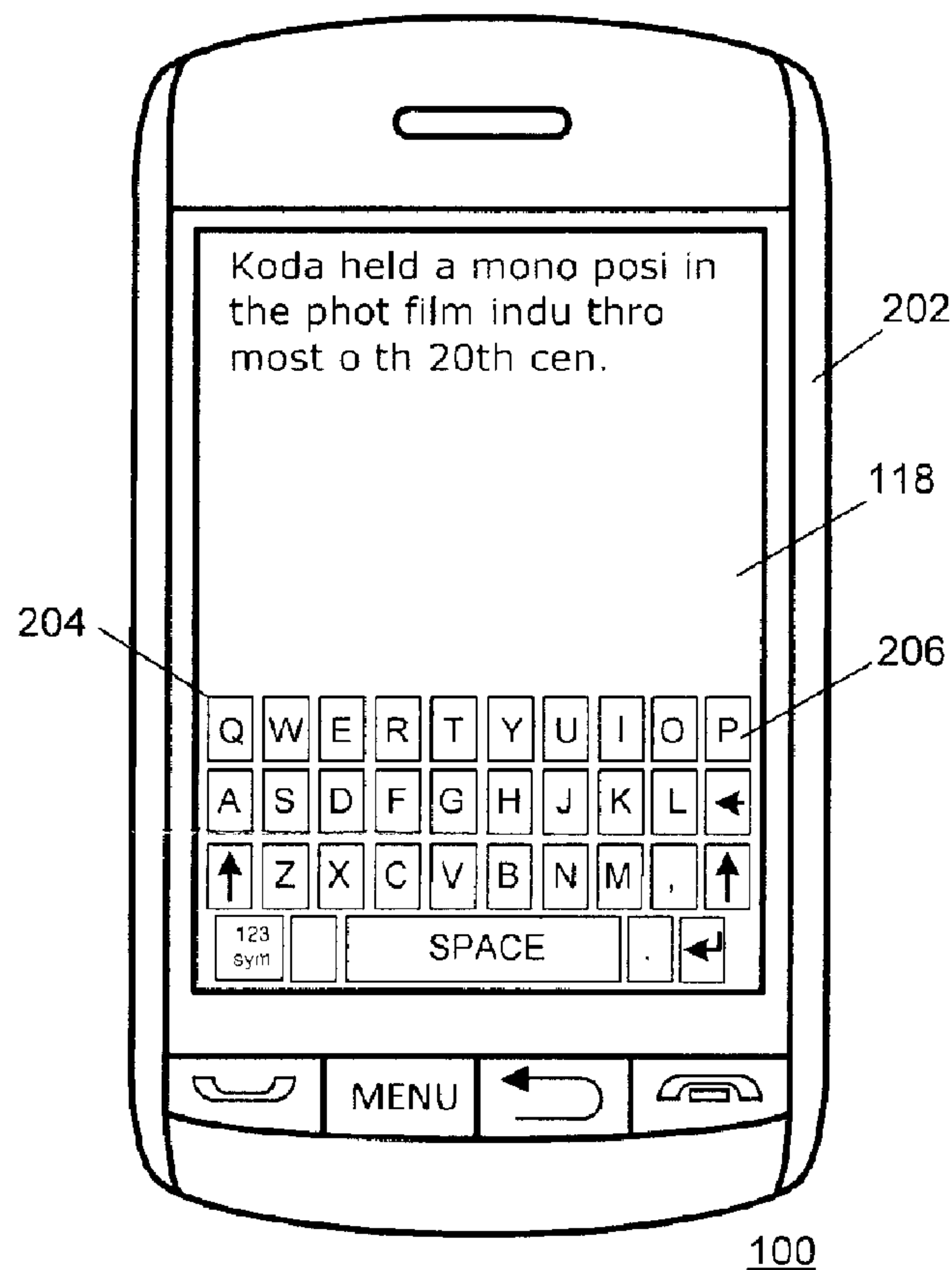




(86) Date de dépôt PCT/PCT Filing Date: 2012/04/30  
(87) Date publication PCT/PCT Publication Date: 2013/10/30  
(45) Date de délivrance/Issue Date: 2016/06/07  
(85) Entrée phase nationale/National Entry: 2013/06/26  
(86) N° demande PCT/PCT Application No.: CA 2012/050273  
(87) N° publication PCT/PCT Publication No.: 2013/163717

(51) Cl.Int./Int.Cl. *G06F 3/01* (2006.01),  
*G06F 17/27* (2006.01)  
(72) Inventeurs/Inventors:  
PASQUERO, JEROME, CA;  
STONEHOUSE, NOEL JOHN ORLAND, CA;  
LEGG, DANIEL JAMES, US;  
GRIFFIN, JASON TYLER, CA  
(73) Propriétaire/Owner:  
BLACKBERRY LIMITED, CA  
(74) Agent: BORDEN LADNER GERVAIS LLP

(54) Titre : APPAREIL ELECTRONIQUE POUR LA DESAMBIGUISATION BIDIRECTIONNELLE DE TEXTES AXEE SUR  
LE CONTEXTE ET METHODE CONNEXE  
(54) Title: ELECTRONIC DEVICE AND METHOD FOR A BIDIRECTIONAL CONTEXT-BASED TEXT DISAMBIGUATION



(57) Abrégé/Abstract:  
A system and method for a bidirectional context-based text disambiguation is provided.



## **ABSTRACT**

A system and method for a bidirectional context-based text disambiguation is  
5 provided.

# **ELECTRONIC DEVICE AND METHOD FOR A BIDIRECTIONAL CONTEXT-BASED TEXT DISAMBIGUATION**

## **FIELD OF TECHNOLOGY**

5   **[0001]** The disclosed and claimed concept relates generally to electronic devices and, more particularly, to an electronic device having a keyboard and a text input disambiguation function that can employ contextual data.

## **BACKGROUND**

10   **[0002]** Electronic devices, including portable electronic devices, have gained widespread use and may provide a variety of functions including, for example, telephony, text messaging, web browsing, or other personal information manager (PIM) functions such as a calendar application. Portable electronic devices include several types of devices such as cellular telephones (mobile phones), smart telephones (smart phones), Personal Digital Assistants (PDAs),  
15   tablet computers, or laptop computers, with wireless network communications or near-field communications connectivity such as Bluetooth® capabilities.

20   **[0003]** Portable electronic devices such as smart phones, tablet computers, or PDAs are generally intended for handheld use due to their small size and ease of portability. A touch-sensitive input device, such as a touchscreen display, is desirable on handheld devices, which are small and may have limited space for user input or output devices. Improvements in electronic devices with displays are desirable.

