Detecting Missing Content Queries in an SMS-Based HIV/AIDS FAQ Retrieval System

Edwin Thuma[1,2], Simon Rogers[1], and Iadh Ounis[1]

[1] School of Computing Science, University of Glasgow, G12 8QQ, UK
[2] Department of Computer Science, University of Botswana, P/BAG 0022, Botswana
thumae@dcs.gla.ac.uk,{simon.rogers,iadh.ounis}@glasgow.ac.uk

Abstract. Automated Frequently Asked Question (FAQ) answering systems use pre-stored sets of question-answer pairs as an information source to answer natural language questions posed by the users. The main problem with this kind of information source is that there is no guarantee that there will be a relevant question-answer pair for all user queries. In this paper, we propose to deploy a binary classifier in an existing SMS-Based HIV/AIDS FAQ Retrieval System to detect user queries that do not have the relevant question-answer pair in the FAQ document collection. Before deploying such a classifier, we first evaluate different feature sets for training in order to determine the sets of features that can build a model that yields the best classification accuracy. We carry out our evaluation using seven different feature sets generated from a query log before and after retrieval by the FAQ retrieval system. Our results suggest that, combining different feature sets markedly improves the classification accuracy.

Keywords: Frequently Asked Question, Missing Content Queries, Text Classification

1 Introduction

Mobile phones have emerged as the platform of choice for providing services such as banking [18], payment of utility bills [26] and learning (M-Learning) [1] in the developing world. This is because of their low cost and high penetration in the market [1, 7]. In order to take advantage of this high mobile phone penetration in Botswana, we have developed an SMS-Based HIV/AIDS FAQ retrieval system that can be queried by users to provide answers on HIV/AIDS related queries. The system uses, as its information source the full HIV/AIDS FAQ question-answer booklet provided by the Ministry of Health (MOH) in Botswana for its IPOLETSE[1] call centre. This FAQ question-answer booklet is made up of 205 question-answer pairs organised into eleven chapters of varying sizes. For example, there is a chapter on “Nutrition, Vitamins and HIV/AIDS” and a chapter on “Men and HIV/AIDS”. Below is an example of a question-answer pair entry that can be found in Chapter Eight, “Introduction to ARV Therapy”:

[1] http://www.hiv.gov.bw/content/ipoletse

Question : What is the importance of taking ARV therapy if there is no cure for AIDS?

Answer : Although ARV therapy is not a cure for AIDS, it enables you to live a longer and more productive life if you take it the right way. ARV therapy is just like treatment for chronic illnesses such as diabetes or high blood pressure.

For the remainder of this paper, we will refer to a question-answer pair as the FAQ document and the sets of all 205 FAQ documents as an FAQ document collection. The users' SMS messages will be referred to as queries.

Because of the constrained nature of mobile phone displays, our system does not return a ranked list of FAQ documents for each SMS query. Instead our system adopts an iterative interaction strategy that we proposed in [24]. However, we differ with our earlier strategy by returning the whole FAQ document to the user at each iteration and not the question part only. For example, for each SMS query sent by the user, the system ranks the FAQ documents in the FAQ document collection. The top ranked FAQ document is returned to the user. If the user is satisfied that this FAQ document matches the SMS query, the user respond with "YES" or remain idle and the interaction terminates. If the user is not satisfied, they reply with "NO", and the system then displays the next highest ranked FAQ document and the process is repeated until the user respond with "YES".

One key problem in this domain is that there is no guarantee that there will be a relevant FAQ document for all user [22, 23] queries. This may result in users iterating with our system for longer, wasting time and their SMS credit. It is for this reason that we propose to deploy a classifier that can detect those queries for which there are no relevant FAQ documents in the collection. In their earlier work, Yom-Tov et al. [25] defined Missing Content Queries (MCQs) as those queries for which there are no relevant documents in the collection and non-MCQs as those that have the relevant documents. In this work, we will use this definition by Yom-Tov et al. Detecting *MCQs* can be useful for both the user and the information supplier in an SMS Based HIV/AIDS FAQ Retrieval System. The user can be informed if the FAQ document collection does not contain the relevant FAQ document for the user query rather than returning irrelevant FAQ documents [16]. On the other–hand, the information supplier can note the kind of information need that is of interest to the user but not addressed by the current information source (FAQ document collection) [25]. Armed with this knowledge, the information supplier can update the FAQ document collection by adding or modifying FAQ documents entries.

