Received 28 January 2016
Revised 3 April 2016
Accepted 4 April 2016

A lexicon based approach for classifying Arabic multi-labeled text

Ismail Hmeidi, Mahmoud Al-Ayyoub, Nizar A. Mahyoub and
Mohammed A. Shehab
Jordan University of Science and Technology, Irbid, Jordan

Abstract

Purpose – Multi-label Text Classification (MTC) is one of the most recent research trends in data mining and information retrieval domains because of many reasons such as the rapid growth of online data and the increasing tendency of internet users to be more comfortable with assigning multiple labels/tags to describe documents, emails, posts, etc. The dimensionality of labels makes MTC more difficult and challenging compared with traditional single-labeled text classification (TC). Because it is a natural extension of TC, several ways are proposed to benefit from the rich literature of TC through what is called problem transformation (PT) methods. Basically, PT methods transform the multi-label data into a single-label one that is suitable for traditional single-label classification algorithms. Another approach is to design novel classification algorithms customized for MTC. Over the past decade, several works have appeared on both approaches focusing mainly on the English language. This work aims to present an elaborate study of MTC of Arabic articles.

Design/methodology/approach – This paper presents a novel lexicon-based method for MTC, where the keywords that are most associated with each label are extracted from the training data along with a threshold that can later be used to determine whether each test document belongs to a certain label.

Findings – The experiments show that the presented approach outperforms the currently available approaches. Specifically, the results of our experiments show that the best accuracy obtained from existing approaches is only 18 per cent, whereas the accuracy of the presented lexicon-based approach can reach an accuracy level of 31 per cent.

Originality/value – Although there exist some tools that can be customized to address the MTC problem for Arabic text, their accuracies are very low when applied to Arabic articles. This paper presents a novel method for MTC. The experiments show that the presented approach outperforms the currently available approaches.

Keywords Label-set dimensionality, Lexicon-based multi-label classification, ML-Accuracy, Multi-label data, Single-label data

Paper type Research paper

1. Introduction

Text classification (TC) is a branch of data mining and machine learning, which extracts useful information then uses it to build models that define this information for analyzing data behavior (Hotho *et al.*, 2005; Mendoza, 2012). The need for an automatic procedure to handle this job motivated the development of automatic tools to classify and organize documents written in all natural languages. Arabic TC is especially difficult because Arabic morphology is so complex. Also, the custom of omitting diacritics in texts written for adults produces many ambiguities in our database (Al-Harbi *et al.*, 2008).

International Journal of Web
Information Systems
Vol. 12 No. 4, 2016
pp. 504-532
© Emerald Group Publishing Limited
1744-0084
DOI 10.1108/IJWIS-01-2016-0002

There are several machine-learning techniques which can ease the process of classifying a text. Corpus-based TC is considered as the method of extracting information found in a training set, and then categorizing documents of a testing set, which is used to evaluate the performance of a particular attempt at classification (Hotho *et al.*, 2005). Another appealing method is unsupervised TC, which classifies documents according to some inherent characteristics or features that those documents may share (Aggarwal and Zhai, 2012). Unfortunately, till now, this approach produced lower accuracy in comparison with the supervised method according to recent research, even when it makes use of pre-processing techniques such as stemming, feature selection, feature extraction and many other ways to improve the work of such unsupervised systems (Joorabchi and Mahdi, 2010).

A lexicon-based TC can help match the vocabularies associated with each label stored in the lexicon with text vectors found in documents and classifying them accordingly. The way in which that lexicon is built can be a valuable factor in boosting the accuracy of the unsupervised classification, especially when it is automated (Kim *et al.*, 2014).

TC can be tweaked to behave according to a particular goal. For example, to assign each training document to exactly one class (this is called a single-label classification). Another possibility is assigning more than one class to each document according to the diverse categories that its content may share (this is called a multi-label classification) (Duwairi and Al-Zubaidi, 2011). Most of the work on TC in the past was done using a single-label classification, yet with the expectation of working with a multi-label approach in the future. But, it will be a great challenge to define the multiple categories that documents may contain (Ezzat *et al.*, 2012).

There are several complications that make working with multi-label data difficult. Such difficulties arise because of the huge dimensionality of label sets used to annotate the data. The method of evaluation used to estimate the accuracy of classification in the multi-label environment may differ. Thus, a few methods were presented to transform the multi-label data into a form appropriate to work with like problem transformation (PT) methods that convert the multi-label data into a single labelled one (Ezzat, 2010; Ezzat *et al.*, 2010; Yamamoto and Satoh, 2014).

Mainly, three different methods in the literature are presented for categorizing multi-label data sets. The following section is a brief discussion of the data sets.

Most of the work in the literature shows the corpus-based techniques to classify multi-label data, although these methods were used when classifying single-label data as well (Ezzat, 2010). In this study, it is worthy to use the corpus-based method as a baseline to compare the results of our work with. This technique transforms the ordinary text (training data) into a bag-of-words (BoW) vectors, and then it extracts features that can help build a model and classify unseen data (testing data) (Xu *et al.*, 2012). Besides, the same classifiers which are used in the traditional corpus-based single-label classification also can be conducted in multi-label classification process, such as SVM, NB, KNN and DT etc., but by taking into consideration that some pre-processing techniques must be applied on the multi-label data to make it appropriate for classification. PT methods are one example of pre-processing techniques that aim to unify the multi-label structure of the data set into a single-label form capable of being classified.

Unfortunately, a small number of studies showed classifying multi-label data using unsupervised and semi-supervised methods (Ezzat *et al.*, 2012; Xu *et al.*, 2012). Lexicon-based method is considered one of the methods that counts on building lexicons (domain-based lexicons) depending on extracting useful features (lexical terms) introduced in this study and considered the first one in the multi-label environment, although lexicon-based techniques are introduced in literature but in media data not yet in text applications (Xu *et al.*, 2012).

Many applications currently are derived from TC. These applications use the benefits of TC and machine learning classification to help them work appropriately and efficiently. Spam filtering, e-mail routing, language and authorship identification (Efstathios, 2009; Cheng *et al.*, 2011), genre classification, sentiment analysis (Al-Kabi *et al.*, 2013; Abdulla *et al.*, 2013; Al Shboul *et al.*, 2015), bioinformatics, scene classification and article triage are examples of applications extended from TC (Ezzat, 2010; Zhang *et al.*, 2007). Some multi-label applications in real life would be very helpful in representing the hidden information in the data. Multi-label sentiment analysis is an example of such applications that signifies sentiment terms in more than the two famous polarities, positive or negative labels, and can identify other extra labels such as related domains to the former polarities, or even explore if the sentiment text is in a dialect or in Modern Standard Arabic (Abbasi *et al.*, 2008). Another real-life application of multi-label classification is the process of classifying media data (video, images or audio data) according to annotating these data into multiple classes such as in Xu *et al.* (2012) and Feng (2010). Moreover, it is interesting when the articles in news Web pages are organized in a multi-class way so that people can find their interests more easily and accurately; hence, this property adds more to the multi-label history. As a result, a multi-label concept can be a fruitful environment for expressing information and data in more detail that can be helpful in our real life.

As the number of Arabic speakers using the internet has surpassed 150 million, there is an increasing call for advanced technology such as machine-learning and text-mining applications, in addition to other natural language processing techniques capable of handling Arabic text efficiently and effectively (Saad, 2010; Document classification, 2014). The methods of natural language processing developed for English and Chinese, languages with the largest number of internet users, do not work well for the Arabic language, which belongs to a totally different language family. The complexity of Arabic morphology combined with the custom of omitting all short vowels from newspapers and other text written for adults produce a source of much ambiguity[1].

The rest of the paper is organized as follows: Section 2 describes all the relevant works in the multi-label field. Section 3 presents the different approaches that deal with multi-label data (the corpus- and lexicon-based methods). Section 4 describes the detailed methodology of classifying multi-label articles using a lexicon approach. Then, in Sections 5 and 6, all the results from classifying multi-label data using the lexicon-based tool are presented. In Sections 7 and 8, all the explanation and assessment of the results are taken into account. In Section 9, we describe an enhancement technique for improving the accuracy of classification. Finally, in Section 10, the conclusion of the entire study is represented and all the plans to continue this work in the future.

2. Literature review

2.1 Multi-label classification

A small number of studies dealt with the multi-level data set as a new and different path of the typical single-based classification. There are research projects in that field conducted with an English data set by Duwairi and Al-Zubaidi (2011) who presented a hierarchical classification method based on modifying the original KNN classifier and making use of a label or class representative. Each category is represented by one unique document from the overall training set. This document is chosen by making the entire training set elect the most representative and the closest document, and then select the important features of that document. The classification process takes place when the new document is compared with the class representative not the entire set of training documents, taking in consideration that the data are represented in a hierarchical manner. Duwairi and Al-Zubaidi (2011) also presented a new similarity technique that depends on calculating the term frequency of a particular document and matching it with the class in hand. This similarity measurement outperformed the state-of-the-art similarity methods such as Jacard, Cosine and Dice, giving the highest classification accuracy.

