An Enhanced Modified Application on Mobile Searching (SMS-Search)

Pavan Kumar Doppalapudi
Research Scholar,
Shri Venkateswara University ,Gajraula, (UP)

Dr. Rajeev Kumar
Dept. of Computer Science & Engineering,
Shri Venkateswara University, Gajraula, (UP)

Abstract - In 21st century, SMS-supported mobile searching applications are rapidly more widespread around the globe, particularly in current consequence among users with High-end smart phones. So, web exploration is one of the main and valuable of these applications. SMS (Simple Mail Service) supported web exploration takes unstructured queries and returns web snippets via SMS. This application permits users with low-priced mobile phones and no data plan to hit into the enormous amount of information on the web. SMS-supported web exploration is a demanding problem since the channel is both tremendously restricted and costly. Preferably, together the query and the response fit within an SMS. This paper presents the blueprint and execution of SMS-Search, an SMS-supported search application that facilitates users to achieve exceptionally crisp (one SMS message) search responses for queries across random keywords in one round of communication. SMS-Search is intended to balance existing SMS-supported search applications that are moreover inadequate in the keywords they identify or engage an individual in the loop.

Specified an unformed search query, SMS-Search, uses a usual search engine as a back-end to extract a number of search responses and uses a mixture of information retrieval procedures to take out the most suitable 140-byte snippet as the final SMS-search reply. We illustrate that SMS-Search returns suitable responses for 62.5% of Mahalo search queries in our experiment set; this precision rate is high given that Mahalo make use of a human to respond the identical queries. We have also installed a pilot edition of SMS-Search for make use of with a little featured group and a bigger scale open beta in Moradabad to investigate our application and contribute to our understanding.

Keywords: SMS, Mahalo, Caching, Unstructured search query

I. ENTHUSIASM

The outstanding development of the smart phone market has provoked the blueprint of new types of mobile information applications. With the development of Twitter, SMS GupShup and other community messaging networks, the precedent few years have observed a rising occurrence of Short-Messaging Service(SMS) based applications and services. SMS-supported applications are also more and more frequent in poor nations. Regardless of the growing influence of mobile devices with the introduction of "smart phones", a significant portion of mobile devices in current scenario are still straightforward economical devices with restricted processing and communication potential. Due to a combination of social and economic issues, voice and SMS will expected prolong to stay the main communication channels existing for a non-trivial portion of the people in current scenario. For any SMS-supported web service, efficient *SMS-supported search* is an important building block. SMS-supported search is a speedily rising worldwide market with over 12million subscribers as of July 2008 [Critical Mass: The Worldwide State of the Mobile Web. 2008]. An SMS message is limited to 140 bytes, which severely confines the amount of information in a search reply. SMS-supported search is also non-interactive due to the search reply time; anecdotally [Pump up the volume: An assessment of voice-enabled web search on the i-phone], existing SMS-supported search engines acquire on the order of tens of seconds [Google SMS, Yahoo One Search, Bing, msn] to several minutes per response [Mahalo]. Still not including the 140-byte SMS size restriction, extended conventional web search to mobile devices is a difficult problem due to the small form issue and highly intermittent environment. Unlike desktop search, clients on mobile devices hardly ever have the comfort of iteratively refining search queries or filtering through pages of outcomes for the information they want. In this paper, we concentrate on the difficulty of *SMS-supported search*: *how does a mobile client efficiently explore the web using one round of communication where the search reply is constrained to one SMS message?*

Despite the fact that we do not know the internals of existing SMS Search algorithms, we can examine from the client interface and credentials that existing automated applications for SMS web search such as Google SMS and Yahoo! One Search support clients to enter queries for an amount of pre-defined keywords. These pre-

defined keywords are either identified during the utilization of particular keywords within the search query such as "describe" or "planet" (e.g. Yahoo SMS: "describe planet") or have specific parsers to find out which of the keywords is intentional (e.g. querying "RAW" to Yahoo! One Search is a query for information about "intelligence unit"). Mahalo [Mahalo], a latest SMS-supported search engine, employs peoples to explore the web and respond queries in an SMS reply. TradeNet, now called eSoko [Esoko], is a mobile market place platform in Ghana that would need an SMS based search service as well. Immediate access to small bits of information is the enthusiasm at the back all of the existing SMS based search applications and of this effort. None of the existing automated SMS search applications is an absolute resolution for search queries across random keywords. Similar to conventional web search queries, SMS search queries undergo the *long extended trend*: there exists an extended of search queries whose keywords are not accepted (e.g. "what are schooling gift ideas?" or "what chemicals are in mosquitoes repellent"). We confirm that this undoubtedly is the instance in our sample of Mahalo questions where only 27% of the queries in our data set are verticals (as defined by Google SMS) and 73% are long extended. In this paper, we explain the blueprint and execution of SMS-Search, an SMS-supported search engine specifically intended for the long extended of search queries that are stretched across a large collection of keywords. These keywords correspond to the queries in a mobile search workload that are not responded by existing domain specific verticals. SMS-Search is intended to incorporate into an existing SMS search application to respond queries for unsupported long extended keywords. Given a query, SMS-Search employs a conventional search engine as a back-end to extract a number of search outcomes and take out the suitable 140 bytes as the SMS search reply. Section 5.2 further elucidates how vertical and long extended queries are defined in this effort.