Previous work on *MCQs* detection by Yom-Tov et al. [25] , Hogan et al. [13] and Leveling [17] relied on classification to detect queries that do not have relevant FAQ documents in the FAQ document collection. In their work, Yom-Tov et al. and Hogan et al. trained their classifiers using features generated by query difficulty estimators. Leveling [17] on the other-hand generated the features for training the classifier during the retrieval phase on the training data and examined the top five documents (e.g scores for top five documents).

In this work, we will follow the work in [13, 16, 17, 25] by tackling the detection of $MCQs$ in an HIV/AIDS FAQ retrieval systems as a binary classification problem. This paper attempts to contribute to the current state of the art in missing content detection for an SMS-Based FAQ retrieval system by first analysing and evaluating different feature sets in order to determine the best combination of features that can be used to build a model that would yield the highest classification accuracy. We will carry out a thorough evaluation using two different datasets. The first dataset is a collection of HIV/AIDS documents and a query log of HIV/AIDS related $MCQs$ and $non-MCQs$ collected in Botswana over a period of 3 months. In Section 3.2, we describe how we collected this query log. The other dataset is a collection of FAQ documents from the Forum for Information Retrieval Evaluation (FIRE)[2] English monolingual SMS-Based FAQ retrieval training data and the associated query log (SMS queries). The FIRE2012 dataset has 7251 FAQ documents in total. We use two different datasets in order to be able to make a general conclusion. Seven different feature sets will be created for these datasets. The first feature set (baseline) will be made up of the actual query strings of $non-MCQs$ and $MCQs$ as features for the training and testing instances. The second feature set will be made up of features deployed by Leveling [17] while the third feature set will be made up of the features deployed by Hogan et al. [13] and some additional query difficulty predictors. We then create four additional feature sets by combining the previous three feature sets in order to determine whether combining these feature sets would yield a better classification accuracy. Three different classifiers in WEKA [9], namely Naive Bayes [15], RandomForest [2] and S-SVC (Support Vector Classification) [4] will be trained and tested on this feature sets to evaluate their effectiveness in classifying $non-MCQs$ and $MCQs$. WEKA is an open-source data mining software. In this paper, we also investigate whether increasing the size of the training set would yield a better classification accuracy.

The rest of this paper is organised as follows: We survey related work in Section 2, followed by a description of our methodology in Section 3. In Section 4 we describe our experimental setting. We then present experimental results and evaluation in Section 5, followed by conclusions in Section 6.

2 Related Work

Earlier work on the detection of missing content queries ($MCQs$) was first introduced by Yom-Tov et al. [25] on their investigation of the applications of query difficulty estimation. In their experiment, they artificially created 166 $MCQs$ by deleting the relevant documents for 166 queries from the TREC-8 collection that had a 200 description-part queries and 200 title-part queries. They then trained a tree-based estimator to classify $MCQs$ and $non-MCQs$ using the complete set of 400 queries. In their experiment, they used a query difficulty estimator trained by analysing the overlap between the results of the full query and the results of its sub-queries to pre-filter easy queries before identifying $MCQs$

[2] http://www.isical.ac.in/ fire/faq-retrieval/2012/data.html

with a tree-based classifier. Their results suggest that identifying $MCQs$ can be improved by combining the MCQ classifier (tree-based classifier) with a query difficulty estimator. They also reported that when the MCQ classifier is used alone to detect $MCQs$, it groups together easy queries and $MCQs$ thus yielding worse results. They suggested that this can be alleviated by pre-filtering easy queries using a query difficulty estimator.

Hogan et al. [13] combined 3 different lists of $MCQs$ generated through three different approaches and then applied a simple majority voting approach to identify $MCQs$ in an SMS-Based FAQ retrieval setting. The first list of candidate $MCQs$ was generated using an approach proposed by Ferguson et al. [8] for determining the number of relevant documents to use for query expansion. In this approach, a score for each query was produced based on the inverse document frequency (IDF) component of the BM25 for each query without taking into consideration the term frequency and the document length. First, the maximum score possible for any document was calculated as the sum of the IDF scores for all the query terms. Following this approach, documents without all the query terms will have a score less than the maximum score. A threshold was then used to determine if a query should be added to the list of candidate $MCQs$. They added queries that had all their document scores below 70 % of the maximum score to this list.