Another study is presented by Al-wedyan *et al.* (2011) by including association rules in the process of building the classifier. This technique is considered as the first in Arabic studies that increased the classification accuracy when it is used with the traditional classifiers such as decision trees then compared it with the famous Naïve Bayes and support vector machine (SVM) classifiers. In total, 5,121 multi-labelled Arabic documents were collected from Saudi newspapers and used in their experiments. The results showed that the association rules technique gave the highest classification accuracy when it is compared with SVM and NB.

Ezzat (2010), in his thesis, presented so many contributions in the field of multi-label classification. The first one is by discovering the high performance of multi-label classification with respect to large amounts of data. Then, he introduced a new method that uses the correlations between a chain of binary classifiers to enhance the performance of classification and get rid of overfitting problems. Finally, he presented another technique that depends on pruning the label set and sub-sampling them to scale up the algorithm to handle large amounts of data. All the mentioned techniques have been evaluated by some assessment methods appropriate for the multi-labeled data set classification such as Hamming Loss, Exact Match and others. His work had a great impact on enhancing the performance of the multi-label classification process.

Ezzat *et al.* (2012) provided an unsupervised topic analyzer tool that does not depend on either a training data set or any other classification methods. A large multi-label Arabic data set is considered in this study. This data set is very large (about 5,962 documents) because it was constructed by collecting the newspaper and magazine on the Al Jazeera website. Their topic analyzer system works by using feature selection methods that are used to assign labels to the test data. This technique registered a high score for classification accuracy when compared with corpus-based methods.

Zhang and Zhou (2007) presented a multi-label classification technique that depends on modifying the traditional KNN classifier. An unseen data set is classified by comparing each paper to its calculated KNN neighbors for each category within the label set configuration. Then, according to some information gain and maximum posterior gene function measurements, the unseen instances are assigned to the probable label set.

In their experiment, a data set of images of yeast is used to measure the classification accuracy of their method. Their mechanism outperformed the other traditional multi-label classifiers such as Rank-Svm, Adtboost.Mh and Boostexter.

Finally, Elhawary and Elfeky (2010) present a system for subjectivity and sentiment analysis of Arabic Web pages. What is interesting about this system is that it contains a supervised multi-label classifier to assign a tag to each Web page from the set {review, forum, blog, news, shopping store}. Although this classifier is used in a different setting (sentiment analysis), its generalization to general multi-label classification makes it similar to Ahmed *et al.* (2015).

2.2 New aspects in multi-label classification

There are multiple ways to deal with multi-label data sets and classify them appropriately. Such techniques differ in nature depending on whether they use either PT methods or algorithm adaptation methods. Using the first method, we have to transform the data set to a single-labeled one to make it easy to work with, especially if we want to apply machine-learning methods to the documents. While, the other technique suggested adapting a particular single-label classifier to accept the multiple classes found in each instance with the data set. A research project (Elhawary and Elfeky, 2010) presented six methods called PT methods (PT1-PT6) to transform their multi-label corpus, which is a biological data set, into a single-label data set. These methods are considered preprocessing techniques that should be carried out before any learning techniques are applied to the transformed data. The other techniques, the algorithm adaptation techniques, also presented by Elhawary and Elfeky (2010), depend on modifying the algorithms to deal directly with multi-label data sets. Vogrinčič and Bosnic´ (2011) followed the same path as that used by Elhawary and Elfeky (2010) to classify their multi-label data set using an ontology. They made a comparison between classifying the data using PT techniques and classifying the same data set directly using corpus-based multi-label classifiers such as the back-propagation multi-label neural network (BP-MLL) and multi-label KNN algorithms. The results showed that the corpus-based methods outperform the techniques that depend on PT methods.

2.3 Lexicon-based classification

The single-label corpus-based learning has always demonstrated great accuracy when compared with unsupervised and semi-supervised techniques such as lexicon-based classifications. Researchers apply and adjust their lexicons to reach a point at which the accuracy of their systems gets the same or as good as the corpus-based techniques. But, unfortunately, till now, all the studies that use lexicons have lower accuracy than the corpus-based methods. One of these research projects that presented a semi-supervised technique which uses a part of the corpus to construct features and terms to build a domain-based sentiment lexicon is that of Lau *et al.* (2011). This technique depends on the document-term and term-term relationships to extract that information. This automatic method gave lower accuracy when it was compared with the manual method, as it depends on the frequency with which a term is mentioned in the corpus and whether that term is relevant. So their new technique took advantage of the parsing method used by the linguists, then follows it up with the automatic method used in the corpus to get better accuracy.

Another method that makes use of three main predefined dictionaries (HowNet, Dalian University of Technology and National Taiwan University sentiment dictionaries) to get sentiment features and use them as a basis to classify their multi-sentiment micro-blogs' data set. Their study showed that the multi-label sentiment classification using the Dalian University dictionary has better results than the other two dictionaries (Liu and Chen, 2015).

Another study was carried out by Weichselbraun *et al.* (2013). They described a new way to sentimentally classify documents using a context-aware sentiment lexicon. By first identifying the ambiguous sentiment terms (the terms that are rated as both positive and negative on the first pass) in the training data and put it in a contextualized lexicon and then identifying the sentiment terms and putting them in a sentiment lexicon. The classification method starts by defining sentiment terms in the testing data, then any term that is found within the contextualized lexicon or not included in the sentiment lexicon is considered as neutral. This technique enhances the sentiment classification and produces better accuracy when it is time to exclude ambiguous sentiment terms.

Maite *et al.* (2011) also built a lexicon-based system called SO-CAL (the Sentiment Orientation Calculator), designed to perform multi-label sentiment classification with a constructed lexicon. This lexicon is constructed and annotated manually by expert linguists through several stages. The SO lexicon included all those parts of speech that can influence the expression of the sentiment of the sentence, adjectives, adverbs, verbs and nouns. The textual corpus included 400 annotated reviews. The results from Maite *et al.* (2011) showed that larger lexicons built automatically do not necessarily produce higher performance than a manual lexicon, which may contain useless terms that have a negative effect on the sentiment accuracy. SO-CAL, however, demonstrated higher accuracy in comparison with other manual and automatically constructed lexicons mentioned in their review of the literature.

More recently, Abdulla *et al.* (2014) published another study using lexicons constructed from Arabic text, both manually and automatically. The manual one was built using 300 seed words (English words) from the SentiStrength Web page, and then they translated those words into Arabic, giving them the same polarity found in that site. After that, the synonyms of each word were added to the lexicon expanding it to 4,815 words. They compared two different automatic methods: the direct translation of a one well-known English lexicon, gave approximately 9,100 different words, and the other method was carried out using a balanced labeled portion of their corpus. Then they applied the term frequency (TF) weighting method to extract features and use them as sentiment entries for the automatic lexicon. Finally, they augmented the manual and the automatic lexicons to see how that can affect the accuracy of the classification process. As expected, the integrated technique gave the higher accuracy among them all, as it accumulated the advantages of the manual technique, the accurate human annotated words and the advantages of the huge amount of the automatic lexicon entries.

Finally, Musto *et al.* (2014) carried out a research project focused on the feasibility of an unsupervised approach to sentiment classification. They used four lexical resources to get the sentiment terms with their polarity: MPQA, Senti-Word-Net, Senti-Net and Word-Net-Affect all considered the state-of-the-art lexical resources in their domains and they tried to compare between them. The data set was collected from two Web pages, Sem-Eval- 2013 and Stanford-Twitter-Sentiment (STS). The total size of the data set is 1,614,000 tweets. Their approach is based on classifying the data into three sentiment polarities (positive, neutral and negative) by dividing each tweet into small micro-phrases, then retrieving each sentiment term of the entire phrase from a particular lexical resource, and getting its polarity accordingly. Finally, they accumulated the whole sentiment polarity of the entire tweet to get the total orientation. Their results showed that MPQA and Senti-Word-Net outperformed the rest of the lexical resources; this comes from the subjective terms that each lexical resource have to enhance the classification process.

2.4 Feature selection methods applied to improve classification

Neumayer *et al.* (2011) presented a study that compares different feature selection techniques. The authors divided these methods into two groups, corpus-based and unsupervised feature selection techniques. In addition, the authors presented a merged set of feature selection methods. The experiments were performed using an SVM and a fivefold cross validation assessment strategy that were applied on RCV1-v2 test collections. The results showed that the best accuracy was given by the combined feature selection method.