II. SMS-SEARCH SETBACK

SMS-Search uses a grouping of well-known information retrieval procedures to deal with the suitable information mining problem. SMS-Search is intended for *unstructured queries* supporting a related design as a customary search engine query. The key proposal of SMS-Search is that significant SMS queries normally include a term or a group of successive terms in a query that supplies a *clue* as to what the client is searching for. The clue for a query can either be unambiguously supplied by the client or automatically derived from the query. SMS-Search uses this clue to deal with the information mining trouble as follows: Known the top N search replies to a query from a search engine, SMS-Search mines *snippets* of content from within the neighbourhood of the clue in each reply page. SMS-Search scores snippets and ranks them across a Multiplicity of metrics. The design of our SMS-Search application (Figure 1) consists of a query server that handles the authentic search query and outcomes, and an SMS gateway that is accountable for communication between the mobile clients and the query server. The client is a user with a mobile phone who launch an SMS message to the short code (a particular number, shorter than full telephone numbers used to deal with SMS and MMS messages) for our application, which reaches at the SMS gateway and is then send out to our server for processing. At our query server the query is then sent to a common search engine and answer pages are downloaded. The query server mines the outcomes from the downloaded pages, and returns them to the SMS gateway that sends it back to the client that issued the call.

Identified Verticals vs Long Extended

Assured SMS based queries are most excellent responded by querying known verticals (e.g., Train, directions, agriculture). Google SMS search may be observed as a covering around its search application for a fixed collection of known groups such as mobile numbers, agriculture, train information, address etc. Our whole SMS search application consists of suitable parsers and vertical redirectors for a small number of known groups (mobile numbers, agriculture, addresses). For example, *trainenquiry.com* and *sulekha.com* are examples of such verticals for weather and yellow pages. For these groups with existing verticals, producing an SMS search reply needs a straightforward alteration of the query into a suitable database query (or filling a form) at the related vertical web portal. The centre of SMS-Search is to hold long extended queries that do not fit into verticals. Managing queries for verticals is a comparatively easy if monotonous process and suffers from speedily retreating returns. Our entire SMS search application supports a fundamental set of vertical keywords that is a subset of the Google SMS keywords. As an element of this structure, the SMS-Search algorithm is positioned at the back of a categorizer that first resolves whether a specified query belongs to an employed vertical based on the existence of defined keywords or if it should be sent to SMS-Search.

SMS-Search Exploration Problem

SMS-Search is intended for unstructured search queries across random keywords. Given any query, SMS-Search first employs an existing search engine as a back-end to acquire the top few search outcome pages. Using these pages the remaining problem is to parse the textual content to acquire the suitable search answers that can be reduced to one SMS message. This is basically a problem of suitable reduced snippets of text across the outcome pages that form candidate SMS search answers. We define a *snippet* as any uninterrupted stream of text that fits inside an SMS message. SMS-Search uses a clue for every query as a significant

clue to mine every outcome page for suitable text snippets. A *clue* refers to either a word or a collection of consecutive words within the query that is approximately analytic of what type of information the client is searching for. Specified that the clue will emerge in the answer page, SMS-Search uses the neighbourhood of the text around the clue to mine for suitable textual snippets. To elucidate our problem and algorithm we presume that the clue is given by the client unambiguously, but afterward we talk about how the clue may be automatically mined from usual language questions. The SMS-Search exploration problem can be described as follows: Given an unstructured SMS search query in the form of <query, clue> and the textual content of the top N search answer pages as returned by a search engine for "query", mine a reduced set of text snippets from the answer pages that fit within an SMS message (140 bytes) that give a suitable search reply to the query. Where the clue is a word or collection of words found within the query. This problem definition unambiguously presume that the clue is specified for every query. Existing SMS-supported search applications like Google SMS have a similar unambiguous prerequisite where a keyword is specified as the very last word in a query; the differentiation being that the existing applications only maintain a fixed set of keywords whereas SMS-Search allows random clues.

Our Contribution : To the best of our knowledge we are the first to handle issues relating to SMS based automatic question-answering. We address the challenges in building a FAQ-based question answering system over a SMS interface. Our method is unsupervised and does not require aligned corpus or explicit SMS normalization to handle noise. We propose an efficient algorithm that handles noisy and clarity.

III. SMS-SEARCH EXPLORATION ALGORITHM

In this part, we depict our algorithm to mine snippets for any given unstructured query of the form <query, clue>.

Fundamental Design

Search queries are essentially vague and a universal technique to disambiguate queries is to use supplementary contextual information, from which, the search is being performed. Insecurely, the word "context" is any supplementary information linked with a query that gives a helpful clue in providing a targeted search answer for a query [Chen, and Kotz, 2000, Lawrence, 2000]. Similar to these mechanisms, SMS-Search uses an unambiguous clue so the snippet mining algorithm can classify the estimated position of the preferred information in a search answer page. We encourage the use of a clue using a straightforward instance of a long extended query. Consider the query "Barack Obama wife" where the word "wife" correspond to the clue. When we give this query to a search engine, most search outcome pages will contain "Sonia" or "Sonia Gandhi" or "Sonia Sharma" or "Sonia Rakesh Sharma" within the neighbourhood of the word "wife" in the text of the page. For this query, to conclude any of these as suitable search answers, SMS-Search will explore the neighbourhood of the word "wife" in every outcome page and look or frequently occurring *n-grams* (where n corresponds to one to five repeated words). For example, "Sonia Gandhi" is a n-gram that is a 2−gram. A straightforward algorithm to conclude the acceptable answer to this example query is to yield all accepted n-grams within the neighbourhood of the clue and grade them based on different metrics (frequency, distance etc.). On the other hand, outputting frequently occurring n-grams as search answers is only suitable when the authentic search answer for a query is a $1 - 5$ word answer. For a number of general SMS-search queries, the authentic suitable search answer is embedded in a sentence or a collection of few sentences. In such cases, we require to mine entire snippets of text as a search answer as contrasting to just n-grams. SMS-Search makes a unambiguous distinction between *n-grams* and *snippets*. Although both correspond to uninterrupted sequences of words in a document, a n-gram is particularly short in length (1 – 5 words), while a text snippet is a sequence of words that can fit in a single SMS message. In our SMS-Search algorithm, n-grams are employed as an middle unit for our statistical techniques while snippets are employed for the final grading since they are suitably sized for SMS. We subsequently depict the fundamental SMS-Search algorithm. Consider a search query (Q, C) where Q is the search query including he clue word(s) C . Let $O_1, \ldots O_N$ correspond to the textual content of the top N search answer pages to Q. Given (Q, C) and $O_1 \ldots O_N$, the SMS-Search snippet mining algorithm consists of three steps: *Neighbourhood Mining:* For each outcome page Oi, SMS-Search explores for every appearance of the clue word C and mines a textual neighbourhood around the clue that corresponds to a candidate snippet of length covering one SMS message in either side of the clue. For each snippet, we mine all exclusive n-grams with up to 5 words. The selection of the limit 5 is provoked by the reality that the Linguistic Data Consortium (LDC) [Linguistic data consortium] publishes the web occurrence of each n-gram with up to 5 words. *N-gram Grading:* We grade the n-grams based on three metrics: distance to the clue, frequency of occurrence, and average grade of the outcome page. We also use the comparative infrequency of a n-gram on the web to standardize the n-gram grading.