The second list of candidate $MCQs$ was generated by training a k-nearest-neighbour classifier to identify $MCQs$ and $non-MCQs$. The features used to train this classifier included query performance estimators (Average Inverse Collection Term Frequency (AvICTF) [12], Simplified Clarity Score (SCS)) [11], the derivates of the similarity score between collection and query (SumSCQ, AvSCQ, MaxSCQ) [27], result set size and the un-normalised BM25 document scores for the top five documents. Their classifier achieved 78 % accuracy on the FAQ SMS training data using a leave-one-out validation. The third list of candidate $MCQs$ was generated by simply counting the number of term overlaps for each incoming query and the highest ranked documents (For example, if the query consists of more than one term and had only one term in common with the document, that query was marked as a MCQ). Hogan et al. used the held-out training data to evaluate their approach and they concluded that combining the three lists of candidate $MCQs$ through a simple majority voting yielded better results.

Leveling [17] viewed the detection of missing content queries in an SMS-Based FAQ retrieval setting as a classification problem. In his approach, he trained an IB1 classifier as implemented in TiMBL [6] using numeric features generated during retrieval phase on the training data (FIRE 2011 SMS-Based FAQ retrieval monolingual English data) to distinguish between $MCQs$ and $non-MCQs$. The features used for training were comprised of the result set size for each query, the raw BM25 document scores for the top five documents (5 features), the percentage difference of the BM25 document scores between the consecutive top 5 documents (4 features), normalised BM25 document scores for the top five retrieved documents (5 features) and the term overlap scores for the SMS query and the top 5 retrieved documents (5 features). Their approach essentially yielded a binary classifier that can determine whether a query is a MCQ or a $non - MCQ$. This approach is much simpler compared to the approach proposed by Hogan et al. [13] because it relies on a single classifier instead of relying of several classifiers. Leveling evaluated this approach using a leave-one-out validation approach which is supported by TiMBL and reported a classification accuracy of 86.3 % for $MCQs$ with the best performing system. Such a high classification accuracy for $MCQs$ resulted in a very low classification accuracy of 56.0 % for $non - MCQs$.

Misclassification of a large number of $non - MCQs$ can be costly in an HIV/AIDS FAQ retrieval system. In this work, our goal is to minimise this misclassification accuracy. In order to achieve this, we differ with previous work, by first analysing and evaluating different feature sets in order to determine the best combination of features that would yield the best classification accuracy.

3 Methodology

We begin Section 3.1 by outlining our research questions, followed by Section 3.2 where we describe how we collected and identified *non-MCQs* and *MCQs*. We then describe how we created the training and testing instances in Section 3.3.

3.1 Research Questions

- **R1 :** Which types of features produce the highest classification accuracy when classifying $MCQs$ and $non - MCQs$?
- **R2 :** Does combining different types of feature sets, produce a better classification accuracy when classifying the $MCQs$ and the $non{-}MCQs$, compared to classifying using any individual feature set?
- **R3 :** Does increasing the size of the training set for the $MCQs$ and the $non - MCQs$ yield a higher classification accuracy?
- **R4 :** Do we get comparable classification accuracy when these feature sets are generated using a different dataset?

3.2 Collecting and Identifying Missing and Non-Missing Content Queries

A study was conducted in Botswana to collect SMS queries on the general topic of HIV/AIDS. In this study, 85 participants were recruited to provide SMS queries. Having provided the SMS queries, they then used a web-based interface to find the relevant FAQ documents from the FAQ document collection using the SMS queries. This provided us with SMS queries linked to the appropriate FAQ documents in the collection. In total, 957 SMS queries were collected of which 750 ($non{-}MCQs$) could be matched to an FAQ document in the collection. The remaining 207 ($MCQs$) did not match anything in the collection. In this work, we investigate how to detect these $MCQs$ in an Automated HIV/AIDS FAQ retrieval system. In order to investigate the robustness of our approach, we also used a second dataset of 707 SMS queries (540 $non - MCQs$ and 167 $MCQs$) that we randomly selected from the FIRE2012 English Monolingual SMS query dataset. This dataset had 4476 SMS queries. We selected only a fraction of these SMS queries to use in our experimental evaluation because we had to manually correct them for spelling errors.

The main difference between the two datasets (HIV/AIDS and FIRE2012 datasets) is that the FIRE2012 dataset covers several topics (Railways, telecommunication, health, career counselling and general knowledge e.t.c) while the HIV/AIDS dataset only has one topic, HIV/AIDS. Also, the $MCQs$ for the HIV/AIDS dataset are the on-topic (related to HIV/AIDS only) while the $MCQs$ for the FIRE2012 dataset has both the on-topic and the off-topic $MCQs$. Both the HIV/AIDS and the FIRE2012 SMS queries were manually corrected for spelling errors so that such a confounding variable does not influence the outcome of our experiments. In the next section, we describe how we created the training and testing instances using this query log to answer the above research questions.