Abbasi *et al.* (2011) produced a research study that includes sentiment classification by identifying n-gram features, with the help of a feature selection technique called feature relation network (FRN). The FRN model platform used a rule-based multivariate feature selection approach that focuses only on semantic information and the influence of the syntactic association of the different n-gram features. FRN is used to enhance the classification process of two test data groups, digital camera and automobile 1- to 5-star reviews. There were three types of feature selection methods in the literature – univariate, multivariate and hybrid feature selection methods – used to compare their FRN model with. The results showed that the FRN model outperformed these methods in the classification of sentiment reviews.

3. Approaches in multi-label classification of Arabic data

In this section, we focus on two approaches to classifying Arabic multi-label data: Corpus based and lexicon based classification methods. Although there are several works that include a corpus based method to classify multi-label data in the literature (Ezzat, 2010), as far as we know, the amount of multi-label classification of Arabic data (MLCAD) research is limited. The other approach that we will describe in this section is called the lexicon-based method. This technique is considered unique, especially in the Arabic language field. Classifying multi-label data using a lexicon takes place after collecting the Arabic multi-label data set described in our previous work (Ahmed *et al.*, 2015).

3.1 Corpus-based learning on multi-label classification of Arabic data

The general concept of machine learning involves three different approaches: supervised or corpus-based with an annotated corpus, unsupervised and semi-supervised (Hotho *et al.*, 2005). The corpus-based method aims to build classification models from a part of the corpus called the training data, and then the other part of that corpus is maintained as a testing data that are used by the classifier to predict the classes of the unseen data. In the multi-label environment, dealing with corpus-based technique differs from that in the single-label one. There are two main approaches in the corpus-based multi-label classification: PT methods and Algorithm Adaptation methods (Ezzat, 2010). The first technique begins by transforming the multi-label data into single-label data, and then it uses the traditional base classifiers to classify unseen data. The second technique adapts some base classifiers such as KNN and SVM to deal directly with multi-label data (Ahmed *et al.*, 2015; Ezzat, 2010). In our work, we chose to work with the first approach (i.e. the PT method) to carry out our baseline experiment. The next subsections will introduce some of these techniques:

3.1.1 Label combination method. This technique considers all the labels within each label-set related to their examples as one single atomic (combined) label. This method gives good results in the literature comparing it with the others (Ahmed *et al.*, 2015).

3.1.2 Binary relevance method. Each label in the label-set will be assigned a binary classifier that treats each label as: present (+ve) or absent (−ve) depending on whether it exists in the label-set within each example (a complete record contains one example from the data and a set of labels) (Ezzat, 2010).

3.1.3 Ranking and threshold method. This method decomposes each label in the label-set environment, and then appends them to their related instances. After that, each label with a confidence value greater than a particular threshold remains while the rest are discarded (Ezzat, 2010).

Other PT techniques are described in detail in Ezzat (2010) and Ahmed *et al.* (2015), which is considered as a state-of-the-art description of techniques in a multi-label environment.

3.2 Lexicon-based learning in multi-label classification of Arabic data

This technique is the main contribution in this study, and it depends on the combination of lexicons to classify multi-label Arabic data as depicted in Figure 1. Lexicons are dictionaries that store special words called words or phrases; these terms represent a meaningful purpose such as sentiment terms or terms from some other special domain (Maite *et al.*, 2011). The following subsections explain in detail all the stages for constructing a lexicon-based multi-label Arabic TC system.

3.2.1 Data set description. In this approach, we collected 51,114 Arabic articles from the *BBC News* website using a crawler. The crawled data set contains many labels (67 labels, to be specific). Because some labels have a very small number of articles, it might be difficult for our approach to automatically build lexicons for them. So we narrow

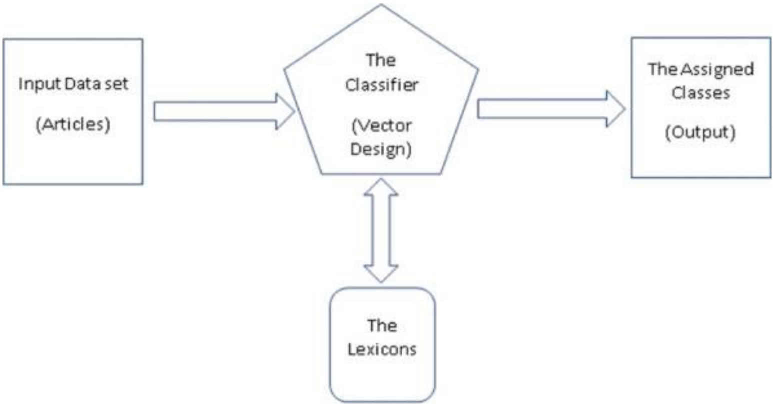


Figure 1.
The Lexicon-based multi-label Arabic TC system architecture

down the number of labels by excluding any label with fewer than 100 examples. Once the pruning process is complete, 8,800 examples with 35 labels were maintained and kept for use in this approach. Table I shows the different labels remaining to be used.

The remaining data set (i.e. the 8,800 examples) includes single- and multi-label instances. Next, it is divided and used for different purposes in this research. This will be clarified in the next paragraphs.

3.2.1.1 Multi-label data. In total, 4720 of the 8,800 examples, along with their relevant labels, are segregated to be treated as a pure multi-label corpus. We divided these data into two groups using a 70/30 split: one for training and the other for testing. The training data included 3,310 examples, and they are exploited to construct the lexicons.

The label "In Arabic"	The label "In English"
اسيا	Asia
افريقيا	Africa
افغانستان	Afghanistan
اقتصاد و تنمية	Economy and Development
الاتحاد الاوروبي	European Community
الامم المتحدة	United Nations
الخليج	Gulf
السعودية	Saudi Arabia
السودان	Sudan
الصراع الفلسطيني الاسرائيلي	Israeli–Palestinian Conflict
الصين	China
العراق	Iraq
المملكة المتحدة	United Kingdom
الولايات المتحدة	United States
اليمن	Yemen
انتفاضات العالم العربي	Arab World Uprising
اوروبا	Europe
ايران	Iran
باكستان	Pakistan
تجارة و اعمال	Trade and Business
تركيا	Turkey
تكنولوجيا	Technology
ثقافة وفنون	Arts & Culture
روسيا	
رياضة	Sport
سوريا	Syria
سياسة	Politics
صحة و تغذية	Health and Nutrition
علوم	Sciences
علوم و تكنولوجيا	Science and Technology
قضايا الشرق الاوسط	Middle East issues
كرة القدم	Football
لبنان	Lebanon
ليبيا	Libya
مختارات	Misc

Table I.
Distribution of the 35 labels left to be considered after pruning

While, 1,410 examples are included in the testing data and they are used to evaluate the system accuracy.

The reason behind choosing a pure set of the multi-label data is to explore the correlation between the labels within the same label set (i.e. the labels that share the same instance in the multi-label environment) and how that affects the performance of the classification process.

3.2.1.2 Single-label data. Our crawler encountered some Web pages on the BBC site that has only one label in its "related subject" area, and it collected just as it collected the other multi-labelled data (Ahmed *et al.*, 2015). The idea behind the existence of the single-label data in the multi-label environment demonstrates the presence of one label and the absence of the others, so indeed, they can be considered as multi-label data with just the absence of the rest of the labels. To explain this idea more, suppose that there are three instances, the first one includes four labels: politics, economic, culture and sport, whereas the second one contains three labels: politics, sports and culture, and the third contains only the politics label considered as a single-label instance. So the third instance actually is collected with the multi-label data, but the absence of the three other labels makes it a single-label instance. Moreover, by including the single-label data set in the training set to only construct a different kind of lexicon will help us see if there is any influence that can improve the accuracy of the classification process. The single-label data includes 4,080 examples with the same labels that are mentioned in Table I.

3.3 Lexicon construction

Building lexicons seems reasonable in some situations such as in the field of sentiment analysis because of the low number of the labels used specially with using many methods for this purpose, but building a domain-based lexicon needs more human effort and time to extract and annotate the relevant terms. In this study, we construct our lexicons automatically to avoid those complications, although it is less accurate than doing it manually. If we depend on human annotations for building 35 different lexicons, then we need 35 different experts, which is hard to afford. On the other hand, in our approach, one domain-independent method is used to build our lexicons.

TF is an example that depends on the recurrence of terms to define and extract features that are suitable for classifying documents in an unsupervised way (Abdulla *et al.*, 2014; Alwajeeh *et al.*, 2014). We depend on the training data mentioned in the last session to build the 35 lexicons. So we implemented a small tool that counts the term frequency of each label's examples, which were separated as described in the data set section. Assigning an appropriate threshold to select the most frequent terms helped us to build each lexicon for a particular label. So if our TF tool is capable of doing that process automatically and in one click, we can get all the useful terms that will be used as entries for all the lexicons. The shortcoming of the automatic way is the occurrence of some irrelevant terms that may result in a misclassification problem and consequently will affect the accuracy of the classification system.

