A PRELIMINARY STUDY OF PROSODY-BASED DETECTION OF QUESTIONS IN ARABIC SPEECH MONOLOGUES

Omair Khan, Wasfi G. Al-Khatib* and Lahouari Cheded

King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia 31261

December 2010

The Arabian Journal for Science and Engineering, Volume 35, Number 2C

--

The following is a preliminary study of prosody-based detection of questions in Arabic speech monologues. The study aims to address the challenge of identifying questions in Arabic speech by analyzing prosodic features. The research focuses on the detection of questions using a preliminary study of prosody-based methods.

The study is based on a preliminary framework that employs a combination of various computational techniques to detect questions in Arabic speech. The methods used include support vector machines (SVMs), decision trees, and Bayesian networks. The results show promising accuracy in question detection, especially when using a combination of prosodic features.

The study concludes with a discussion on the potential applications of the developed methods in various domains, such as natural language processing, speech recognition, and Arabic language technology.

*Corresponding Author:
E-mail: wasfi@kfupm.edu.sa

Paper Received September 25, 2010; Paper Revised December 22, 2010; Paper Accepted December 29, 2010
ABSTRACT

Prosody features have been widely used in many speech-related applications, including speaker and word recognition, emotion and accent identification, topic and sentence segmentation, and text-to-speech applications. Languages other than Arabic have received a lot of attention in this regard. An important application of prosodic features which is investigated here is that of identifying question sentences in Arabic monologue lectures. To our best knowledge, this is the first attempt at addressing question detection from spoken lectures in any language. To this end, we developed a small corpus made of 1028 utterances that were extracted from 15 Arabic spoken lectures. We approach this problem by first segmenting the continuous speech (recorded lectures) into sentences using both intensity and duration features. Prosodic features are, then, extracted from each sentence. These features are used as input to four different classifiers to classify each sentence into either a question or a non-question sentence.

Our results suggest that questions are cued by more than one type of prosodic features in spontaneous Arabic speech. We classified questions with an accuracy of 77.43%. A feature-specific analysis further reveals that energy and fundamental frequency ($F0$) features are mainly responsible for discriminating between question and non-question sentences. In terms of classification, we found that a Bayes Network performs better than support vector machines, multi-layer perceptron neural networks, or decision trees on our dataset. Removal of correlated features through Correlation-based Feature Selection produced more efficient and accurate results than the complete feature set.

Key words: question detection, prosodic analysis, audio monologues, Arabic lectures, learning algorithms
A PRELIMINARY STUDY OF PROSODY-BASED DETECTION OF QUESTIONS IN ARABIC SPEECH MONOLOGUES

1. INTRODUCTION

There has been a huge increase in the amount of data being generated and stored as computers and Internet usage are increasingly becoming part of our everyday life. This data is present in text as well as in audio and video formats. Before the arrival of digital subscriber line (DSL) and broadband connections, people used to limit themselves to text and image data. Now, with broader bandwidths available, there has been an increase in audio and video content on the Internet. Online video and audio sites such as YouTube, Google video, etc., are among the most visited sites on the Internet. Audio and video content is widely shared through file-sharing peer-to-peer networks. Multimedia content now constitutes the bulk of the Internet traffic in the form of IP-telephony, video and audio conferencing, Internet radio stations, music stores, lecture sites, etc.

Audio data form an important segment of the multimedia data present on, and transmitted through, the Internet. It includes online lectures, music, radio programs, podcasts, news, Text-To-Speech (TTS) systems that translate textual websites into audio for visually-impaired people, etc. This content is present in downloadable as well as in streaming formats. Prominent non-Western languages on the web include Chinese, Japanese, Arabic, Korean, and Turkish. As of 2010, Arabic is the seventh most widely used language on the Internet.

In the Arabic language, huge repositories of lectures on cultural topics, e.g., Islamic faith, are available. These stores are using various indexing schemes. An interesting such scheme is the semantic one where all lectures can be either manually or automatically transcribed using speech recognition techniques. However, neither of these two approaches is viable at this point. Manual transcription is a very laborious process requiring sizeable manpower and lengthy time periods, and consequently entails a high financial cost. In addition, Arabic speech recognition has not matured enough to achieve high accuracy rates on such unrestricted domains, as compared to English. Therefore, one needs to look for other reasonable semantic content that can be automatically extracted. One such semantic content is that of questions posed within these lectures, whether by the speaker or the audience. In many such lectures, a question-and-answer session usually follows the main speech. These questions form a sizeable and very useful knowledge-base on various issues, which users may utilize for different purposes. For example, one can look for the opinions of different scholars on a certain contemporary issue. Normally, the lectures are monologues, i.e., questions and answers are spoken by the same person. Therefore, the problem can be approached from a prosodic point of view, where the intonation of the voice is used for cues to distinguish a question from an answer. We employ prosody to identify question segments in the speech in order to achieve high success rates. These segments can be later utilized by speech processing techniques in order to recognize important words and carry out the semantic indexing based on them.

In this paper, we report on our findings with regard to identifying and employing prosodic features in detecting question and non-question segments in a monologue. This study is an extension of our earlier investigation reported in [1]. The paper is organized as follows: Section 2 gives a literature survey on this topic; after that, Section 3 presents the set of features considered in this study; this is followed by Section 4 describing the feature extraction process; the classification technique employed and the experiment design are shown in Sections 5 and 6, respectively; Section 7 presents the results and their discussion; And finally, our conclusions and future work are included in Section 8.

