Text messaging and retrieval techniques for a mobile health information system

Ademola O. Adesina
Department of Computer Science, University of the Western Cape, South Africa

Kehinde K. Agbele
Department of Computer Science, University of the Western Cape, South Africa

Ademola P. Abidoye
Department of Computer Science, University of the Western Cape, South Africa

Henry O. Nyongesa
Department of Computer Science, University of the Western Cape, South Africa

Abstract
Mobile phones have been identified as one of the technologies that can be used to overcome the challenges of information dissemination regarding serious diseases. Short message services, a much used function of cell phones, for example, can be turned into a major tool for accessing databases. This paper focuses on the design and development of a short message services-based information access algorithm to carefully screen information on human immunodeficiency virus/acquired immune deficiency syndrome within the context of a frequently asked questions system. However, automating the short message services-based information search and retrieval poses significant challenges because of the inherent noise in its communications. The developed algorithm was used to retrieve the best-ranked question–answer pair. Results were evaluated using three metrics: average precision, recall and computational time. The retrieval efficacy was measured and it was confirmed that there was a significant improvement in the results of the proposed algorithm when compared with similar retrieval algorithms.

Keywords
frequently asked question (FAQ); information retrieval (IR); mobile health (mHealth); mobile Health Information system; question and answer (Q&A) system; short message service (SMS)/text message

1. Introduction
The short message service (SMS)-based information retrieval is a way of accessing information necessitated by the rapid development of mobile telecommunication. The technology is characterized by instant access to information as a response to SMS enquiries. The SMS-based information request is considered unique because of the restricted size available for the reply, so only a few results can be returned for any given query. A mobile retrieval search system enables the user to obtain extremely concise and appropriate responses from queries across arbitrary topics. Users may be forced to rephrase or reformulate the query if their answers are not made available in the preliminary pages of the search response. Unfortunately, there is a limit to what a mobile phone user can download compared with what is downloadable from a desktop system or microcomputer. Mobile search users rarely employ the advanced search feature of the search engine but prefer to expend extra energy in reformulating the query [1, 2].
In the meantime, advances in mobile communication have generated the concept of text messaging. This computer-mediated communication has its own peculiarities, whereby users have their own patterns of writing, inventing new abbreviations and using non-standard orthographic forms [3]. This language is not in alignment with the pattern of traditional natural languages because of the restriction in the number of characters permissible to be written on the mobile device, bandwidth of digital communication and limited memory capacity of the mobile phone [4, 5]. The SMS communication provides a platform where messages can be delivered even when the recipient is engaged in voice communication or is otherwise unable to attend to a call. The text message has created a social network environment for peer groups where information is shared. For example, in mobile health (mHealth) technology, SMS has played a significant role in bridging the gap in communication not only between the patients and health workers [6–8] but also between healthcare workers and medical resources, for example, the medical library [9].

SMS communication is used for information access and retrieval in healthcare-related applications. Its introduction minimizes visits of physicians to patients, and is useful for drug prescriptions, consultancy services, appointment reminders, health and prevention reports, bills and other forms of information. This leads to better understanding and education about healthcare issues, and in turn, reduces the cost of healthcare provision [10–13]. For instance, in South Africa, SMS is used to remind tuberculosis patients to take their drugs. The tuberculosis drug Rifafol needs to be taken daily and on a consistent basis to be effective. SMS texts which are written in English and local languages – Afrikaans and Xhosa – are sent at a pre-determined time daily to the patient. This is done for a period of six months for a complete treatment [14]. In addition, in 2008, SIMPill was implemented in South Africa to remind tuberculosis patients of their medication. A Subscriber Identity Module (SIM) card is placed on the bottle top, which sends an SMS text every time the bottle is opened. A reminder in the form of an SMS text is sent to the patient or relative if the bottle is not opened at the expected time [15].

