريينإنهذاالرقمهوالأعلىمنذ 1996.
وتوقعمحللونبريطانيونأنيرتفعالرقممعز يادةمستوىركودأسعار المساكنوتفاقم
الركودالذييغرقفيهالاقتصادالمحلي.
وتؤكدأوساطاقتصاديةفيبريطانيأنالبلادلمتشهدمثيلاللركودالحاليمنذ 60
عاما,
فيماذكر تصحفبريطانيةأنأعدادالعاطلينستتجاوز العامالمقبلثلاثةملايين.
ووفقالمجلسالمقرضينالعقاريينقفز عددحالاتالاستيلاء علىالمساكنبسببالعجز
عنسدادالقروضالعقاريةبنسبة 50% فيالأشهر الثلاثةالأخير ةمن 2008
مقار نةمعالفتر ةنفسهامن 2007 ليصلإلى 10400 حالة

Fig. 2 Example of Arabic Text

التي	ارتفع	البالغ	مؤشر	ارتفاعا	استمرار	وباعتبار
الذي	الحد	البلاد	نسبة	الأخيرة	الارتفاع	اقتصادية
إلى	الربع	البيع	نفسها	الأذهان	الإطلاق	الاستيلاء
إن	الرقم	الرهن	وبينت	الجمعة	الحصيلة	الاقتصاد
ألا	السنة	الفترة	وتفاقم	الحالي	العاطلين	البريطاني
ألف	الشهر	المالية	وتوقع	قروض	العقاري	العقاريين
ألفا	العام	المقبل	وتؤكد	للأرقام	العقارية	القروض
أن	العجز	المقبلة	ورغم	للركود	القياسية	بالتفصيل
أنه	القرن	أوائل	وضع	للمكتب	الماضي	بالمفاجئ
أنهم	اليوم	بأكمله	وعزا	لمجلس	الماضية	بالولايات
حين	أثناء	رغم	ووفقا	متوقعا	المرتفعة	الاقتصادي
على	أرقام	ركود	ويعد	مجلس	المساكن	البريطانيين
عن	أزمة	زال	ويعيد	محللون	المستوى	التخفيضات
عنه	أزمت	زيادة	يبعد	مستوى	المسجلة	المستويات
غير	أسعار	سبقه	يرتفع	مقارنة	أظهرت	المقرضين
في	أعداد	سداد	يعاني	المتحدة	بريطاني	إحصاءات
فيما	أعلى	سنوي	يعتقد	المجلس	بريطانيا	بريطانيون
فيه	أقساط	عاما	يغرق	المحلي	قطاعات	للمقرضين
لا	آخر	عدة	يقل	الوطني	لعجزهم	للإحصاء
لم	بسبب	عدد	ارتفعت	إجمالي	مساكنهم	والمسجل
لها	بلغت	فترة	الإنفاق	أوساط	ملحوظا	عمليات
ما	بنسبة	فقدان	الأخير	بالنظر	ووصف	فجرت
مع	تجارة	فقدوا	الأزمة	تراجعا	نشرت	ذكرت
من	تزيد	قفز	الأشهر	حالات	بريطانية	سجلت
منذ	تشهد	كان	الأعلى	حيوية	تسعينيات	مثيلا
منه	تمديد	كبيرة	الثلاثة	ملايين	ستتجاوز	حملت
هذا	ثلاثة	كثيرا	الركود	منازلهم	سيحصل	حدود
هذه	جميع	كشف	السابق	صحف	سيفقدون	حذروا
هو	حالة	ليصل	القائمة	عقارية	للاقتصاد	وفي

Fig 3. Text after Tokenization

3.2 Feature Extraction

Feature extraction is one of the most important phases in text processing systems. A well representative features leads to a more accurate with higher performance system.There are many representation proposed for Arabic text. The most know and used are stemming. Light stemming which is based on stripping prefixes and suffixes from a word, and root-based stemming which are based on extracted the canonical form of the word, the form which the word was derived from.The problem with root-based method is that the semantic is not preserve. That is multiple words with the same root and different meaning are joined under the same root. This will affect the performance of the text processing task in mind.In the other hand, light-stemmer performs better, but having the infix may treat words with the same meaning as different (e.g. singular and plural for of a word)a Hidden Markov Model HMM is presented, which extract the morphological weight of an Arabic word and, then transfer the weight into a unified form which combines weights with the same meaning together.