Snippet Grading: We define the grade of any snippet as a collective summation of the top-few graded n-grams within the snippet. Among all snippets, we conclude the top-graded snippet(s) as the search answer. We now

intricate upon these three steps.

Neighbourhood Mining

Given a query (Q, C) and its outcome pages $O_1, \ldots O_N$, SMS-Search first mine snippets around the clue C in each of the pages. For each page Oi, we find every happening of the clue C in the page. Each snippet is up to 140 bytes long and the clue is as cantered as much as the neighbouring text allows. We establish that delineating snippets by sentence boundaries direct to many corner cases due to sound that could distort the statistical outcomes. The snippets are then combined if they overlap to keep away from double counting into *snippet tiles* (Figure 2). These snippet tiles structure the foundation of all additional measurements and calculations, and it is only within these snippet tiles that the final outcome is mined. From a realistic perspective of not needing to download a number of pages and having sufficient variety to mine n-grams and snippets, we establish that a value of N = 10 works fit. We mine the text from these web pages by filtering out all characters, hypertext tags, and non-ASCII ciphers. The outcome is simple text that is similar

to what would be provided by an ordinary browser. Clue Original Text Snippets N-Grams Snippets Tiles comparatively often and in close proximity of the clue "wife" for the top search answer pages to the query "Rajiv Gandhi wife". As a result, this n-gram is extremely considerable for the search answer for the query. For each exclusive n-gram in any snippet, we calculate three autonomous measures.

N-gram grading:

N-gram grading is a significant step in the SMS-Search snippet mining algorithm. Given any snippet mined around the clue, the first step in our n-gram grading algorithm is to assemble all probable n-grams in the snippet. Table 1 demonstrates briefly how the n-grams are produced. The objective of the n-gram grading algorithm is finding the n-grams that are most probable to be associated to the accurate answer. The underlying principle of our n-gram grading algorithm is that any n-gram that satisfies the subsequent three properties is potentially associated to the suitable answer for a query with a specified clue:

1. the n-gram become visible very often around the clue.
2. the n-gram become visible very close to the clue.
3. the n-gram is not a universally used popular word or expression.

As an example, the n-gram "Sonia Gandhi" is not a universally used expression and come into sight

Table 1: Slicing example for the text "the quick fox jumped over the lazy dog", clue = "over".

N-Gram	Occurrence	Least distance
"the"	2	1
"the quick"	1	3
"the quick fox"	1	2
"quick fox jumped"	1	1

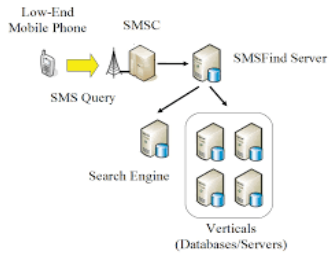

Fig.1. Basic Low-level Mobile to vertical Databases.

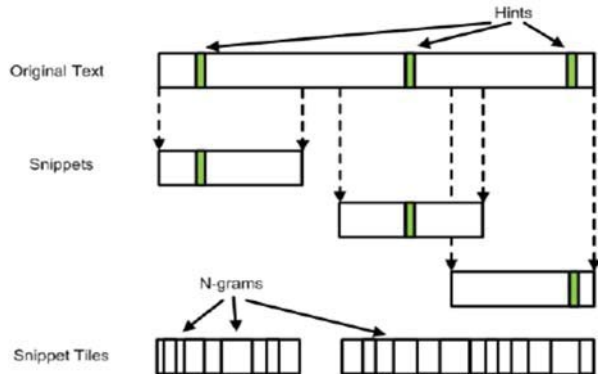

Figure-2-Snippet-creation-aggregation-into-snippet-tiles-and-n-grams.

Occurrence - The number of times the n-gram arises across all snippets.

Average grade - The summation across every happening of a n-gram of the Page Grade of the page where it occurs, divided by the n-gram's raw occurrence. This is to include the grading scheme of the fundamental search engine in our overall grading function. (Some n-grams have a raw average grade of less than 1 because the page enclosing the search outcomes is allotted a grade of 0.)
Least distance - The least distance between a n-gram and the clue across any happenings of both. Spontaneously, this metric shows the proximity of the clue defined by the client is to the search query. It is used as an element of our overall grading function to let the client clue to disambiguate two otherwise likewise graded n-grams. An instance of the metrics we have at this point is shown in Table 2. In this instance, the question "wearable computer for head-mounted display" should return the answer "Google Glass" as highlighted in the table. Note that this illustration is precisely in the range of queries that we are concerned in, it is too exceptional for a conventional miner and universal enough to be demonstrable by our scheme. From the catalogue of n-grams we can scrutinize that after slicing, most of the top outcomes are highly significant to the query according to our metrics.