Generally, an information retrieval process begins when a user enters an inquiry into the search engine with the expectation of getting reasonable feedback. An inquiry is a formal statement of the user’s information needs expressed in a formal language [16]. The feedback represents the list of sensible answers to the inquiry made by the information seeker. This list is made available as a result of a comparison between the keyword terms of the query statement [17, 18] and the repository of the answers in a database [19, 20]. Natural language and social network communication (SNC) languages can be used as query statements to source information on the search engines. SMS as an example of SNC is a preferred form of communication for many youths [21–23]. For this communication paradigm, building automated question-answering systems has proved difficult because of the various means the users have employed of representing formal language. For example, in the datasets collected for the experiment, tomorrow as an English word is expressed in more than 20 SMS versions – tomoz, tomorro, tomorrw, tomor, mora, morr, mor, morn, tom, 2mora, tomoro, 2morrow, tmw, 2mornew, 2mrow, 2morow, 2mrrw, 2moz, 2mw, amoro, tomorrow, 2moro, tmrw and tomrv. The translation of SMS variants into the Standard English form (tomorrow) is of utmost importance in SMS normalization. This is the stage of correcting the erroneous forms in which the SMS text appears (SMS normalization is, however, beyond the scope of this paper). The freedom of SMS writing poses a great challenge to its normalization. Other natural language processing techniques like stemming, regression, classification, clustering and stop word identification are almost impossible until there is a correction of the noisy form of the SMS.

Using a Google search as a benchmark, the typical results of an SMS-based search can be considered using a query sentence – wn d u intt arv thrpy – extracted from an English query ‘when do you initiate antiretroviral therapy?’ Google responds only with a normalized form of thrpy as therapy and translates the abbreviation arv to antiretroviral. This is a usual experience for SMS information seekers. Google appears to be the best web search engine in terms of average precision and response time [24]. When used, the SMS query results mostly take the form of Garbage In Garbage Out, and as such are not helpful to the SMS user. Normally, when a user mistypes an input query, the system will suggest an alternative query sentence, in order to continue the semantic-based search [25]. Sometimes, suggestions made by the search engine are far from the intent of the SMS user, for example, in the search that was performed, wn d u were joined together as wndu.

There is a need to clean-normalize the SMS in order for it to play a role in question answering (QA) systems [20, 26–29]. An SMS-based QA retrieval system accesses information in the form of questions and answers with the use of SMS services on the mobile phone platform. The QA systems may appear in four guises. The first is natural language processing – this is a situation whereby users send a query in natural language for enquiries on phones or mobile devices, and the answers are returned in natural language. The Google web search system uses the text of links to index documents [30] and processes the query. For example, using a query sentence ‘when do you initiate antiretroviral therapy?’ will return a long list of documents about antiretroviral, because the search system has found references that include the word therapy that most frequently point to documents discussing antiretroviral therapy. The second is human intervention – messages are sent in the form of natural language to a particular agent. Normally, the agent, who is an expert, gives the
answer to the request. This is mostly common with expert systems, like MYCIN, where enquiries are made so as to determine the kind of ailment and treatment procedure. MYCIN is a computer program designed to provide attending physicians with advice comparable to that which they would otherwise get from a medical consultant. To use MYCIN, the attending physician must sit in front of a computer terminal that is connected to a DEC-20 (one of Digital Equipment Corporation’s mainframe computers) where the MYCIN program is stored. When the MYCIN program is evoked, it initiates a dialogue. The physician types answers in response to various questions. Eventually MYCIN provides a diagnosis and a detailed drug therapy recommendation [31].

2. Related work

It is crucial that relevant answers are provided for users when the enquiry is made, otherwise this can lead to query abandonment or further iteration of the request [36]. Hence it is important for the search to present the most relevant document in the FAQ collection as the answer to the SMS-based request. Burke et al. [37] used a natural-language-processing question answering system that uses FAQ files as its knowledge base. The technique is based on four assumptions used to convert the FAQFINDER system: (1) organizing the FAQ file in QA format; (2) setting the information locality within the QA pair; (3) determining the question’s relevance within the QA in order to find the match; and (4) possessing a general knowledge of the languages for question matching. The user’s query terms are matched with the FAQ files. The FAQFINDER search process was limited to a small set of FAQ files that are likely to have the best match to the user’s query.