The 35 lexicons are built using the previously described tool but how they are built depends on whether we are working on a multi- or a single-label classification.

3.3.1 The multi-label-based lexicons. As we mentioned in the multi-label data set subsection, 3,310 examples were prepared to construct the multi-label 35 lexicons. Also two kinds of lexicons are extracted: non-stemmed and stemmed lexicons to explore the accuracy of classification as a consequence of these two methods. A simple light stemmer is included in the TF tool and it only removes the stop words besides any prefixes found in each word. We end up with 70 lexicons, two groups, one with 35 stemmed multi-label lexicons and the other with 35 non-stemmed ones.

We have some observations from the resulting lexicons, which can thoroughly affect the classification process as well as the total accuracy:

- Some common terms are found in many other labels' lexicons and for sure they will return multiple labels for those terms, this certainly will support the multi-labelling property and further will explain the correlation between labels, but unfortunately it is a kind of a two-edged sword that will severely affect the classification process and add labels irrelevant to the results. This problem will be fixed later in the implementation phase.
- The number of terms in each lexicon is unbalanced, ranging between 35 and 800. This problem will cause some labels to dominate throughout the classification and produce a bias that is undesirable when estimating the accuracy. The reason behind this problem appears to lie in the process of setting the threshold of the TF that could return the unbalanced number of terms. So we tried to calibrate the TF values to be compatible with each lexicon and decrease the bias problem as far as possible.

3.3.2 The single-label-based lexicons. In working with lexicons of this type, 4,080 examples are exploited to build another two kinds of methods, stemmed and non-stemmed lexicons. A pure single-label data set is used in this process to construct these lexicons and follow the same procedure and structure as explained earlier in the multi-label-based lexicon section, so a total of 70 lexicons are built (35 stemmed and 35 non-stemmed lexicons). Our goal in working with this kind of date was to see the differences between the extracted terms that come from them in comparison with those from the multi-label data, and how they can affect the classification and the accuracy of our tool. There are some observations from the built lexicons:

- The number of entries in the constructed lexicons is more stable than that in the multi-label lexicons, and consequently a low bias is encountered.
- Most of the terms are relevant and may raise the accuracy of our system classification.
- A lower number of lexicons' entries are encountered than those in the multi-label lexicons (200 terms maximum and 150 minimum).
- The single-label lexicons have the same structure as the multi-label ones.

3.4 Characteristics of the data set and lexicon entries

Some features have been collected to increase our knowledge about the gathered data and the constructed lexicons. These statistics help us figure out how the number of words in the data set and the number of terms extracted for each lexicon affect the performance of the classification system that we present here. The features are:

- a detailed number of labels in each label mode (multi-label and single label data set);
- the number of training and testing data in each label mode;

- total number of words contained in the training and testing data set for each label mode;
- the total number of terms related for all the constructed lexicons considering the original and the stemmed lexicons; and
- the average number of words in each article.

Table II summarizes all the features mentioned regarding the related statistics.

4. The lexicon-based multi-label classification system

In this phase, we show in detail how the classification system works and all the steps of performing the necessary operations. Most lexicon-dependent classification systems described in the literature are focused on sentiment analysis and calculate the negative and positive terms in a particular text (Abdulla *et al.*, 2014; Maite *et al.*, 2011; Abdulla *et al.*, 2014). In this study, we followed the same techniques but by modifying it to be compatible with the nature of the domain-based lexicons.

4.1 The vector design

In this subsection, we describe the construction of the classification systems step by step.

4.1.1 Pre-processing step. This pre-processing method is used as a first step in most text-processing systems. The point is to get rid of unnecessary information that is found in the text and make it appropriate for classification (Neumayer *et al.*, 2011). The following steps show the different pre-processing techniques which are performed by our classification system to all the text that we want to classify:

- The text is tokenized.
- The article letter “الـ‘/’The” is removed.
- Letters are normalized such as converting “أ”/“Hamza” to “ا‘/’Alef”.
- All stop words such as “في, كان, إن”/“That, Was, In” are removed.

These four operations are performed as the text from each article in the data set is entered. Steps 2 to 4 are considered as a light stemming process that is accumulated within our system. Then each example (each row in the testing data set) represents the Arabic text with its relevant labels, so that the entire label-sets are kept and indexed with their related instances for the classification process.

The characteristics	Multi-label mode	Single-label mode
Total number of labels used	35	35
Number of training data	3,310	4,080
Number of testing data	1,410	–
Total number of words in the training data	826,744	1,021,28
Total number of words in the testing data	359,989	–
Total number of terms in the original lexicons	6,239	6,443
Total number of terms in the stemmed lexicons	7,445	7,721
Average number of words in each article	251	250

Table II. The collected characteristics of the data set and the lexicons

4.1.2 The classification operation. In this step, the main classification operation is carried out, and a lot of work is done to the entered text as follows:

- Each word in the entered text is searched in all lexicons, the tool returns its matched labels from the header (column name that contains the label name) of its related lexicon.
- The process in Step 1 is repeated for all of the terms within the text of the single example in hand.
- The total number of occurrences of each label found in Step 2 is counted.
- All the predicted labels are organized and prioritized according to their count value in an ascending order.
- Because the maximum label dimensionality (i.e. is the space of labels in a particular example, and the maximum label dimensionality will be the maximum number of labels contained in a particular label-set) in the data set is equal to 5, the system only considers the top five labels that have the greatest count value. This step is very important in narrowing the dimensionality of the predicted label-set, and moreover, it will discard the unnecessary labels that contain lower count value.

The lexicon-based multi-label Arabic TC algorithm clarifies an algorithm representing all the former steps.

The following example demonstrates the former operations.

The entered text: “يعتبر رونالدو احسن لاعب دفاع كرة” “قدم من ضمن المرشحين دوليا”

```
Input: Data set (Articles) D, Domain based lexicon Lex
Output: Hamming-Loss, Exact-Match, ML-Accuracy.
Load D
Begin
For each file in Data set                        //where file is an article record
Initial List<string> MaxWords          //where MaxWords is a list containing
                                         the top 5 labels
ListWords := Tokenize (file)    //where ListWords is a list of the tokenized file
Initial List<string,double> KeyWords       //where KeyWords is a list of the
                                             returned labels
    For each word in ListWords
    IF stemming THEN                   //stemming: if the stemming option is on
    Word:= DoLightStemming(word)
    END IF
    Search for word in Lex
    Get L (word)
    value := Count (L)
    KeyWords. ADD (L, value)
    END For
Sort KeyWords in Descending Order
For x:= 1 to 5
MaxWords.ADD(KeyWords.word)
END For
Calculate Hamming-lose(MaxWords,file.L)       //where file.L is the original label
                                                set of the file
Calculate Exact-Match(MaxWords, file.L)
```

```
    Calculate ML-Accuracy(MaxWords, file.L)
END For
END
```

The classification and scoring assignment are shown in Table III.

The process of retrieving each term appearing in the lexicons can be seen more clearly in Figure 2. The underlined italic terms demonstrate the retrieved lexical terms that are transmitted from the sports and politics lexicons. The null label signifies that no label is captured for the relevant terms.

4.1.3 Evaluation process. We estimate the classification here by using three measurements that are familiar in a multi-label classification environment. ML-Accuracy, Hamming Loss, Exact Match and Execution Time are those measurements that we consider to evaluate our system. The following steps show the mechanism for evaluating the classification process automatically:

- For each tested example in the data set, and for the relevant data set constructed in the previous section, the original and the predicted label-set are compared. This can be accomplished by constructing a matrix of all the 35 labels before and after the classification operation as Table IV shows.
- Use the binary representation (0/1) to express the absence (0) or the presence (1) of a particular label within the label-set and then fill the matrix accordingly.

Suppose we have an example with the original data set: (سياسة, العراق) and the predicted label set is: (سياسة, اليمن, الامم المتحدة, العراق, سوريا), the matrix representation will be as shown in Table IV.

Now the example is ready to be evaluated using the four mentioned measurement metrics (Ezzat, 2010) as shown in Table IV.

So suppose that D is the training multi-label data set, N is the total number of examples, L is the total number of labels, Y is the label-set annotated by the human experts, Y' is the label-set after the prediction process, then we compute the metrics as follows.

Exact Match: In this metric, it calculates the number of the label-sets in which the original and the newly constructed label-sets are exactly the same. (i.e. Y = Y') averaged by the total number of examples in the tested data set. For the former example, this metric equals to zero, as Y ≠ Y'.