3.2.1 HMM Extraction Model

Hidden Markov Model is one of the most important machine learning models in speech and language processing [30]. HMM is a probabilistic sequence classifier, given a sequence of units (in our case letters) and its job is to compute the probability distribution over possible labels and choose the best label sequence [30].The Hidden Markov Model is a finite set of states, and a set of transitions between states that are taken based on the input observations. Each of which is associated with a probability distribution [31]. Weights are augmented; where each transition is associated with a probability of how likely state a transit to state b. Transitions among the states are governed by a set of probabilities called transition probabilities. In a particular state an outcome or observation can be generated, according to the associated probability distribution. It is only the outcome, not the state visible to an external observer and therefore states are ``hidden" to the outside; hence the name Hidden Markov Model [31].

Count of Words	Word
7	الركود
4	العقاريين
4	العام
4	الماضي
4	حالة
4	البريطاني
4	المساكن
3	ألف
3	مقارنة
3	الرقم
3	مستوى
3	حالات
3	الاستيلاء
3	المقرضين
3	الارتفاع
2	البريطانيين
2	الاقتصاد
2	بالتفصيل
2	الربع
2	الأشهر

Fig 4.Text after After Stop word removal

Hidden Markov Model is used to extract Arabic word weights. HMM is represented by a set of states and a set of transitions from one state to another. A given word is tested through the model by using states as the letters of the word, and the transition from start state 0 to end state will represent the full word. The model will output the path which yields the highest probability. There are two probability matrices, the state transition probability matrix, and the emission probability matrix. State transition matrix will provide the probability of going from state i to state j. Furthermore, the emission probability matrix will provide the probability of emitting an observation in a given state i, observations are the alphabets of the Arabic language plus a special character called "Shadda" "شدة", a total of 31 observation is considered.

Elements of the proposed Hidden Markov Model are:

1. A set of N states $S = s_1 s_2 ... s_N$ representing the number of states of the model, each state represent one letter of a word, and a path from state s_i to s_j represent a word. N=172 states.
2. A transition probability matrix A. $A = a_{11} a_{12} ... a_{nn}$, where a_{ij} represents the probability of moving from state i to state j, $A = \{a_{ij}\}$. That is going from one letter to the next in a given word.
3. A sequence of K observation $O = o_1 o_2 ... o_k$ each drawn from the vocabulary $V = v_1, v_2, ... v_V$, V represents Arabic letters plus some special letters. The number of observation symbols in the alphabet, M=31.
4. A sequence of emission probabilities $E = e_i(o_k)$, each sequence expresses the probability of an observation o_k being generated from a state i.
5. A special start state and a final (end) state S_0, S_F which are not associated with observations. The proposed model has one null initial state and multiple end states. End state can be the last state of any valid weight states, or a suffix state.

Example, a word "مسلمون" (Muslims) has six letters, adding a null state it should start with state 0 and goes up to six states depending on A (transition probability), and E (emission probability). There could be multiple correct paths for the word, but the only one with the highest probability will be accepted as a valid solution. In the case of our example, a path of states 0, 8,15,16,17,170,171. state 0 as a starting null state, state 8 will represent the letter "" and it is considered as a part of the prefix states group, and it only prefixes a noun, so the word will be identified as a noun. State 15 up to 17 represent the weight "F3L" (فعل=سلم) and it is also the root of the word. States 170, and 171 (ون) are the suffixes of the word, and it is special for plurals. Other words are found in the same way.