Mogadala et al. [38] used a language modelling (LM) approach to match noisy SMS text with the right FAQ. The team developed a dictionary-based approach for SMS text normalization. The cleaned SMS text is then matched with the FAQ using an LM method before the corresponding response to the query is released. Mogadala et al.’s [38] experiments use a combination of SMS datasets of English, Hindi and Malayalam languages with their corresponding FAQs in different combinations for the monolingual task, and FAQs in Hindi and the English language for the cross-lingual task. In both sets of experiments, the percentage of the languages is continuously varied in order to retrieve information from their FAQ databases using English SMS queries. The FAQs are divided into three different collections: (1) the questions only; (2) the answers only; and (3) combinations of questions and answers of the three languages. The results show that developed LM questions outperform both answers and combinations of questions and answers for matching SMS queries.
LM model does not give consideration to synonyms. It is word-dependent. This means that any other answer that could be chosen in the FAQ answer dataset may not be considered.

Hogan et al. [26] identified SMS-based FAQ retrieval systems as having three steps: (1) SMS normalization; (2) retrieval of ranked results; and (3) identification of out-of-domain query results. In order to normalize the SMS FAQ queries, a set of transformation rules was created and the corpora were manually annotated. The tokens were aligned with the original text messages to give a one-to-one correspondence between the original and corrected tokens. The documents and SMS questions underwent the same pre-processing. In the research of Hogan et al. [26], each SMS token was examined (if it remains unchanged) and then the corrected token was substituted. A set of candidate lists was generated, and the best candidate in the context was selected as the correction. The best candidate was selected using three methods: (1) manually annotated data was used as a correction rule to get the best transformation for the SMS tokens and the frequency of use of the correction rules became a criterion for calculating the normalized weights of the replacement of SMS token in the corpus; (2) candidate corrections were created by consonant skeletons — a consonant skeleton is the withdrawal of vowels from a word, leaving only the ‘consonant’ of the word, for example, ‘medicine — mdcn’, ‘health — hlth’ [39], and the mapping between the consonant skeletons and the words produces additional correction candidates for the query words; and (3) candidates were generated when all words in the corpus were compared with the prefix of the question words, to confirm that there was a truncation. The three methods produced replacement candidate lists, which were merged by adding their weightings from their term frequency. The token scores were calculated using the maximum product of that weight and the n-gram score of the corrected token. Hogan et al. [26] used a manual annotation of the dataset, but this may be cumbersome for large corpora. The experiment was performed on monolingual English SMS datasets with different retrieval engines (Solr, Lucene and a combination of the two search engines) and approaches. The best result from the candidate list was retrieved by ranking the weighted scores of a list of question–answer pairs. The evaluation of the results involved comparing out-of-domain results when tested on the two search engines. The SMS normalization approach is token-based and all the tokens were processed.

Kothari et al. [19] designed an automatic FAQ-based question answering system. The method involved promoting SMS query similarity to FAQ-questions. This was done through a combinatorial search approach. The search space consisted of combinations of all possible dictionary variations of tokens in the noisy query. The combinatorial search system modelled an SMS query as a syntactic tree matching so as to improve the ranking scheme after candidate words had been identified. Initial processing of noise removal was introduced so as to improve the information retrieval efficiency. The model involved the use of a dictionary, and mapped the SMS query to the questions in the corpus. The noise removal step was, however, computationally expensive [40]. The system developed by Kothari et al. [19] did not involve training SMS data on text normalization. It had the advantage of handling semantic variations in question formulation but the method failed to discuss the choice of homophonic words in the context of automatic speech recognition. Kothari et al. [19] depended on a scoring function for the choice of selecting FAQ questions. Thus, in cases where there is a tie over the score function, it would be difficult to rank the question, and other factors, such as the proximity measurement of the SMS query and FAQ token as proposed by Jain [41] and Joshi [42].