2. LITERATURE SURVEY

Prosody is widely used in many speech-related applications. Researchers have used prosodic features for dialog act detection [2–4], topic segmentation [5,6], sentence boundary detection [7–11], emotion detection [12–14], sentence disambiguation [15], detection of speech disfluencies [16], removal of mistakes from the output of speech recognizers [17,18], etc.

In the above approaches, researchers have used various types of classifiers such as decision trees, support vector machines (SVM), artificial neural networks such as MLP-NN or RBF, and simpler classifiers such as k-NN, linear classifiers like LDC, Boosting algorithms like AdaBoost, etc. Some authors such as Lee and Narayanan [14] have combined the output from multiple classifiers to give a single result.

Most researchers have used decision trees for prosody feature selection and classification tasks. Their advantage

---

1 http://www.internetworldstats.com/stats7.htm
2 Examples include http://www.islamweb.net and http://www.islamway.com
is that analysis of feature usage is easier because we can see exactly the usefulness of each feature for classification purposes. Mostly, CART- style decision trees have been used by researchers in the field.

Prosodic models have been discussed extensively in the field of Text-To-Speech (TTS) systems. In this field, various models for different languages can be found in [19] for English, [20] for Mandarin Chinese, and [21] for German. However, these are models to generate prosody from text to speech. Nevertheless, they give us important information about how prosody is used in different languages.

In the English and German languages, it has been shown that yes/no questions have a final rising, while question-word questions have a final falling tone [22]. It was found that besides utterance final position, question intonation has an effect on the pitch of the whole utterance, especially when dealing with question-word questions.

For the Chinese language, the spectral balance of the final syllable is a more useful prosodic feature than the pitch at the end of the utterance, which is the most useful prosodic feature of the English language [23]. In Swedish, question intonation has been primarily described as marked by a raised topline and a widened F0 range on the focal accent [24]. Studies in the Danish language have found that, in addition to the final rising tone, other features such as the onset F0 and the overall pitch range also help distinguish a question from a statement [25].

Shriberg et al. [3] report that for the English language, questions have higher F0 means and higher end gradients than statements. Questions are shorter in duration, have a lower percentage of frames in continuous regions, and have less variability in the speaking rate than do statements. Declarative and yes-no questions typically have a final F0 rise. Wh-questions often fall in F0 as do statements. Wh-questions show a higher average F0 than statements.

With respect to the problem of question detection in speech, Yuan et al. were among the earliest researchers to report their findings with respect to question detection in Chinese and English conversational speech [26]. The data used in their study consist of the Mandarin Chinese corpus of telephone speech (LDC96S24) and its transcripts (LDC96T16). Their decision tree classifier, C4.5, achieved an error rate of 14.9% with respect to a 50% chance-level rate. Quang et al. detected questions from French meeting recordings, achieving an accuracy rate of 75% [27]. Their data consists of recordings of 13 meetings involving 3 to 5 speakers. Their durations range from 15 to 60 minutes. They were classified into 3 categories: recruitment interviews, project discussions in a research team, and brainstorming-style talking. Another early work in this regard is that of Liscombe et al. [28]. This research was carried out on a corpus of spoken tutor dialogues for the development of intelligent tutoring spoken dialogue systems. Prosodic, lexical, syntactic, and student- and task-dependent information from student turns were extracted. Using decision trees, they were able to predict student question-bearing turns at a rate of 79.7%. Another related work by Ananthakrishnan et al. was able to achieve 71.9% accuracy for the binary (yes/no) question vs non-question classification for identifying question-bearing turns in spontaneous multi-party speech using lexical and prosodic evidence [29]. They used the ICSI Meeting Corpus data, consisting of 72 hours of speech collected from 75 research meetings by various groups within the institute [30]. Boakye et al. also used the ICSI Meeting Corpus data to investigate the automatic detection of English questions in meetings using lexico-syntactic, turn-related, and pitch-related features [31]. They report that lexico-syntactic features are most useful, with turn- and pitch-related features providing a combined and complementary information. The best precision and recall values achieved were 76.23% and 66.84%, respectively. Kathol et al. employ lexical, speaker, and dialog act tag information to detect question-answer pairs in meetings [32] without using any prosodic information. Their data set consists of 1435 utterances extracted from 14 meetings.

It is worth mentioning here that our domain, viz, spoken monologues, is different from all of the aforementioned research work in question detection, regardless of the language. Arabic speech data available from the Linguistic Data Consortium1 (LDC) or from the European Language Resources Association2 (ELRA) consist of telephone and microphone conversations, telephone and microphone-based read and spontaneous speech, and broadcast news, none of which could be used in our study here.

