

**METHOD AND SYSTEM FOR INCREMENTAL SEARCH WITH REDUCED
TEXT ENTRY USING A REDUCED KEYPAD WITH OVERLOADED KEYS**

Inventors:

Sashikumar Venkataraman

Rakesh Barve

Pankaj Garg

5 Pranav Rajanala

Murali Aravamudan

Ajit Rajasekharan

10 ABSTRACT

[0001] In accordance with one or more embodiments of the invention, a method and system are provided for incremental search with reduced text entry using a reduced keypad with overloaded keys where results are retrieved as user types in a “multiple term prefix input query” (501) by sequentially pressing once on each overloaded key embedding a character that constitutes one of the multiple terms of the query string. The method described in the present invention performs, during its indexing of the search space, a many-to-many mapping from the alphanumeric space of terms to numeric strings corresponding to the various prefixes of each alphanumeric term constituting the query string. A numeric string is one in which each alphanumeric character in the string is replaced by its corresponding numeric equivalent based on, in one embodiment, the twelve-key reduced keypad. This mapping scheme enables the system to incrementally retrieve all results matching the ambiguous alphanumeric input query, as the user types in each character. User does not have to explicitly specify the termination of each term to assist the system in disambiguating the

15
20

input query - input query could be prefix substrings from multiple terms. The system leverages off the multiple term prefixes constituting the input query to disambiguate it. The multiple term prefix based disambiguation method described in the present invention reduces the amount of text and steps user has to go through to enter a multiple term input query and
5 retrieve results, in comparison to existing character and word based disambiguation systems.

PRIOR ART

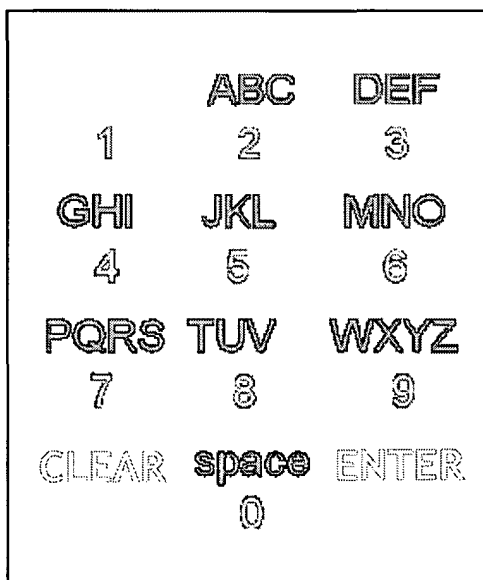
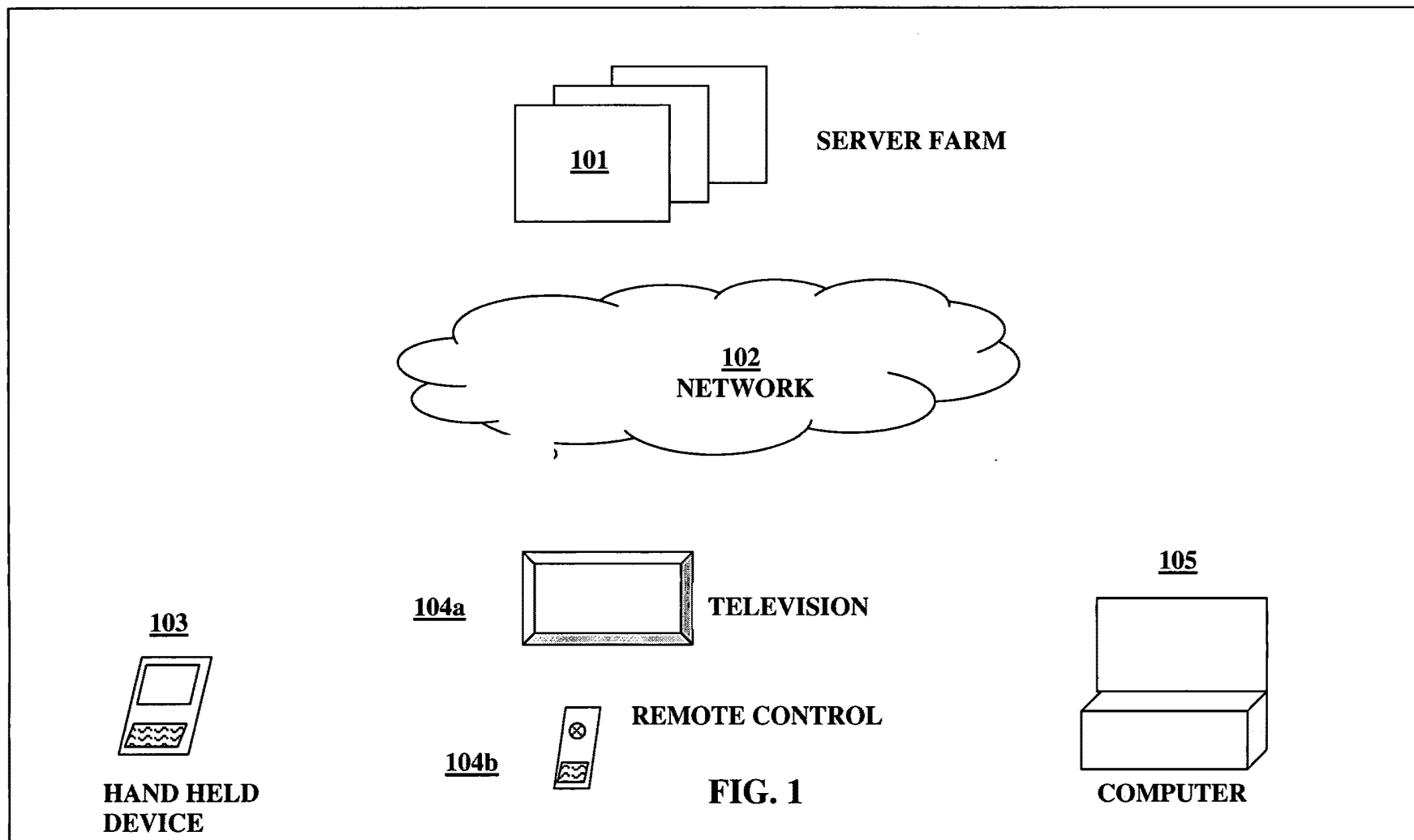