An n-gram count-based algorithm developed by Jain [41] took into account various n-grams in order to calculate the score of questions from the corpus. This was similar to the approach used to develop SMSql (Section 5). The score of different FAQ questions from the candidate sets was then calculated. The maximum score among the set was therefore returned with its corresponding answer in the FAQ database. Two factors were considered that led to an enhancement in evaluating the FAQ score in the candidate set. They were the proximity of the SMS query and FAQ tokens, and a comparison of the question sentence length of the matched tokens from the SMS query to the FAQ questions under consideration [41, 42]. When the algorithm was evaluated on many real-life FAQ datasets from different domains, the results showed significant improvement in terms of the accuracy compared with Kothari et al. [19]. This approach did not give consideration to synonyms, that is, it was word-dependent as answers were chosen only from the FAQ answer dataset.

Chen et al. [4] proposed SMSFind as another type of SMS-based information retrieval model. This was designed to deliver the final search response to a normalized SMS query. It used a conventional search engine in its back end to provide an appropriate answer for the SMS request. SMSFind used translated SMS queries. Typically, the arrangement contained an SMS term or a collection of consecutive terms in a query that provides a hint as to what the user is looking for. The hint, provided by the user or automatically generated from the document, was used to address the information extraction problem. SMSFind used this hint to address the problem as follows: given the top search responses to a query from a search engine, SMSFind extracted snippets of text from within the neighbourhood of the hint in each response page. SMSFind scored snippets and ranked them across a variety of metrics. The hint extracted was used to determine the answer to the request. It was scored based on a top-n list for each page. The highest score was released as an answer to
the request [4]. The use of hints in the algorithm was considered as a supervised learning approach [43, 44] and it was expensive to generate and store. The research never considered the contextual information of the searches and the searching was limited to the constituent of the hint.

The research presented shares similarities in the area of application, that is, health-related matter, with the research of Anderson et al. [45] and Masizana-Katongo et al. [46], but the two research groups used SMS parsing technique to query the search engine after the SMS token had been disambiguated using context-free grammar. In our approach, the SMS term was taken as a query while the FAQ was considered as a document for the SMS-based retrieval method. While their research was applied to a multilingual scenario, ours considered the monolingual scenario. The system flowchart of the SMS-based information retrieval system is discussed in the next section.

3. System flowchart of the proposed SMS-based retrieval algorithm (SMSql)

The flowchart in Figure 1 shows a typical SMS-based retrieval system. The flowchart is drawn in order to understand, evaluate and design the SMS-based information retrieval system. The SMS terms are sent into the SMS normalization/translation process. The normalized SMS terms are compared with the FAQ-query terms in the FAQ English database query set. The FAQ query terms are statistically selected as the keywords.

The FAQ-queries are ranked according to the extent of the similarities of the SMS-queries and FAQ-queries in the ranked list of pre-defined FAQ queries. The result of the SMS request is presented to the texter based on the user’s judgement, that is, is it satisfactory or not? However, there is room for a reformulation process to take place. Dissatisfaction of the users of information leads to disengagement from the search system [47–49]. Expected results and satisfactory answers may not be presented to the user if the FAQ-query dataset does not have the query terms in common.

![Figure 1. Flowchart of SMS-based retrieval system (SMS question locator – SMSql).](image-url)
4. Research problem and methodology

This section discusses the research problem and the approach adopted to solving the problems.

4.1. Research problem

The research sets out to investigate ways of receiving an accurate response when an SMS text is used as a query to a FAQ database server in order to garner advice on a specific health domain. The research focused on two research questions which are:

(1) How should the design and development of an SMS-based information access system be achieved?

(2) How is the retrieval efficiency of the developed SMS-based information access system determined?

4.2. Research methodology

The experiment involved the use of FAQ database consisting of over 350 sampled questions on issues of HIV/AIDS – drug administration, prevention, control and support, counselling, food prescription, awareness, sex education, and education and training. The FAQ dataset comprised English words and HIV/AIDS terminologies. Out of these sampled questions, about 200 questions were extracted from Ipollete call centres [50] and the remainder were gathered from over 15 related websites. Ipollete data is a question resource of 205 most frequently asked questions about HIV/AIDS and antiretroviral therapy. The booklet [50] was prepared by the Ministry of Health, Botswana. Several other SMS-based retrieval process researches on the HIV/AIDS FAQ system make use of Ipollete documents [16, 36, 46, 51]. The related websites have vast information on the HIV/AIDS epidemic in FAQ forms on aspects of drug administration, therapy, sex education, food and nutrition, physical exercise and treatment. The data was collected over a period of 20 months.