3.3 Creating Training and Testing Instances for Missing Content and Non-Missing Content Queries

In this work, we used the $non - MCQs$ and $MCQs$ that were collected and categorised as described in Section 3.2. Seven different feature sets were created for our experimental investigation and evaluation. In order for us to be able to answer research question ***R1***, we created three different feature sets, $fSet1$, $fSet2$, $fSet3$ as described below :

fSet1 : Instances in this feature set were represented by a vector of attributes representing word count information from the text contained in the query strings. Below is an example of an instance, first represented as a query string and then as a vector of attributes representing word count information of this query string.

Query String : what does aids stand for?

Word Count : 23 1,159 1,212 1,488 1,591 1.

In our example above, the attributes in this vector are separated by commas and each attribute is made up of two parts, the attribute number, and the word count information. For example the attribute *"23 1"* denotes that the term *what* is attribute number 23 in the string vector and this term only appear ones.

fSet2 : For this feature set, training and testing instances were created using the approach proposed by Leveling [17]. In particular, numeric attributes generated during retrieval phase of the FAQ documents by the aforementioned $non - MCQs$ and $MCQs$ were used in this feature set. For each query, we performed retrieval on the FAQ Retrieval Platform described in Section 4.1 to extract attributes for identifying $non - MCQs$ and $MCQs$. In total, we created 957 instances for the feature set ($fSet2$) with the HIV/AIDS SMS queries and 707 instances for the feature set ($fSet2$) with the FIRE2012 SMS queries. These instances were assigned their corresponding class label ($non - MCQs$) and ($MCQs$).

fSet3 : For this feature set, the training and testing instances were created using eight different query difficulty estimation predictors. Seven of these predictors were pre-retrieval predictors and these were : Average Pointwise Mutual Information (AvPMI) [10], Simplified Clarity Score (SCS) [11], Average Inverse Collection Term Frequency (AvICTF) [12], Average Inverse Document Frequency (AvIDF) [10] and the derivatives of the similarity score between collection and query (SumSCQ, AvSCQ, MaxSCQ) [27]. One post-retrieval predictor was used, the Clarity Score (CS) [5]. For each query, the FAQ Retrieval Platform described in Section 4.1 was used to generate the score for each query difficulty estimation predictor.

Combined Feature Sets : We created four additional feature sets by combining the above feature sets ($fSet1$, $fSet2$ and $fSet3$) in order to answer research question, ***R2:***. The feature sets were simply combined by merging the corresponding instances. These four additional feature sets were : *fSet1+fSet2, fSet1+fSet3, fSet2+fSet3 and fSet1+fSet2+fSet3*

To enable us to answer research question ***R3***, we randomly split the feature set ($fSet1 + fSet2 + fSet3$) 10 times into training and testing sets. For each training/testing split, we created two training sets, one containing 50 % of the data (instances) and the other containing 75 % of the data. The training set with 75 % of the data was the superset of the training set with 50 % of the data. The remaining 25 % of the data was made the testing set. In total, we had 20 different training sets, 10 containing 50 % of the data and the other 10 containing 75 % of the data. The training data with 50 % of the data shared the same testing set with its superset containing 75 % of the data.

4 Experimental Setting

We begin Section 4.1 by describing the HIV/AIDS FAQ Retrieval Platform used for generating the features for the training and testing instances, followed by a description on how we train and classify $non-MCQs$ and $MCQs$ in Section 4.2.

4.1 FAQ Retrieval Platform

For our experimental evaluation, we used the Terrier-3.5[3] [19], Information Retrieval (IR) platform with BM25 [21]. All the HIV/AIDS and the FIRE2012 FAQ documents used in this study were first pre-processed before indexing and this involved tokenising the text and stemming each token using the full Porter [20] stemming algorithm. To filter out terms that appear in a lot of FAQ documents, we did not use a stopword list during the indexing and the retrieval process. Instead, we ignored the terms that had low Inverse Document Frequency (IDF) when scoring the documents. Indeed, all the terms with term frequency higher than the number of the FAQ documents (205) were considered to be low IDF terms. Earlier work in [17] has shown that stopword removal using a stopword

[3] http://terrier.org/

list from various IR platforms like Terrier-3.5 can affect retrieval performance in SMS-Based FAQ retrieval. The normalisation parameter for BM25 [21] was set to its default value of $b = 0.75$.