FIG. A



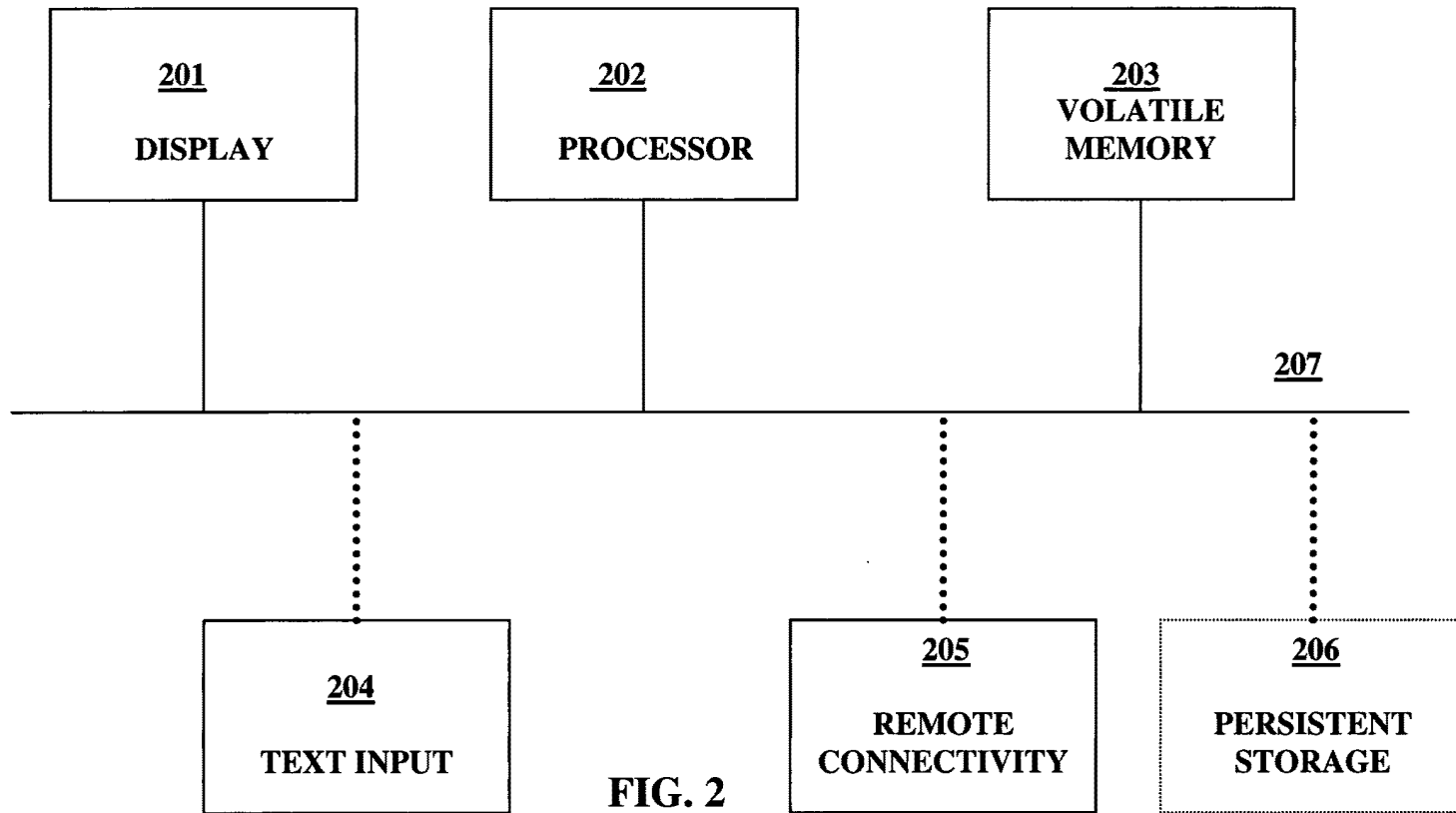


FIG. 2