For the purpose of keyword extraction, 350 sampled questions used in this experiment were presented, individually, to about 140 students of the University of the Western Cape. One-hundred sample booklets were recovered from the students. These were used for analysis. The students were educated on the purpose of the research. The likely terms, idioms or words that can be recognized as the keywords per FAQ query sentence were identified by the students. The terms, idioms or words with the greatest frequency were selected as the keywords. This was achievable by statistically selecting keywords and idioms from the query corpus in the FAQ query-set gathered earlier. The keywords and idioms are combinations of words or phrases that give a reasonable meaning to each query. From the keyword phrases, idioms can be derived. For example, a query sentence ‘when do you initiate antiretroviral therapy’ has three statistically selected words – initiate, antiretroviral and therapy – as the idiom equivalents or keyword phrases. An idiom on the other hand is a collection of words with a specific semantic meaning as a group, which may not yield the same meaning when interpreted individually as words and not collectively as a phrase [52].

From the FAQ query set, 20 questions were statistically selected for our analysis. These questions were translated to SMS shorthand by the students of the University of the Western Cape. A set of 20 questions from 10 respondents yielded the 200 SMS-query formats used in our dataset, that is, each query had 10 respondents. A large collection of data was necessary in order to reduce the tendency towards bias in the SMS writing. Extraction of the best match question–answer pairs in the FAQ server was the ultimate goal. Keywords for each question were extracted based on the frequency level from more than 10 respondents. This meant that each question had a maximum of 10 SMS variants, that is, the number of respondents. In summary, there were 65 questions with similar keywords and idiom equivalents from the 350 questions originally gathered for this experiment.

At this stage it is important to note that stop words were less important parts of the keyword phrases and were discarded. Stop words are very common words that appear frequently in text and carry little or no semantic meaning in an expression [53]. Stop words affect the retrieval effectiveness because they have high frequency and tend to diminish the impact of frequency differences among less common words, affecting the weighting process [54]. It is therefore recommended that high-frequency word n-grams that occur in many words be eliminated before computing the similarity coefficient. Weighting the remaining n-grams using an inverse frequency coefficient, that is, assigning the highest values to the least frequently appearing n-grams, will ensure that matches between less frequent n-grams contribute more to word similarity than matches between frequent n-grams [55].

The retrieval efficiency results of the two algorithms – term frequency-inverse document frequency (tf-idf) and SMSql – are used as the basis for judging of the efficient algorithm. The scale of relevance judgement needed in calculating the retrieval efficiency was placed on a scale of 5, where excellent = 5; very good = 4; good = 3; moderate = 2; and poor = 1. The judgement was based on the first five FAQ sets of queries that emerged from the various ways in which SMS questions were sent into the search engine. This approach is similar to that of Mogadala et al. [38], where cleaned SMS texts...
5. Proposed algorithm – SMS question locator (SMSql)

This section discusses the SMSql algorithm over the SMS FAQ search and retrieval system for mobile communication. The normalized idiom equivalents or keyword phrases extracted from the SMS query were matched with words present in our FAQ corpus. The algorithm considered similarity in words between the SMS query and the FAQ database, the length of the two sentences and the order in which the words were placed. The length of the query sentence was given priority. For easy identification each question (with its corresponding answers) had a unique code. Keyword isolation and identification led to further derivation of idioms. The interpretation of the wordings was individually done but considered collectively for FAQ query selection.

