

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
2 July 2009 (02.07.2009)

PCT

(10) International Publication Number
WO 2009/079875 A1

(51) International Patent Classification:
G06F 17/30 (2006.01)

(21) International Application Number:
PCT/CN2007/071233

(22) International Filing Date:
14 December 2007 (14.12.2007)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicants (for all designated States except US):
SHANGHAI HEWLETT-PACKARD CO., LTD [CN/CN]; 112, jianguo Road, chaoyang District, HP Building, Beijing 100022 (CN). **HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P.** [US/US]; 20555, S.H.249, Houston, Texas 77070 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **ZHANG, Li** [CN/US]; 1501, Page Mill Rd, Palo Alto, California 94304-1100 (US). **XIONG, Yuhong** [US/US]; 1501, Page Mill Rd, Palo Alto, California 94304-1100 (US). **WANG,**

Wei-Chun [CN/CN]; 112, jianguo Road, chaoyang District, HP Building, Beijing 100022 (CN).

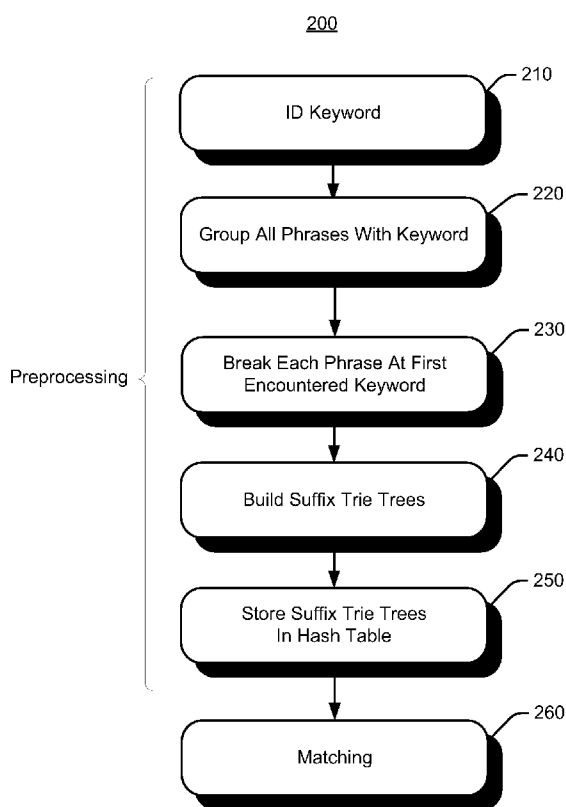
(74) Agent: **SHANGHAI PATENT & TRADEMARK LAW OFFICE, LLC**; 435 Guiping Road, Xuhui, Shanghai 200233 (CN).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM),

[Continued on next page]

(54) Title: SYSTEMS AND METHODS FOR EXTRACTING PHRASES FROM TEXT



(57) Abstract: Systems and methods for extracting phrases from text are disclosed. In an exemplary embodiment, a method may include preprocessing desired phrases into at least one phrase indexing data structure for efficient matching. The method may also include scanning text to construct a hash table including keys and corresponding entries. The method may also include locating suffix trie trees for each word in the hash table. The method may also include matching each position in the hash table against the suffix trie trees, and outputting phrases matched in the scanned text.

Fig. 2



European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI,
FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL,
PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM,
GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report*

SYSTEMS AND METHODS FOR **EXTRACTING PHRASES FROM TEXT**

BACKGROUND

[0001] Although there are a large number of websites on the Internet or World Wide Web (www), users often are only interested in information on specific web pages from some websites. Given the sheer size of Internet, it has become increasingly important to tailor searches and product recommendations to the user's personal interests/preferences. It is burdensome for both the user and the entity seeking information to ask the user to specify his/her personal interests/preferences. In addition, those interests/preferences may change over time. It is therefore generally considered more convenient to automatically discover user interests/preferences from the web pages the user visits. To enable such a tool, information contained in web pages visited by the user needs to be identified and extracted.

[0002] This information may include extracting movie titles, entity names (e.g., sports teams), or other information from web or text documents. On the surface, the problem of matching movie titles (or other information) appears to be a string matching problem. But actually, it has special characteristics and the standard string matching algorithms do not work well.