3. FEATURE SET

In order to employ prosody, one needs to determine the set of features that must be examined in order to determine whether these “discriminating” features significantly contribute to the success of the classification process. Based on the features that have been investigated in the literature review, we decided to study the following ones:

1. Pitch Features: Pitch features are considered the most important set of features in determining whether a particular sentence is a question sentence. Most of the pitch features are not straightforward to extract because of the ways in which pitch is used by different speakers and in different speaking contexts. We use

---

1 http://www.ldc.upenn.edu/
2 http://www.elra.info/
pitch to recognize tone but the problem is that pitch is never fully measurable. Therefore, we always have missing data values. There are also problems of discontinuities in pitch tracking such as doubling errors and pitch halving [8]. These errors are estimated by a lognormal tied mixture model of \(F_0\), which computes a set of speaker-specific pitch range parameters [33]. Then, in many approaches, these values are processed through median filtering, linear stylization before being used for feature extraction. This filtering and linear stylization of \(F_0\) helps us in the robust extraction of \(F_0\) features such as the value of \(F_0\) slope at a particular point, the maximum or minimum stylized \(F_0\) within a region, the average slope, the speaker specific values, etc. Pitch detection algorithms can be broadly divided into two types: those that estimate pitch in the time domain and those that do so in the frequency domain. Some of the pitch detection algorithms can be found in Yin [34], Praat [35], Momel [36], and ESPS [37]. Praat and ESPS are the most widely used tools in this field.

2. **Energy Features:** Several studies have proposed different ways to measure the energy of voice signals. Some authors claim that the spectral balance approach, which involves measuring the energy distribution over the frequency spectrum, is vital for detecting stress and accent [38]. Tsao mentioned that question intonation in Chinese is ‘a matter of stress’ [26].

3. **Duration Features:** The term “duration” refers to the length of a particular constituent of speech. Duration is widely regarded to be an important prosodic cue for the detection of questions. Studies have revealed that the utterance’s final syllable is usually longer in question intonation than in statement intonation, whereas the other syllables in question intonation tend to be shorter [26]. Therefore, Yuan and Jurafsky have extracted three duration features, namely, the duration of final syllable, the average duration of other syllables, and the length of the whole utterance.

4. **Pause Features:** Pause features are very important for dividing an utterance into sentences. More often than not, the signal energy is used to calculate the pause features but some authors have used other measures, e.g., in [9] they use perceptual loudness extracted from audio files encoded using MPEG-1.

5. **Speaking Rate Features:** Considerable improvement in the recognition can be made simply by taking speaking rate into account [39]. Speaking rate is defined differently by different people. The two most common definitions are word-per-minute (WPM) and syllable-per-second (SPS) [40]. However, since we don’t have any information about either of these two entities, we cannot, therefore, use either of them as a measure.

### 4. FEATURE EXTRACTION

Before extracting features, we segment the speech into separate sentences using pause as a criterion. For detecting pauses, we use intensity and energy in the sound signal. If during a certain signal duration, the intensity of the signal is below a certain minimum intensity used as a threshold, then that duration is considered a pause. Currently, the default value for minimum duration is 0.6 seconds and 59 dB for maximum intensity. We have marked the sentence interval at the boundaries of the pause with a time margin of 0.1 second at both ends of the sentence segments. For speech segmentation and feature extraction, we use a speech phonetics application software called Praat [35]. For segmenting the speech, we have modified the program written by M. Lennens\(^1\).

After segmentation, we identify questions in the speech and label each question according to its type, viz yes/no, wh, rhetorical, or embedded. Non-question sentences are labeled automatically by the segmentation program. These sentences are then fed into our feature extractor that extracts the relevant features from those segments. We have modified the program called ‘Prosodic Feature Extraction Tool For Praat’ by Huang et al. [41]. This tool requires vowel, syllable, and word information of the audio signal in order to extract features. Since we do not have this information, we had to modify this tool to suit our needs.

One of the restrictions of using this tool is that there should only be one speaker in one audio file. In Praat, pitch and energy are calculated by the frame. The default length of each frame is 0.01 second or 10 milliseconds. The start, end, and duration of objects in Praat are measured by the index or the number of frames in the waveform.

For detecting pitch, Praat uses a method described in [35]. This method is more accurate, noise-resistant, and robust than methods based on cepstrum or combs, or the original autocorrelation methods. The reason why other methods were invented was the failure to recognize the fact that if one wants to estimate a signal’s short-term autocorrelation function, \(r_{xx}(\tau)\), on the basis of a windowed signal, \(r_{yw}(\tau)\), one should divide the autocorrelation function of the windowed signal by the autocorrelation function of the window, \(r_{ww}(\tau)\):

\[
r_{xx}(\tau) = r_{yw}(\tau) / r_{ww}(\tau)
\]  

---

\(^1\)http://www.helsinki.fi/~lennes/praat-scripts/
Due to the lack of word information, it was only possible to use $F0$ and energy as our word features. Pattern features were also used. Each sentence was divided into three separate segments: the starting 20 frames, the ending 20 frames, and the middle portion of the utterance. The energy and pattern features, $F0$, as well as their linear and logarithmic ratios and differences were all computed for each of these segments. Overall, 144 features were extracted. These 144 features were then reduced to 123 numeric features after removing basic symbolic pattern features. Basic pattern features had information in the nominal form and that information was then used to calculate the derived pattern features. Details of the number of features in each feature set are given in Table 1.