Table 2: list of top 5 n-grams results for the query "wearable computer for head-mounted display" and their associated raw metrics prior to normalization.

N-Gram	Occurrence	Least distance	Average Grade
Google	12	1	1.2
Glass	12	1	1.14
Google Glass	11	1	0.89
Wearable computer	10	2	1.08
Head-mounted	9	2	1.76

Filtering n-grams: Before grading n-grams, we filter the set of n-grams based on the three measures: occurrence, average grade and least distance. A n-gram should have a least occurrence and should be within a definite least distance of the clue to be measured. For N = 5, we set a least frequency bound of 3 and a least distance threshold of 10; we prefer these thresholds experimentally based on manual investigation of n-grams across sample queries. Correspondingly, we pay no attention to all n-grams with a very low average Page Grade. Grading n-grams: Correlating comparative significance to any of the metrics or idealistically grading based on a particular metric is not suitable. To combine the different metrics into a single metric we carry out two straightforward steps. First, we standardize the raw occurrence, average grade, and least distance to a homogeneous distribution within the range between 0 and 1. We indicate the three standardized scores of a n-gram *s* as occr(s), avg-grade(s), least-dist(s). Second, the overall grading score of a n-gram *s* is a linear combination of the three standardized grades:

grade(s) = accr(s) + avggrade(s) + leastdist(s) …….. (1)

Figure 3: Mahalo number of queries per group as available on Mahalo's website [Mahalo].

To Communicate with more than one through programs

We use the grading score to grade all n-grams related with a query. We need to regard as one significant feature

in the standardization of the occurrence score. If two n-grams, *t* have the same occurrence measure but if n-gram *s* has a much lesser web occurrence than n-gram *t* (*s* is unusual than *t*), then *s* desires to be higher graded than *t*. We use the "Web 1T 5-gram Version 1" dataset from the LDC to attain the occurrence for any n-gram and calculate its standardized occurrence.

Snippet Grading Algorithm

If the reply to a query is a very small answer of a few words, then the finest SMS based search answer is to output all the top-graded n-grams related with a query. Though, given a query, it is hard to conclude whether the reply is embedded in a single n-gram or is truly a grouping of multiple n-grams. In this scenario we yield the most excellent snippet under the expectation that the reply is embedded in the snippet and the client can understand it. The n-gram grading algorithm cannot be extended to grading snippets since approximately all of the snippets are exclusive, and their occurrence measure will be 1. We enlarge the n-gram grading algorithm to compute the grade of snippets based on the n-grams present within a snippet. In addition, different snippets may enclose different number of n-grams that may bring in a prejudice in the grading function. To keep away from such a prejudice, we bring in a top-R n-grams based grading function. Snippet grade: Consider a snippet S with a set of n-grams $O = \{o_1, \ldots o_m\}$ with corresponding n-gram grades $grade(o_1), \ldots grade(o_m)$. Let $o_{i1}, \ldots o_{iR}$ correspond to the top R graded n-grams in O . Then the grade of snippet S is:

In other words, we define the grade of a snippet based on the collective grade of the top-R graded n-grams within the snippet. In practice, we select R = 5. In the *snippet grading* stage, we conclude the maximum graded snippet is the most significant answer to the original query. Remember that our snippets were intended to be under 140 bytes, but the snippet tiles may in fact be longer depending on overlaps during the integration process. To find the most excellent snippet, we first split each snippet tile into snippets using a 140 byte sliding window across every snippet tile that compliments word restrictions. We then score each snippet based on the summation of the top R n-grams and return the top scoring snippet as the final outcome. In the estimation, we show that grading n-grams first before grading snippets is important for enhanced precision; in other words, straightforwardly scoring snippets from the web page outcomes without using n-grams performs very disappointingly in practice.

IV. EXECUTION

The foundation SMS-Search algorithm is executed in only 597 lines of ASP code and uses widely accessible parsing libraries. We have not executed any optimizations or caching, but SMS-Search normally returns outcomes within 4-10 seconds while running on a 2.4Ghz Intel Core i3 PC with 4 GB of RAM and a 2 Mbps broadband connection. This response time is dominated by the time essential to obtain query outcome from Google and download web pages referred to by the outcomes. To install SMS-Search as a search application we employed a front-end to send and receive SMS messages. We setup a SMS short code, and direct all SMS requests and replies to and from our server machine running the application across a Samba 75 GSM modem and Kannel an open source SMS Gateway [Kannel]. As a verification of idea, and to advance the overall usability of our organization we have also employed interfaces to several basic verticals as a part of the application including: agriculture, describe, local business results, and news. Each of these interfaces to verticals is under 200 lines of ASP code.

V. ASSESSMENT

Entirely assessing such an application is non-trivial; a huge set of reasonable queries must be asked, and replies must be evaluated either by evaluators or the clients themselves. We obtain a practical set of queries by transforming actual Mahalo queries. We then assess our structure performance in complete words to explore the restrictions of our structure and probable enhancements. In our assessment, the replies returned by SMS-Search are evaluated acceptable *if the all of the exact answer words emerge anywhere in the returned snippet*. The accurate answer ere first determined by three evaluators who came up with exact answers by manual scrutiny of each question, referring to the Mahalo reply for reference. Occasionally even the Mahalo reply was considered wrong by the evaluators and a different reply was concluded online and used in its place. There were also queries that had a number of suitable answers, we consider all of those "accurate" in our assessment. This is the straightforward rating structure we use throughout the assessment. We do not at present grade our scores based on other significant features such as readability or precision. For a full-fledged question/reply structure, including methods to advance readability [Kanungo, and Orr, 2009] would likely be essential.