[0003] There are several well-known algorithms for string matching. The Knuth-Pratt-Morris and Boyer-Moore algorithms match a single string pattern

against an input string. The Aho-Corasick and Set-wise Boyer-Moore algorithms match multiple patterns. When applied to human language text, these algorithms usually work at character level. That is, the basic elements that form the patterns and the input text are considered to be characters. This means that the alphabet (the set of elements) is relatively small. In the case of English, there are twenty-six letters, plus a few other characters (e.g., space and apostrophe). Some of the algorithms take advantage of the small size of the English alphabet by pre-computing the actions for each element. For example, the Boyer-Moore algorithm pre-computes, for each element in the alphabet, the number of characters in the input text that can be skipped and stores these skip values in a table. However, these algorithms are limited to use with a specific language. In addition, a large number of comparisons are needed to effectively extract the desired information and therefore the algorithms are generally considered to be inefficient for these purposes.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] **Figure 1** is a high-level diagram of an exemplary networked computer system in which systems and methods for extracting phrases from text may be implemented.

[0005] **Figure 2** is a flowchart illustrating exemplary operations for extracting phrases from text.

[0006] **Figure 3** is a flowchart illustrating exemplary matching operations in more detail.

[0007] **Figure 4** is a flowchart illustrating additional exemplary preprocessing operations.

DETAILED DESCRIPTION

[0008] Systems and methods for extracting multi-word patterns or “phrases” from a text (e.g., a web page or document) are disclosed. In exemplary embodiments, a database is constructed with the desired phrase, and in some embodiments, related phrases that often appear in the same text when the desired phrase is used. The systems and methods may then be implemented for matching phrases in the text.

[0009] Exemplary systems and methods are described herein with reference to the particular application of extracting movie titles from text. By knowing which movie the user is interested in, links can then be provided for the user to, e.g., rent the movie, read reviews, or discover similar movies. It is noted, however, that the systems and methods disclosed herein may also be implemented for a wide variety of other applications (e.g., for extracting from text song names, sports team names, companies, schools, government agencies, etc.).

[0010] The disclosed multi-word pattern matching approach is different from the standard string matching problem. Take the English language as an example. At the word level, it is well known that the frequency of words follows a Zipf

distribution. Zipf's law states that, in a structured set of texts of natural language the frequency of any word is inversely proportional to its rank in frequency. Accordingly, the most frequent word occurs twice as often as the second most frequent word, which occurs twice as often as the fourth most frequent word, and so forth.

[0011] The systems and methods disclosed herein take advantage of this distribution by comparing each unique word (not each occurrence of a word) against the least frequent word in the phrase (e.g., a movie title). This approach greatly reduces the number of comparisons needed to identify the phrase in the text. In addition, when working at the word level, the set of elements in the patterns and text are all the possible words in one or more language. Accordingly, the systems and methods described herein can work with multiple languages at the same time.

[0012] In addition, text can be compared against a very large set of phrases. This large element set is handled by using hash tables to store the mapping from the words to data structures (st(w), pt(w) and hash table), making the comparison very efficient. This set-wise, multi-word matching problem also enables special characteristics (e.g., word frequency comparisons) that the string matching algorithms do not take advantage of.

[0013] **Figure 1** is a high-level illustration of an exemplary networked computer system 100 (e.g., via the Internet) in which systems and methods for extracting phrases from text may be implemented. The networked computer

system 100 may include one or more communication networks 110, such as a local area network (LAN) and/or wide area network (WAN), for connecting one or more websites 120 at one or more host 130 (e.g., servers 130a-c) to one or more user 140 (e.g., client computers 140a-c).

[0014] The term “client” as used herein (e.g., client computers 140a-c) refers to one or more computing devices through which one or more users 140 may access the network 110. Clients may include any of a wide variety of computing systems, such as a stand-alone personal desktop or laptop computer (PC), workstation, personal digital assistant (PDA), or appliance, to name only a few examples. Each of the client computing devices may include memory, storage, and a degree of data processing capability at least sufficient to manage a connection to the network 110, either directly or indirectly. Client computing devices may connect to network 110 via a communication connection, such as a dial-up, cable, or DSL connection via an Internet service provider (ISP).

[0015] The operations described herein may be implemented by the host 130 (e.g., servers 130a-c which also host the website 120) or by a third party system 150 (e.g., servers 150a-c) in the networked computer system 100. In either case, the servers may execute program code which enables comparison of phrases in one or more documents (e.g., a web page in website 120). The results may then be used to assist the user 140.