## **BRIEF DESCRIPTION OF THE DRAWINGS**

25   **[0004]** Embodiments of the present disclosure will now be described, by way of example only, with reference to the attached Figures, wherein:

30   **[0005]** FIG. 1 is a block diagram of an example of a portable electronic device in accordance with the present disclosure;

**[0006]** FIG. 2A is a schematic depiction of the portable electronic device, in accordance with the present disclosure;

**[0007]** FIG. 2B is a schematic depiction of a memory of the portable electronic device, in accordance with the present disclosure;

**[0008]** FIG. 3A is a flowchart of an example candidate selection routine, in accordance with the present disclosure;

**[0009]** FIG. 3B is a flowchart of an example context-based optimization routine, in accordance with the present disclosure; and

5 **[0010]** FIG. 4 is a schematic depiction of an example context-based optimization routine, in accordance with the present disclosure.

### **DETAILED DESCRIPTION**

**[0011]** The present disclosure describes a context-based text disambiguation method. The method, which is performed by at least one processor, comprises  
 10 receiving an input text comprising a set of string objects, which may include ambiguous objects in the sense that some of the string objects represent, for example, incomplete or unrecognizable words of a selected language. Next, the processor identifies a set of candidate word objects corresponding to at least a first one of the string objects and a second one of the string objects. Each  
 15 candidate word object represents, for example, a complete or recognizable word of the selected language. The processor then outputs a selected word object, for example, in place of a first one of the string objects, as a function of a contextual comparison between one or more candidate word objects corresponding to the first string object and one or more candidate word objects  
 20 corresponding to the second string object.

**[0012]** A block diagram of an example of a portable electronic device 100 is shown in FIG. 1. The portable electronic device 100 includes multiple components, such as a processor 102 that controls the overall operation of the portable electronic device 100. The portable electronic device 100 presently  
 25 described optionally includes a communications subsystem 104 and a short-range communications 132 module to perform various communication functions, including data and voice communications. Data received by the portable electronic device 100 is decompressed and decrypted by a decoder 106. The communications subsystem 104 receives messages from and sends messages to  
 30 a wireless network 150. The wireless network 150 may be any type of wireless network, including, but not limited to, data wireless networks, voice wireless networks, and networks that support both voice and data communications. A power source 142, such as one or more rechargeable batteries or a port to an



external power supply, powers the portable electronic device 100.

**[0013]** The processor 102 is coupled to and interacts with other components, such as Random Access Memory (RAM) 108, memory 110, a display 112. In the example embodiment of FIG. 1, the display 112 is coupled to a touch-sensitive overlay 114 and an electronic controller 116 that together comprise a touch-sensitive display 118. The processor 102 is also coupled to one or more actuators 120, one or more force sensors 122, an auxiliary input/output (I/O) subsystem 124, a data port 126, a speaker 128, a microphone 130, short-range communications 132, and other device subsystems 134. User-interaction with a graphical user interface (GUI) is performed through the touch-sensitive overlay 114. The processor 102 interacts with the touch-sensitive overlay 114 via the electronic controller 116. Information, such as text, characters, symbols, images, icons, and other items that may be displayed or rendered on a portable electronic device, is displayed on the touch-sensitive display 118 via the processor 102. The processor 102 may interact with an orientation sensor such as an accelerometer 136 to detect direction of gravitational forces or gravity-induced reaction forces so as to determine, for example, the orientation of the portable electronic device 100. The processor 102 may interact with a GPS module 152 in order to determine the geographical location of the portable electronic device 100.

**[0014]** To identify a subscriber for network access, the portable electronic device 100 uses a Subscriber Identity Module or a Removable User Identity Module (SIM/RUIM) card 138 for communication with a network, such as the wireless network 150. Alternatively, user identification information may be programmed into memory 110.

**[0015]** The portable electronic device 100 includes an operating system 146 and software programs or components 148 that are executed by the processor 102 and are typically stored in a persistent, updatable store such as the memory 110. Additional applications or programs may be loaded onto the portable electronic device 100 through the wireless network 150, the auxiliary I/O subsystem 124, the data port 126, the short-range communications subsystem 132, or any other suitable subsystem 134.

**[0016]** A received signal, such as a text message, an e-mail message, or web

page download, is processed by the communications subsystem 104 and input to the processor 102. The processor 102 processes the received signal for output to the display 112 and/or to the auxiliary I/O subsystem 124. A subscriber may generate data items, for example e-mail messages, which may be transmitted  
 5 over the wireless network 150 through the communications subsystem 104, for example.

**[0017]** A front view of an example of the portable electronic device 100 is shown in FIG. 2A. The portable electronic device 100 includes a housing 202 in which the touch-sensitive display 118 is disposed. The housing 202 is an  
 10 enclosure that contains components of the portable electronic device 100, such as the components shown in FIG. 1.

**[0018]** A keyboard 204 may be a physical keyboard within the housing 202, or a virtual keyboard rendered as a GUI displayed on the touch-sensitive display 118 as illustrated by the example embodiment of FIG. 2A. As shown in FIG. 2A,  
 15 the keyboard 204 is a GUI rendered on the touch-sensitive display 118 and has a QWERTY keyboard layout. In alternate example embodiments, other keyboard layouts such as QWERTZ, AZERTY, Dvorak, or the like, may be utilized. Similarly, reduced keyboards having two or more characters associated with certain keys, such as a reduced QWERTY keyboard layout, can be contemplated. For example,  
 20 a reduced QWERTY keyboard may be provided in which the letters Q and W share a single key, the letters E and R share a single key, and so forth.

**[0019]** The keyboard 204 may be rendered in any suitable program or application such as a web browser, text messaging (e.g., SMS), email client, contacts, calendar, music player, spreadsheet, word processing, operating  
 25 system interface, and so forth for text input. Other information such as text, characters, symbols, images, and other items may also be displayed, for example, as the keyboard 204 is utilized for data entry. The keyboard 204 includes a plurality of keys 206, each key associated with at least a character or a function as indicated by indicia displayed thereupon.

**[0020]** The memory 110 is depicted schematically in FIG. 2B. The memory 110 can be any of a variety of types of internal and/or external storage media such as, without limitation, RAM, ROM, EPROM(s), EEPROM(s), and the like that provide a storage register for data storage such as in the fashion of an internal  
 30



storage area of a computer, and can be volatile memory or nonvolatile memory.

**[0021]** As can be understood from FIG. 2B, the memory 110 includes, for example, data stored and/or organized in a number of tables, sets, lists, and/or otherwise. Specifically, the memory 110 includes a word list 202 and a  
 5 contextual data table 204. Stored within the word list 202 are a number of word objects 208 and frequency objects 210. The word objects 208 generally are each associated with a frequency object 210. The word objects 208 are generally representative of complete words.

**[0022]** Associated with substantially each word object 208 is a frequency  
 10 object 210 having frequency value that is indicative of the relative frequency within the relevant language of the given word represented by the word object 208. In this regard, the word list 202 includes a plurality of word objects 208 and associated frequency objects 210 that together are representative of a wide variety of words and their relative frequency within a given vernacular of, for  
 15 instance, a given language. The word list 202 can be derived in any of a wide variety of fashions, such as by analyzing numerous texts and other language sources to determine the various words within the language sources as well as their relative probabilities, i.e., relative frequencies, of occurrences of the various words within the language sources.