Table 1. Sizes of Feature Sets

<table>
<thead>
<tr>
<th>Feature Set</th>
<th>No. of Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full</td>
<td>123</td>
</tr>
<tr>
<td>$F0$-only</td>
<td>55</td>
</tr>
<tr>
<td>Energy-only</td>
<td>38</td>
</tr>
<tr>
<td>Patterns-only</td>
<td>31</td>
</tr>
<tr>
<td>$F0$-Energy</td>
<td>92</td>
</tr>
<tr>
<td>$F0$-Pattern</td>
<td>77</td>
</tr>
<tr>
<td>Energy-Pattern</td>
<td>64</td>
</tr>
</tbody>
</table>

- **$F0$ Features:** The mean and variance of the logarithmic $F0$ values are used to normalize $F0$ features and calculate the overall baseline and topline pitch of the speech. Pitch contour is stylized and several $F0$ values are computed from it. $F0$ features are mainly about the pitch range of a sentence. They include the minimum, maximum, mean, and the last $F0$ values of a specific region such as starting, middle, and ending segments of the sentence. These features are also normalized by the baseline $F0$ values, the topline $F0$ values, and the pitch range using linear difference and log difference. They also include features which compare the pitch ranges of the starting, ending, and middle segments of the sentence.

- **Energy Features:** The energy features are based on the intensity contour produced by Praat. Similar to $F0$ features, range features are also calculated for energy using various normalization methods.

- **Pattern Features:** These are the maximum, minimum, average pitch, and energy slopes in the sentences and the ratios of these values in the starting, middle, and ending segment of the sentence. Pitch and energy slope features indicate rising, falling, and unvoiced patterns of the stylized pitch values or the intensity contour in the sentence. We included the slope difference and dynamic patterns (i.e., falling, rising, and unvoiced) across the three segments of the sentence.

Raw pitch values were calculated using Praat’s autocorrelation method. These values were then smoothed out and the voiced/unvoiced regions were determined and stored in TextGrid. Praat’s pitch stylization function was used to stylize raw pitch values over voiced regions. Pitch slope values were then generated from stylized pitch contours.

Since there was no intensity stylization function in Praat, intensity values were stylized by the pitch stylization function. Stylization was performed over the entire intensity contour, whereas in the case of pitch, this stylization was applied only to voiced regions.

### 5. Classification

The features that have been identified in the previous section are then used for the classification task. However, using the complete set of features increases the computational cost of the classification as well as adds noise to the data set [3]. For a fast classification process, feature selection techniques are used to reduce the raw feature set size since removing redundant features will improve both the definition of a class and the interpretation of classification results.

The best way to select a feature subset is to do it manually using both an understanding of the application domain and the nature of the features used. However, there are many automatic methods for carrying out this task. These methods involve searching the right set of features using some feature utility criteria. If the feature set is non-trivial, the number of distinct feature sets grows exponentially. If the performance of the selected feature set is evaluated on the classification task by learning algorithms, this makes the task of appropriately selecting a feature subset computationally expensive.

We have manually generated subsets of the complete feature set and compared them with each other using decision trees. We found that $F0$ and energy features are the most important feature types for question identification. Overall, there is a high degree of correlation between these features such that the absence of a particular feature type is compensated for by the presence of other features [1].

As in many other applications, the relevant feature set is not known a priori. Therefore, it was necessary to collect as many features as possible for domain representation. Feature selection schemes used here are explained in
Section 5.1.

In order to compare the effectiveness of the feature selection schemes used here, the feature sets chosen by each technique were tested with four learning algorithms – a decision tree learner (C4.5 release 8), Bayesian Network, Support Vector Machines (SVM), and Multi-Layer Perceptron Neural Networks (MLP-NN). These algorithms were chosen because they represent quite different approaches to learning and they are relatively fast, with the exception of MLP-NN. In addition, most question detection research was carried out using decision trees for classification. With the availability of other classification techniques, it was therefore natural to investigate the viability of these alternative techniques and harness their power to achieve a better classification performance than had previously been obtained. These learning algorithms and their parameter settings are briefly explained in Section 5.2.

5.1. Feature Subset Selection Schemes

Feature reduction strategies hinge on two feature selection types: unsupervised and supervised. Principal Component Analysis (PCA) and Locally Linear Embedding, are unsupervised feature extraction algorithms, where the obtained lower dimensions are not necessarily subsets of the original coordinates. On the other hand, the supervised feature selection strategy selects subsets of the existing features [42]. Relevant and important subsets of features are selected depending on the specific supervised feature selection technique used. For an effective selection, the selection criterion is combined with a search strategy such as forward selection, backward elimination, bidirectional search, etc. to search through the feature space.

For the purpose of our experiment, we have chosen one optimal unsupervised feature selection technique, which is the PCA algorithm, and one optimal supervised feature selection technique, which is the correlation-based feature selection (CFS) scheme [43]. In our experiment, we have used PCA with eigenvectors that reflect 95% of the variance in the data. This has narrowed down the number of features from 123 to 25 and, by so doing, actually improved the performance of classification.

As for our chosen optimal supervised correlation-based feature selection scheme (CFS), it analyses subsets of features instead of single ones. Its heuristic is given by Equation 2, which assigns high scores to subsets that are highly correlated with the class and have low inter-correlation.

\[
M_s = \frac{k \bar{r}_{cf}}{\sqrt{k + k(k - 1)r_{ff}}}
\]  

(2)

Here, \(M_s\) is the heuristic “merit” of a feature subset \(S\) containing \(k\) features, \(\bar{r}_{cf}\) is the average feature-class correlation, and \(\bar{r}_{ff}\) is the average feature-feature inter-correlation [43]. The feature-feature correlation in Equation (2) can be estimated by the symmetrical uncertainty, \(SU\), given in Equation (3) after CFS discretizes the numeric features using the technique proposed by Fayyad and Irani [44].

\[
SU = 2\left[\frac{H(X) + H(Y) - H(X, Y)}{H(X) + H(Y)}\right]
\]

(3)

\(H(X), H(Y),\) and \(H(X, Y)\) represent the entropy of features \(X\), \(Y\) and their combined entropy, respectively.