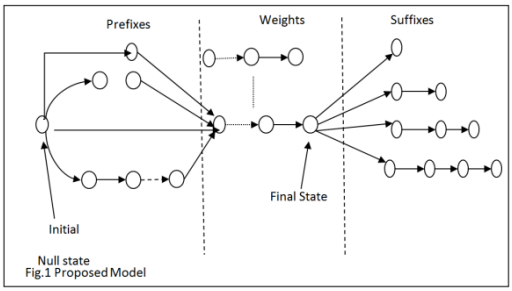

Fig.5 Proposed Model

First, we define the number of states S in the system. A total of 172 states were identified as prefixes, weights, or suffixes. Prefixes are represented by 15 states. States are logically divided into three groups that identify the set of prefixes state group, the set of weights states group, and the set of suffixes states group. Weights are represented by 82 states, and suffixes are represented by 75 states. We start with one initial null state, and multiple end states. An end state is the last state of any valid end state of a weight, or a suffix state. Observations are 2 8Arabic letters added to it shadah (شدّة), Alef maqsora (ى) and Taa (ة), and we distinguish between Alef and Hamza (أand إ). A total of 31 observations is embedded. Figure 1 shows the proposed model design.

A word may or may not have a prefix. Prefixes are of length up to 7, for example the word (وبالاستخدام). The word has a prefix of length 7. A word without prefixes or suffixes could be of size 3-4-5-6-7 with infixes. A word may or may not have a suffix. Suffixes could be of length up to 4. For example, the word (فعاليّات) has a suffix of length four letters.

For example a word (اجتمعنا) will have one prefix, and two suffixes, leaving 4 letters to represent the pattern (افتعل), which is a verb in the past tense. Table.1 shows other examples of word decoding.

The input to the learning algorithm would be unlabeled sequence of observations O and a vocabulary of potential hidden states S which simply means the word, and the correct path of states it should have. Standard algorithms for HMM training are Forward-backward, and Baum Welch algorithm. The Algorithm will train both the transition probabilities "*A*" and the Emission Probability E of the HMM. Generally, the learning problem is how to adjust the HMM parameters, so that the given set of observations (words) is represented by the model in the best way for the weight-root extraction system. The Forward-Backward Algorithm was used to train our system.

3.3 Feature Reduction

The resulting feature space of a text analysis system is usually very large. Feature reduction is an important step to reduce features and, to eliminate noisy components.

3.3.1 Synonym Merge Reduction

A semantic approach is presented to reduce feature space. The main idea behind the approach "synonym merge" is to preserve important terms from being excluded. Term-document matrix is constructed to apply merge over all training set. A dictionary of synonyms [32] is used, were terms with similar meaning are giving one group code. Checking for terms synonym then merging terms with same group id into one feature.

Algorithm (construct synonym tree)
A synonyms tree is constructed from list of synonyms in such a way:
- Null node having 28 branches constituting the Arabic alphabet.
- Each of the 28 node have 28 branches and so on until a word is constructed from root node to the n-1 node (n is the number of letters in a word)
- A leaf node contains the group ID.

Algorithm (synonym check)
Do for all terms
- From root select branch which contains the next word letter.

Continue until last term letter.
- If word does not belong to a synonym group the return null.
- If group found then return group ID.
- Merge all features with the same synonyms group id.

The result of the synonym check algorithm is a semantically reduced feature space. The resulting features are processed with a feature selection method to produce the final feature space that is used for later processing such as classification and clustering.

3.3.2 Term Frequency- Inverse Term Frequency

Term-Frequency-Inverse-Term-Frequency (TFIDF) is formulated as:

$$tf - idf(t,d) = tf(t,d) \times idf(t)$$

Where the term frequency refers to the number of occurrences of term *i* in document and the inverse document frequency is a measure of the general importance of the term (obtained by dividing the total number of documents by the number of documents containing the term and then taking the logarithm of that quotient).
The *inverse document frequency*:

$$idf(t) = log\frac{|D|}{|\{d: t \in d\}|}$$

With,
- $|D|$: the total number of documents in the document set.
- $|d: t \in d|$: number of documents where the term *t* appears.

A high weight in tf–idf is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents; the weights hence tend to filter out common terms. The tf-idf value for a term will be greater than zero if and only if the ratio inside the idf's log function is greater than 1. Depending on whether a 1 is added to the denominator, a term in all documents will have either a zero or negative idf, and if 1 is added to the denominator a term that occurs in all but one document will have an idf equal to zero.