20 **[0023]** The portable electronic device 100 also includes a contextual data table 204 stored in the memory 110. The contextual data table 204 can be said to have stored therein a number of string objects and associated context data.

**[0024]** Specifically, the contextual data table 204 comprises a number of key  
 25 objects 214 and, associated with each key object 214, a number of associated contextual objects 216. In the example embodiment, in which the English language is employed on the portable electronic device 100, each key object 214 is a word object 208. That is, a key object 214 in the contextual data table 204 is also stored as a word object 208 in one of the word list 202. Each key object 214 has associated therewith one or more contextual value objects 216 that are  
 30 each representative of a particular contextual data element.

**[0025]** The contents of the contextual data table 204 are obtained by analyzing the language objects 100 and the data corpus from which the language objects 100 and frequency objects 210 were obtained. A particular

contextual object 216 is associated with a particular key object 214 when there is any statistically significant coincidence between the two objects, that is, when there is some statistically significant likelihood that the particular key object 214 would appear in the context of the particular contextual value 216, or vice versa.

- 5 **[0026]** One example of a context is that in which a particular key object 214 follows or precedes, to a statistically significant extent, a particular word. For instance, it could be determined that the key word "POSITION" occurs, to a statistically significant extent, after the context word "MONOPOLY," and that the key word "20th" occurs, to a statistically significant extent, before the context
- 10 word "CENTURY." Depending upon the configuration of the contextual data table 204, such a context might be limited to a particular context object that immediately precedes or follows a particular key word, or it might include a particular context object that precedes or follows a particular key object by either exactly, or a maximum of, one, two, three, or more words. In some
- 15 embodiments, a context includes more than one word, for example, the words "happy new", are statistically likely to appear before the word "year." In some embodiments, some contextual associations are weighed more heavily than others, the weights depending, for example, on the correlation level characterizing the particular association.
- 20 **[0027]** Another example of a context is that in which a particular string object is, to a statistically significant extent, a first word in a sentence. In such a situation, the identified context might be that in which the particular string object follows, to a statistically significant extent, one or more particular punctuation marks such as the period ".", the question mark "?", and the
- 25 exclamation point "!". In such a situation, the contextual object 216 would be the particular punctuation symbol, with each such statistically significant punctuation symbol being a separate contextual object 216.

- [0028]** The contextual objects 216 can each be stored as a hash, i.e., an integer value that results from a mathematical manipulation. For instance, the
- 30 contextual object 216 "MONOPOLY", while being a word object 208, can be stored in the contextual data table 204 as a hash of the word "MONOPOLY". The key objects 214, such as the word "POSITION" can similarly each be stored as a hash.



**[0029]** In an example embodiment, the memory 110 includes a text disambiguation routine for resolving input texts comprising string objects. String objects may occur, for example, when the user enters the text using a reduced keyboard, wherein some keys correspond to more than one character. In  
 5 systems with full keyboards, the user can, in the interest of time, deliberately input partial (and therefore ambiguous) string objects, relying on the disambiguation routine to automatically resolve the ambiguities and to correctly complete the string objects.

**[0030]** In an example embodiment, the user inputs a text comprising a  
 10 plurality of string objects. For instance, the user desires to input the following text: "Kodak held a monopoly position in the photographic film industry throughout most of the 20th century". In the interest of time, the user inputs the following text, instead: "Koda held a mono posi in the phot film indu thro most o  
 15 th 20th cen." The text comprises numerous string objects, i.e., partial strings of characters that correspond to two or more complete words. The user can use any number of characters for each string object: only the first character (e.g., "o"), the first two, three or more characters (e.g., "th", "cen", "posi"), or the entire word (e.g., "held", "most" and "20th").

**[0031]** The processor 102 receives the input text and breaks it into a plurality  
 20 of string objects, where each string object includes at least one character. For simplicity and without limiting the generality, each string of characters within the input text is referred to herein as a string object, whether a particular string of characters happens to correspond to a complete word or not.

**[0032]** In an example embodiment, the processor 102 processes each string  
 25 object using a candidate selection routine, as illustrated in the flowchart in FIG. 3A. The candidate selection routine consults the memory 110 to identify at 302 one or more complete word objects 208 that correspond to the current string object. A complete word object 208 is said to correspond to a string object when, for example, the string object is either a prefix of the complete word  
 30 object 208 or would be substantially identical to the entirety of the complete word object 208. In some embodiments, the candidate selection routine anticipates that the user could have mistyped a string object and also identifies complete word objects 208 that have prefixes similar to the string object. For example, the routine may identify a complete word object 208 "sound" to

correspond to the string objects "soi" or "soin," anticipating a potential typo, since characters "i" and "u" are closely positioned on a QWERTY keyboard.

**[0033]** Next, the candidate selection routine selects at 304, among all the identified complete word objects 208, those objects that are associated with frequency objects 210 having relatively high frequencies. For example, the candidate selection routine selects N complete word objects 208 that associated with N frequency objects 210 having the highest frequencies. N could be any number and may vary from one string object to another. That is, the candidate selection routine can select as few as one complete word object 208 for one string object, and as many as a hundred complete word objects 208 for another string object. In some embodiments, the number of selected complete word objects 208 is not limited, and the selection routine can keep processing objects until it is interrupted, for example, by an input from the user. In some embodiments, in addition to frequencies, the candidate selection routine takes into account the length of the complete word objects 208, as it relates to the length of the string object, for example. For instance, shorter complete word objects 208 may be preferred when the string object is shorter.

**[0034]** Once one or more complete word objects 208, hereinafter referred to as "candidates," are selected, they are temporarily stored at 304 in the memory 110. The candidate selection routine then ends, and the processor 102 can run the routine on another string object.

**[0035]** At any point in time, for example, when all the string objects within the input text have been processed by the candidate selection routine, the processor 102 selects a set of two or more string objects and processes that set of string objects with a context-based optimization routine. The context-based optimization routine obtains from the memory 110 all candidates stored for each of the string objects in the set, and prioritizes the candidates of each string object based on the number of contextual associations between each candidate and the candidates of other string objects within the set.

**[0036]** The processor 102 can run the context-based optimization routine several times, each time including a different, potentially overlapping, set of string objects. The processor 102 does not have to wait for all string objects to be processed by the candidate selection routines before it starts running the



context-based optimization routines. Generally, once any set of two or more string objects have been processed by the candidate selection routines, the processor 102 can run the context-based optimization routine on that set.