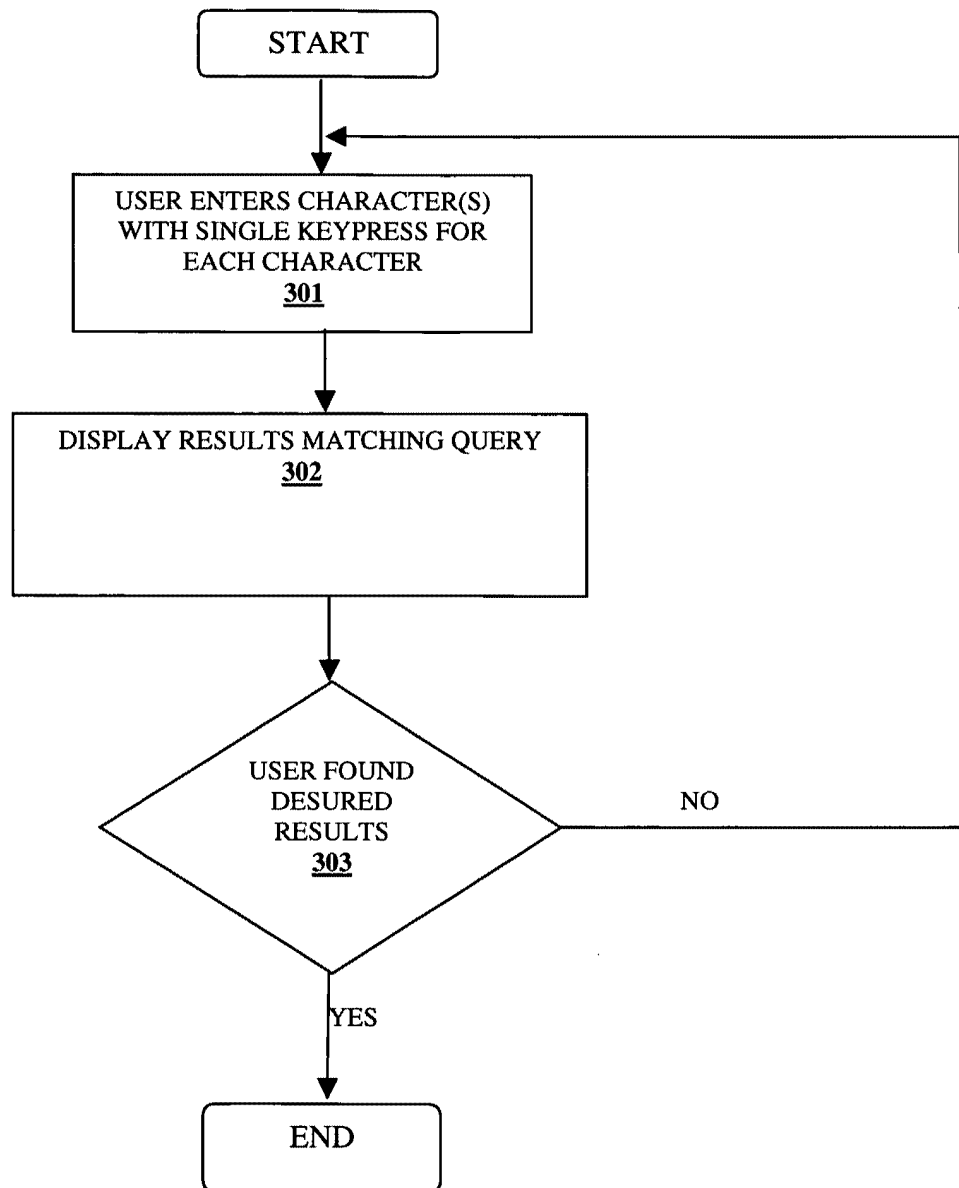


FIG. 3