[0016] The term “server” as used herein (e.g., servers 130a-c or servers 150a-c) refers to one or more computing systems with computer-readable storage. The

server may be provided on the network 110 via a communication connection, such as a dial-up, cable, or DSL connection via an Internet service provider (ISP). The server may be accessed directly via the network 110, or via a network site. In an exemplary embodiment, the website 120 may also include a web portal on a third-party venue (e.g., a commercial Internet site) which facilitates a connection for one or more servers via a back-end link or other direct link. The servers may also provide services to other computing or data processing systems or devices. For example, the servers may also provide transaction processing services for users 140.

[0017] When the server is “hosting” the website 120, it is referred to herein as the host 130 regardless of whether the server is from the cluster of servers 130a-c or the cluster of servers 150a-c. Likewise, when the server is executing program code for extracting phrases from text, it is referred to herein as the server 150 regardless of whether the server is from the cluster of servers 130a-c or the cluster of servers 150a-c.

[0018] The program code may execute the exemplary operations described herein for extracting phrases from text. In exemplary embodiments, the operations may be embodied as logic instructions on one or more computer-readable medium. When executed on a processor, the logic instructions cause a general purpose computing device (e.g., server 150) to be programmed as a special-purpose machine that implements the described operations.

[0019] For purposes of illustration, the components and connections depicted in the figures may be used to extract movie titles from text (e.g., documents or web pages on the Internet 110), although the systems and methods are not limited to such an application. By knowing which movie the user is interested in, links can be provided for the user to conveniently rent the movie, to read reviews of the movie, discover similar movies, etc.

[0020] There are already fairly complete movie databases 155 available, such as the Internet Movie Database (IMDB). These databases typically contain the movie title, directors, and actors/actresses. Other information may also be provided by these databases 155. There are two major challenges in efficient and accurate extraction. First, there are a large number of movies. For example, the IMDB database contains more than 120,000 movies. It is very expensive to search for each movie title that may be contained in the text. Second, a movie title is often a meaningful word or phrase which needs to be distinguished from other words in the text. And when multiple movies have the same name, it needs to be determined which movie, if any at all, the title really corresponds to. Exemplary operations for enabling these features are described in more detail now with reference to Figures 2-4.

[0021] **Figure 2** is a flowchart illustrating exemplary operations 200 for extracting phrases from text. First, the movie titles are preprocessed (operations 210-250) into one or more phrase indexing data structures for efficient matching. Then those phrase indexing data structures are used to expedite the matching

process (Fig. 3). In operation 210, a keyword is identified for each movie title. In exemplary embodiments, the keyword is the word that appears least frequent in a typical text.

[0022] In operation 220, all of the phrases are grouped by keyword. For example, all the movie titles ($t_1, t_2, \dots, t_k$) with the same keyword w are grouped. In operation 230, each phrase is broken at the first encountered keyword. For example, break each t_i at the first encountered keyword w in t_i , obtaining p_i and s_i . That is, $t_i = p_i \bullet w \bullet s_i$, where $\bullet$ indicates concatenation. In operation 240, suffix trie trees $st(w)$ and $pt(w)$ are built. For example, a suffix trie tree $st(w)$ is built at the word level, on s_i 's for $i = 1$ to k . Then the order of words in p_i is reversed, obtaining q_i , and a suffix trie tree $pt(w)$ is built on q_i 's for $i=1$ to k . In operation 250, the suffix trie trees $st(w)$ and $pt(w)$ are stored in the phrase indexing data structures and are indexed by keyword w . The phrase indexing data structures serve as an overall index of the document. In operation 260, matching may commence, as described in more detail with reference to Figure 3.

[0023] **Figure 3** is a flowchart illustrating exemplary matching operations 300 in more detail. In operation 310, text is scanned (e.g., sequentially). In operation 320, a hash table is constructed in memory. The hash table includes keys (e.g., the words contained in the text) and entries (e.g., the list of positions of the words in the text). The hash table is used in operations 330 and 340.

[0024] In operation 330, the suffix trie trees $st(w)$ and $pt(w)$ in the phrase indexing data structures built in preprocessing are located for each word w in the

hash table. In operation 340, each position in the hash table is matched against the suffix trie tree $st(w)$ (wherein the matched set is referred to as L_1) and against the suffix trie tree $pt(w)$ (wherein the matched set is referred to as L_2). An entry of the hash table records all the positions of a word's appearance in the document. By checking each position, the algorithm basically goes through the whole document. The specific manner in which the document is indexed is not important and any suitable process may be implemented. In operation 350, phrases matched in the text are output. Continuing the above example, the intersection between L_1 and L_2 is the set of titles matched in the document.