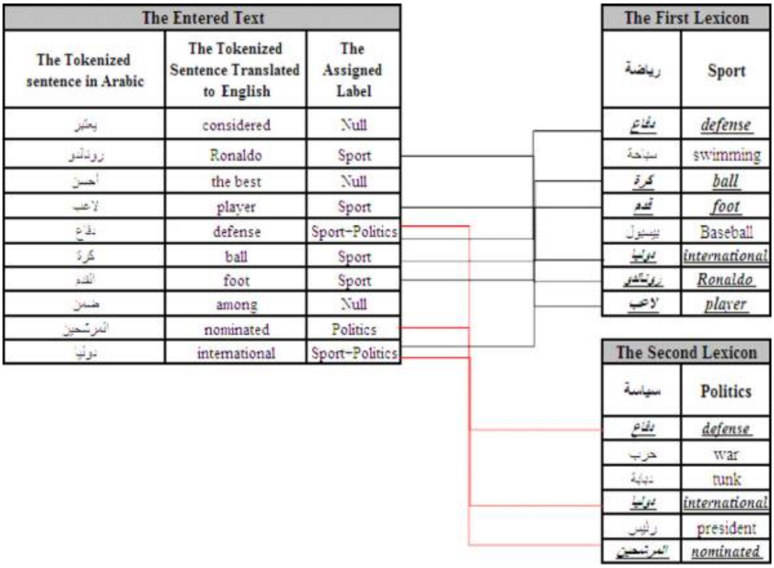
$$Exact\ Match = \frac{1}{N} * \sum_{i=1}^{N} 1_{y'_i = y_i} \quad (1)$$

For example, suppose that there are only four labels in our data set: "سياسة"/"Politics", "رياضة"/"Sport", "اسيا"/"Asia", and "تكنولوجيا"/"Technology", and three testing unseen instances are been used to evaluate the lexicon based system: X1: [Y1 (1, 0, 1, 0), Y'1(1, 0, 0, 0)], X2: [Y2 (1, 0, 1, 1), Y'2 (1, 0, 1, 1)] and X3: [Y3 (1, 1, 1, 0), Y'3 (1, 0, 0, 1)]. Where Y represents the actual label-set of each instance and Y' represents the label-set constructed by our system. Then the resultant Exact Match will be calculated using the following equation:

$$ExactMatch = 1/3\,(1) = 0.333$$

Table III.
An example for how the system will classify the entered text

Sentence in Arabic	دوليا	المرشحين	ضمن	قدم	كرة	دفاع	لاعب	احسن	رونالدو	يعتبر
Sentence in English	international	nominated	among	foot	ball	defense	player	the best	Ronaldo	considered
Classes	C S+P	CP	Cnull	CS	CS	C S+P	CS	Cnull	CS	Cnull
Count	CS = 6		CP = 3		Class sport = 66% while Class politics = 34%					



The Entered Text		
The Tokenized sentence in Arabic	The Tokenized Sentence Translated to English	The Assigned Label
يعتبر	considered	Null
رونالدو	Ronaldo	Sport
أحسن	the best	Null
لاعب	player	Sport
دفاع	defense	Sport–Politics
كرة	ball	Sport
القدم	foot	Sport
ضمن	among	Null
المرشحين	nominated	Politics
دوليا	international	Sport–Politics

The First Lexicon	
رياضة	Sport
دفاع	*defense*
سباحة	swimming
كرة	*ball*
قدم	*foot*
بيسبول	Baseball
دوليا	*international*
رونالدو	*Ronaldo*
لاعب	*player*

The Second Lexicon	
سياسة	Politics
دفاع	*defense*
حرب	war
دبابة	tunk
دوليا	*international*
رئيس	president
المرشحين	*nominated*

Figure 2.
The retrieval process of each lexical term from politics and sport lexicons

Hamming Loss: This metric calculates the errors (i.e. the mismatch in labels Y ≠ Y’) found in the 35 labels for every tested example, and their label-sets over all the 35 labels in the matrix. So, for the former matrix, this metric equals 3/35. Then after calculating this metric for the entire test set, it will be divided by the total number of examples.

$$Hamming\,Loss(D) = \frac{1}{N * L} * \sum_{i=1}^{N} |y'_i \Delta y_i| \tag{2}$$

Using the same example that has been applied to the Exact Match metric, the resultant Hamming Loss will be calculated using the following equation:

$$HammingLoss = 1/3\,(4/4) = 0.333$$

ML-Accuracy: Here, this metric calculates the number of labels that are correctly classified by the system, divided by the sum of the number of misclassified labels with the correctly classified ones (in the earlier matrix it equals 2/5) averaged by the total number of examples after calculating this metric for all the tested data.

$$ML\text{-}Accuracy(D) = \frac{1}{N} * \sum_{i=1}^{N} |y_i \cap y'_i| / |y_i \cup y'_i| \tag{3}$$

Using the same example that has been applied to the Exact Match metric, the resultant ML-Accuracy will be calculated using the following equation:

$$\text{ML-Accuracy} = 1/3\,(1/2 + 3/3 + 1/4) = 0.583$$

Table IV.
The matrix representation of the original and the predicated label set of the example in hand

The 35 labels`	اسيا	افريقيا	اليمن	اوروبا	العراق	الامم المتحدة	رياضة	سوريا	سياسة	. . .	Label 35
Y (The original label-set)	0	0	0	0	1	0	0	0	1	. . .	0
Y’ (The predicted label-set)	0	0	1	0	1	1	0	1	1	. . .	0

Execution Time: After the system completes its work for the whole tested data set, then the time duration in seconds of executing the classification process is estimated (i.e. Total time = End time of the classification − Beginning time of the classification).

5. Experimental setup and results

In this section, we will explain in detail all the output from the constructed system and its results as a consequence of the classification process, taking into consideration that the experiments run in an i5 core processor system with 8 GB RAM (internal memory) to provide high performance when the classification system is applied. The operating system used is Windows 7 which is also suitable for our experiments.

5.1 Lexicon-based multi-label classification toolkit

The classification system is built using the C# programming language. This environment is a good choice to construct our system with. In particular, it can handle a huge amount of data without expressing the classical out of memory issues.

The toolkit provides two main features: the first one affords more than just loading the testing data such as: stemming the entered data without maintaining this process individually, the ability of selecting the preferable lexicons so that you can make the experiments depending on multiple lexicon types at once (i.e. either selecting the original multi-label lexicon, the original single-label lexicon, the stemmed multi-label lexicon or the stemmed single-label lexicon). The previous lexicon types work according to the nature and structure (the way in which the lexicons were constructed) of the contained terms, as described in the lexicons subsection:

- *The original/multi lexicon*: This is the one in which the original (not stemmed) multi-label data are used to extract its terms.
- *The original/single lexicon*: This is the one in which the original (not stemmed) single-label data are used to extract its terms.
- *The stemmed/multi lexicon*: This is the one in which the stemmed multi-label data are used to extract its terms.
- *The stemmed/single lexicon*: This is the one in which the stemmed single-label data are used to extract its terms.

All those lexicons have the ability to be selected individually or with other desired ones to increase the scanning dimensionality and to explore the effects of merging multiple kinds of lexicons.

The other feature that our system provides is the accumulative growing of results with each example taken for classification. Each evaluation metric, such as ML-Accuracy and Hamming Loss, is made available after each testing article is processed.

5.2 The lexicon-based experiments

After finishing the baseline experiments in the last section, it is the time to perform several experiments on the same data set. The different settings of our system are carried out according to the assignment of the different lexicon types: the stemmed and the original (lexical terms from the original data, i.e. without stemming) types, each of which is constructed from the training data set as explained earlier in the lexicon construction section, while the other part of the data set, the testing of 1,410 records, will be used for the classification and the evaluation steps.

5.2.1 Results from the stemmed lexicons. This type of lexicon is captured as a result of extracting lexical terms from the stemmed training data, the 3,310 training records, and they are grouped into two essential sets the multi- and single- lexicons. These two sets are grouped according to the kind of the training data set (the pure multi- and single-data) as we described in the lexicon construction section. Table V shows the results that were gained from the setting of both, the stemmed multi- and the stemmed single lexicons, individually during the classification process.

Several observations come to mind from Table V as follows:

- The best ML-Accuracy is gained from classifying the testing data using the stemmed multi-label lexicon (25.4 per cent).
- Our system failed to find exact matching of the label-sets using both types of stemmed lexicon.
- The best Hamming Loss is preserved by the stemmed multi lexicon, giving 0.103 value of this metric.

Finally, our system is suffering from the Execution Time, making the stemmed single lexicon the worst case with this metric (370,360 s).

5.2.2 Results from the original lexicons. To see the effects of the original terms (i.e. without stemming) in the article text, another type of lexicon comes into view. As we clarify in the lexicon construction section, this lexicon is also grouped into two sets: multi and single sets that depend on the nature of the extracted data (pure single- and multi-label). Table VI describes the results behind the experiment when it is applied on the original lexicons.