Statistics

Our set of queries consists of a mass of 10,000 actual SMS search queries scraped from Mahalo's [Mahalo] unrestricted website on August 22, 2012. These queries are in Normal Language query layout. In comparison to the logs evaluated earlier studies we establish in our statistics an average of 8.90 words per query and 43.17 characters per query, in contrast to 3.05 and 18.48 respectively. The TREC [Trec question answering track] query/reply track datasets are based on search engine queries, and TREC-9 datasets were authentic clients' queries. Conversely, it is ambiguous whether they were collected from mobile devices. From earlier

studies it is obvious that mobile search logs have significant differentiations across lots of characteristics depending on the search means and device [Kamvar, and Baluja, 2006, Kamvar, Kellar, Patel, and Xu, 2009]. Queries from Mahalo are mobile (either voice or SMS) and the replies are created by a peoples and returned via SMS. Consequently, we selected to use the Mahalo dataset due to its high practicality for query types and their allocation across keywords in a mobile background. We merely use all 10,000 queries for our comprehensive investigation of query length and keyword distributions. Since our assessment consists of both rewriting, and manual evaluation of queries and replies, both of which are effort intensive, we were not capable to carry out comprehensive assessment with the full set of 10,000 queries. In its place, an arbitrarily chosen 2,000 query subset is used for our more comprehensive manual assessments. Out of these 2,000 queries, we found through manual investigation that 1526 are long extended.

Query Keywords and Recognizing the Long Extended

In the majority of query log studies, the queries are classified according to keyword to give an illustration of the types of queries being put forwarded. Even though this type of classification is valuable for that point, for our assessment these categories do not facilitate in choosing whether queries are part of the long extended of queries we

Wish to assess. Figure 3 demonstrates the available queries per group on Mahalo's website [Mahalo categories]. This structure of classification by keyword is similar to earlier studies [Kamvar, and Baluja, 2006, Kamvar, Kellar, Patel, and Xu, 2009]. From these groups it is ambiguous whether keywords such as "Entertainment" and "Travel" could map to verticals or are excessively extensive and should be directed to our structure. Moreover dividing keywords into sub-keywords (using Mahalo's available breakdowns) exposes that several of these finer granularity of sub-keywords are straightforwardly map able to verticals: e.g. Yellow Pages" and "Definitions" may be mapped straightforwardly to data sources "wikipedia.com" with little endeavour. Other sub-keywords are still too extensive for a straightforward vertical (e.g. the sub-keywords of "Health such as "Travel" or World Governments"). Each query from Mahalo has an related set of keywords it belongs to as allocated by Mahalo. Table 3 lists the top 10 most vastly represented sub-keywords that identified in our illustration (127 keywords total), the proportion of queries belonging to these sub-keywords, and whether there are equivalent verticals that are executed in any existing SMS search methods. The sum of the proportions of the entire table do not add up to 100 because queries may belong to more than one keyword. Certain keywords such as "Celebrities" have online databases such as Internet Movie Database (IMDB) that could effortlessly be used as a resource of structured information to respond queries, and executing verticals on top of these would not be difficult. Using the sub-keywords, we detached queries from our 2,000 query mass that do not have accessible verticals in any existing automated SMS search structure to recognize the long extended queries (1526 long extended queries).

Table 3: Mahalo top sub-keywords by proportion and existence of probable verticals.

Topic	% of Total Questions	Existing Verticals?
Celebrities	12.4	No
Yellow Pages	9.8	Yes
Movies (not show times)	10.1	No
Music	9.2	No
Definitions	11.1	Yes
Relationships	8.3	No
Food & Drink	6.5	No

These remaining queries are precisely long extended questions SMS-Search is intended to answer. Several examples are listed in Table 4. The query allocation across sub-keywords and the variety of queries in general proposes that mobile user information requirements even on low-end devices are more varied than formerly demonstrated [Kamvar, Kellar, Patel, and Xu, 2009]. This may be due to the more significant input modality of voice or the user's observation of the structure as being extremely intellectual. We reschedule a more concluded relationship between these mobile search patterns and those in the literature to prospective studies.

Baseline Assessment

We carry out a few baseline assessments of our application using our sample of 2,000 queries from

Mahalo containing both vertical and long extended queries. We alter these queries assuming the client understands how to use the application and is prepared to enter keywords along with a clue word rather than a normal language query. For each Mahalo query we rewrite the query exclusively by removing and rearranging words given that we have understanding of how the application works and be familiar with what would likely be a superior alternative for the clue word. As an example, "what are the symptoms of HIV Infection" is rewritten as "HIV Infection symptoms" and the entire query is submitted to Google with "symptoms" used as the
clue word. Using these queries we find that our application (including redirection to available verticals) outcomes in 62.5% correct answers. In contrast, Google SMS returns accurate outcomes for only 9.5%

Questions :

Where is the world largest railway platform ?
Which of the disciplines is Nobel Prize awarded?
Who was Archimedes?
First China War was fought between which country ?
What is the dimensions of football court ?
On which date the World Red Cross and Red Crescent Day is celebrated ?
Film and TV institute of India is located at which place ?

Table 4: Example queries from Mahalo (together with spelling errors).