One of the methods adopted in arriving at a ranked list was assigning weights to the relevant terms. This showed the degree of importance of the terms (tokens) in the documents. Weight difference was needed for the following reasons: (1) to measure the degree of similarity between the FAQ terms and SMS query terms; (2) to determine the length and specificity of the query sentences; and (3) to determine the number of relevant FAQ documents (sentences), that is, the number of query sentence terms. A weight function/value of 1 was used to measure the similarity between the FAQ terms and the SMS query terms. A weight function/value of 2 was used to confirm the FAQ query sentence length. The keyword terms that were available in the FAQ sentence (and non-matching) were assigned 2. This was important if there was to be a tie in the weight function between FAQ terms and SMS query. The FAQ query sentence with lower sum of non-matching terms was considered as the chosen FAQ query sentence. Figure 2 provides a step-by-step description of the SMSql algorithm.

SMSql processed the input sentence word-by-word from left to right. When the first SMS word (target word) was found, then the context window was built. This window was formed by the words placed just before and after the target word present in the FAQ database. The window size used in our system was 3, which included the target word and one word to its left and right, following the claim by Huang et al. [56] and Michelizzi [57] that words further away from the target word are less likely to be related than words close to the target word.

When a FAQ file was chosen as the query was being issued, the system iterated through the QA pairs in the file, comparing each question against the user’s question and computing a score based on the weight function. We defined a scoring function for assigning a score to each statistically selected keyword phrase in the question corpus Q, where SMS token $s_i$ had been normalized to English term $t$ in the dictionary. Therefore, there was a similarity measure $\hat{s}$, between $s_i$ and $t$ such that $\hat{s}(s_i, t) > 0$ and this was denoted in the equation as $s_i \approx t$ in the equation. The score function measured how closely the question matched the SMS question string $S$.

Consider a query term $q$ in every FAQ query sentence in the overall dataset $Q$ as $q \in Q$. In the particular query sentence for each token SMS string $s_i$, the scoring function chose the term from $q$ having the maximum weight. Then the weights $n$ of the chosen terms were summed together, which gave the score:

$$\text{Score}(Q) = \sum_{i=1}^{n} \left[ \max_{t \in Q \text{ and } s_i \approx t} w(s_i, t) \right]$$

The goal was to efficiently find the best matches to the query in the FAQ. The five selections with the highest scores were selected and were returned to the user. Each question from the FAQ file was matched against the user’s question and then scored. Figure 3 illustrates the step-by-step description of the tf-idf algorithm.

**Figure 2.** The SMSql algorithm.
Using the tf-idf algorithm we were able to perform the ranking of the FAQ query for the set of SMS queries given by 10 SMS users over 20 questions. This was ranked and represented as relevance of the questions based on the SMS enquiries for this approach.

### 6. Performance evaluation

*Average precision* and *average recall* were the means of the *precision* and *recall* values obtained respectively from the set of top \( k \) \((k\) was the size of FAQ query document\) existing in FAQ datasets after each relevant FAQ query was retrieved, and this value was then averaged over information needs. That is, the set of relevant FAQ query documents for an information need \( q_j \in Q \) is \( \{d_1, \ldots, d_m\} \) and \( R_{jk} \) is the set of ranked retrieval results from the top result until the FAQ query document \( d_k \) is retrieved.

The best way to test the performance of different retrieval strategies is by using a simulation experiment. In this setting, a sample of queries is available and the documents that are relevant to each query have already been statistically identified. The performance of each automatic system can then be compared with a known standard of optimal performance. Systems are rated according to their ability to rank the relevant documents higher than the documents that are not relevant. While one can give a number of arguments about how and why this test setting does not reflect reality, no better methods for evaluating performance have been developed [58].

The efficiency of the retrieval mechanism was determined by the system retrieval and learning performance. The best retrieval strategy depends greatly on the length and specificity of the query because a complex data-driven retrieval strategy has little success with short queries and limited amounts of information [59]. Users of search engines have been accustomed to using short queries with keyword combinations owing to the restriction of the interface and inner mechanism of the search engine [59]. However, the detail that they provided might be vital to obtain good results for longer, more precisely defined queries where little vocabulary is shared by relevant documents, so that the system may be required to have some language understanding capability in order to discover relevant answer documents [60].