Table.1 Resulting Features

Feature English	TFIDF	Count	Feature Arabic
Britain	0.244	11	بريطان
Recession	0.200	9	ركود
Real estate	0.178	8	عقار
state	0.156	7	حالة
loan	0.156	7	قرض
number	0.133	6	رقم
year	0.133	6	عام
Housing	0.111	5	مساكن
Economy	0.111	5	اقتصاد
council	0.111	5	مجلس

3.4 Dewey index-Based categorization system

The Dewey Decimal Classification (DDC) is a proprietary system of library classification. The DDC attempts to organize all knowledge into 10 main classes [33] (table 1). The ten main classes are further subdivided into ten divisions, and each division into ten sections, giving ten main classes, 100 divisions and 1000 sections. The system is made up of seven tables and ten main classes, each of which is divided into ten secondary classes or subcategories, each of which contain ten subdivisions.Three levels of classes were used in this research and the indexes were filtered to match the feature format which is the morphological weight of the terms.

Table.2: The main 10 classes

Class ID	***Class Name***
000	Computer science, information and general works
100	Philosophy and psychology
200	Religion
300	Social sciences
400	Language
500	Science (including mathematics)
600	Technology and applied Science
700	Arts and recreation
800	Literature
900	History, geography, and biography

Table 3 Shows an example document after classification, the document was classified into three levels; the main class was Economics and the second subclass was "Land economics", and the theird level class was "Real estate"

Table 3: the result of classifying a document

Level 1	Level 2	Level 3
330 Economics	333 Land economics	333.33 Real estate

4. EXPERIMENT AND RESULTS

The HMM model was trained with 15 million Arabic words. Those words constitute all possible different forms that a

word could have. Words were generated by the aid of Arabic dictionary [34,35]. Based on word root, and possible weights for those roots, different forms of a word were generated. The generated words were attached to different prefixes and suffixes following Arabic morphological rules. Following are the procedure to produce the Hidden Markov Model parameters:

1- Collect words' roots and patterns for those roots from Arabic dictionaries.
2- Generate different forms of a word using morphological rules.
3- Add suffixes and prefixes to resulting words.
4- Use the final result to train the model using forward-Backward algorithm.

The result is two matrices, one for state transition probability, and the other for observation emission probability. State transition matrix will provide the probability of going from state s_i to state s_j. Emission probability matrix will provide the probability of emitting a letter E in state s_i. These matrices are used as inputs for the Viterbi algorithm to decode a given word. The algorithm was altered to give all possible paths, and not only the one with the highest probability. In order to extract the correct path further rules have to be applied, which are:

1- End states must not be before the last state of any valid weight (pattern).
2- Prefix and suffix matching table must be applied. For example prefix "ي" does not match suffix "ت".
3- Check the Bi-Gram generated matrices probability if the first and the last letters of the word are probable prefix and suffix.

The words were decoded using Veterbi Algorithm. Those words (Training Set) were extracted from different documents. Following are the processing procedure of a text in order to extract weights and roots:

1. Tokenizing words and eliminating all punctuation.
2. Hamza must all be normalized to one shape "أ".
3. Altered veterbi algorithm is used to decode the words, and find all possible paths.
4. Apply the weights correctness rules, and prefix-suffix matching table.
5. Select the path with highest probability.
6. States which belongs to the weights' states are identified, thus extract the root.

Table 4 shows an example of the text decoding.

Input string	تتناول	يشعرون	ازداد
State Transition	0-3-37-38-39-40-41	0-5-7-8-9-102-103	0-2-25-26-27-28
Probability of Emission	0.000616	0.000104	0.000001
Probability of State transition	0.006813	0.003	0.011867
Expected	تتفاعل	يفعلون	افتعل

An in house Arabic documents data set was used in this experiment. The documents were tokenized to produce bag of words by eliminating unwanted characters. Then stop words were removed from the resulting documents. The resulting features was processed using the suggested HMM to extract words morphological weights. Then synonyms merge technique was used to reduce number of features and to preserve the text semantic. The presented categorization algorithm based on training a BPNN with the dewy-indexes as inputs and dewy-classes as an output. A comparison between suggested BPNN Dewy-index classifier and k-nearest neighbor classifier yield a higher performance. Using f-measure as a performance measure the k-nearest neighbor classifier has a 0.89 F1-measure and our presented algorithm scored 0.95 F1-measure. Having a well-defined list of indexes which represent the classes improves the identification of the proper class for a given document. Furthermore, the suggested approach has the flexibility of single and multi-labeling. Originally multiple labels will be produced, but the class with the highest probability (features) will be assigned.