4.2 Training and Classifying Missing Content and Non-Missing Content Queries

Three different classifiers in WEKA, namely Naive Bayes, RandomForest and C-SVC were deployed in our experimental evaluation. Evidence from previous works suggest that randomforest and support vector classifiers achieve excellent performance compared to naive bayes across a wide variety of binary classification problems and evaluation metrics [3]. We used three classifiers on the labelled feature sets created in Section 3.3 to train and classify $non-MCQs$ and $MCQs$. For each feature set, we created 10 random splits of training and testing sets. For each training/testing split, each training set was made up of 75 % of the data while the remaining 25 % of the data was for testing. All the attributes in these training and testing sets were scaled between -1 and 1. Different Kernels were used for C-SVC. A linear kernel was used for the feature sets with a large number of attributes (String) and a Radial Basis Function (RBF) kernel was used on the feature sets with few attributes.

The regularization parameter C and the kernel parameter γ for the RBF kernel were chosen through a grid-search strategy [14]. This involved performing a 10-fold cross validation on the labelled data with various pairs of (C, γ) and selecting the pair that gave the best classification accuracy. The same grid-search strategy was deployed to select the parameters for RandomForest. The C and the γ parameter for the RBF kernel were set to 1.0 and 0.9 respectively while the C parameter for the linear kernel was set to 0.7. For RandomForest, we set the number of trees to 10 for each feature set while the number of random features for creating the trees varied and were 5 and 10 for $fSet3$ and $fSet2$ respectively and 30 when using $fSet1$. In this experimental evaluation, we will define the $non-MCQs$ as the positive class and the $MCQs$ as the negative class. Table 1 shows a confusion matrix for the outcome of this two class problem.

5 Experimental Results and Evaluation

Table 2, summarises the overall classification accuracy for all the feature sets. The sensitivity measures the proportion of the actual positive instances (recall for TP) correctly classified as $non-MCQs$. The specificity measures the proportion of the actual negatives instances (recall for TN) correctly classified as $MCQs$.

Table 1. Confusion matrix for a 2-class problem

		Predicted Class	
		$non - MCQs$ (+ve)	$MCQs$ (-ve)
Actual Class	$non - MCQs$ (+ve)	True Positive (TP)	False Negative (FN)
	$MCQs$ (-ve)	False Positive (FP)	True Negative (TN)

Table 2. The overall (for the 10 random splits) classification accuracy of all the feature sets.

Dataset	Feature Set	Classifier	Sensitivity	Specificity	Accuracy(%)	ROC area	Kappa
HIV/AIDS	$fSet1$	NB	0.935	0.546	**85.06**∗	**0.84**∗	**0.522**∗
		RF	0.983	0.406	**85.79**∗	**0.857**∗	**0.481**∗
		C-SVC	0.959	0.454	**84.95**∗	**0.833**∗	**0.4812**∗
*FIRE*2012	$fSet1$	NB	0.957	0.431	**83.31**⋄	**0.807**⋄	**0.457**⋄
		RF	0.974	0.341	82.41	0.782	0.3935
		C-SVC	0.956	0.449	**83.59**⋄	**0.832**⋄	**0.4709**⋄
HIV/AIDS	$fSet2$	NB	0.953	0.058	75.97	0.604	0.016
		RF	0.935	0.121	75.86	0.639	0.072
		C-SVC	0.999	0.005	78.37	0.502	0.0055
*FIRE*2012	$fSet2$	NB	0.836	0.593	77.86	0.767	0.4107
		RF	0.937	0.443	82.09	0.793	0.4333
		C-SVC	0.941	0.437	82.23	0.811	0.43382
HIV/AIDS	$fSet3$	NB	0.891	0.473	**80.04**⋆	**0.796**⋆	**0.3821**⋆
		RF	0.937	0.348	**80.98**⋆	**0.777**⋆	**0.337**⋆
		C-SVC	0.969	0.251	**81.40**⋆	**0.748**⋆	**0.2867**⋆
*FIRE*2012	$fSet3$	NB	0.95	0.311	79.97	0.748	0.3199
		RF	0.958	0.389	82.37	0.737	0.4146
		C-SVC	0.978	0.380	82.63	0.759	0.4619
HIV/AIDS	$fSet1 + fSet2$	NB	0.923	0.304	**78.89**⊛	**0.694**⊛	**0.2672**⊛
		RF	0.989	0.271	**83.39**⊛	**0.813**⊛	**0.3465**⊛
		C-SVC	0.952	0.449	**84.33**⊛	**0.841**⊛	**0.4647**⊛
*FIRE*2012	$fSet1 + fSet2$	NB	0.876	0.581	**80.62**◃	**0.771**◃	**0.4596**◃
		RF	0.981	0.329	82.74	0.836	0.3939
		C-SVC	0.961	0.479	**84.72**◃	**0.857**◃	**0.5097**◃
HIV/AIDS	$fSet1 + fSet3$	NB	0.900	0.507	**81.50**⊛	**0.828**⊛	**0.4274**⊛
		RF	0.975	0.425	**85.89**⊛	**0.903**⊛	**0.4837**⊛
		C-SVC	0.953	0.493	**85.37**⊛	**0.866**⊛	**0.5083**⊛
*FIRE*2012	$fSet1 + fSet3$	NB	0.954	0.347	81.05	0.717	0.3643
		RF	0.989	0.383	**84.58**◃	**0.83**◃	**0.4655**◃
		C-SVC	0.948	0.521	**84.72**◃	**0.735**◃	**0.5256**◃
HIV/AIDS	$fSet2 + fSet3$	NB	0.903	0.435	80.15	0.774	0.2672
		RF	0.944	0.324	80.98	0.774	0.3230
		C-SVC	0.959	0.271	80.77	0.776	0.2854
*FIRE*2012	$fSet2 + fSet3$	NB	0.893	0.563	**81.52**•	**0.807**•	**0.4705**•
		RF	0.948	0.479	**83.78**•	**0.812**•	**0.4869**•
		C-SVC	0.954	0.593	**86.88**•	**0.862**•	**0.6001**•
HIV/AIDS	$fSet1 + fSet2 + fSet3$	NB	0.919	0.464	**82.03**⊛	**0.804**⊛	**0.4191**⊛
		RF	0.979	0.314	**83.49**⊛	**0.887**⊛	**0.3754**⊛
		C-SVC	0.948	0.502	**85.16**⊛	**0.871**⊛	**0.5072**⊛
*FIRE*2012	$fSet1 + fSet2 + fSet3$	NB	0.913	0.545	**82.60**◃	**0.794**◃	**0.4871**◃
		RF	0.972	0.413	**84.02**◃	**0.864**◃	**0.4653**◃
		C-SVC	0.965	0.515	85.86	0.866	0.5504