After computing a correlation matrix, CFS applies a heuristic search strategy to find a good subset of features according to Equation (2) [45]. The most commonly-used search strategies in feature space are forward selection and backward elimination.

Forward selection starts with an empty set of features and progressively adds relevant features to it. Backward elimination starts with all the available features in the set and progressively eliminates some from it.

There are other sophisticated search strategies that are used to search for the right subset of features in the data set. These include genetic search, bidirectional search, beam search, etc. However, these strategies are more computationally expensive and their results are not significantly better than those of the forward or backward selection methods [46]. For these reasons, we have chosen the forward selection as the search method for our feature subset selection process. By applying this technique, we significantly reduced our features from 123 to 15 while appreciably increasing the classification performance.

5.2. Learning Algorithms

Some studies have shown that no single learning approach is clearly superior to all of the others in all cases and for all of the problems it addresses. In fact, different learning algorithms often produce similar results for the same problem. The choice of the learning scheme depends on the data itself [46]. We have compared four classifiers for the task of classifying three data sets. We have chosen classifiers from different families of learning algorithms, such
as statistical classifiers (Decision Trees), classifiers that perform comparatively better with scarce data (Support Vector Machines), neural networks (Multi-Layer Perceptrons), and basic learning algorithm (Bayes Networks).

5.2.1. Decision trees

Decision trees are attractive in classification due to the fact that, in contrast to neural networks, they represent rules by which humans can interpret and, hence, know the reasons for a particular classification of data. Two well-known programs for constructing decision trees are C4.5 [47] and CART (Classification and Regression Tree) [48]. We have used the C4.5 decision tree for classification. In our experiment, we used a confidence factor value of 0.25. The confidence factor is used for pruning purposes. The minimum number of instances per leaf is 2. Three folds of data are used for reduced-error pruning. One fold is used for pruning, the rest for growing the tree.

5.2.2. Multi-layer perceptron neural networks (MLP-NN)

Multi-layer perceptron neural networks (MLP-NNs) are supervised feedforward neural networks trained with the standard back propagation algorithm that relies on the use of a desired response [49]. With one or two hidden layers, they can approximate virtually any input-output map and have been widely used for pattern classification. They have been shown to approximate the performance of optimal statistical classifiers in difficult problems. Most neural network applications involve MLP-NNs.

In our experiment, all the nodes in the network are of the sigmoid type. The learning rate is decreased by dividing the starting learning rate by the epoch number. This helps to stop the network from diverging from the target output, as well as improve the general performance. The number of nodes in the hidden layer, \( N_H \), is calculated by the heuristic

\[
N_H = \left( N_{\text{features}} + N_{\text{classes}} \right) / 2
\]

with a learning rate of 0.3 and 500 epochs used for training.

5.2.3. Support vector machines (SVM)

These are supervised learning methods relying on kernel-based algorithms designed to maximize a particular mathematical function, given a set of labeled training data. For classification purposes, SVMs operate by finding a hyper surface in the space of all possible inputs. This hyper surface will attempt to split the positive examples from the negative ones [50].

In our experiment, we use an implementation of John Platt’s sequential minimal optimization algorithm [51] to train a support vector classifier. This implementation globally replaces all missing values and transforms nominal attributes into binary ones. It also normalizes all attributes by default, in which case, the coefficients in the output are based on the normalized data and not on the original data as this is important for the interpretation of the classifier’s output. An SVM classifier with a homogeneous polynomial kernel type is used in our experiment:

\[
K(x, y) = <x, y>^P
\]

with \( P = 1 \).

5.2.4. Bayes Net

Let \( U = \{x_1, x_2, \ldots, x_n\} \), \( n \geq 1 \) be a set of variables. A Bayesian network \( B \) over a set of variables \( U \) is a network structure \( B_S \), which is a directed acyclic graph (DAG) over \( U \) and a set of probability tables \( B_P = \{p(u | pa(u)) : u \in U\} \) where \( pa(u) \) is the set of parents of \( u \) in \( B_S \). A Bayesian network computes a conditional probability distribution \( P(U) = \prod_{u \in U} p(u | pa(u)) \) [46].

In our experiment, a simple estimator is used to estimate the conditional probability tables of a Bayes network once the structure has been learned. It estimates probabilities directly from data. We use a hill climbing algorithm to search network structures. Initially, the network operates as a Naive Bayes Network, that is, a network with an arrow from the classifier node to each of the other nodes. After a network structure is learned, a Markov Blanket correction is applied to the network structure. This ensures that all nodes in the network are part of the Markov blanket of the classifier node. The Bayes score type is used to determine the quality of a network structure.

6. EXPERIMENT DESIGN

First and foremost, it is vitally important to point out at this juncture that, to our best knowledge, this research work is the first of its kind regarding the Arabic language. As such, we had to first laboriously develop our own speech dataset as no audio/speech database was publicly available for benchmarking purposes. In building our own
corpus, we managed to include 15 Arabic audio lectures by three different male speakers with a sample rate of 8 KHz and having mono channel. Since none of the speakers of those lectures is a female, the gender distribution of our data is clearly biased in our study. Any extension of the data corpus has to address this issue and acquire lectures from female speakers so as to remove this bias.