5. CONCLUSION

Arabic is a highly inflected language. The wide range of word forms and the large variety of prefixes and suffixes complicate the extraction of precise features for a text processing system. Therefore, a preprocessing technique is needed to unify similar words in to a single feature before further processing the text. Arabic words are structured in well-known patterns called morphological weights thus "weights" are used as a feature representation model for an Arabic text. Weights are closer to stems, except some of the prefixes that belong to the weight will not be removed. The presented approach is based on Hidden Markov Model. Each state in a model is considered as a letter of a word, a word is represented by consecutive states, from start to end.

The resulting features are usually high therefore a semantic method based on grouping weights with similar meaning and different states (single, plural, past, present) into unified ones.

a Dewey-index based classification system is presented which reveals a promising results.

6. REFERENCES

[1] R. Nisbet, J. elder, G. Miner, "Handbook of statistical analysis and data mining applications", academic Press, Elsevier, 2009.

[2] R. Feldman, and J. Sanger, "The text mining handbook", Cambridge university press, 2007.

[3] G. Salton, and M. Mcgill, "An Introduction To Modem Information Retrieval", Mcgraw-Hill, 1983.

[4] G. Wei, X. Gao, and S. Wu, "Study of Text Classification Methods for Data Sets With Huge Features", 2nd International Conference on Industrial and Information Systems, 2010.

[5] M. Shafiei, S. Wang, R. Zhang, E. Milios, B. Tang, J. Tougas, and R. Spiteri, "Document Representation And Dimension Reduction For Text Clustering", IEEE, 2007.

[6] F. Thabtah, M. Eljinini, M. Zamzeer, and W. Hadi, "Naïve Bayesian Based on Chi Square to Categorize Arabic Data", Communications of the IBIMA, Volume 10, 2009.

[7] R. Al-Shalabi, and R. Obeidat, "Improving KNN Arabic Text Classification with N-Grams Based Document Indexing", INFOS2008, March 27-29, 2008 Cairo-Egypt.

[8] S. Alsaleem, "Automated Arabic Text Categorization Using SVM and NB", International Arab Journal of e-Technology, Vol. 2, No. 2, June 2011.

[9] G. Kanaan, M. Yaseen, R. Al-Shalabi, B. Al-Sarayreh, and A. Mustafa, "Using EM for Text Classification on Arabic", 2nd International conference on Arabic language resources & tools, April, , Cairo, 2009

[10] M. El-Kourdi, A. Bensaid, and T. Rachidi, "Automatic Arabic Document Categorization Based on the Naïve

Bayes Algorithm", Informatics and Systems (INFOS), 2010 The 7th International Conference on, Cairo

[11] R. Al-Shalabi, G. Kanaan, and M. Gharaibeh, "Arabic Text Categorization Using kNN Algorithm", 6th International Conference on Advanced Information Management and Service (IMS), 2010, Seoul.

[12] H. Noaman, S. Elmougy, A. Ghoneim, and T. Hamza, "Naive Bayes Classifier Based Arabic Document Categorization", The 7th International Conference on Informatics and Systems (INFOS), 2010.

[13] S. Al-Harbi, A. Almuhareb, A. Al-Thubaity, M. S. Khorsheed, A. Al-Rajeh, "Automatic Arabic Text Classification", 9es Journées internationales d'Analyse statistique des Données Textuelles, JADT 2008.

[14] F. Harrag, E. El-Qawasmah, and A. Al-Salman, "Stemming as a Feature Reduction Technique for Arabic Text Categorization", 10th International Symposium on Programming and Systems (ISPS), 2011.

[15] Z. S. Zubi, "Using Some Web Content Mining Techniques for Arabic Text Classification", RECENT ADVANCES on DATA NETWORKS, COMMUNICATIONS, COMPUTERS, 2009.