[0025] Other embodiments are also contemplated. For example, preprocessing may be modified to improve the accuracy of matches by reducing false positives. In an exemplary embodiment, director and actor/actress information contained in the movie databases may be utilized in the preprocessing as described in more detail with reference to Figure 4.

[0026] **Figure 4** is a flowchart illustrating additional exemplary preprocessing operations. The basic idea is that when searching for a movie title, director/actor/actress name associated with the movie may also be used. By finding more relevant names, there is a higher confidence that the title is a true match.

[0027] In operation 410, a list of related phrases is obtained from the database (e.g., associated people such as director/actor/actress for each movie). In operation 420, the relevance of the related phrases to the desired phrases is determined. For

example, if the related phrases include persons, the relevance of each person is identified for each movie. In many cases, the relevance is already indicated in the movie database. When it is not, heuristics may be used to determine the relevance (e.g. the number of movies each person is associated to).

[0028] In operation 430, top related phrases are selected for each movie according to the relevance (e.g., x directors, y actors, and z actresses, where x, y, and z are some predefined constants such as $x=1$, and $y=z=2$). Either the top related phrases, or those meeting a predetermined threshold are selected. In an exemplary embodiment, the last name is indexed by the movie title. When a title is matched in the text, the list of related phrases (e.g., names) is retrieved and searched in the text. The confidence is then given by the number of times each related phrase (e.g., names) appears in the document. For example, the billing position may be used to determine the top related phrase wherein the billing position correlates to the importance of each role in the IMDB database. If billing position is not available, a weighted sum of the number of movies and TV events each actor/actress appears in may be used. Still other methods may also be implemented. In operation 440, matched results may be filtered using the top related phrases, e.g., to reduce false positives. Other factors may also be used to help decide whether a particular phrase refers to a movie title. For example, in a web document the number of overlapping words between a movie title and the HTML title may be used. Or for example, the number of times a movie title

appears in the text may be used. Each factor may be assigned a weight and combined using a heuristic formula.

[0029] It is understood that the embodiments shown and described herein are intended only for purposes of illustration of exemplary systems and methods and are not intended to be limiting. In addition, the operations and examples shown and described herein are provided to illustrate exemplary implementations for extracting phrases from text. It is noted that the operations are not limited to those shown. Other operations may also be implemented. Still other embodiments of for extracting phrases from text are also contemplated, as will be readily appreciated by those having ordinary skill in the art after becoming familiar with the teachings herein.

[0030] In addition to the specific embodiments explicitly set forth herein, other aspects and implementations will be apparent to those skilled in the art from consideration of the specification disclosed herein.

CLAIMS:

1. A method for extracting phrases from text, comprising:
preprocessing desired phrases into at least one phrase indexing data structure for efficient matching;
scanning text to construct a hash table including keys and corresponding entries;
locating suffix trie trees for each word in the hash table;
matching each position in the hash table against the suffix trie trees; and
outputting phrases matched in the scanned text.
2. The method of claim 1 wherein preprocessing further includes identifying a keyword for each of the desired phrases.
3. The method of claim 2 wherein the keyword is a word appearing least frequent in a typical text.
4. The method of claim 2 wherein preprocessing further includes grouping the desired phrases by keyword.
5. The method of claim 2 wherein preprocessing further includes breaking each phrase at a first encountered keyword.

6. The method of claim 1 wherein preprocessing further includes building the suffix trie trees and storing the suffix trie trees in the at least one phrase indexing data structure.
7. The method of claim 1 wherein preprocessing further comprises reducing false positives using related phrases.
8. The method of claim 7 further comprising determining relevance of the related phrases to the desired phrases.
9. The method of claim 9 further comprising selecting top related phrases for each desired phrase based on relevance.
10. A system for extracting phrases from text, comprising:
 - at least one phrase indexing data structure residing in memory, the at least one phrase indexing data structure including desired phrases;
 - a hash table constructed in memory, the hash table including keys and corresponding entries; and
 - program code for locating suffix trie trees for each word in the hash table and matching each position in the hash table against the suffix trie trees to match phrases in a scanned text.