Here are some observations from the results in Table VII.

- The highest ML-Accuracy is produced by the single-label lexicon which outperforms the other lexicons by increasing the accuracy of classification to (27.5 per cent).
- The Exact Match still produces a zero value which does not change the relationship of this metric with the other kinds of lexicons.

Table V.
The results from both sets of the stemmed lexicons

Lexicon type	ML-Accuracy	Exact Match	Hamming Loss	Execution Time (in seconds)
Stemmed single-label data	0.2343	0	0.105	370,360
Stemmed multi-label data	0.254	0	0.103	343,380

Table VI.
The results from both sets of the original lexicons

Lexicon type	ML-Accuracy	Exact Match	Hamming Loss	Execution Time (in seconds)
Stemmed single-label data	*0.275*	0	0.099	365,700
Stemmed multi-label data	0.267	0	0.113	377,580

The transformation method	The basic classifier	ML-Accuracy	Hamming Loss	Exact Match	Execution Time
LC	SMO	0.021	0.074	0	3,600.386
	NB	0.016	0.075	0	146.846
	IBK	0.019	0.075	0	34.124
	J48	*0.092*	0.077	0	656.008
BR	SMO	0.014	0.07	0	1,647.958
	NB	0.008	0.074	0	314.639
	IBK	0.024	0.087	0	3,641.265
	J48	0.022	0.075	0	1,489.542
RT	SMO	0.019	0.077	0	15,116.007
	NB	0.014	0.07	0	1,647.958
	IBK	0.024	0.087	0	106.487
	J48	0.014	0.075	0	361.04

Table VII. The baseline experiment results (in the MEKA system)

- The Hamming Loss metric gets its highest value with the multi-label lexicons about (0.113).
- Finally, the Execution Time is reaching its worst values with the multi-label lexicon (377,580 s).

5.3 The baseline (corpus-based) experiment

A baseline experiment must be performed to compare our work performance and efficiency. The MEKA multi-label classification system is an open source Java library that is considered an extension of the popular Waikato Environment for Knowledge Analysis (WEKA). MEKA had a great impact when we used it to classify Arabic multi-label data in our corpus-based multi-label classification work (Ahmed *et al.*, 2015). Accordingly, using the same data set mentioned in the last section (especially in the multi-label data subsection), we set up the same 3,310 training and the 1,410 testing data that were used in the lexicon-based system to be convenient with MEKA system. Also, we select the three main types of PT, in particular: the Label Combination (LC) method, the Binary Relevance (BR) method and the Ranking and Threshold (RT) method, each of which is tested with four classifiers in MEKA (SMO, NB, IBK and J48). Besides, we depend on the same evaluation metrics that were applied on the lexicon-based system (ML-Accuracy, Exact Match, Hamming Loss and Execution Time) to assess the accuracy and performance of the classification in MEKA.

5.3.1 Results from the baseline experiment. As a consequence to the former experiment settings and running, Table VII demonstrates all the measurements' readings that resulted from the MEKA system. The table reveals many observations as follows.

- The first one is that J48 outperforms the other classifiers and gives the best ML-Accuracy (9.2 per cent) when it is applied with the LC PT method.
- The worst accuracy is produced by NB when it is used with the BR transformation method (0.8 per cent).
- Also, no classifier succeeds in getting an Exact Match in any classification process whatever PT is used.

- The worst Hamming Loss score appears when IBK is used with BR and the RT PT methods (0.087), while the best appears when SMO is working with BR and NB with the RT transformation methods (0.007).
- Finally, the longest Execution Time is observed when SMO is working with the RT PT method (15116.007 sec.), while the best time is measured as 34.124 s by IBK when used with the LC transformation method.

6. Assessment of the stemmed lexicon-based system results

6.1 ML-Accuracy

From the results of Tables V and VII, we can infer that our system succeeded in producing a higher ML-Accuracy than the MEKA baseline experiments. This remarkable outcome can be interpreted in the following points:

- The great dimensionality of words that can be found in an article for sure plays the main role to increase the ML-Accuracy, taking into consideration that this property is not found in any other classification systems such as MEKA.
- The richness of this gives us the opportunity to capture more labels in the ML-Accuracy. That gives the opportunity to catch as more as possible labels to represent the correct predicted label-set.
- The ML accuracy is increased gradually as the label dimensionality is also increased during classification, as the ML-Accuracy depends on the number of true positives while evaluating each row of data.
- Within the lexicon-based environment, using the stemmed multi lexicon gives the higher ML-Accuracy rather than the stemmed single lexicon, that comes from the correlation property of the multi-label data and its extracted lexicon. With this property, it is possible to find the same lexical term existing in multiple labels' lexicons, for example, the term "دفاع" which means "Defence" is found within the "سياسة"/"politics" label and "رياضة"/"sport" label. Then the morphological ambiguity in words will increase the probability of encountering more matching labels and hence raising the ML-Accuracy.

6.2 Hamming Loss

From the former tables also we noticed that SMO and NB give the best Hamming Loss results comparing them with our system. The difference is about 3 per cent more in our system than that in MEKA. The following points can explain the main reasons behind this result:

- Because in our system we consider only the top five predicted labels that have the greatest count values over all the others (i.e. fuzzy method) cause a label-set with two or three labels to have a harmful effect on the value of the Hamming Loss metric (because the rest of the labels within the label-set are known in advance that they will cause the mismatch).
- The unbalanced number of terms within the lexicons makes the labels of the lexical items with large values dominate the others, and the probability of encountering a mismatch will increase as well.
- Within our system environment, the difference in this metric is very small and negligible between the two stemmed lexicons (about 0.2 per cent). So the missing

correlation property in the single-label lexicons causes the absence of one or more correlated labels in the predicted label-set.

6.3 Exact Match

Because it is hard to encounter the correct full label-set, especially when we assigned the full range of the predicted label-set which equals the maximum label-set dimensionality (i.e. equals five), this metric always gives a zero value. To avoid this problem, we should use a threshold mechanism that can govern the dimensionality of the predicted label-set. Although MEKA has many threshold schemes when they classify the data, as Tables VI and VII show, the Exact Match in all the cases gives zero value which represents the failure of the system to capture an Exact Match.

6.4 Execution time

The time of classifying and evaluating the testing data is remarkably huge in our lexicon-based system in comparison with that in the MEKA system. This time is consumed because of the following reasons:

- The property of scanning each word in the article text to retrieve the label depends precisely on the number of words in the article, which affects the time spent. The more words to scan, the more time it takes to execute.
- Accessing the lexicon database for a particular word in hand takes time that linearly increases as the same word exists multiple times in the text and in multiple lexicons.
- Another reason is because the classification in MEKA does not depend on the huge number of words in the article but on the number of words remaining after making the string-to-word vector filter that will define a particular number of terms used in the classification process.

7. Assessment of the original lexicon-based system results

In this section, we will discuss the results from the original lexicon system only, as we examined the results from the baseline system in detail earlier when we compared them with the stemmed lexicon.

7.1 ML-Accuracy

The original lexicons, either the multi or the single type, remarkably give the greatest value in comparison with its siblings of the other lexicon type (i.e. the stemmed one). There is one property that is present in the original type and absent in the stemmed one, the diversity of the lexical terms' formats tend to increase the number of data domains (i.e. the variety of labels). For example, the word "تضارب"/"conflict" that is within the science lexicon and the word "تضارب"/"venture" that is found in the economics lexicon, which are both originally not stemmed words, tend to return two different labels, but if we stem those words, then a single word results "تضارب"/"the past tense of the word conflict" and hence a single label will be returned accordingly. This property caused the original lexicon to score better than the stemmed lexicon. We also noticed that the lexical terms of the single-label lexicon work well with the original data to produce the highest ML-Accuracy value. Finally, it seems that the correlation property of the multi-label data does not work well to increase the ML-Accuracy when it is used with the original data.

7.2 Hamming Loss

This metric reaches its minimum value over all the other lexicons when the original single-data lexicon is conducted. The highest number of lexical terms in the original single-data lexicon (6443 terms) over the multi-label one (that contains 6,239 terms) increases the possibility of producing a winning label while searching for a term match, hence decreasing the mismatch cases (i.e. decreasing the Hamming Loss factor).

7.3 Exact Match

As this metric produces a zero value all the time whatever the lexicon type used, the Exact Match measurement proved the failure of our system and the MEKA System to capture a one single positive result. The same analysis of this failure has been explained in the stemmed lexicon section.

7.4 Execution Time

The original lexicons record the highest Execution Time over the stemmed lexicons and the baseline experiments. This time delay is due to the huge text dimensionality that a non-stemmed article may possess. Several mechanisms can be used to increase the performance of the lexicon-based system.