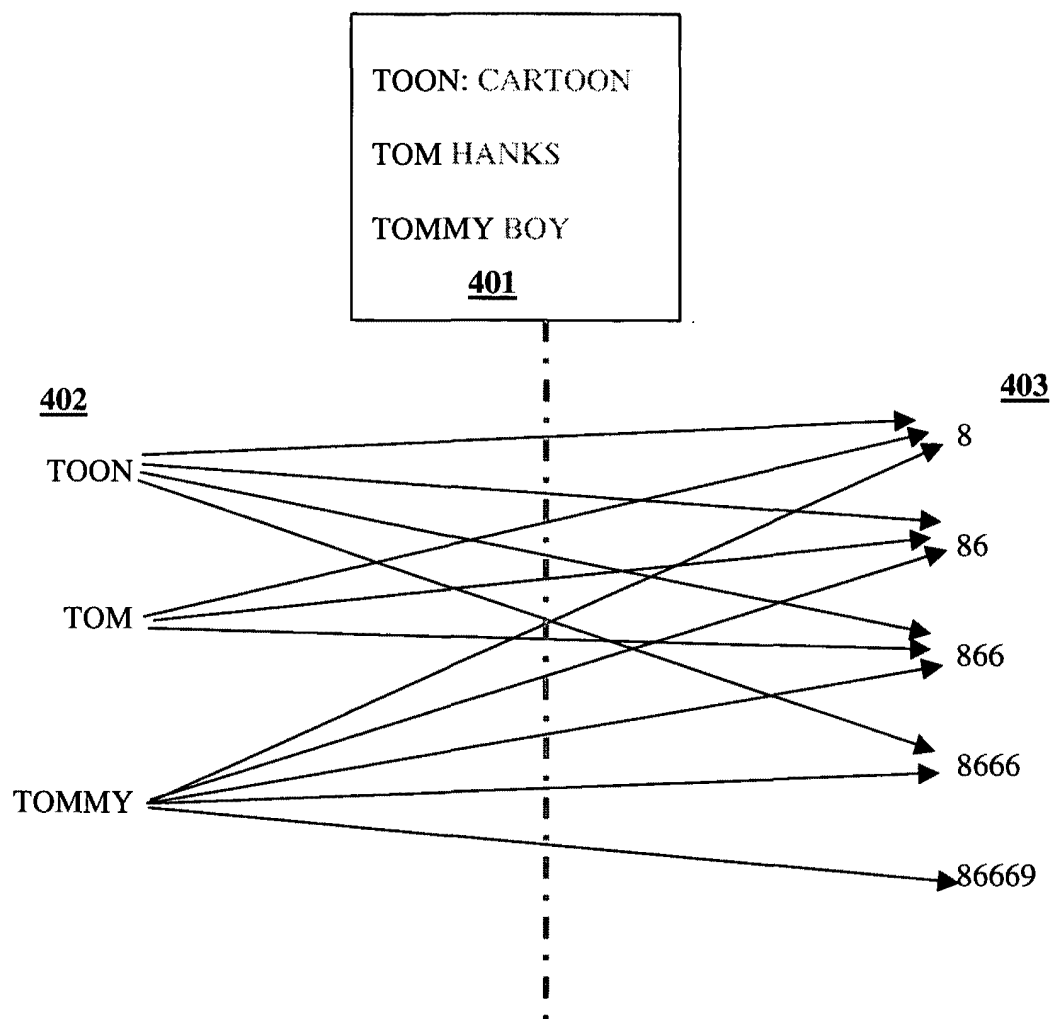


FIG. 4

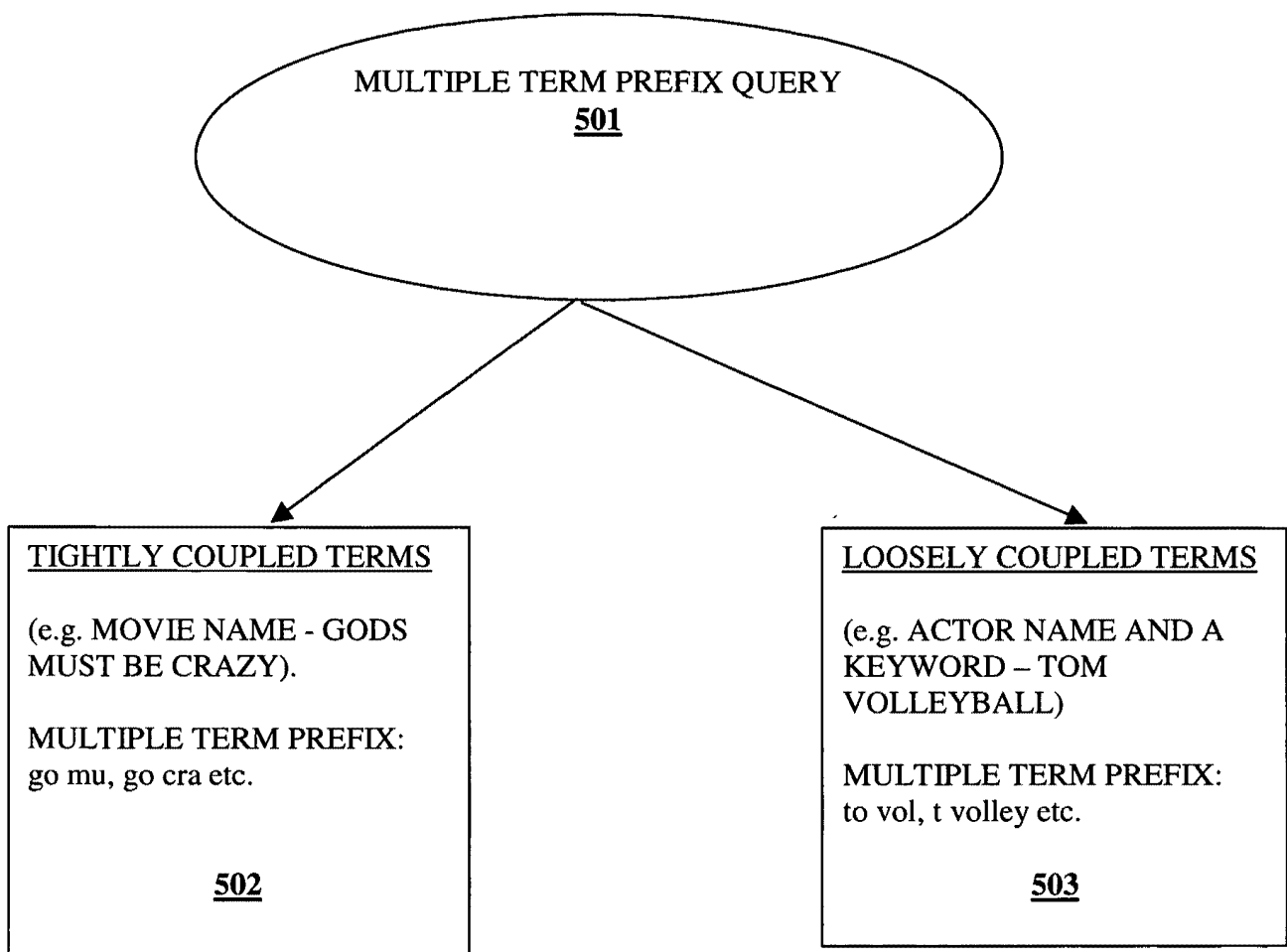