Therefore retrieval efficiency can be calculated through *precision* and *recall*. The learning performance involves performing the same set of experiments with a pre-determined number of iterations with the same dataset a particular number of times. To conduct the evaluation, the following steps were taken:

- We took a sample of 20 SMS coded FAQ query sentences. (*Mostly they were a set of queries that had greater representation from the data collected from the respondents, and were determined statistically.*)
- Each query was designed to retrieve the five best answers. The results were verified by experienced users using datasets applied at the beginning of our experiment and their corresponding answers.
- The retrieval efficiency was measured using precision and recall.
- The computational speed of the two algorithms was compared in order to determine the technique that is faster.

**Precision** is the relative number of correct constituents (FAQ query) retrieved from those retrieved as relevant. Hence the value must be as high as possible for good parsing. A constituent is considered to be correct if it matches a constituent in the Gold Standard (the structure representing the ideal analysis which the parsing results intended [61]).
Precision = \frac{\text{Number of relevant FAQ queries}}{\text{Number of retrieved FAQ queries}} \quad (2)

Recall is the relative number of correct constituents compared with the gold standard parse. It shows how many relevant answers were actually retrieved out of the possible answers – the higher the recall value is, the better the algorithm performance is.

The two metrics, precision and recall, which are inversely related, are computed using unordered list of FAQ query sets [62]. They are based on the user’s relevance assessments following the retrieval process [60]. Therefore, the automatic handling of the various forms of user queries requires not only a large database of QA pairs but also the technology to match the user query to the FAQ documents in the database [40].

7. Results

The evaluations for this experiment were carried out in duplicate. They are presented as follows:

7.1. Comparison between tf-idf and SSql algorithms in terms of average precision and average recall

The results of the experiment are given in Figures 4 and 5, where the average precision and average recall are plotted against a set of selected queries. If 10 SMS users send the same SMS query sentence to the search engine, the SMS writing of each user varies. The average precision and average recall were taken for the 10 SMS users (for each query sentence) based on the relevance judgement. Overall, the performance of SMS queries in the SSql and tf-idf techniques appeared very difficult to confirm. Thus there was a need to perform a statistical test on the results and confirm the significance.

![Figure 4. Average precisions for SMS queries in the SSql and tf-idf algorithms.](image4)

![Figure 5. Average recalls for SMS queries in the SSql and tf-idf algorithms.](image5)
The significance test was adopted to reject the null hypothesis, $H_0$, which means that there is no difference between the results of the two methods. This is done by comparison of the mean precision values across all the queries. The $t$-test was used to compare the mean scores for same group (of 10 users) and same condition or method on two different occasions, or when there are matched pairs [63]. A paired-samples $t$-test was conducted to evaluate the average precision for SMSql and tf-idf algorithms. There was a statistically significant difference in the performance of SMSql (mean = 3.55, standard deviation = 0.9987) and tf-idf (mean = 2.90, standard deviation = 1.2524); $t(19) = -3.577, p > 0.005$ ($p$-value = 0.002) at a confidence interval of 95%.

### 7.2. Comparison between tf-idf and SMSql algorithms in terms of average computational time

These metrics (average precision and recall) may not be sufficient to prove the algorithms. The system was further tested by employing a timing computation of the retrieval system. The two results were compared using the computational time. The results are shown in Table 1. The two algorithms were also compared in terms of computational speed. The computational speed was measured by estimating the time it took to retrieve the best five FAQ documents for each SMS query sentence. The total time used by the 10 users was then averaged and was taken as the average computational time for the SMS query iteration. The estimate was taken for the two algorithms and the average computational time was recorded for the methods. This is similar to the work described by Pudil et al. [64] using simple feature selection. The methods of Pudil et al. [64] show similar performance and differ only in computational efficiency. The objective of the comparison was identifying sub-optimal search methods. This can be achieved by considering computational time and efficiency.

In order to demonstrate clearly the effectiveness of each method, the selection of a feature set from the data showing high statistical dependencies provides a more discriminating test [64]. The execution time to generate results was compared for the two algorithms. The system of calculating the execution time can be constructed out of sequential programs but are typically built from concurrent programs called tasks [65].