It can be seen from Table 2 that the different feature sets yield fairly reasonable recall rates for the TP instances. In particular, the recall rates for TP (sensitivity) ranges from 0.891 to 0.999 for the HIV/AIDS dataset and and 0.836 to 0.989 for the FIRE2012 dataset. To put these values into perspective, these translate to between 668 and 749 correctly classified instances from a total of 750 instances for the HIV/AIDS dataset. In contrast, our classifiers did not perform well for the TN instances. Fairly low recall rates (specificity) for the TN instances were observed. Depending on the feature set and the classifier used, the specificity ranged from 0.005 to 0.546 for the HIV/AIDS dataset and from 0.311 to 0.593 for the FIRE2012 dataset. These values translate to between 1 and 113 correctly classified TN instances from a total of 207 TN instances for the HIV/AIDS and between 52 and 99 from a total of 167 for the FIRE2012 dataset. Our empirical evaluation suggests that all the feature sets performed well for the $non-MCQs$ (TP instances). For the $MCQs$ (TN instances), the best performing feature set only yielded roughly 50% classification. When we compare our results with previous works, we observed that our classifiers performed fairly poorly in the detection of $MCQs$.

To answer research question ***R1***, we used an unpaired t-test to analyse the classification accuracy between the following 10 random splits, ($fSet1$ and $fSet2$), ($fSet1$ and $fSet3$), and ($fSet2$ and $fSet3$). $fSet1$ provided a significantly higher classification accuracy (unpaired t-test, $p < 0.05$) compared to the other feature sets as denoted by $*$ for the HIV/AIDS dataset and $\diamond$ for the FIRE2012 dataset. Also observed were significantly higher (unpaired t-test, $p < 0.05$) Kappa statistic and ROC area (AUC) for $fSet1$. The kappa statistic measures the agreement of prediction with the true class and a value of 1 signifies total agreement and a value of 0 signifies total disagreement. The ROC area on the other-hand signifies the overall ability of the classifier to identify $MCQs$ and $non-MCQs$. The best classifier has an area of 1.0 and a classifier with an area of 0.5 or lower is considered ineffective.