11. The system of claim 10 wherein the keyword for each of the desired phrases is identified.

12. The system of claim 12 wherein the keyword is a word appearing least frequent in a typical text.

13. The system of claim 12 wherein the desired phrases are grouped by keyword.

14. The system of claim 12 wherein each phrase is broken at a first encountered keyword.

15. The system of claim 10 wherein the suffix trie trees are stored in the hash table.

16. The system of claim 10 wherein the hash table includes keys and corresponding entries, the keys being words contained the text and the corresponding entries being the list of positions of the words in the text.

17. The system of claim 10 wherein the program code executes a preprocessing operation to improve accuracy of matches by reducing false positives using related phrases.

18. The system of claim 10 wherein phrases extracted from text are movie titles, book titles, album names, or sport team names.

19. The system of claim 10 wherein a desired phrase is extracted from a large database.

20. A system for extracting movie titles from web-based text, comprising:
memory means for storing at least one phrase indexing data structure including desired movie titles;
table means for storing keys and corresponding entries; and
means for locating suffix trie trees for each word in the hash table and matching each position in the hash table against the suffix trie trees to match movie titles in a scanned web-based text.

21. The system of claim 20 further comprising preprocessing means for improving accuracy of matches of movie titles in the web-based text by reducing false positives using phrases related to the movie titles.