Input, Output	**% Correct**
Mixed, Snippets	**62.5**
Long Extended, Snippets	**56.4**
Long Extended, N-Grams	**23.7**
Long Extended (w/clue), TF-IDF Snippets	**19.9**
Long Extended (w/out clue), TF-IDF Snippets	6.1%
Long Extended Unmodified Queries, Snippets	15.9%
Long Extended Unmodified Queries, N-Grams	6.7%

Example for SMS Gateway – Adv. Ver. Of SMSs

of these queries. We note that the low performance of the Google SMS could be due to a multiplicity of causes that we discuss in Section 7, and the consequence is presented here only for reference. What is more motivating is if we eliminate the queries that are forwarded to existing verticals. To focus on the core SMS-Search algorithm, we think about only the performance of the SMS-Search algorithm on the 1526 long extended query subset. All additional assessment in this section is carried out with this set of 1526 long extended queries. Table 5 summarizes the outcomes of our assessment. We find that for the long extended queries SMS-Search returns 56.4% correct results. Moreover, if we think about only the uppermost n-grams returned rather than the whole snippet, the performance fall to 23.7%. These outcomes largely specify that the raw performance of SMS-Search has significant room for enhancement, and returning snippet replies usually results in enhanced performance. We observed before that there are concerns with readability, but what do the snippet outcomes in fact look like? A representative set of examples is shown in Table 6 for both accurate and inaccurate outcomes. Contrasted to the Mahalo individual written replies we examine that the readability of our snippets is poor and could be enhanced; On the other hand, finding the most favourable snippet structure is a separate problem where existing methods could be useful [Kanungo, and Orr, 2009].

Is Refinement Using N-grams Profitable?

Because we are returning snippets rather than n-grams, it is normal to request whether n-grams are essential or if grading snippets alone would execute just as well. To confirm that the middle refinement phase using n-grams is profitable we compare beside a straightforward per-word TF-IDF approach. In the first part of this trial we collect SMS sized snippets straightforwardly using a 140 byte sliding window across all documents. In this second part of the trial, we do the same thing excluding the snippets (d_j) are scored based on the summation of the standard TF-IDF scores as the metric for each of the i words (o_i) of the query. We find that with merely TF-IDF, the accurate outcome is returned in only 6.1% of queries. The outcomes of the per-snippet correctness are

shown in Table 5 for assessment. With the clue word used in combination with TF-IDF 19.9% of queries are appropriately returned. On the whole, both raw TF-IDF and TF-IDF with a clue word return inferior outcomes than with the middle refinement using n-grams.

Returning Several Snippets

It is imaginable that for the queries that are not replied well, such as unclear queries or clarifications, somewhat longer snippets or additional outcomes could improve our results. The application may permit several SMS messages to be requested in the form of a "supplementary" link. To investigate this prospect we trialed with returning several snippets. We compare two dissimilar snippet selection techniques to improve outcome multiplicity. In both techniques, we first arrange the snippets by grade. The first technique is to return a snippet for each of the top graded n-gram outcomes. The second technique returns only snippets for the top graded n-gram outcome, but needs that the returned snippets are from different occurrences of the clue. We find that as more outcomes are returned, the number of queries replied boost slightly (1 - 5%) for each supplementary reply returned. Both techniques demonstrate a similar rate of enhancement, but the first technique of maximizing n-gram multiplicity constantly performs enhanced by around 10%.

VI. RELATED WORK

There has been comparatively small research on SMS-supported search. In this part, we take somewhat superior vision of the problem space and compare SMS-Search with mobile search applications, query/reply (Q/R) applications from the Text Retrieval Conference (TREC) community [Text retrieval conference], and text summarization systems. Distribution of the metadata and the requests to relatively few nodes suffices to achieve a high probability of a match. Moreover, the strategy is robust. Even if some of the randomly chosen nodes are subverted or non-operational, the probability of a match is high. Moreover, it is not easy for a small group of nodes to subvert the iTrust mechanisms to censor, filter, or subvert information.

Mobile Search Features

Mobile search is a primarily different search standard than conventional desktop search. Yet, we carry on to view mobile web search as either an expansion of the desktop search model for high-end phones (such as PDA/iPhone) or somewhat limited version through XHTML/WAP on low-end devices. Mobile search in both of these settings be different from conventional desktop search in a number of ways as revealed in latest studies by Kamvar et al. [Kamvar, and Baluja, 2006, Kamvar, Kellar, Patel, and Xu, 2009]. The first study [Kamvar, and Baluja, 2006] found that the mobile search click-through rate and the search page views per query were both significantly inferior in contrast to desktop search. Implication, most mobile search clients be inclined to use the search application for short time-periods and are either satisfied with the search engine snippet replies or do not find what they were searching for. The study also establish that the determination of mobile users was incredibly low signifying that the huge mass of mobile searchers move toward queries with a specific keyword in mind and their search frequently does not direct to investigation. The second study [Kamvar, Kellar, Patel, and Xu, 2009] showed that the multiplicity of search keywords for low-end phone clients was a lot fewer
than that of desktop or iPhone-based search. This outcome recommends that the information requirements are extensive, but are not satisfied by the information applications available on low-end phones. We find supported indications in our analysis of Mahalo queries that mobile query multiplicity across keywords is elevated given a more easy-to-read input modality such as voice. As a whole, these studies signify a imperative need for rethinking the existing mobile search model for low-end mobile devices.

VII. CONCLUSION

In this paper, we have demonstrated SMS-Search, an computerized SMS-supported search response application that is customized to work across random keywords. We find that a mixture of straightforward Information Retrieval algorithms in combination with existing search engines can make available practically precise search responses for SMS queries. Using queries across random keywords from a real-world SMS query/reply service with human-in-the-loop responses, we demonstrate that SMS-Search is capable to answer 62.5% of the queries in our test collection. We also demonstrate how SMS-Search is integrated into an overall SMS-supported search application, and reveal its applicability in an open public beta in Muradabad, India. Although more influential Information Retrieval and Natural Language Processing (like LISP) techniques are bound to enhance performance, this work correspond to a venture into an open and realistic research domain. In addition, our techniques may be realistic to other constrained Information Retrieval over unstructured documents away from SMS-supported web search.

REFERENCES

[1] Rudy Schusteritsch, Shailendra Rao, Kerry Rodden. 2005. Mobile Search with Text Messages: Designing the User Experience for Google SMS. CHI, Portland, Oregon.