**[0037]** The flowchart in FIG. 3B illustrates a context-based optimization routine run by the processor 102, in accordance with an example embodiment. The context-based optimization routine begins by receiving at 310 a set of two or more string objects. At 312, the routine obtains, for each string object, all of its candidates. The candidates are obtained from the memory 110 where they have been temporarily stored for each string object by a corresponding candidate selection routine. At 314, the context-based optimization routine identifies, for each candidate of each string object, the number of contextual associations between that candidate and any of the candidates of any other string object. At 316, the routine updates the priorities of each candidate of each string object in the set based, for example, on to the number of contextual associates of that candidate. The context-based optimization routine then ends.

**[0038]** FIG. 4 illustrates the context-based optimization routine performed by the processor 102 on a set of three string objects, in accordance with an example embodiment. In the example embodiment, the input text consists of at least six string objects 402 referred to, for simplicity, as A, B, C, D, E and F. It is assumed that at least the string objects A, B and C have already been processed by the candidate selection routine, and as a result, candidates 404 for each of the three words have been produced and stored in the memory 110. For example, candidates A1 and A2 have been stored for the string object A, candidates B1 and B2 have been stored for the string object B, and candidates C1 and C2 have been stored for the string object C.

**[0039]** The context-based optimization analyzes, for example, eight possible permutations 430-437 of the candidates 404, each permutation forming a different combination of three candidates. For each permutation 430-438, the optimization pass determines whether any candidates within the permutation is contextually associated with any another candidate, that is, whether one of the candidates corresponds to a key object 214 and another candidate corresponds to a contextual object 216 associated with the key object 214. For each found contextual association, the optimization process increases the relative priorities of both the candidate corresponding to the key object 214 and the candidate



corresponding to the contextual object 216.

**[0040]** For example, if the candidate C2 is contextually associated with the candidate B2 and its context corresponds to the position of C relative to B, the priorities of both candidates are increased, e.g., by 1, in each permutation wherein both C2 and B2 are present (permutations 433 and 437). Further, if C2 is also contextually associated with A1, then in permutations wherein both A1 and C2 are present (permutations 431 and 433) the priorities of both candidates are increased, as well. It will be noted that for each contextual association the routine determines whether the associated candidates are positioned within the input text according to the particular context. That is, if the context defining the association of A1 and C2 dictates that A1 must immediately precede C2, the routine would determine that there was no contextual association in the above example, because A1, while preceding C2, does not precede it immediately in the input text, as it is separated by another word.

**[0041]** The context-based optimization routine ends when all the permutations for the given set have been analyzed. The updated priorities for each candidate are stored, for example, in the memory 110, to be used in a subsequent run of the context-based optimization routine.

**[0042]** It will be noted, that instead or in addition to the described permutation technique, in some example embodiments, the processor 102 employs within the context-based optimization routine any other technique that considers contextual associations within the set of string objects and prioritizes the candidates of each string object based on the contextual associations. In one example, the processor 102 increases the priority of two contextually associated candidates only once, regardless to the number of permutations containing the two candidates. In another example, the processor 102 takes into account the correlation strength values corresponding to each contextual association. In yet another example, the candidate selection routine temporarily stores, along with the candidates, their respective frequency values. Consequently, in some example embodiments, the priority of each candidate is based on a combination of its intrinsic frequency value and its extrinsic, context-based associations with other string objects.

**[0043]** In an example embodiment, the processor 102 runs the context-based

optimization routine one or more times, each time processing a different set of two or more string objects. Continuing the previous example, at any time after the string object D has been processed by the candidate selection routine, the processor runs the context-based optimization routine on the set of string  
 5 objects B, C and D. The priority values updated during the previous run of the context-based optimization routine (where it processed string objects A, B and C) are maintained and are built upon at this subsequent run. Thus, the candidate B2 starts with the priority of "+2" and that priority will be further increased if, for example, it is contextually associated with any of the candidates of the string  
 10 object D. The processor then runs the context-based optimization routine on the set of string objects C, D and E, then D, E and F, and so forth.

**[0044]** At any point in time when the processor 102 determines that a particular string object will not be further optimized, i.e., it will not be a part of another set to be processed by the context-based optimization routine, the  
 15 processor 102 finalizes that particular string object. Finalizing a string object comprises selecting among its candidates the best candidate, for example, the candidate with the highest priority.

**[0045]** For instance, if the processor 102 runs the context-based optimization routine in the sliding-window manner described above, the processor 102  
 20 finalizes the string object A immediately after the set A, B, and C is processed. To finalize the string object A, the processor 102 selects the best candidate among A1 and A2. Since A1 and A2 have accumulated priorities of +2 and +0, respectively, A1 is selected as the best candidate for the string object A. Similarly, the processor 102 finalizes the string object B after the set B, C and D  
 25 is processed, finalizes the string object C after the set C, D and E is processed, and so forth.

**[0046]** Thus, it will be noted the processor 102 advantageously employs a bidirectional disambiguation method, since a given string object is disambiguated based on information related to string objects both preceding and  
 30 succeeding it in order. For example, in the above-illustrated example, the string object C participates in three different sets: A-B-C, B-C-E, and C-E-D). Consequently, the priorities of candidates C1 and C2 could be affected by contextual information belonging to any of the words A through D.



**[0047]** It will be noted that the context-based optimization is not limited to processing three string objects at a time, and can similarly process any number of words, such as complete sentences, paragraphs, and so forth. For example, the processor may run the context-based optimization only once, based on a set  
5 that includes all the string objects in the input text. The context-based optimization pass is also not limited to two candidates per string object, and can process string objects having one candidate in the same set with string objects having tens or hundreds of candidates.

**[0048]** In an example embodiment, when all string objects are finalized, the  
10 processor 102 displays the finalized words, i.e., the best candidate for each word, on the touch-sensitive display 118. Alternatively, the processor 102 does not wait for all words in the input text to be finalized and outputs one or more words as soon as they become finalized. Continuing the previous example, A1, the best candidate for the string object A, is output as soon as the string object  
15 A is finalized, that is, immediately after the processor 102 runs a context-based optimization on the set A, B and C.

**[0049]** In an example embodiment, the processor 102 outputs to the user several versions of finalized words. For example, the processor 102 first performs context-based optimizations based on one technique (e.g., using  
20 permutations and not using correlation levels) and then based on another technique (e.g., not using permutations but considering correlation levels). Depending on the technique used, different candidates may end up being selected as best candidates for some string objects. Consequently, the processor 102 displays several alternative versions on the touch-sensitive display 118. In  
25 an example embodiment, the processor 102 automatically replaces some or all string objects with the selected best candidates. In other example embodiments, the processor 102, after displaying one or more best candidates to the user, receives user's selection of the desired best candidates and replaces with the string objects with the selected best candidates. In some embodiments, the user  
30 can indicate which individual candidates are correct and which are incorrect. The processor 102 can then rerun the candidate selection routine and/or the context-based optimization routine based on that indication, for example, by "fixing" the correct candidates and finding, based on the fixed candidates, replacements for the incorrect candidates.