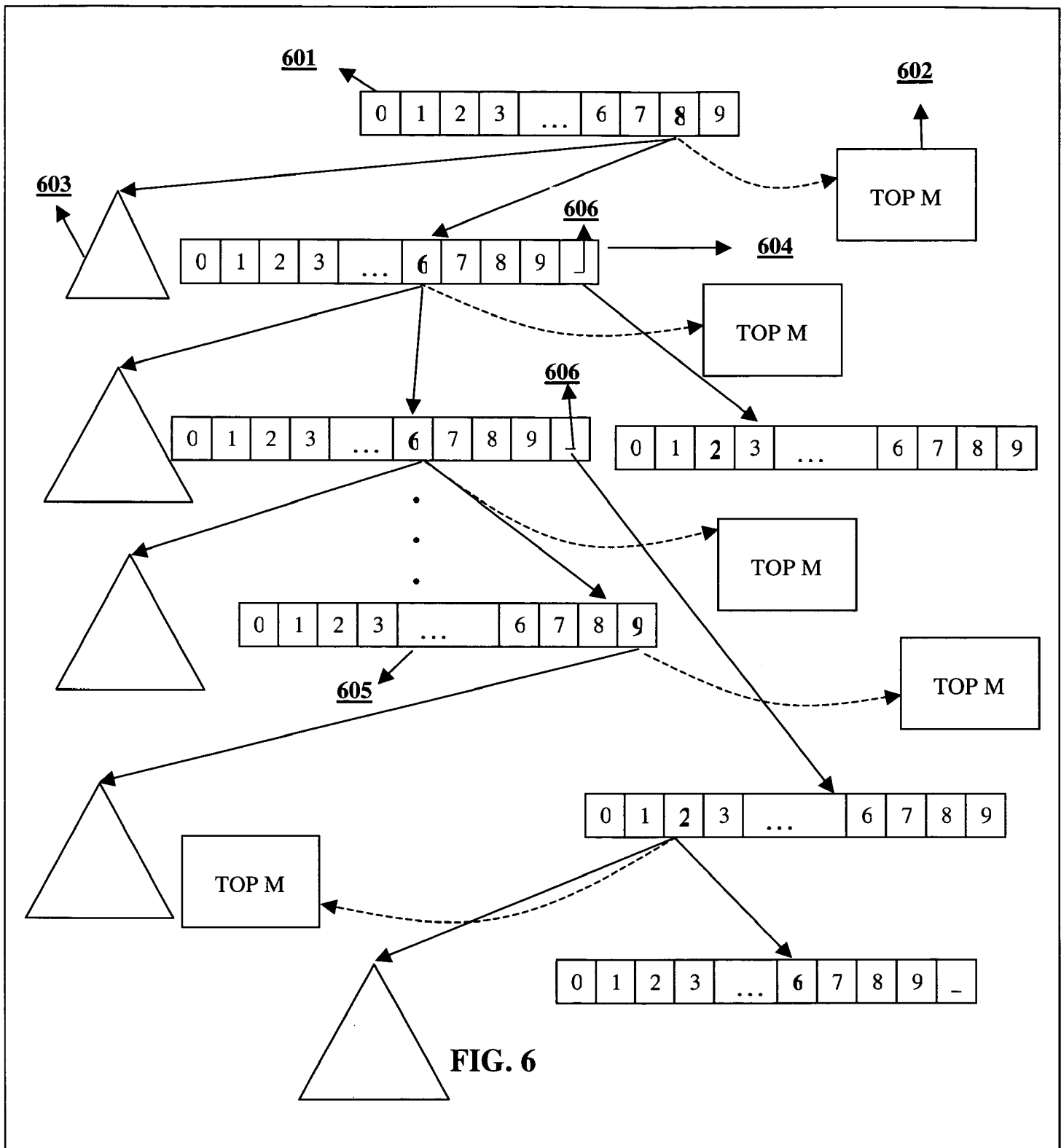