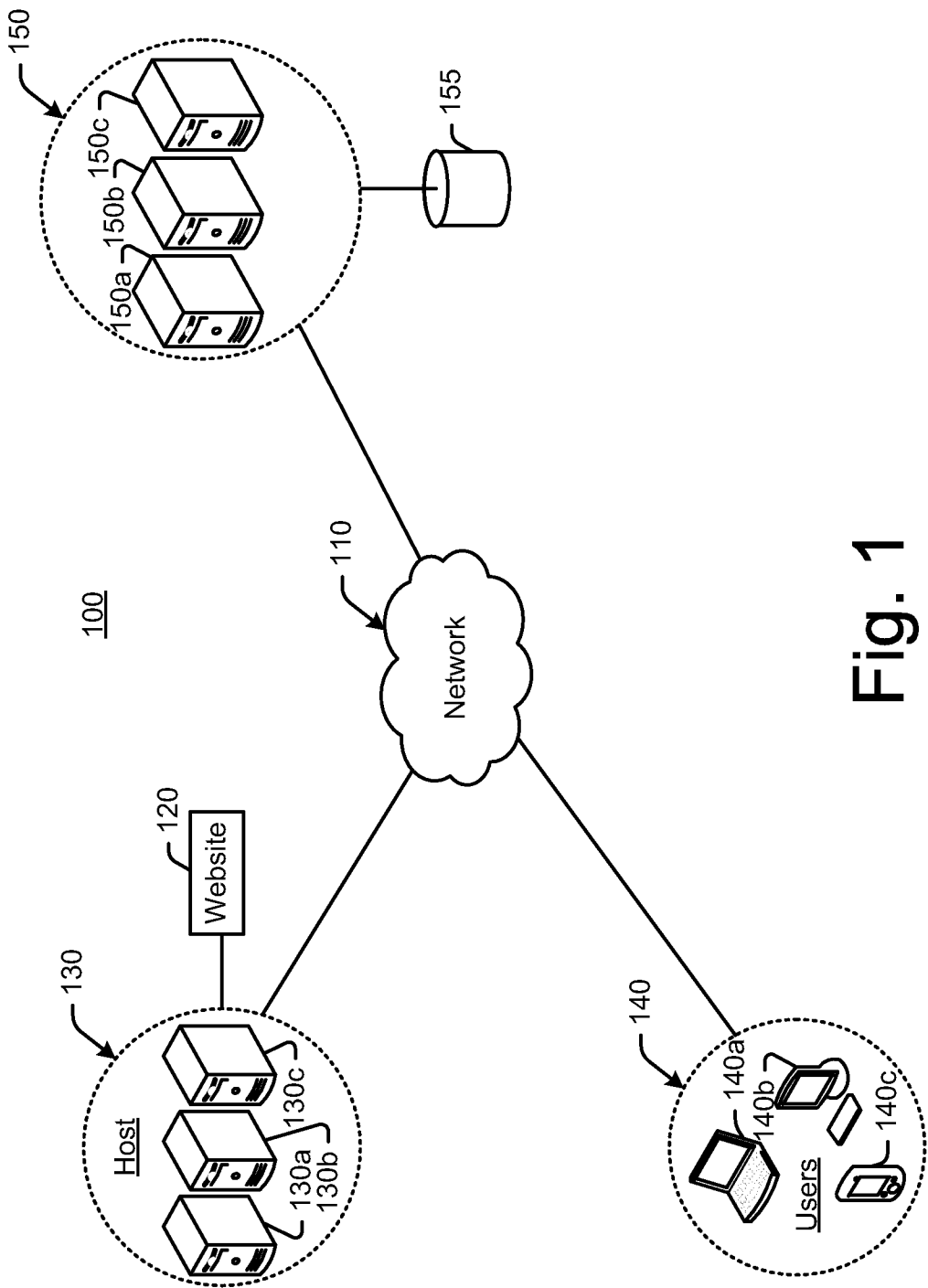
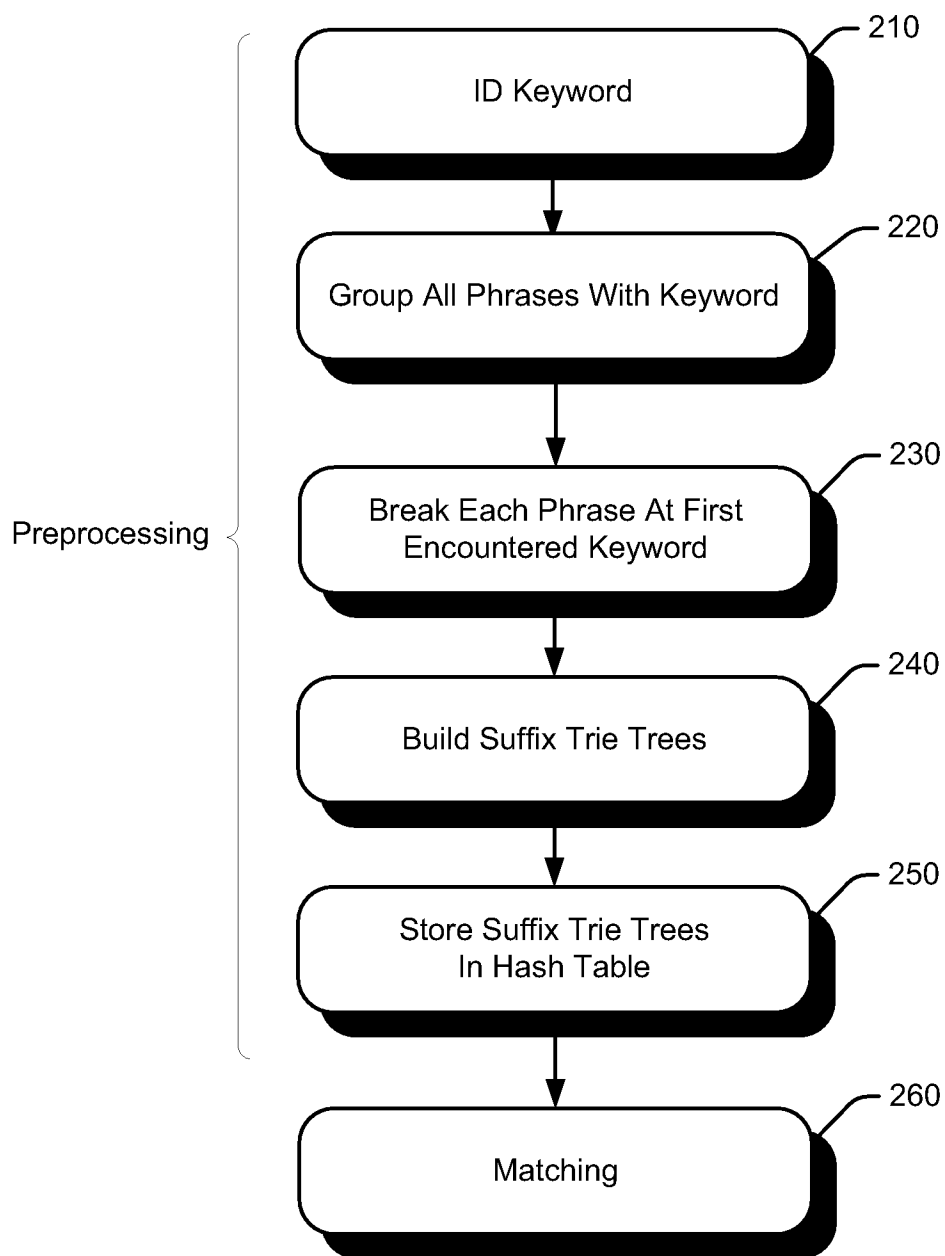