We had in total 12571 sentences out of which only 514 could be classified as one of the three question types that were considered in this study. These lectures had 152 yes/no or wh questions, 24 rhetorical questions, and 338 embedded questions. In order to keep an equal a priori probability of question and non-question classes during the training stage, we had to downsample the non-questions to 514. Therefore, at the end, we had 514 questions and 514 non-question sentences. A total of 144 features were extracted from each of these 1028 sentences, 123 of which were numeric. Although we had transcribed three different types of questions in our corpus, we could not, however, downsample the data any further because of the small size of our dataset.

The performance of a particular learning scheme with the training data set is not a true reflection of its performance on an unseen (or validation) data set. When a large amount of data is available, a large data set is used for both validation and testing. However, in the event that the data is scarce, as is the case in this study, cross-validation is used. In cross-validation, a number of folds, \( n \), needs to be specified. The data set is randomly reordered and then uniformly split into \( n \) folds of equal size. Every iteration uses one fold for testing and the other \( (n - 1) \) folds are used for training the classifier. The test results are then collected and averaged over all folds. This improves the accuracy of the cross-validation estimate. The folds can be purely random or be slightly modified from one run to the next so as to create the same class distribution in each fold as that in the complete dataset.

In this work, a single ten-fold cross-validation scheme was used. We needed to use cross validation for both parameter and error estimation. We applied two attribute selection techniques to our data set. This yielded three separate data sets: a first data set containing all of the 123 numeric features, a second data set with a reduced number of 15 features obtained with the Forward Selection method using the Correlation-Based Feature selection scheme, and finally, a third data set with a reduced number of 25 features obtained with the Principal Component Analysis method.

Ten runs of experiments were carried out both with and without feature selection on all of the classification schemes. The experimentation environment used in this work is WEKA [46]. The experiment consists of the following tasks:

1. The percentage of correct classification results from 10 ten-fold cross-validation runs of these experiments were computed and averaged.
2. For each training/testing set split, the set dimensionality was reduced by attribute selection methods before being passed onto the learning algorithm used.
3. Attribute selection was performed using the training data and each learning scheme was tested on the test data using the selected features.
4. The statistics, such as the number of attributes selected and the size of the decision tree used, were all recorded.

In this experiment, the dependent variables are the accuracy, training time, tree size, and feature set sizes, whereas the independent variables are the attribute selection schemes and learning algorithms.

7. RESULTS AND DISCUSSION

In this section, the performances of the various learning algorithms used here for the task of question/ non-question identification, namely SVM, MLP-NN, and Bayes Net, are compared to each other using some criteria based on classification accuracy and time taken by each, as well as benchmarked against the performance of the Decision Tree classifier reported in our earlier work [1].

The first task of assessing and comparing the classification accuracy of the 4 learning algorithms entails a number of tests based on 6 different measures, as shown below. The accuracy was calculated using a Percent Correct measure with a confidence level of 0.05 (for a two-tailed distribution).

Table 2 shows the results of our accuracy comparison.
Table 2. Results for Percent Correct

<table>
<thead>
<tr>
<th>Dataset</th>
<th>(C4.5)</th>
<th>(Bayes Net)</th>
<th>(SVM)</th>
<th>(MLP-NN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFS</td>
<td>75.54</td>
<td>77.43</td>
<td>71.63</td>
<td>76.70</td>
</tr>
<tr>
<td>PCA</td>
<td>65.74</td>
<td>67.94</td>
<td>74.01</td>
<td>72.39</td>
</tr>
<tr>
<td>Full</td>
<td>75.48</td>
<td>77.18</td>
<td>75.21</td>
<td>76.15</td>
</tr>
</tbody>
</table>

○, ● statistically significant improvement or degradation, respectively

We can see from Table 2 that, overall, the Bayes Net outperforms all other classifiers, except in the case of the PCA dataset, where it is itself outperformed by both SVM and MLP-NN. With the Bayes Net, both CFS and Full dataset yield a similar and good accuracy, while at the same time, outperforming the PCA. For MLP-NN, CFS and Full dataset perform better than PCA. Overall, it can be concluded that of the 3 datasets used, the best one is the Full dataset since it performs consistently well with all three classifiers. Next, comes the CFS dataset which gives good results with all classifiers except with SVM. The PCA dataset though gives good results only with the SVM classifier.

The kappa statistic, shown in Table 3, measures the agreement between the predicted class and the true one, with the value of 1.0 indicating complete agreement.

Table 3. Kappa Measure

<table>
<thead>
<tr>
<th>Dataset</th>
<th>(C4.5)</th>
<th>(Bayes Net)</th>
<th>(SVM)</th>
<th>(MLP-NN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFS</td>
<td>0.51</td>
<td>0.55</td>
<td>0.43</td>
<td>0.53</td>
</tr>
<tr>
<td>PCA</td>
<td>0.31</td>
<td>0.36</td>
<td>0.48</td>
<td>0.45</td>
</tr>
<tr>
<td>Full</td>
<td>0.51</td>
<td>0.54</td>
<td>0.50</td>
<td>0.52</td>
</tr>
</tbody>
</table>

○, ● statistically significant improvement or degradation, respectively

Here too, we can see that both SVM and MLP-NN outperform the Bayes Net on the PCA dataset only, but are both outperformed by it (Bayes Net) on the other two datasets (CFS and Full).