FIG. 5



702

The Koala Brothers: Archie's Loose Tooth; Pe

The Koala Brothers: George's Day Off; Archie

The Koala Brothers: Alice Rides Again; Ned ti

Charlie and Lola: I'm Not Feeling Well

Rockos Modern Life: Junk Junkies; Day of the

Rockos Modern Life: Born to Spawn; Uniform

Charlie and Lola: It's My Book

INPUT TERM: 5(J)

701

FIG. 7

802



The Koala Brothers: Archie's Loose Tooth; Pe
The Koala Brothers: George's Day Off; Archie
Rockos Modern Life: Born to Spawn; Uniform
The New Adventures of Winnie the Pooh: Ee
Charlie and Lola: My Wobbly Tooth
Charlie and Lola: The Most Wonderfulest Pic
Lizzie McGuire: Educating Ethan

INPUT TERM: 5 86(J TO)

801

FIG. 8

902

Tom and Jerry Kids: Circus Antics; Tres Sheil

Tom and Jerry Kids: No Biz Like Snow Biz; M

Tom and Jerry Kids: Cleocatra; Me Wolfenste

Tom and Jerry Kids: Zorrito; Deep Sleep Droc

Tom and Jerry Kids: Who Are You Kitten?; B

Tom and Jerry Kids: Catch That Mouse; Goo

Tom and Jerry Kids: Father's Day; Scourge o

INPUT TERM: 5 866(J TOM)

901

FIG. 9

BACKGROUND OF THE INVENTION

Field of Invention

[0002] The present invention relates to a method and system for incremental search with reduced text entry using a reduced keypad with overloaded keys where results are retrieved
5 as user types in the “multiple term prefix input query” by sequentially pressing once on each overloaded key embedding a character that constitutes one of the multiple terms of the query string.

Description of Related Art

[0003] Of the many user interface deficiencies stifling the widespread adoption of search
10 on television and mobile phones, text entry is perhaps the most noteworthy. The limited size of the keypad combined with the overloading of each key with multiple alphabets makes text input quite cumbersome.

[0004] The 12-key keypad interface (Fig. A) present in most cell phones, and recently in television remote controls, has multiple alphabets overloaded on the same key. Text entry
15 using this keypad has required either a disambiguation step where user presses each key multiple times (e.g. to choose B, user presses the “2” key twice), or a vocabulary based completion choice for each word (e.g. the T9 text input). Neither of these methods are optimal interfaces for incremental search (where user gets results dynamically as each search term character is typed), in terms of the number of steps to get to the result. The deficiency of
20 the multi-press interface is the very characteristic of the interface itself – too many key strokes. The key drawback of applying a vocabulary based word completion interface for search is the introduction of an additional step where the user has to make a choice from a list of all possible search queries generated by the ambiguous text input. Furthermore vocabulary

based word disambiguation systems are primarily used for composition applications (as opposed to search applications) where user explicitly disambiguates each word by a word completion action before proceeding to the next word in the composition. Given that most input queries are multiple words, an explicit word completion action during search is an
5 additional burden on the user.