The percentage of improvement between SMSql and tf-idf was 4.1%, that is, $(0.073 − 0.07)/0.073*100\% = 4.1\%$. The results proved that the SMSql is 4.1% better than the tf-idf approach using the computational time as the metric value. From Table 1, the average time taken ($t_n$) for an SMS query number ($Q_n$) for each SMS request by the user was taken for each of the algorithms, and the results presented. The score of each query sentence was calculated sequentially and then ordered to generate the result. The average time for each iteration of the SMS queries, SMS $Q_1$–$Q_{20}$, for each algorithm was taken. The results show the time spent in generating responses to requests made in this experiment.

<table>
<thead>
<tr>
<th>SMS query no. ($Q_n$)</th>
<th>Average computational time for 10 iterations per SMS query</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>tf-idf ($s$)</td>
</tr>
<tr>
<td>1</td>
<td>0.085</td>
</tr>
<tr>
<td>2</td>
<td>0.077</td>
</tr>
<tr>
<td>3</td>
<td>0.086</td>
</tr>
<tr>
<td>4</td>
<td>0.074</td>
</tr>
<tr>
<td>5</td>
<td>0.085</td>
</tr>
<tr>
<td>6</td>
<td>0.037</td>
</tr>
<tr>
<td>7</td>
<td>0.069</td>
</tr>
<tr>
<td>8</td>
<td>0.067</td>
</tr>
<tr>
<td>9</td>
<td>0.077</td>
</tr>
<tr>
<td>10</td>
<td>0.068</td>
</tr>
<tr>
<td>11</td>
<td>0.074</td>
</tr>
<tr>
<td>12</td>
<td>0.088</td>
</tr>
<tr>
<td>13</td>
<td>0.067</td>
</tr>
<tr>
<td>14</td>
<td>0.075</td>
</tr>
<tr>
<td>15</td>
<td>0.082</td>
</tr>
<tr>
<td>16</td>
<td>0.067</td>
</tr>
<tr>
<td>17</td>
<td>0.078</td>
</tr>
<tr>
<td>18</td>
<td>0.068</td>
</tr>
<tr>
<td>19</td>
<td>0.059</td>
</tr>
<tr>
<td>20</td>
<td>0.074</td>
</tr>
<tr>
<td>Total</td>
<td>1.457</td>
</tr>
<tr>
<td>Average</td>
<td>0.073</td>
</tr>
</tbody>
</table>
8. Conclusion

It is imperative to link information seekers to information sources by matching the query with the description of the content that is associated with the indexed information segments in the database. This paper investigated a situation where SMS text is used to make an enquiry from the search engine on health-related matters. Information seekers are not patient enough to source results beyond the preliminary download pages, and unfortunately, the noisy form in which SMS appears is not sufficient to provide accurate results until it is normalized. This creates a great challenge to the information seeker whenever SMS is used to search for information. An algorithm was developed to use the normalized form of SMS text for information search. The developed algorithm considered the similarity in words between the SMS query and the FAQ database, the length of the two sentences as well as the order in which the words are placed. The retrieval efficiency of the developed algorithm was compared with the existing algorithm in an SMS-based FAQ system. The query-term similarity between the SMS and FAQ was used to present the five best ranked relevant results. Three parameters – average precision, recall and computational time – were used for the basis of proving that the developed method was better when the retrieval efficacy was considered. Statistically, there was significant difference in the performance of the two techniques and also the computational speed of the developed algorithm proved to be better by 4%. The developed algorithm has been proved to produce fast results for the normalized SMS-query.

Acknowledgements

Our appreciation goes to Professor Isabella M. Venter for her contribution in the early stages of producing the manuscript.

Funding

The authors would like to acknowledge the Senate Research Committee of the University of the Western Cape, Bellville, South Africa for funding.

References

[11] Kaplan WA. Can the ubiquitous power of mobile phones be used to improve health outcomes in developing countries. Global Health 2006; 2(9).
Burke RD, Hammond KJ, Kulyukin V, Lytinen SL, Tomuro N and Schoenberg S. Question answering from frequently asked


Evans RE. The impact of voice characteristics on user response in an interactive voice response system 2009, Masters dissertation, Rice University, Houston, TX.