8. Enhancement method

We saw in the last section how the ML-Accuracy is getting better using the lexicon-based approach. Although the accuracy in the lexicon-based system outperforms that in the corpus-based method, it is still not good enough to use in real-life applications. So in this section we will present a threshold method for the sake of improving ML-Accuracy as well as the Hamming Loss.

8.1 Adaptive threshold

Applying threshold functions is considered a useful way to control the resultant label-set in the output of the classification so that neither too large nor too small set is considered (Ezzat, 2010). Specifically, for a specific example and a set of fractional values, each representing the "confidence/probability" that this example belongs to one of the labels under consideration, we want to use the threshold value to transform these fractional values into binary ones (zeros or ones) to determine which labels should be assigned to this example. This can be done in two methods (Ezzat, 2010). In the first one, each label has its own threshold, while in the second one, all the labels within the output confidence are subjected to just one threshold. In our work, we use the second method.

- Abdulla *et al.* (2014), in their work, applied an adaptive threshold method that manually selects the correct TF score then returns the most relevant lexical terms of their sentiment lexicon. This technique seems promising regarding the enhancement in their classification accuracy, but applying this method in our work seems very expensive, as 35 lexicons must be adapted. So in our work, the lexicon is ready to use and we tried to automatically apply the threshold method from that point (i.e. after lexicons construction).

8.1.1 Using the training data set. In this step and for the sake of extracting a particular threshold value for all the 35 labels found, the training set is used in this process to assign the right threshold value for all the labels. Taking into consideration that we have two different sets of the training data, the one which is used as a pure multi-training set and the other as a pure single0-training set, we calculate the thresholds separately for each training set.

8.1.2 Calculating the TF for each example in the training set. In this step, each instance (i.e. each article text) is scanned in every label lexicon to search and count the number of the terms for the label in hand (i.e. the TF for the lexical terms found in the article text). Then from the label set of the example in hand, we see if that label is present within the label set or not. If it exists, then we assign a "1" value and if not a "0" value will be given. This process will be done for all the 35 labels in our lexicons. Table VIII demonstrates the final representation of each label with its TF score and its binary value that represents the presence/absence of the label within the label-set.

8.1.3 Sorting the TF values. For each label in hand, all its TF values are sorted in an ascending order and accordingly all the fields in Table VIII are organized as well.

8.1.4 Calculating the errors. In this step and for each label table, we calculate the errors related to the correctly and the incorrectly classified articles regarding to the presence/absence field in Table VIII. The following steps will show this process in detail:

- A virtual line will be drawn such that any sets above this line must be the corrected classified articles (i.e. their presence/absence value is 1), and any sets below that line must be the incorrectly classified articles (i.e. their value is 0).
- According to the movement of that line from top to bottom of the sorted table, the error will be calculated such that the number of records which contradicts what is mentioned in Step 1 is considered as an error.
- By the movement of the virtual line from one record to the other, we calculate how many errors were found and select the smallest one.
- Finally, the location of the smallest error (represented by the location of the virtual line at that point) is maintained so that taking into consideration two TF values, the upper and lower TF values (i.e. the values above and below the line).
- Calculate the average sum of both TF values such that:

$$Threshold = (TFupper + TFlower)/2 \tag{4}$$

- Performing Steps 1 to 5 for all the 35 labels to calculate their thresholds.
- Then a new count value is obtained by:

$$Countnew = Countlabel/Thresholdlabel \tag{5}$$

8.1.5 An illustrative example of how to calculate errors and threshold values. Now we reorganize Table VIII as shown in Figure 3.

The instance (article) no.	TF of "Asia" lexicon terms	Presence/Absence
1	100	0
2	95	1
3	88	1
. . .	. . .	. . .
3310	15	0

Table VIII. Assigning the TF score and binary value for label "اسيا"/"Asia" in all its relevant examples of the training set

By following the steps of calculating the errors:

- *Step 1*: We made a line that begins moving from instance number 1 until it reaches to instance number 8.
- *Step 2*: According to the line movement, we have to check if all the fields of "presence/absence" column above the line are zeros, and below that line all of them are one.
- *Step 3*: In the case of our example (i.e. when the virtual line is located after instance number 4), we notice that there are two errors that contradict the assumption made in Step 2 (i.e. at instance numbers 3 and 7). So the total error is equal to 2. When the line resides after the instance number 6, then the errors will be at instances number 3, 5 and 7. So the total error then is equal to 3.
- *Steps 4 and 5*: Accordingly, the smallest error appears when the line resides after the instance number 4, and we should calculate the threshold according to the following equation.

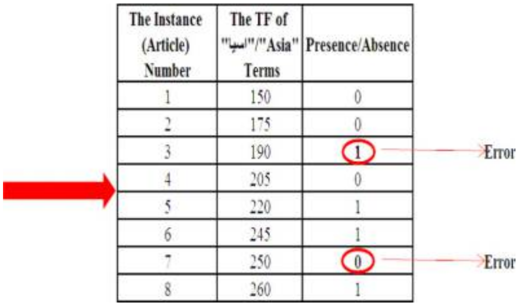
$$\text{Thresholdlabel} = (\text{TFupper} + \text{TFlower})/2 = 205 + 220 = 425/2 = 212.5$$

- *Step 6*: Perform Step 1 to 5 to calculate the thresholds for the other labels.
- *Step 7*: Suppose that the Steps 1 to 5 are performed for the label A1 ("اسيا"/"Asia"), and the performance output for this label is equal to 10. By applying the following equation, the output of label A1 equals to:

$$\text{Countnew} = \text{Countlabel}/\text{Thresholdlabel} = 10/212.5 = 0.0471$$

8.2 Experimental setup

We made some modifications to our lexicon-based system to calculate the threshold values of all labels. The threshold process is done with respect to the four lexicons types for creating four different sets of threshold values. Accordingly, each label is getting four different threshold values depending on the four types of lexicons. Now the same testing data set is used for the experiment to see how the adaptive threshold method influences the classification process.



The Instance (Article) Number	The TF of "اسيا"/"Asia" Terms	Presence/Absence
1	150	0
2	175	0
3	190	1
4	205	0
5	220	1
6	245	1
7	250	0
8	260	1

Error
Error

Figure 3.
An example showing how to calculate the proper threshold using the adaptive method

8.3 Results and observations

Table IX shows all the readings resulting from applying the adaptive threshold technique after the experiments on the four types of lexicons were completed. There are several observations recognized during the experiment and after the results appear:

- The ML-Accuracy is remarkably enhanced and increased rapidly for all the four lexicons. The amount of this increment varies from one lexicon type to another. For example, the stemmed-single lexicon got 5 per cent more accuracy than before, while in the stemmed-multi lexicon, the accuracy increased by only 1.9 per cent above the old value and the original-multi obtains a 3 per cent accuracy increment, then finally the top lexicon which gave the highest ML-Accuracy score, exactly 4.2 per cent more than the old value, is earned by the original-single lexicon.
- The Hamming Loss readings also made good progress by decreasing its amount in all the four types of lexicons. The best Hamming Loss value is obtained by the original single lexicon which maintains 0.731 score. This score approaches the smallest value (0.7 score) of this metric with the baseline experiment.
- The Exact Match still returns a zero value as before with no change.
- The Execution Time increased in these experiments showing the greatest time of 550,800 s, more by one day (i.e. 86,400 s) than the old value which was obtained earlier by the original-single lexicon.

8.4 Assessment of the results

Integrating the idea of adaptive thresholding into the proposed system improved its effectiveness. This is achieved by choosing the most proper threshold values which allow for more suitable labels to have a better chance of appearing at the top of the results. Moreover, this technique gives a chance to capture more accurate labels within each resultant label-set and hence enhances the accuracy and the Hamming Loss measurements. Also keeping track of the TF value at which the smallest error occurs gives us a chance to avoid large values of Hamming Loss and then enhances this metric accordingly. It consumes more time to perform the calculation of the new TF values, then applying them to construct the threshold values, and finally uses them within the classification process.

9. Conclusion and future work

This paper presents a study that involves a lexicon-based system to classify the documents in an Arabic multi-label data set. Several experiments are made to show the ability of automatically building lexicons to classify multi-label Arabic text data in

Lexicon type	ML-Accuracy	Exact Match	Hamming Loss	Execution Time (in seconds)
Stemmed-Single label data	0.28	0	0.989	*435,600*
Stemmed-Multi label data	0.263	0	0.10	514,800
Original-Single label data	*0.315*	0	*0.731*	543,600
Original-Multi label data	0.296	0	0.873	550,800

Table IX. The results of the adaptive threshold experiments

comparison with a corpus-based approach using the MEKA open-source tool. A total of 8,800 documents in a multi-label BBC Arabic data set, mainly 7,390 training records (multi-label and single label data) against 1,410 testing records, are used in the former mentioned experiments. The training data are harnessed to extract two kinds of lexicons, depending on the type and structure of the data used (original and stemmed), each of which produces two groups of lexicons, the multi- and single-label groups. In general, the results show that our lexicon-based system provided a remarkably high ML-Accuracy comparing it with the corpus-based approach that uses the MEKA tool. To be precise, the original single-label lexicon produces the highest ML-Accuracy value among all the lexicon types. This study could be considered as a good start in researching Arabic and other languages.