[16] R. Duwairi, "Arabic Text Categorization", the international Arab Journal of information Technology, vol. 4, No. 2, April 2007

[17] R. A. El-Khoribi and M. A. Ismael, "An Intelligent System Based on Statistical Learning for Searching in Arabic", AIML Journal, Volume (6), Issue (3), September, 2006

[18] M. J. Bawaneh, M. S. Alkoffash and A. I. Al Rabea, "Arabic Text Classification using K-NN and Naive Bayes", Journal of Computer Science 4 (7): 600-605, 2008.

[19] A. Mesleh, "Support Vector Machines based Arabic Language Text Classification System: Feature Selection Comparative Study", 12th WSEAS Int. Conf. on Applied Mathematics, Cairo, Egypt, December 29-31, 2007.

[20] Tarek F. Gharib, Mena B. Habib, and Zaki T. Fayed, "Arabic Text Classification Using Support Vector Machines", The International Journal of Computers and Their Applications ISCA, vol.16, no.4, pp. 192-199, Dec 2009

[21] Saeed Raheel, Joseph Dichy, Mohamed Hassoun, "The Automatic Categorization of Arabic Documents by Boosting Decision Trees", Fifth International Conference on Signal Image Technology and Internet Based Systems,2009.

[22] R. Mohamed, J. Watada, "An Evidential Reasoning Based LSA Approach to Document Classification for Knowledge Acquisition", IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), 2010.

[23] B. Al-Shargabi, W. AL-Romimah, and F. Olayah, "A Comparative Study for Arabic Text Classification Algorithms Based on Stop Words Elimination" , ISWSA '11 Proceedings of the International Conference on Intelligent Semantic Web-Services and Applications, 2011.

[24] W. Musa H. Salam, J.. Al-Widian, "Performance of NB and SVM Classifiers in Islamic Arabic Data" , ISWSA '10 Proceedings of the 1st International Conference on Intelligent Semantic Web-Services and Applications.

[25] J. Alwedyan, W. Hadi, M. Salam, H. Y.Mansour, "Categorize Arabic Data Sets Using Multi-Class Classification Based on Association Rule Approach", ISWSA '11 Proceedings of the International Conference on Intelligent Semantic Web-Services and Applications, 2011.

[26] L. Khreisat, "Arabic Text Classification Using N-Gram Frequency Statistics a Comparative Study", Conference on Data Mining, 2006.

[27] B. Al-Salemi and M. J. Ab-Aziz, "Statistical Bayesian Learning for Automatic Arabic Text Categorization", Journal of Computer Science 7 (1): 39-45, 2011.

[28] S. Raheel and J. Dichy, "An Empirical Study on the Feature's Type Effect on the Automatic Classification of Arabic Documents", CICLing 2010, LNCS 6008, pp. 673–686, 2010. Springer-Verlag Berlin Heidelberg 2010.

[29] M. N. Al-Kabi, S. I. Al- Sinjilawi, A Comparative Study of The Efficiency of Different Measures To Classify Arabic Text", University of Sharjah Journal of Pure & Applied Sciences Volume 4, No. 2, 2007.

[30] L. Hao, and L. Hao, "Automatic Identification of StopWords in Chinese Text Classification", International Conference on Computer Science and Software Engineering,2008.

[31] R.B. Myerson, "Fundamentals of social choice theory", Discussion Paper No.1162, 1996.

[32] L. S. Larkey, L. Ballesteros, and M. E. Connel, Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis, in Proc. of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 275 – 282, 2002.

[33] G. Zheng, and G. gaowa, "The Selection of Mongolian Stop Words", IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS), 2010.

[34] F. Zou, F. L. Wang, X. Deng, S. Han, and L. S. Wang, "Automatic Construction of Chinese Stop Word List", Proceedings of the 5th WSEAS International Conference on Applied Computer Science, Hangzhou, China, April 16-18, 2006 (pp1010-1015).

[35] S. Khoja, and R. Garside. "Stemming Arabic text", Computer Science Department, Lancaster University, Lancaster, UK, 1999.