A comparison between $fSet2$ and $fSet3$ was also made using unpaired t-test and it was observed that $fSet3$ gives a better classification accuracy for the HIV/AIDS dataset as denoted by $\star$ in Table 2. No significant difference in classification accuracy was observed between $fSet2$ and $fSet3$ for the FIRE2012 dataset. This disparity between the HIV/AIDS and the FIRE2012 dataset when we compare the classification accuracy between $fSet2$ and $fSet3$ suggest that the retrieval scores and word overlap information (used in $fSet2$) are not good discriminators for the on-topic $MCQs$ (research question ***R4***). Although $fSet2$ did not perform well for the on-topic $MCQs$ (TN instances, HIV/AIDS dataset), it performed well for the off-topic $MCQs$ (TN instances FIRE2012 dataset) as depicted by higher specificity values.

There was a significantly higher classification accuracy observed when $fSet1$ was combined with the other feature sets (research question ***R2***). This is denoted by $\circledast$ and $\triangleleft$ in Table 2 for the HIV/AIDS and FIRE2012 dataset respectively (unpaired $t-test < 0.5$ for $((fSet1 + fSet2)$ and $fSet2)$, $((fSet1 + fSet3)$ and $fSet3)$ and $((fSet1 + fSet2 + fSet3)$ and $(fSet2 + fSet3))$. Similar findings were observed when $fSet3$ was combined with $fSet2$ as denoted by $\bullet$, (unpaired $t-test < 0.5$ for $((fSet2 + fSet3)$ and $fSet3)$)

A paired t-test was used to analyse whether increasing the size of the training instances increases the classification accuracy (***R3***). The results, as shown in Table 3, indicate that there is a significant difference in classification accuracy as denoted by $*$ (paired t-test, $p < 0.05$) when the training set in increased by 25% from the original 50% of the data to 75% of the data.

6 Conclusions

This study set out to determine the best classifier to deploy in our already existing SMS-Based HIV/AIDS FAQ retrieval system for detecting $MCQs$ and

Table 3. The overall classification accuracy for ($fSet1$ + $fSet2$ + $fSet3$). One training set contains 50% of the data (instance) and the other contains 75% of the data. All the values depicted, range from 0 to 1 except the accuracy which is expressed as a percentage.

Dataset	Feature Set	Training Set Size	Classifier	Sensitivity	Specificity	Accuracy(%)	ROC area	Kappa
$HIV/AIDS$	$fSet1 + fSet2 + fSet3$	50%	NB	0.924	0.454	82.24	0.816	0.4192
			RF	0.976	0.271	82.34	0.86	0.3213
			C-SVC	0.96	0.478	85.58	0.889	0.5075
$FIRE2012$	$fSet1 + fSet2 + fSet3$	50%	NB	0.898	0.473	79.77	0.757	0.3984
			RF	0.972	0.305	82.47	0.837	0.3509
			C-SVC	0.943	0.462	82.88	0.832	0.4598
$HIV/AIDS$	$fSet1 + fSet2 + fSet3$	75%	NB	0.923	0.493	**82.98**$*$	**0.822**$*$	**0.4526**$*$
			RF	0.983	0.266	**82.75**$*$	**0.884**$*$	**0.3281**$*$
			C-SVC	0.956	0.56	**87.04**$*$	**0.886**$*$	**0.5747**$*$
$FIRE2012$	$fSet1 + fSet2 + fSet3$	75%	NB	0.92	0.539	**83.02**$*$	**0.798**$*$	**0.494**$*$
			RF	0.972	0.443	**84.72**$*$	**0.882**$*$	**0.4952**$*$
			C-SVC	0.967	0.497	**85.57**$*$	**0.85**$*$	**0.537**$*$

$non - MCQs$. Several research questions were addressed to achieve the above goal. Our result suggest that the most important feature set (***R1***) for building such a classifier is $fSet1$, which is a set of attributes representing word count information from the text contained in the query strings. It also emerged from this study that the classification accuracy of a classifier built using other feature sets ($fSet2$ and $fSet3$) can be improved further by combining these feature sets with ($fSet1$) (***R2***), in particular $fSet3$ (feature sets generated by query difficulty predictors). In future, we will investigate better ways on how to combine these feature sets, in order to improve the classification accuracy.

In addition, we also investigated whether increasing the training set size would yield a better classification accuracy. A significant increase in accuracy, ROC area and Kappa statistic was observed when the training set was increased by 25%. The other finding to emerge from this study is that some feature sets work best for some datasets and perform poorly on other datasets (***R4***). As our results suggest in Table 2, $fSet2$ does not perform well when the $MCQs$ are on-topic ($MCQs$ related to the FAQ document collection) as in the case of the HIV/AIDS dataset. This feature set do however perform well when the $MCQs$ are off-topic ($MCQs$ not related to the FAQ document collection) as in the case of the FIRE2012 dataset. Other feature sets ($fSet1$ and $fSet3$) do however perform well across these different collections.