[2] Sunil Kumar Kopparapu, Akhilesh Srivastava and Arun Pande. 2007. SMS based Natural Language Inter-face to Yellow Pages Directory, In Proceedings of the 4th International conference on mobile technology, applications, and systems and the 1st International

symposium on Computer human interaction in mobile technology, Singapore.
[3] Monojit Choudhury, Rahul Saraf, Sudeshna Sarkar, Vi-jit Jain, and Anupam Basu. 2007. Investigation and Modeling of the Structure of Texting Language, In Proceedings of IJCAI-2007 Workshop on Analytics for noisy Unstructured Text Data, Hyderabad.
[4] E. Voorhees. 1999. The TREC-8 question answering track report.
[5] D. Molla. 2003. NLP for Answer Extraction in Technical Domains, In Proceedings of EACL, USA.
[6] E. Sneiders. 2002. Automated question answering using question templates that cover the conceptual model of the database, In Proceedings of NLDB,
[7] B. Katz, S. Felshin, D. Yuret, A. Ibrahim, J. Lin, G.Marton, and B. Temelkuran. 2002. Omnibase: Uni- form access to heterogeneous data for question answering, Natural Language Processing and Information Systems, pages 230−234.
[8] W. Song, M. Feng, N. Gu, and L. Wenyin. 2007. Question similarity calculation for FAQ answering, In Proceeding of SKG 07, pages 298−301. Aiti Aw, Min Zhang, Juan Xiao, and Jian Su. 2006. A phrase-based statistical model for SMS text nor- malization, In Proceedings of COLING/ACL, pages 33−40.
[9] Grinter, R.E. and Eldridge, M. Wan2tlk?: Everyday text messaging. In Proc. CHI 2003, ACM, 441–448.
[10] GSM Association. SMS (Short Message Service). ttp://www.gsmworld.com/technology/sms/
[11] James, C.L. and Reischel, K.M. Text input for mobile devices: Comparing model prediction to actual performance. In Proc. CHI 2001, ACM, 365–371.
[12] Jones, M., Buchanan, G., and Thimbleby, H. Sorting out searching on small screen devices. In Proc. Mobile HCI 2002, LNCS 2411, Springer, 81–94.
[13] https://www.twilio.com/sms/features
[14] Global - Key Telecoms, Mobile and Broadband Statistics. www.budde.com.au, 2009.
[15] Windows Live Mobile. http://home.mobile.live.com/.
[16] Just Dial. http://www.justdial.com.
[17] D. Harman. Overview of the first text retrieval conference (TREC-1). In First Text Retrieval Conference (Trec-1): Proceedings, page 1, 1993.
[18] D. Carmel, E. Yom-Tov, A. Darlow, and D. Pelleg. What makes a query difficult? In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, page 397, 2006.
[19] H.T. Dang, D. Kelly, and J. Lin. Overview of the TREC 2007 question answering track. In Proceedings of TREC, 2007.
[20] A. Singhal and M. Kaszkiel. A case study in web search using TREC algorithms. In Proceedings of the 10th international conference on World Wide Web, page 716, 2001.
[21] C. Clarke, N. Craswell, and I. Soboroff. Overview of the TREC 2004 terabyte track. In Proceedings of the 13th Text REtrieval Conference, Gaithersburg, USA, 2004.
[22] J. Allan, B. Carterette, J.A. Aslam, V. Pavlu, B. Dachev, and E. Kanoulas. Million query track 2007 overview, 2007.
[23] E. Brill, S. Dumais, and M. Banko. An analysis of the AskMSR questionanswering system. In Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10, page 264. Association for Computational Linguistics, 2002.
[24] C.L.A. Clarke, G. Cormack, G. Kemkes, M. Laszlo, T. Lynam, E. Terra, and P. Tilker. Statistical selection of exact answers (MultiText experiments for TREC 2002). In Proceedings of TREC, pages 823–831, 2002.
[25] M.M. Soubbotin and S.M. Soubbotin. Use of patterns for detection of answer strings: A systematic approach. In Proceedings of TREC, volume 11. Citeseer, 2002.
[26] C. Aone, M.E. Okurowski, and J. Gorlinsky. Trainable, scalable summarization using robust NLP and machine learning. In Proceedings of the 17th international conference on Computational linguistics-Volume 1, pages 62–66. Association for Computational Linguistics, 1998.
[27] C.Y. Lin. Training a selection function for extraction. In Proceedings of the eighth international conference on Information and knowledge management, pages 55–62. ACM, 1999.
[28] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. Placing search in context: The concept revisited. In ACM Transactions on Information Systems, volume 20, pages 116–131, 2002.
[29] R. Kraft, C.C. Chang, F. Maghoul, and R. Kumar. Searching with context. In Proceedings of the 15th international conference on World Wide Web, pages 477–486, 2006.
[30] G. Chen and D. Kotz. A survey of context-aware mobile computing research. Technical report, Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College, 2000.
[31] S. Lawrence. Context in web search. In IEEE Data Engineering Bulletin, volume 23, pages 25–32, 2000.
[32] Linguistic Data Consortium. http://www.ldc.upenn.edu.
[33] X. Li and D. Roth. Learning question classifiers. In Proceedings of the 19th international conference on Computational linguistics-Volume 1, pages 1–7. Association for Computational Linguistics, 2002.
[34] Kannel. http://www.kannel.org/.
[35] T. Kanungo and D. Orr. Predicting the readability of short web summaries. In Proceedings of the Second ACM International Conference on Web Search and Data Mining, pages 202–211, 2009.
[36] J. Yi, F. Maghoul, and J. Pedersen. Deciphering mobile search patterns: a study of yahoo! mobile search queries. In Proceedings of the 17th international conference on World Wide Web, 2008.
[37] TREC Question Answering Track. http://trec.nist.gov/data/qamain.html.
[38] ChaCha Categories. http://www.chacha.com/categories.
[39] J. Mothe and L. Tanguy. Linguistic features to predict query difficulty-a case study on previous trec campaigns. In SIGIR workshop on Predicting Query Difficulty-Methods and Applications, pages 7–10, 2005.
[40] J.Lin and B. Katz. Question answering techniques for the World Wide W
[41] Dr. Hadeel Showket AlObaidy, "Building Ontology Web Retrieval System Using Data Mining," ed. English SMS Query from participants Transl8t! Lingo2Word SCORE Algorithm translation
[42] R. S. Monojit Choudhury, Vijit Jain, Animesh Mukherjee, Sudeshna Sarkar and Anupam Basu, "Investigation and modeling of the structure of texting language," IJDAR, vol. 10, pp. 157-174, 2007.
[43] S. P. Sumalatha Ramachandran, Sharon Joseph, Vetriselvi Ramaraj, "Enhanced Trustworthy and highQuality Infromation Retrieval System fro Web Search Engines," IJCSI International Journal of Computer Science Issues, vol. 5, 2009.
[44] M. K. Wadee S. Alhalabi, Moiez Tapia, "Search Engine ranking Efficiency Evaluation Tool," Inroads-The SIGCSE Bulletin, vol. 39, pp. 97-101, 2007.
[45] Available: http://www.seochat.com/c/a/SearchEngine-Optimization-Help/Mob Wht s hiv whats Wots hiv whats Wt s hiv whats whats Wts hiv Whats What is Whats wt is Wts ile-Search-EngineOptimization/
[46] G. Pass, et al., "A Picture of Search," First Intl. Conf. on Scalable Information Systems, 006.
[47] D. E. Rose and D. Levinson, "Understanding User Goals in Web Search," 2004.