**[0050]** While specific embodiments of the disclosed and claimed concept have been described in detail, it will be appreciated by those skilled in the art that various modifications and alternatives to those details could be developed in light of the overall teachings of the disclosure. For example, the processing  
5 routines described herein (e.g., the candidate selection, context-based optimization, etc.) can be performed in part or in their entirety by a remote device, such as a server. Running computationally intensive tasks on a powerful remote device can be advantageous in terms of speed, as well as power savings. Additionally, all or parts of the stored data described herein (e.g., word list 202,  
10 contextual data table 204, etc.) can also be stored on a remote device, whether or not the computations are done remotely or on the electronic device 100.

**[0051]** Accordingly, the particular arrangements disclosed are meant to be illustrative only and not limiting as to the scope of the disclosed and claimed concept which is to be given the full breadth of the claims appended and any and  
15 all equivalents thereof.

**CLAIMS**

1. A method, performed by a processor, comprising:
  - receiving, by the processor, an input text comprising a set of string
  - 5 objects, each object including at least one character;
  - identifying a first set of candidate word objects corresponding to a first one of the string objects and identifying a second set of candidate word objects corresponding to a second one of the string objects, wherein identifying the first set of candidate word objects corresponding to the first string object and the
  - 10 second set of candidate word objects corresponding to the second string object comprises identifying, within a word database, one or more words that begin with the first string object or the second string object and that have the highest associated frequency values within a language associated with the input text;
  - and
  - 15 outputting a selected word object as a function of a contextual comparison between one or more candidate word objects corresponding to the first string object and one or more candidate word objects corresponding to the second string object.
- 20 2. The method of claim 1, wherein the contextual comparison determines whether one or more candidate word objects corresponding to the first string object are statistically likely to occur in the context of one or more candidate word objects corresponding to the second string object.
- 25 3. The method of claim 2, wherein the first string object precedes the second string object within the input text.
4. The method of claim 2, wherein the first string object is separated from the second string object by at least one other string object.
- 30 5. The method of claim 2, further comprising outputting a second selected word object as a function of a contextual comparison between one or more candidate word objects corresponding to the first string object and one or more candidate word objects corresponding to the second string object.

6. The method of claim 5, wherein identifying the set of candidate word objects corresponding to at least the first one of the string objects and the second one of the string objects occurs before outputting the second selected word object.

5

7. The method of claim 2, further comprising receiving a confirmation input and replacing the first string object with the selected word object upon receiving the confirmation input.

- 10 8. An electronic device comprising a processor, the processor configured to:  
receive an input text comprising a set of string objects, each object including at least one character;  
identify a first set of candidate word objects corresponding to a first one of the string objects and a second set of candidate word objects corresponding  
15 to a second one of the string objects, wherein to identify the first set of candidate word objects corresponding to the first string object and the second set of candidate word objects corresponding to the second string object the processor is configured to identify, within a word database, one or more words that begin with the first string object or the second string object and that have  
20 the highest associated frequency values within a language associated with the input text; and  
output a selected word object as a function of a contextual comparison between one or more candidate word objects corresponding to the first string object and one or more candidate word objects corresponding to the  
25 second string object.

9. The electronic device of claim 8, wherein the contextual comparison determines whether one or more candidate word objects corresponding to the first string object are statistically likely to occur in the context of one or more  
30 candidate word objects corresponding to the second string object.

10. The electronic device of claim 9, wherein the first string object precedes the second string object within the input text.



11. The electronic device of claim 9, wherein the first string object is separated from the second string object by at least one other string object.

5 12. The electronic device of claim 9, wherein the processor is further configured to:

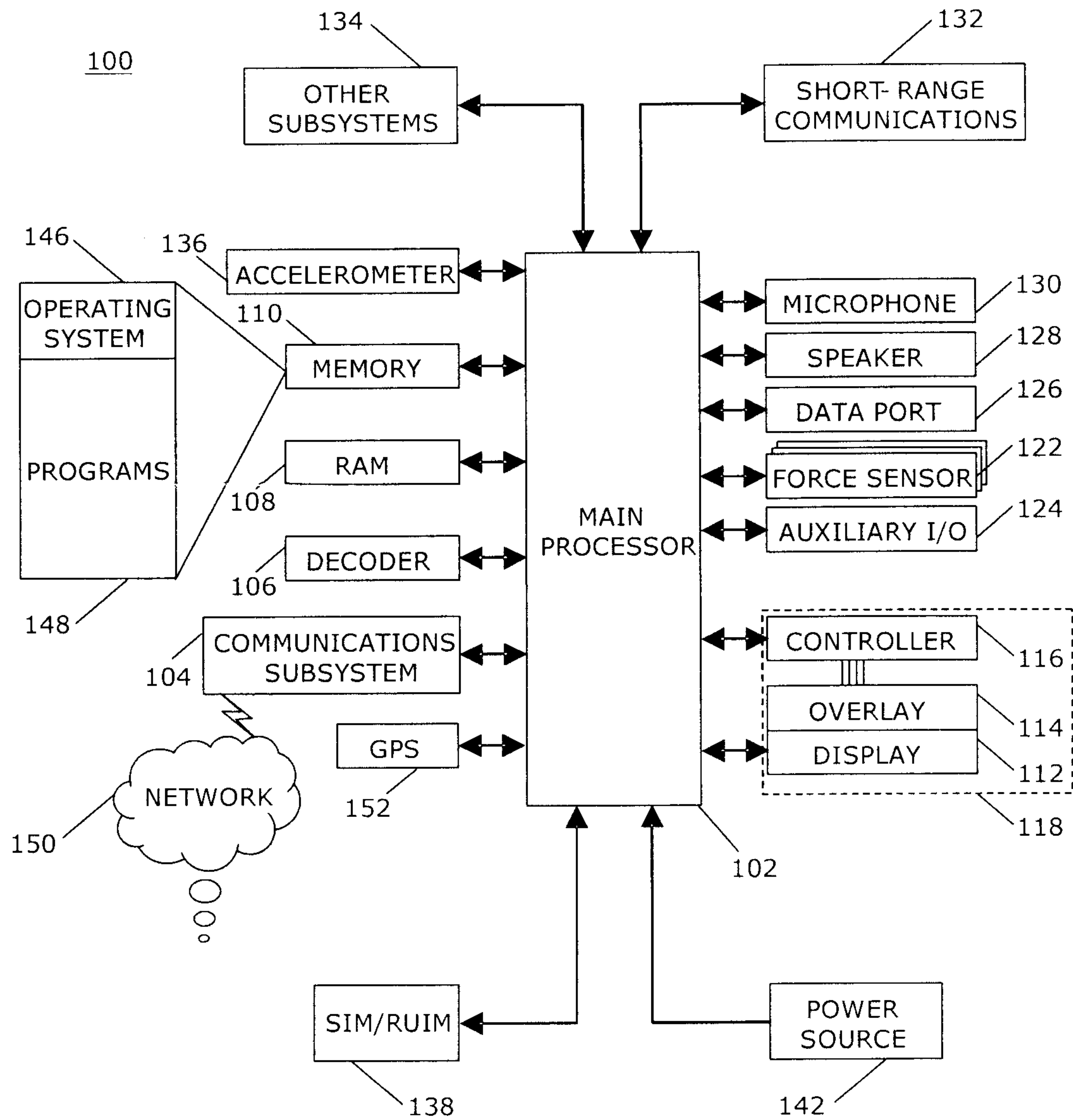
output a second selected word object as a function of a contextual comparison between one or more candidate word objects corresponding to the first string object and one or more candidate word objects corresponding to the  
10 second string object.