References

1. E. Bornman. The Mobile Phone in Africa: Has It Become a Highway to the Information Society or Not? *CONTEMP. EDU. TECH.*, 3(4):278–292, 2012.
2. L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
3. R. Caruana and A. Niculescu-Mizil. An Empirical Comparison of Supervised Learning Algorithms. In *Proc. of ICML*, 2006.
4. C.-C. Chang and C.-J. Lin. LIBSVM: A library for Support Vector Machine. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27, 2011.
5. S. Cronen-Townsend, Y. Zhou, and W.B. Croft. Predicting Query Performance. In *Proc. of SIGIR*, 2002.
6. W. Daelemans, J. Zavrel, K.V.D. Sloot, and A.V.D. Bosch. TiMBL: Tilburg Memory-Based Learner - version 4.3 - Reference Guide, 2002.

7. J. Donner. Research Approaches to Mobile Use in the Developing World: A Review of the Literature. *The Info. Soc.*, 24(3):140–159, 2008.
8. P. Ferguson, N. O'Hare, J. Lanagan, A.F. Smeaton, K. McCarthy, O. Phelan, and B. Smyth. CALRITY at the TREC 2011 Microblog Track. In *Proc. of TREC*, 2011.
9. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA Data Mining Software: an Update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009.
10. C. Hauff, V. Murdock, and R. Baeza-Yates. Improved Query Difficulty Prediction for the Web. In *Proc. of CIKM*, 2008.
11. B. He and I. Ounis. Inferring Query Performance using Pre-Retrieval Predictors. In *Proc. of SPIRE*, 2004.
12. B. He and I. Ounis. Query Performance Prediction. *Info Syst*, 31(7):585 – 594, 2006.
13. D. Hogan, J. Leveling, H. Wang, P. Ferguson, and C. Gurrin. DCU@FIRE 2011: SMS-based FAQ Retrieval. In *Proc. of FIRE*, 2011.
14. C.-W. Hsu, C.-C. Chang, and Lin C.-J. A Practical Guide to Support Vector Classification. 2010.
15. G.H John and P. Langley. Estimating Continuous Distributions in Bayesian Classifiers. In *Proc. of UAI*, 1995.
16. I. Lane, T. Kawahara, T. Matsui, and S. Nakamura. Out-of-Domain Utterance Detection Using Classification Confidences of Multiple Topics. *Aud. Speech, and Lang. Process. IEEE Transact.*, 15(1):150–161, 2007.
17. J. Leveling. On the Effect of Stopword Removal for SMS-Based FAQ Retrieval. In *Proc. of NLDB*, 2012.
18. I. Medhi, A. Ratan, and K. Toyama. Mobile-Banking Adoption and Usage by Low-Literate, Low-Income Users in the Developing World. In *Proc. of IDGD.* 2009.
19. I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma. Terrier: A High Performance and Scalable Information Retrieval Platform. In *Proc. of OSIR at SIGIR*, 2006.
20. M.F Porter. An Algorithm for Suffix Stripping. *Elec. Lib. Info. Syst*, 14(3):130–137, 2008.
21. S. Robertson and H. Zaragoza. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Info. Retr.*, 3(4):333–389, 2009.
22. E. Sneiders. Automated FAQ Answering: Continued Experience with Shallow Language Understanding. Question Answering Systems. In *Proc. of AAAI Fall Symp.*, 1999.
23. E. Sneiders. Automated FAQ Answering with Question-Specific Knowledge Representation for Web Self-Service. In *Proc. of HSI*, 2009.
24. E. Thuma, S. Rogers, and I. Ounis. Evaluating Bad Query Abandonment in an Iterative SMS-Based FAQ Retrieval System. In *Proc. of OAIR*, pages 117–120. 2008.
25. E. Yom-Tov, S. Fine, D. Carmel, and A. Darlow. Learning to Estimate Query Difficulty: Including Applications to Missing Content Detection and Distributed Information Retrieval. In *Proc. of SIGIR*, 2005.
26. M. Zhang M.Y., Dodgson. High-tech Entrepreneurship in Asia: Innovation, Industry and Institutional Dynamics in Mobile Payments. Edward Elgar Publishing, Inc., 2007.
27. Y. Zhao, F. Scholer, and Y. Tsegay. Effective Pre-Retrieval Query Performance Prediction Using Similarity and Variability Evidence. In *Proc. of ECIR*, 2008.