[0005] The core problem in an ambiguous text entry system is “lesser the characters, the harder it is to the disambiguate the results” – the ambiguity decreases as the input character count increases. This fact has the undesirable consequence of diminishing the power of a search engine that has the potential to retrieve results with just a few input characters. As
10 mentioned earlier an additional problem challenging a search system is the majority of search queries are multi word queries - this would force the user to enter in multiple, if not all of the query words before arriving at the desired result. For instance in the case of queries that match results that are not popular (e.g. Oona Hart, an actress who is not very popular), or when the terms of the query are semantically related (loosely coupled terms) to each other
15 (e.g. Tom Hanks volleyball – the user in this case is trying to find the name of the movie with cast Tom Hanks and keyword volleyball), user would be forced to enter more than one term as part of the input query. Some search systems attempt to solve this by brute force match of a single word input query with all the substrings of any term in the search space, assuming optimistically that the input was a sufficiently unique signature substring that could isolate
20 the desired result. These systems, however, increase the number of matched results significantly on average, even in an unambiguous text input scenario thus forcing the user to either scroll through a long list of results or enter more characters in order to reduce the matched results. In ambiguous text input systems, brute force substring matching in the

search space explode the results list rendering this technique useless as a reduced text entry method.

[0006] Hence there is a need for an interface for incremental search that can take ambiguous text input from a reduced keypad interface and (1) accept the input query without an explicit
5 disambiguation step and (2) be able to produce desired results with a reduced amount of text entry even when the input text entry is ambiguous.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] Figure A is prior art incorporated here for elucidative purposes. For a more complete
10 understanding of various embodiments of the present invention, reference is now made to the following descriptions taken in connection with the accompanying drawings in which:

[0008] Fig. 1 illustrates a text entry system in accordance with one or more embodiments of the invention being used in a different range of device and network configurations.

[0009] Fig. 2 illustrates a television configuration to perform text entry.

15 [0010] Fig. 3 illustrates the simple interface for finding results with reduced text entry using an overloaded keypad.

[0011] Fig. 4 illustrates the many-to-many mapping of terms to the numeric equivalents.

[0012] Fig. 5 illustrates the two distinct couplings between multiple terms in a query.

[0013] Fig. 6 illustrates a data structure for retrieving results incrementally for each
20 character using the many-to-many mapping scheme.

[0014] Fig. 7 illustrates the results retrieved when user types in a single character “5” with one key press.

[0015] Fig. 8 illustrates the results retrieved when user types in a single character “3” with one key press. The resultant input string is now “53”.

5 [0016] Fig. 9 illustrates the results retrieved when user types in a single character “7” with one key press. The resultant input string is now “537”.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0017] Preferred embodiments of the present invention and their advantages may be
10 understood by referring to Figs. 1-9, wherein like reference numerals refer to like elements.

[0018] **Fig. 1** illustrates an overall system for performing searches with reduced text entry using a wide range of devices in accordance with one or more embodiments of the invention. A server farm 101 can serve as the source of search data and relevance updates with a network 102 functioning as the distribution framework. The distribution framework
15 could be a combination of wired and wireless connections. Examples of possible networks include cable television networks, satellite television networks, IP-based television networks, wireless CDMA and GSM networks. The search devices could have a wide range of interface capabilities such as a hand-held device 103 (e.g., a phone or PDA) with limited display size and a reduced keypad with overloaded keys, a television 104a coupled with a
20 remote control device 104b having an overloaded keypad (**Fig. A**), and a Personal Computer (PC) 105 with a reduced keyboard and a computer display.

[0019] Fig. 2 illustrates multiple exemplary configurations for search devices in accordance with one or more embodiments of the invention. In one configuration, a search device (e.g., PC 105) can have a display 201, a processor 202, volatile memory 203, text input interface 204 (which can be on-device or through a wireless remote control 104b), remote connectivity 205 to the server 101 through the network 102, and a persistent storage 206. A device configuration for a device such as the hand-held device 103 might not include local persistent storage 206. In this case, the device 103 could have remote connectivity 205 to submit the query to the server 101 and retrieve results from it. Another configuration of the device 103 may not have remote connectivity 205. In this case, the search database may be locally resident on a local persistent storage 206. The persistent storage 206 may be, e.g., a removable storage element too such as SD, SmartMedia, CompactFlash card etc. In a configuration of the device with remote connectivity 205 and persistent storage 206 for search (e.g., television 104a), the device may use the remote connectivity for search relevance data update or for the case where the search database is distributed on the local storage 206 and on the server 101. The most suitable configuration in a memory constrained device is the search data residing remotely on a server. Unlike composition applications where the “most frequently used or popular terms space” are small in size and can be maintained in a local vocabulary, search spaces are larger inherently because people instinctively use “unique word signatures” to recall a document of interest. Hence maintaining search spaces locally may not be practical in most devices that have insufficient local memory – network based search configuration becomes essential. In one or more exemplary embodiments of the invention, a television 104a may have a set-top box with a one-way link to a satellite network. In this configuration, all search data including relevance

updates may be downloaded to the device through a satellite link to perform local searching. In this case the set-top box has sufficient storage capacity to maintain search spaces locally. Local storage is large in this case to circumvent the deficiency of a one-way link.