The True Positive (TP) rate is the ratio of examples classified as members of a particular class over the total number of examples which are true members of that class. It shows the proportion of the true members of that class that are captured by the classifier used. It is equivalent to the “Recall” scheme. Table 4 gives the comparison of the True Positive rates of the three attribute selection schemes used in all of the four learning algorithms studied here.

Table 4. True Positive Rate

<table>
<thead>
<tr>
<th>Dataset</th>
<th>(C4.5)</th>
<th>(Bayes Net)</th>
<th>(SVM)</th>
<th>(MLP-NN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFS</td>
<td>0.80</td>
<td>0.93</td>
<td>0.82</td>
<td>0.80</td>
</tr>
<tr>
<td>PCA</td>
<td>0.67</td>
<td>0.72</td>
<td>0.81</td>
<td>0.72</td>
</tr>
<tr>
<td>Full</td>
<td>0.77</td>
<td>0.83</td>
<td>0.81</td>
<td>0.76</td>
</tr>
</tbody>
</table>

○, ● statistically significant improvement or degradation, respectively

In Table 4, we can see that overall, the Bayes Net performs better than the other classifiers with all datasets, except for the PCA dataset. For this (PCA) dataset, as shown in Tables 2-4, the SVM remains the most consistently performing of the classifiers used.

The False Positive (FP) rate measures the ratio of examples which were incorrectly classified as members of a certain class X, over all examples which are not members of the class X. In essence, both the TP and FP ratios tend to complement each other and both yield a measure of classification robustness. Table 5 shows the values of the “False Positive” rate for the three attribute selection schemes used with all four learning algorithms.

Table 5. False Positive Rate

<table>
<thead>
<tr>
<th>Dataset</th>
<th>(C4.5)</th>
<th>(Bayes Net)</th>
<th>(SVM)</th>
<th>(MLP-NN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFS</td>
<td>0.29</td>
<td>0.38</td>
<td>0.39</td>
<td>0.26</td>
</tr>
<tr>
<td>PCA</td>
<td>0.35</td>
<td>0.36</td>
<td>0.33</td>
<td>0.27</td>
</tr>
<tr>
<td>Full</td>
<td>0.26</td>
<td>0.28</td>
<td>0.30</td>
<td>0.24</td>
</tr>
</tbody>
</table>

○, ● statistically significant improvement or degradation, respectively

We can see from Table 5 that, in the case of the CFS dataset, both the Bayes Net and SVM have significantly higher rates of False Positives than the other classifiers. However, the MLP-NN outperforms all other classifiers for all 3 datasets, in particular for the PCA dataset.
The “Precision” criterion measures the proportion of the examples of the true members of a particular class over those which were classified as members of this class. Table 6 gives the precision figures of the three attribute-selection schemes with all four learning algorithms used.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>(C4.5)</th>
<th>(Bayes Net)</th>
<th>(SVM)</th>
<th>(MLP-NN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFS</td>
<td>0.74</td>
<td>0.71</td>
<td>0.68●</td>
<td>0.76</td>
</tr>
<tr>
<td>PCA</td>
<td>0.66</td>
<td>0.67</td>
<td>0.71○</td>
<td>0.73○</td>
</tr>
<tr>
<td>Full</td>
<td>0.75</td>
<td>0.75</td>
<td>0.73</td>
<td>0.76</td>
</tr>
</tbody>
</table>

○, ● statistically significant improvement or degradation, respectively

Table 6 shows that overall, the MLP-NN has the best performance among all 4 classifiers and for all of the 3 datasets used. For the PCA dataset, the SVM and MLP-NN perform slightly better than the Decision Trees and Bayes Net, which have similar performances. With respect to this criterion, the SVM has been the least performing classifier for the CFS dataset.

The F-Measure is simply expressed by \( \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \), and gives a combined measure for both precision and recall. Table 7 shows all the relevant F-measure values for the three datasets and 4 classifiers used.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>(C4.5)</th>
<th>(Bayes Net)</th>
<th>(SVM)</th>
<th>(MLP-NN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFS</td>
<td>0.76</td>
<td>0.81○</td>
<td>0.74</td>
<td>0.77</td>
</tr>
<tr>
<td>PCA</td>
<td>0.66</td>
<td>0.69</td>
<td>0.76○</td>
<td>0.72○</td>
</tr>
<tr>
<td>Full</td>
<td>0.76</td>
<td>0.78</td>
<td>0.76</td>
<td>0.76</td>
</tr>
</tbody>
</table>

○, ● statistically significant improvement or degradation, respectively

From Table 7, we can see that the Bayes Net performs better than the other learning algorithms on all datasets except for the PCA dataset, on which the MLP-NN and SVM perform better than the other 2 learning algorithms.

These results indicate that for accuracy and a combined measure of precision and recall, the Bayes Net on the CFS dataset gives the best results.