[48] J. Chen, et al., "SMS-Based Contextual Web Search," presented at the Mobiheld '09 Barcelona, Spain, 2009.
[49] S. K. Samanta, et al., "Automatic language translation for mobile SMS," International Journal of Information Communication Technologies and Human Development (IJICTHD), vol. 2, pp. 4358,
[50] Chien-Liang Liu, et al., "Computer assisted writing system," Expert systems and applications, vol. 38, pp. 804-811, 2010.
[51] Critical Mass: The Worldwide State of the Mobile Web. The Nielsen Company, 2008.
[52] P. Bruza, R. McArthur, and S. Dennis. Interactive Internet search: keyword, directory and query reformulation mechanisms compared. In Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, pages 280–287. ACM New York, NY, USA, 2000.
[53] Linguistic Data Consortium. http://www.ldc.upenn.edu.
[54] M. Mitra, A. Singhal, and C. Buckley. Improving atomatic query expansion. In Proceedings of the 21st nnual international ACM SIGIR conference on Research and development in information retrieval, pages 206–214. ACM New York, NY, USA, 1998.
[55] OpenRosa. http://www.openrosa.org/.
[56] M. Paik et al. The Case for SmartTrack. IEEE/ACM Conference on Information and communication Technologies and Development (ICTD), 2009.
[57] G. Pass, A. Chowdhury, and C. Torgeson. A Picture of Search. In First Intl. Conf. on Scalable Information
[58] G. Chen and D. Kotz. A survey of context-aware mobile computing research. Technical report, Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College, 2000.
[59] CommCare. www.dimagi.com/content/commcare.html.
[60] W. Dorda, T. Wrba, G. Duftschmid, P. Sachs, W. Gall, C. Rehnelt, G. Boldt, and W. Premauer. ArchiMed: a medical information and retrieval system. Phlebologie, 38(1):16–24, 2008.
[61] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. Placing search in context: The concept revisited. ACM Transactions on Information Systems, 20(1):116–131, 2002.
[62] FrontlineSMS. http://www.frontlinesms.com/.
[63] GoogleSMS. http://www.google.com/sms.
[64] InSTEDDGeoChat. http://instedd.org/geochat/.
[65] R. Kraft, C. Chang, F. Maghoul, and R. Kumar. Searching with context. In Proceedings of the 15th Systems, 2006.
[66] D. E. Rose and D. Levinson. Understanding User Goals in Web Search. In WWW, May 2004.
[67] X. Shen, B. Tan, and C. Zhai. Context-sensitive information retrieval using implicit feedback. In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pages 43–50. ACM New York, NY, USA, 2005.
[68] J. Teevan, S. Dumais, and E. Horvitz. Personalizing search via automated analysis of interests and activities. In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pages 449–456. ACM New York, NY, USA, 2005. [69]YahooOneSearch. http://mobile.yahoo.com/onesearch.
[69] C. Yu, K. Lam, and G. Salton. Term weighting in information retrieval using the term precision model. Journal of the ACM (JACM), 29(1):152–170, 1982.
[70] www.zapmeta.co.in/SMS+Send+On+ Mobile
[71] SMSGupShup.http://www.smsgupshu p.com/.
[72] Critical Mass: The Worldwide State of the Mobile Web. The Nielsen Company, 2008.
[73] Pump Up The Volume: An Assessment of Voice-Enabled Web Search on the iPhone.
[74] http://www.mcubedigital.com/msearchgro ove/.
[75] GoogleSMS. www.google.com/sms.
[76] YahooOneSearch. http://mobile.yahoo.com/onesearch.
[77] ChaCha. http://www.chacha.com.
[78] T. Sohn, K.A. Li, W.G. Griswold, and J.D. Hollan. A diary study of mobile information needs. In Proceedings of the 26th annual SIGCHI conference on Human factors in computing systems, 2008. [79] Esoko. http://www.esoko.com/.
[79] Google_Squared. http://www.google.com/squared/.