As a future work, other methods can be applied to construct lexicons beside the recent one such as TF/IDF technique. Moreover, increasing the size of the lexicons may produce great progress in ML-Accuracy and the other evaluation metrics. Finally, increasing the performance of our system so that the huge time for classifying the data is reduced must be taken into account, such as speeding up the process of lexical terms retrieval.

Funding

This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

Note

1. www.internetworldstats.com/stats19.htm

References

Abbasi, A., Chen, H. and Salem, A. (2008), "Sentiment analysis in multiple languages: feature selection for opinion classification in web forums", *ACM Transactions on Information Systems (TOIS*), Vol. 26 No. 3, pp. 1-34.

Abbasi, A., France, S., Zhang, Z. and Chen, H. (2011), "Selecting attributes for sentiment classification using feature relation networks", *Knowledge and Data Engineering, IEEE Transactions on*, Vol. 23 No. 3, pp. 447-462.

Abdulla, N., Al-Ayyoub, M. and Al-Kabi, M. (2013), "An extended analytical study of Arabic sentiments", *International Journal of Big Data Intelligence*, Vol. 1 No. 1, pp. 103-113.

Abdulla, N., Majdalawi, R., Mohammed, S., Al-Ayyoub, M. and Al-Kabi, M. (2014), "Automatic lexicon construction for Arabic sentiment analysis", *Proceedings of the Conference on the Future Internet of Things and Cloud (FiCloud), International Conference, IEEE, Barcelona*, pp. 547-552.

Aggarwal, C.C. and Zhai, C.X. (2012), *Mining Text Data*, Springer-Verlag, Berlin Heidelberg.

Ahmed, N., Shehab, M., Al-Ayyoub, M. and Hmeidi, I. (2015), "Scalable multi-label Arabic text classification", *The International Conference on Information and Communication Systems (ICICS 2015)*, Jordan.

Al-Harbi, S., Almuhareb, A., Al-Thubaity, A., Khorsheed, M.S. and Al-Rajeh, A. (2008), "Automatic Arabic text classification", *Proceedings of The 9th International Conference on the Statistical Analysis of Textual Data, Lyon*.

Al-Kabi, M.A., Abdulla, N. and Al-Ayyoub, M. (2013), "An analytical study of Arabic sentiments: Maktoob case study", *Internet Technology and Secured Transactions (ICITST), 8th International Conference for IEEE*, London, pp. 89-94.

Al Shboul, B., Al-Ayyoub, M. and Jararweh, Y. (2015), "Multi-way sentiment classification of Arabic", *Proceedings of the 6th International Conference on Information and Communication Systems (ICICS), IEEE, Amman*, pp. 206-211.

Alwajeeh, A., Al-Ayyoub, M. and Hmeidi, I. (2014), "On authorship authentication of Arabic articles", *Information and Communication Systems (ICICS), 5th International Conference on IEEE, Irbid*.

Alwedyan, J., Hadi, W., Salam, M. and Mansour, H. (2011), "Categorize Arabic data sets using multi-class classification based on association rule approach", *Proceedings of the International Conference on Intelligent Semantic Web-Services and Applications*, *ACM*, np Paper #18.

Cheng, N., Chandramouli, R. and Subbalakshmi, K. (2011), "Author gender identification from text", *Digital Investigation*, Vol. 8 No. 1, pp. 78-88.

Document classification (2014), available at: http://en.wikipedia.org/wiki/Document_classification (accessed 28 June 2014).

Duwairi, R. and Al-Zubaidi, R. (2011), "A hierarchical k-nn classifier for textual data", *The International Arab Journal of Information Technology*, Vol. 8 No. 3, pp. 251-259.

Efstathios, S. (2009), "A survey of modern authorship attribution methods", *Journal of the American Society for information Science and Technology*, Vol. 60 No. 3, pp. 538-556.

Elhawary, M. and Elfeky, M. (2010), "Mining Arabic business reviews", *Proceedings of the IEEE International Conference on Data Mining Workshops (ICDMW), IEEE*, *Sydney*.

Ezzat, H., Ezzat, S., El-Beltagy, S. and Ghanem, M. (2012), "TopicAnalyzer: a system for unsupervised multi-label Arabic topic categorization", *Innovations in Information Technology (IIT), 2012 International Conference IEEE*, Abu Dhabi.

Feng, S. (2010), "Transductive multi-instance multi-label learning algorithm with application to automatic image annotation", *Expert Systems with Applications*, Vol. 37 No. 1, pp. 661-670.

Hotho, A., Nürnberger, A. and Paass, G. (2005), "A brief survey of text mining", *LDV Forum*, Vol. 20 No. 1, pp. 19-62.

Joorabchi, A. and Mahdi, A. (2010), "An unsupervised approach to automatic classification of scientific literature utilizing bibliographic metadata", *Journal of Information Science*, Vol. 37 No. 5, pp. 499-514.

Kim, K., Chung, B.S., Choi, Y., Lee, S.J., Jung, J.Y. and Park, J. (2014), *Language Independent Semantic Kernels for Short-Text Classification*, Elsevier, Amsterdam, Vol. 41, pp. 735-743.

Lau, R., Lai, C., Bruza, P. and Wang, K. (2011), "Pseudo labeling for scalable semi-supervised learning of domain-specific sentiment lexicons", *20th ACM Conference on Information and Knowledge Management*, Glasgow.

Liu, M.C. and Chen, J.H. (2015), "A multi-label classification based approach for sentiment classification", *Expert Systems with Applications*, Vol. 42 No. 3, pp. 1083-1093.

Maite, T., Brooke, J., Tofiloski, M., Voll, K. and Stede, M. (2011), "Lexicon based methods for sentiment analysis", *Computational Linguistics*, Vol. 2 No. 37, pp. 267-307.

Mendoza, M. (2012), "A new term-weighting scheme for Naïve Bayes text categorization", *International Journal of Web Information Systems*, Vol. 8 No. 1, pp. 55-72.

Musto, C., Semeraro, G. and Polignano, M. (2014), "A comparison of lexicon-based approaches for sentiment analysis of microblog posts", *Proceedings of the International Workshop on Information Filtering and Retrieval, DART14*, Pisa, 10 December, pp. 59-68.

Neumayer, R., Mayer, R. and Nørvåg, K. (2011), "Combination of feature selection methods for text categorisation", *Advances in Information Retrieval*, Springer, Berlin Heidelberg, pp. 763-766.

Read, J. (2010), "Scalable multi-label classification", Doctoral dissertation, University of Waikato, Hamilton.

Saad, M. (2010), "The impact of text preprocessing and term weighting on Arabic text classification", Master of science thesis, Computer Engineering Department, Islamic University-Gaza, Gaza.

Said, A., Wanas, M., Darwish, M. and Hegazy, H. (2009), "A study of text preprocessing tools for Arabic text categorization", *The Second International Conference on Arabic Language*, Cairo, pp. 230-236.

Tsoumakas, G. and Katakis, I. (2009), "Multi-label classification: an overview", *International Journal of Data Warehousing and Mining*, Vol. 3 No. 3, pp. 1-13.

Tsoumakas, G., Katakis, I. and Vlahavas, I. (2010), "Mining multi-label data", *Data Mining and Knowledge Discovery Handbook*, Springer, New York, NY, pp. 667-685.

Vogrinčič, S. and Bosnic´, Z. (2011), "Ontology based multi-label classification of economic articles", *Computer Science and Information Systems*, Vol. 8 No. 1, pp. 101-119.

Weichselbraun, A., Gindl, S. and Scha, A. (2013), "Extracting and grounding context-aware sentiment lexicons", *IEEE Intelligent Systems*, Vol. 28 No. 2, pp. 39-46.

Xu, X., Jiang, Y., Xue, X. and Zhou, Z. (2012), "Semi-supervised multi-instance multi-label learning for video annotation task", *Proceedings of the 20th ACM International Conference on Multimedia*, ACM, pp. 737-740.

Yamamoto, S. and Satoh, T. (2014), "Two phase estimation method for multi-classifying real life tweets", *International Journal of Web Information Systems*, Vol. 10 No. 4, pp. 378-393.

Zhang, M. and Zhou, Z. (2007), "ML-KNN: a lazy learning approach to multi-label learning", *Pattern Recognition*, Vol. 7 No. 40, pp. 2038-2048.

Corresponding author
Ismail Hmeidi can be contacted at: hmeidi@just.edu.jo