[0020] Referring to **Fig. 3**, user enters a character using an ambiguous text input interface (**Fig. A**) where a single key press is performed for each character 301. The incremental search scheme displays results that match the input character 302. Since the input is ambiguous the match of results would include the matches for all the ambiguous input characters represented by the single key press. To address this increased match, the underlying search method adopts in addition to a relevance based ordering scheme other criteria such as character count based subspace biasing (incorporate by reference patent “method and system for incremental search with reduced text entry on television where the relevance of results is a dynamically computed function of user input search string character count”). On the resultant list it finally applies an ordering criteria based on the lexical match with single term prefixes, multiple term prefixes, and lastly abbreviation matches. User continues to enter more characters if the desired result is not reached 303.

[0021] **Fig. 4**, illustrates an example of the many-to-many mapping scheme from alphanumeric terms space to numeric string space. The terms “TOON”, “TOM”, “TOMMY” are mapped to the numeric equivalents of their prefix strings “T”(8), “TO”(86), “TOO”(866), “TOON”(8666), “TOMMY”(86669). This many-to-many mapping scheme enables incremental search by enabling even a single character that user typed in to retrieve relevant results, as described in greater detail below. This many-to-many mapping is done during the indexing phase, for all terms that can be used to discover a result.

[0022] Fig. 5 illustrates the two broad categories of multiple prefix terms 501. Tightly coupled terms 502 are those in which the terms used to discover a document are closely related, such the terms from movie title (e.g. **Gods** must be **crazy**, **Harrison Ford**). The terms may be either ordered or unordered (e.g. Ford Harrison or Harrison Ford). Loosely coupled terms 503, on the other hand are terms that are semantically related as in “Tom Volleyball” (Tom hanks playing a role in a movie Cast away with Volleyball being a keyword). The prefix based multiple terms input described in the present invention, in addition to being an intuitive means for reduced text input significantly reduces the size of the result space in comparison to current incremental search techniques where any query substring could match with results. In partial substring based systems the ambiguity in input could significantly increase the result space size, particularly when the number of characters input is low. This would force the user to enter more characters in these systems diminishing the effectiveness of the search engine.

[0023] Fig.6 illustrates an embodiment of the invention where a trie data structure with each node having numerical values form 0-9 are used (601,604,605). Each node has the top M records 602 in “in-memory” storage. These records are returned immediately to user based on the match string. The illustrated picture shows the layout of the data structure for the terms “TOMMY BOY”. The number of prefix terms and the size of the prefix terms used for pre-computing the trie index are determined by the memory availability and computational capabilities of the system (incorporate by reference patent “method and system for performing searches for television content using reduced text input”). In the illustration the size of the prefix terms is 2 and the number of terms for pre-computing the trie index is 2. So when user enters any of the search queries, “8 269”(T BOY), “86 269”(TO BOY), “8 2”(T

B), “86 2”(T BO) the results would be retrieved from the top M records 602 if it is present there. If it is beyond the top M, the records are retrieved from the secondary storage structure 603 if it is present there. If the results are not found here, the system would perform an intersection of the top “N” records ($N \geq M$) for the individual terms, in this case, for example “T BOY”. In a preferred embodiment of the invention, the system would return the results obtained both from the trie walk (where results are obtained either from the top “M” records or from the container) and from the intersection method, where the intersection method would be deployed once the input character count exceeds a threshold. The multiple prefix terms based search is thus realized in one embodiment by the unique combination of a

- (1) trie walk whose strength and efficiency is highest when input character count is low, and
- (2) an intersection whose strength and efficiency is highest when the input character count is above a threshold.

Another key characteristic of this method of multiple prefix terms based search that distinguishes it from other reduced text entry search schemes is the absence of an explicit term/word termination action. The system disambiguates the input by leveraging off the multiple terms (both in the case of loosely coupled and tightly coupled terms).

[0024] Fig. 7 illustrates the results 702 for a single character input “5”(J). User’s intent is to find “tom and jerry” and starts with entering Jerry (unordered input). However, there are multiple matches for different terms due to the input ambiguity (Koala, Lola, Life etc.). These terms are ordered by the popularity of the matching document and the relevance of the matching term to the document.

[0025] Fig. 8 illustrates the results 802 for a multiple term text input, where user is continuing to enter more text “5 86”(J TO) to discover “tom and jerry”. The input text

ambiguity initiates matches for several programs ([**Koala,Tooth**], [**Hallie Todd** in cast for Lizzie McGuire] etc.).

[0026] **Fig. 9** illustrates the case where user discovers the desired result 902 after inputting the multiple prefix string “5 866” (J TOM).