13. The electronic device of claim 12, wherein identifying the set of candidate word objects corresponding to at least the first one of the string objects and the second one of the string objects occurs before outputting the second  
15 selected word object.

14. The electronic device of claim 9, wherein the processor is further configured to receive a confirmation input and replace the first string object with the selected word object upon receiving the confirmation input.

20

1/4



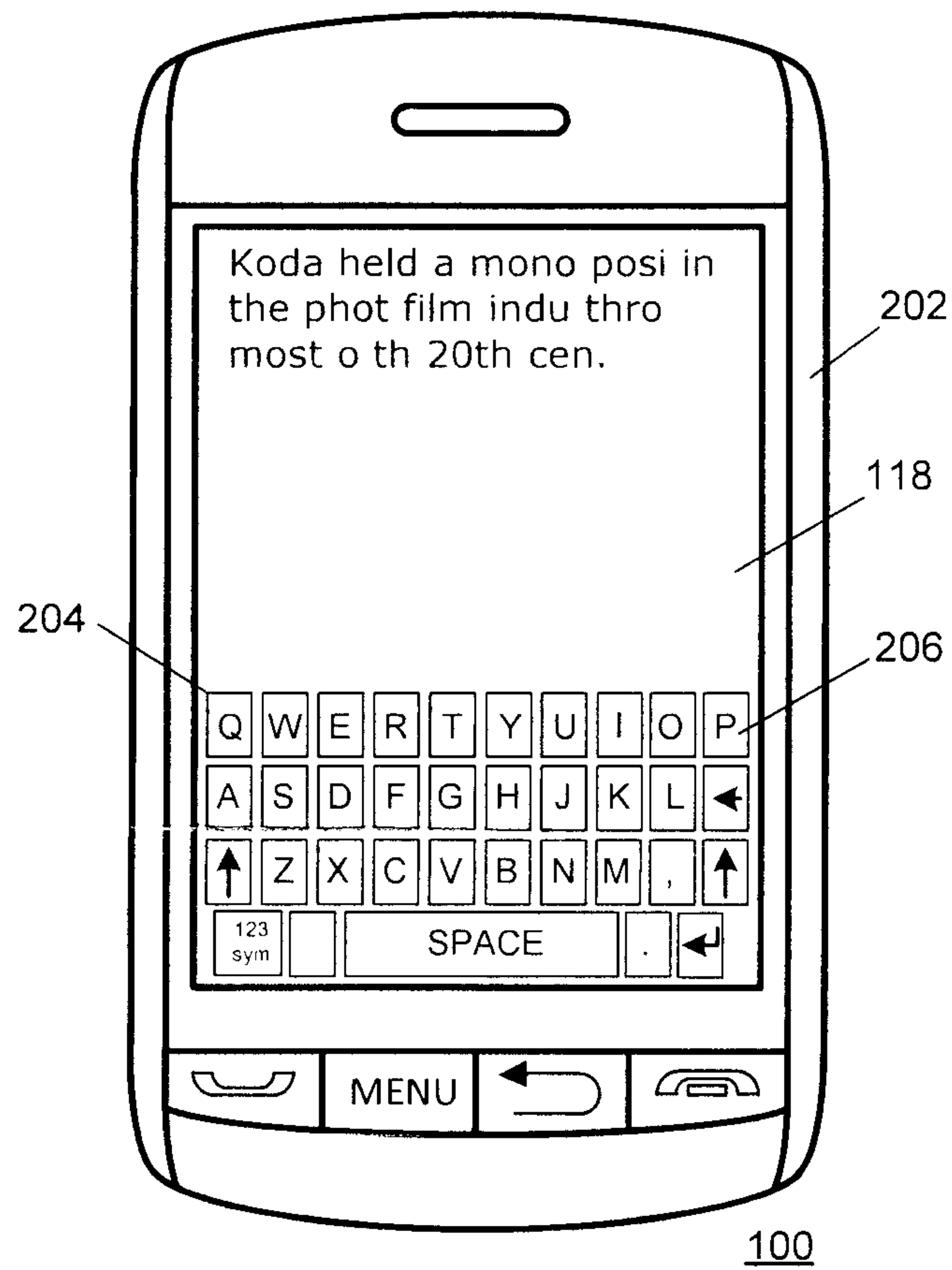


FIG. 2A

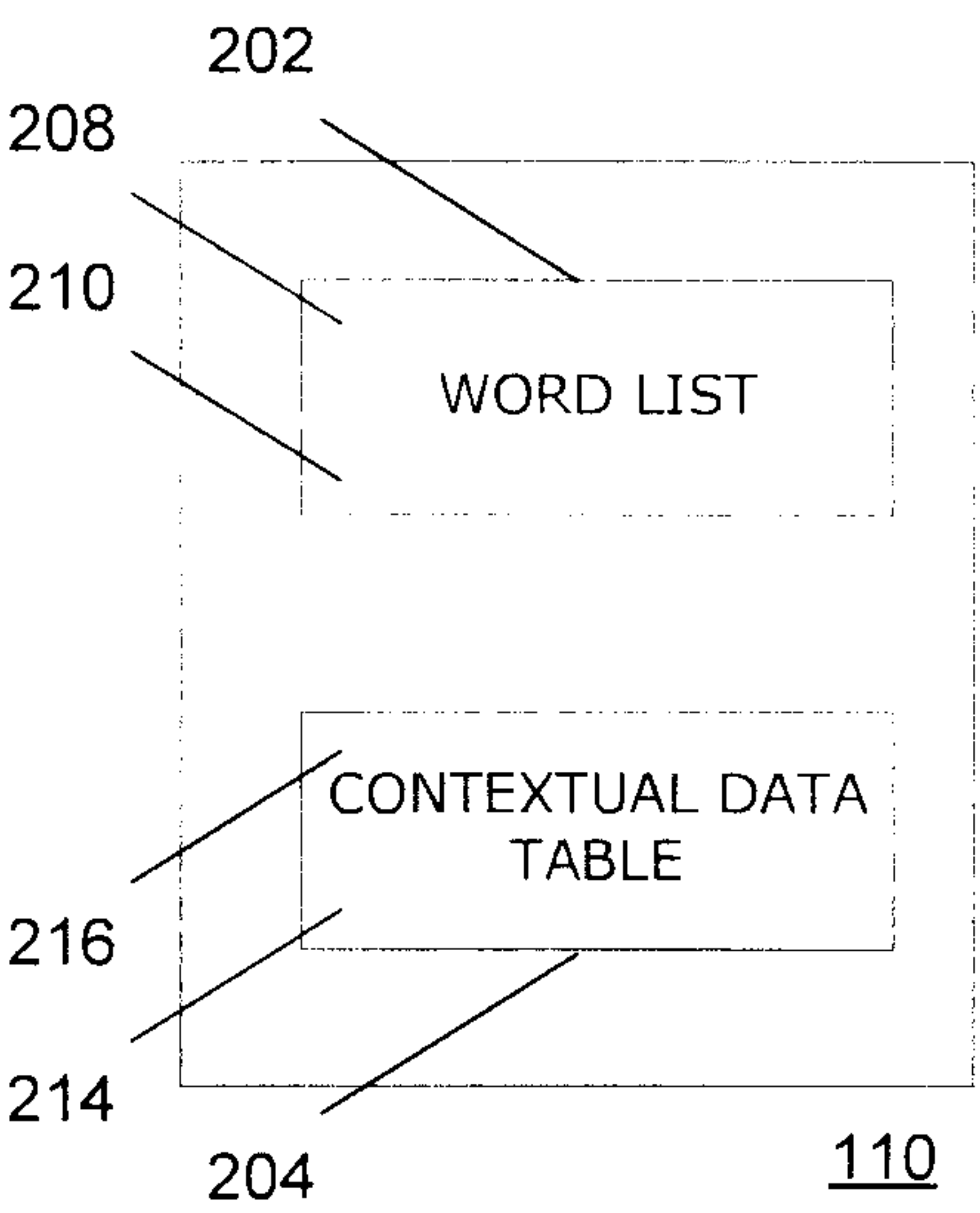


FIG. 2B



3/4

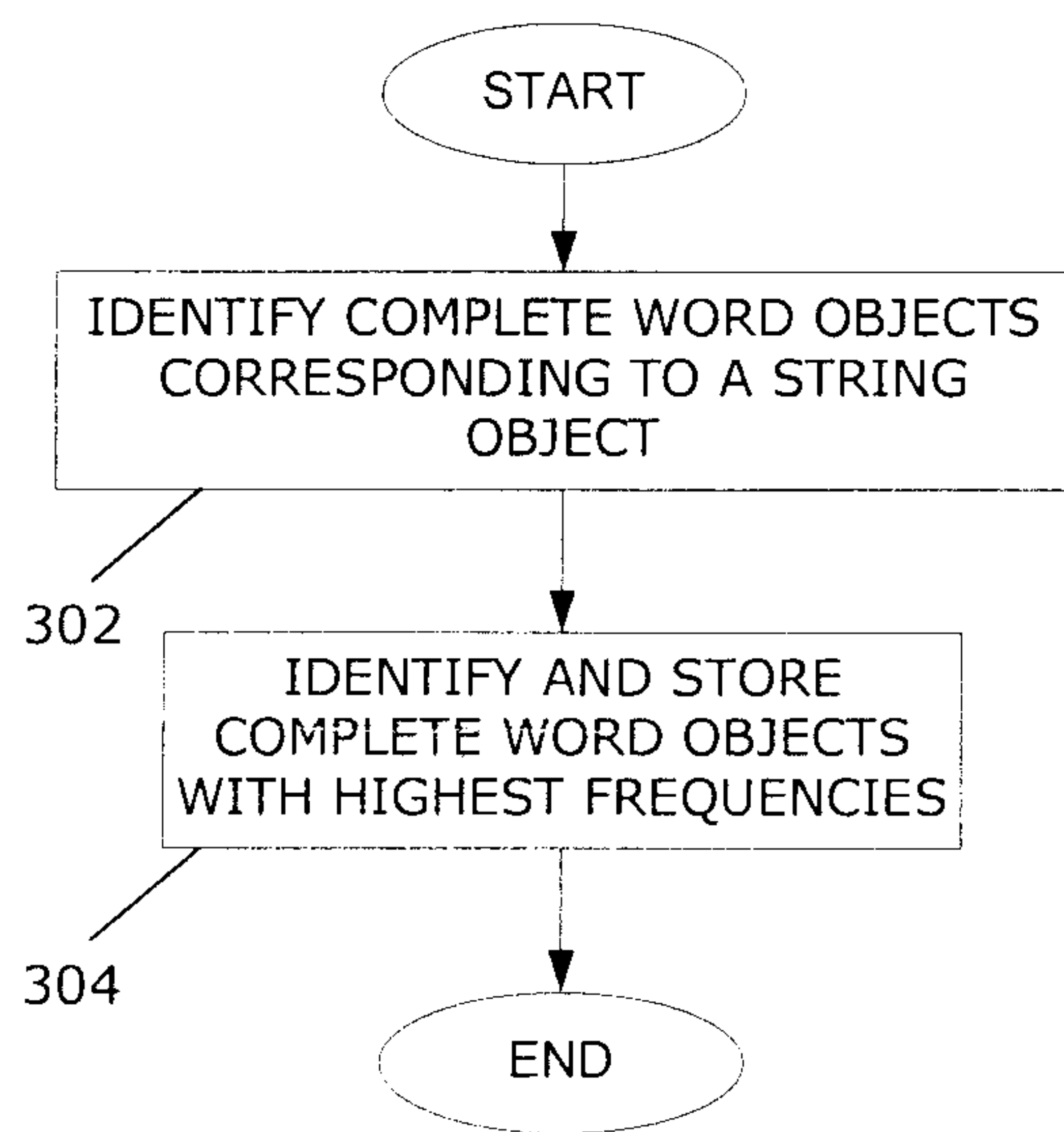


FIG. 3A

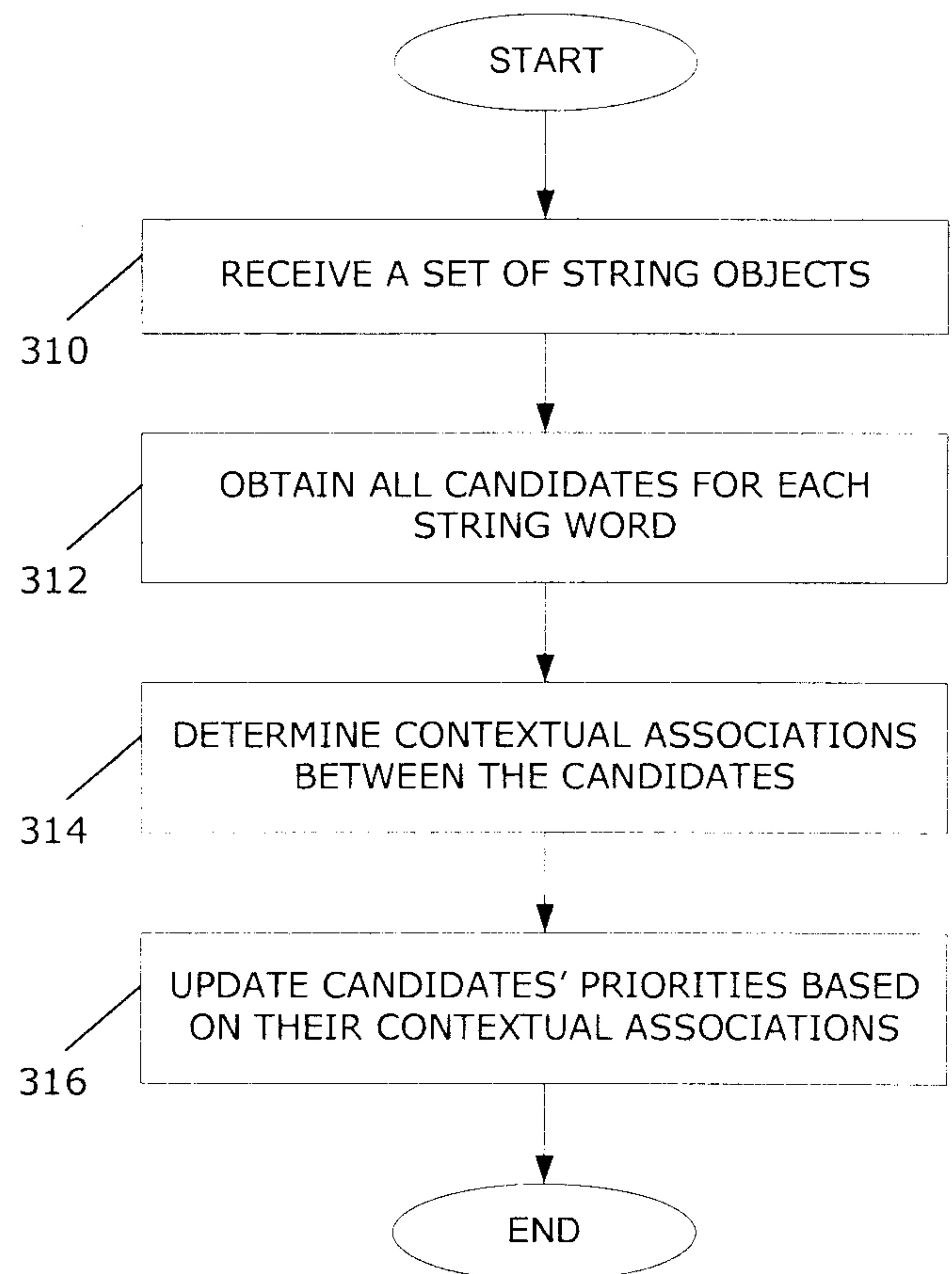


FIG. 3B

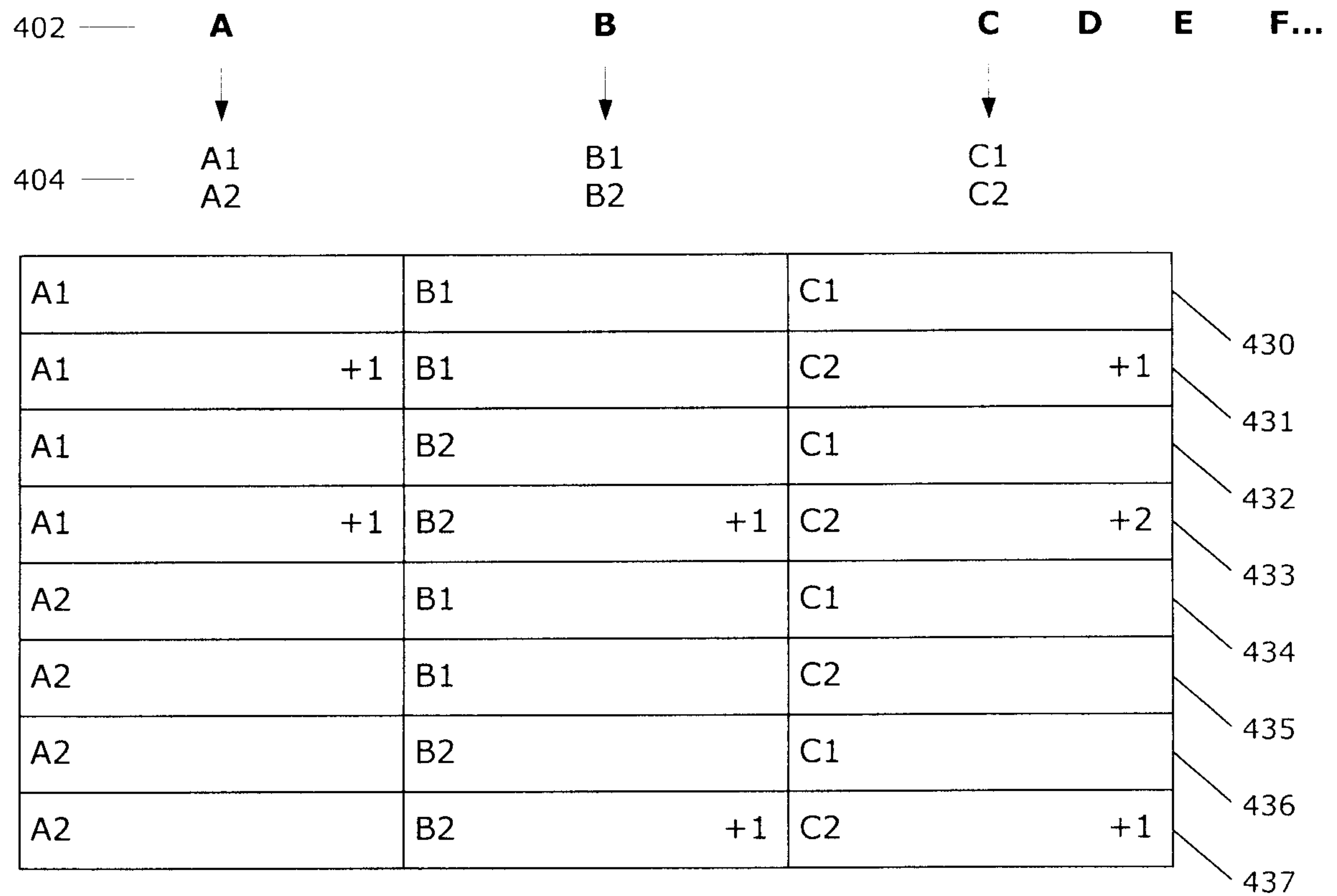


FIG. 4