Fig. 1

2/4

200**Fig. 2**

3/4

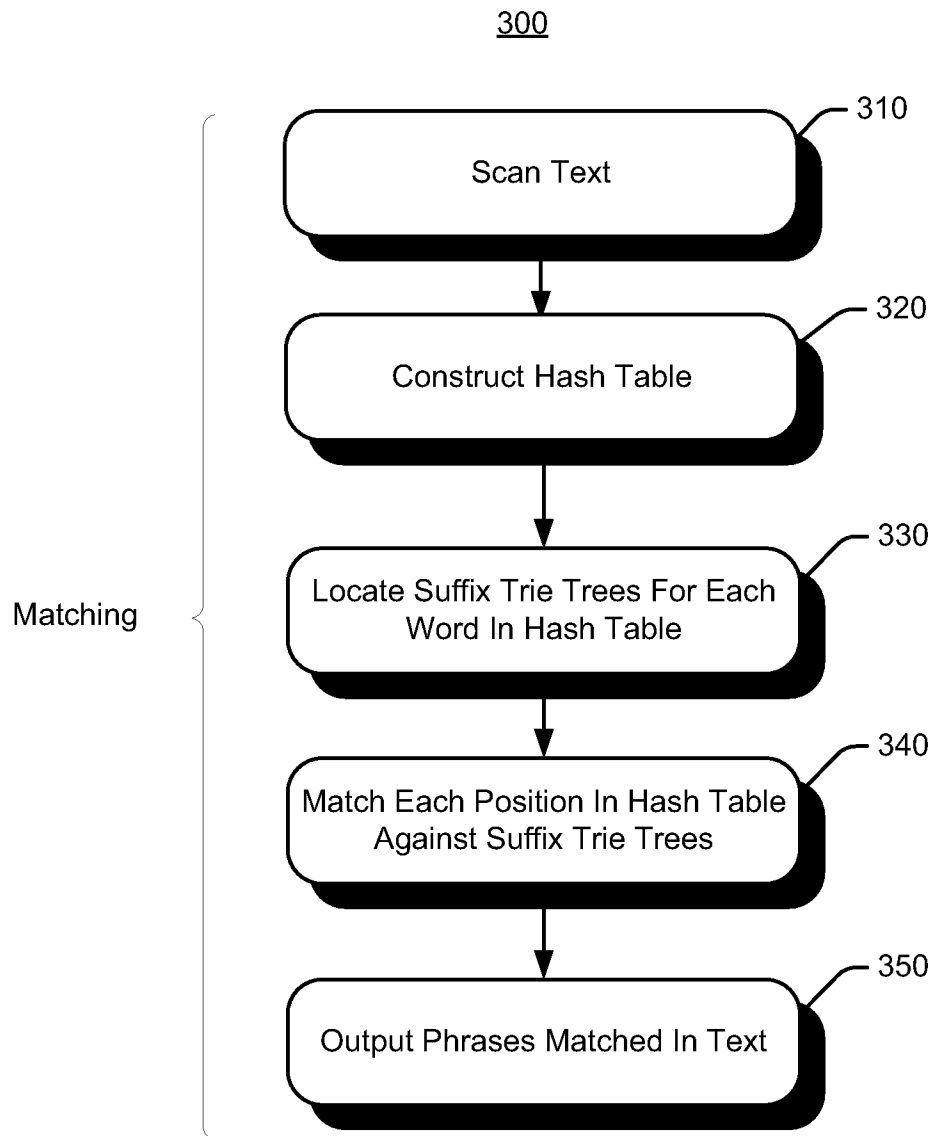
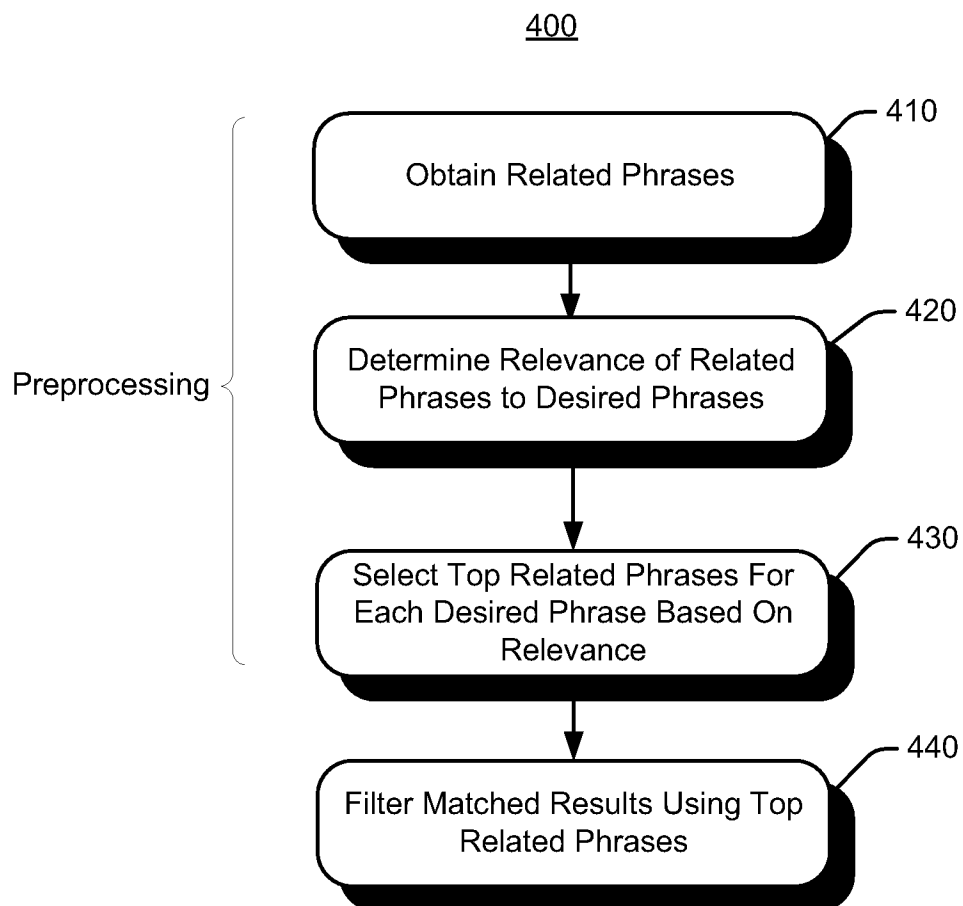