The second task consists of comparing the 4 learning schemes used with 3 different datasets for their training time. Table 8 compares the time each attribute-selection scheme takes for classification.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>(C4.5)</th>
<th>(Bayes Net)</th>
<th>(SVM)</th>
<th>(MLP-NN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFS</td>
<td>0.25</td>
<td>0.09○</td>
<td>0.25</td>
<td>8.22●</td>
</tr>
<tr>
<td>PCA</td>
<td>0.97</td>
<td>0.22○</td>
<td>0.89</td>
<td>68.49●</td>
</tr>
<tr>
<td>Full</td>
<td>2.92</td>
<td>0.56○</td>
<td>3.75●</td>
<td>600.36●</td>
</tr>
</tbody>
</table>

○, ● statistically significant improvement or degradation, respectively

Here too, the Bayes Net outperforms all other classifiers for all 3 datasets used, whereas the MLP-NN comes out expectedly as the worst performing classifier for all 3 datasets. The remaining 2 classifiers (Decision Trees and SVM) show a pretty similar performance for all 3 datasets.

A receiver operating characteristic (ROC) curve is a plot of the sensitivity, or the true positive rate vs the false positive rate as the prediction threshold sweeps through all its possible values. An ROC of 1 indicates perfect prediction, i.e., all positive cases were sorted above all negative ones. However, an ROC of 0.5 is indicative of a random prediction, where there is no relationship between the predicted and the true classifications. The details of the ROC area comparison of the learning algorithms and feature selection schemes are given in Table 9.
Table 9. Area Under ROC

<table>
<thead>
<tr>
<th>Dataset</th>
<th>(C4.5)</th>
<th>(Bayes Net)</th>
<th>(SVM)</th>
<th>(MLP-NN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFS</td>
<td>0.81</td>
<td>0.88 •</td>
<td>0.72</td>
<td>0.86 •</td>
</tr>
<tr>
<td>PCA</td>
<td>0.66</td>
<td>0.74 •</td>
<td>0.74</td>
<td>0.79 •</td>
</tr>
<tr>
<td>Full</td>
<td>0.80</td>
<td>0.86 •</td>
<td>0.75</td>
<td>0.85 •</td>
</tr>
</tbody>
</table>

•: statistically significant improvement or degradation, respectively.

From Table 9, we can see that Bayes Net and MLP-NN perform better for all the datasets. Of all the 3 datasets, except for the SVM, which is the least performing of all of the classifiers used, the CFS dataset gives the best ROC area figures. Hence, based on the above 8 tests carried out to assess both the classification accuracy and training time of the 3 classifiers used here, we can safely say that, of the 3 classifiers studied here, the Bayes Net, coupled with the feature set reduction scheme of CFS, has shown the best performance in terms of both classification accuracy and training time.

8. CONCLUSION AND FUTURE WORK

In this paper, we reported on our preliminary study of question detection in Arabic monologues and presented our findings obtained from it. To our knowledge, this work is the first such attempt carried out in the domain of lectures in any language. In order to carry out our study, we developed a limited-size corpus of 1028 utterances extracted from 15 Arabic audio lectures. We used three different classifiers, viz SVM, MLP-NN, and Bayes Net, and benchmarked their performances against that of our previously-tested Decision Tree classifier with a Full data set. We compared their classification results based on the full feature set and 2 feature subsets selected using different feature reduction schemes, namely a completely statistical technique of feature selection (PCA), and a selection scheme that removes correlated and redundant features (CFS). We found out that the feature set reduced by CFS and classified by the Bayes Net gives the optimum performance in terms of accuracy, prediction, and training time. Although the results did not improve significantly (from 75.54% to 77.43%), we believe that this was due mainly to the limited data set we worked with.

Despite the limited-size corpus used, this work shows that prosody is a powerful information source and tool for the identification of questions in spontaneous speech. Several prosodic cues for questions were identified. These cues should be fully exploited in Arabic ASR systems for question identification.

Various combinations of prosodic features signal the question sentence in Arabic. F0 and Energy features are the most important feature types for question identification. Overall, there is a high degree of correlation between these features, in fact so much so that if one feature type is not present, other features can compensate for it without much loss incurred.

A number of potential areas for future work can be investigated. The single classification scheme studied here can be generalized to a multi-classification one whereby output from different classifiers can be combined or fused to improve classification performance. The development of a larger corpus, taking into account samples from female speech, would allow us to uncover and exploit distinguishing features of different types of questions. Since the ratio of question statements to non-question statements in lectures is very low, a minimum of 100 lectures may need to be processed in order to have a reasonable corpus. This research can also be extended to dialogues and conversations and to identifying other speech acts such as command, accordance, appreciation, appeal, anger, and other types of emotional expressions. Our experiments were conducted on automatically-segmented non-question sentences which do not give perfectly separated sentences. It would, therefore, be interesting to carry out a sensitivity analysis to find out to what extent the results of our experiments would be affected by a more accurate sentence segmentation. Another interesting area of research is the employment of prosodic features directly extracted from a sentence to determine the boundaries of this sentence.

With prosody-based research in the Arabic language still in its nascent state, when compared to its English language-based counterpart, and due to the consequent paucity of results in this area, we feel that this paper offers a research initiative through which our results can be compared with those obtained for English and other languages, and in so doing, uncover any possible similarities between a user’s speaking patterns across different languages. That may lead to building a common body of standard question identification techniques that are applicable across different languages. This will enrich the area of speech-based human-computer interface (HCI).

ACKNOWLEDGMENT

The authors would like to thank King Fahd University of Petroleum and Minerals (KFUPM) for supporting this research work under Project No. FT 070008.
REFERENCES