Fig. 3

4/4

**Fig. 4**

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2007/071233

A. CLASSIFICATION OF SUBJECT MATTER

G06F17/30 (2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC:G06F17/-

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

WPI;EPODOC;PAJ;CNKI;JEE;CPRS: text phrase extract indexing hash w table trie w tree keyword program w code

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 2007140861 A (KONICA MINOLTA MG KK) 07 Jun.2007(07.06.2007) See the whole document	1-21
A	US 2007225968 A1(IBM CORP) 27 Sep.2007(27.09.2007) See the whole document	1-21
A	US 6721728 B2(MCGREEVY M W) 13 Apr.2004(13.04.2004) See the whole document	1-21

☐ Further documents are listed in the continuation of Box C.☒ See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“E” earlier application or patent but published on or after the international filing date

“L” document which may throw doubts on priority claim (S) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&”document member of the same patent family

Date of the actual completion of the international search

05 Sep.2008(05.09.2008)

Date of mailing of the international search report

25 Sep. 2008 (25.09.2008)

Name and mailing address of the ISA/CN

The State Intellectual Property Office, the P.R.China
6 Xitucheng Rd., Jimen Bridge, Haidian District, Beijing, China
100088

Facsimile No. 86-10-62019451

Authorized officer

HUANG,Yuan

Telephone No. (86-10)62411769

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2007/071233

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
JP 2007140861 A	07.06.2007	None	
US 2007225968 A1	27.09.2007	CN 101093504 A	26.12.2007
		JP 2007257390 A	04.10.2007
US 6721728 B2	13.04.2004	AU 2002245542 A1	08.10.2002
		WO 02077872 A2	03.10.2002
		WO 02077872 A3	29.01.2004
		US 2002188599 A1	12.12.2002
		AU 2002245542 A8	20.10.2005