Innovation, Sciences et Développement économique Canada
Office de la Propriété Intellectuelle du Canada

Innovation, Science and Economic Development Canada
Canadian Intellectual Property Office

CA 2861121 C 2020/07/07

(11)(21) 2 861 121

(12) BREVET CANADIEN
CANADIAN PATENT

(13) C

(86) Date de dépôt PCT/PCT Filing Date: 2013/02/01
(87) Date publication PCT/PCT Publication Date: 2013/09/06
(45) Date de délivrance/Issue Date: 2020/07/07
(85) Entrée phase nationale/National Entry: 2014/07/14
(86) N° demande PCT/PCT Application No.: US 2013/024247
(87) N° publication PCT/PCT Publication No.: 2013/130215
(30) Priorité/Priority: 2012/02/29 (US13/408,853)

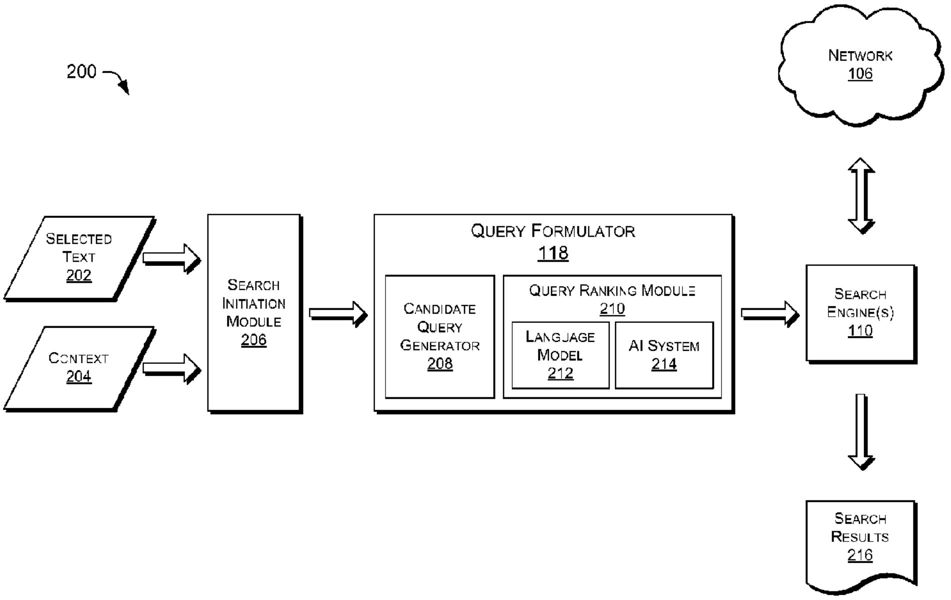
(51) Cl.Int./Int.Cl. *G06F 16/903* (2019.01), *G06F 16/9032* (2019.01)

(72) Inventeurs/Inventors:
BAI, PENG, US;
CHEN, ZHENG, US;
HUANG, XUEDONG DAVID, US;
NI, XIAOCHUAN, US;
SUN, JIAN-TAO, US;
ZHANG, ZHIMIN, US

(73) Propriétaire/Owner:
MICROSOFT TECHNOLOGY LICENSING, LLC, US

(74) Agent: SMART & BIGGAR LLP

(54) Titre : FORMATION DE REQUETE DE RECHERCHE SUR LA BASE D'UN CONTEXTE
(54) Title: CONTEXT-BASED SEARCH QUERY FORMATION



(57) Abrégé/Abstract:
Searching is assisted by recognizing a selection of text from a document as an indication that a user wishes to initiate a search based on the selected text. The user is provided with query suggestions based on the selected text and the query suggestions are ranked based on a context provided by the document. The user may select the text by using a mouse, drawing a circle around the text on a touch screen, or by other input techniques. The query suggestions may be based on query reformulation or query expansion techniques applied to the selected text. Context provided by the document is used by a language model and/or an artificial intelligence system to rank the query suggestions in predicted order of relevance based on the selected text and the context.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
6 September 2013 (06.09.2013)

WIPO | PCT

(10) International Publication Number
WO 2013/130215 A1

(51) International Patent Classification:
G06F 17/30 (2006.01) *G06F 3/048* (2013.01)
G06F 17/20 (2006.01) *G06F 3/14* (2006.01)

(21) International Application Number:
PCT/US2013/024247

(22) International Filing Date:
1 February 2013 (01.02.2013)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
13/408,853 29 February 2012 (29.02.2012) US

(71) Applicant *(for all designated States except US)*: **MICROSOFT CORPORATION** [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(72) Inventors: **BAI, Peng**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **CHEN, Zheng**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **HUANG, Xuedong David**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **NI, Xiaochuan**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **SUN, Jian-Tao**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **ZHANG, Zhimin**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, 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, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, 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, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: CONTEXT-BASED SEARCH QUERY FORMATION

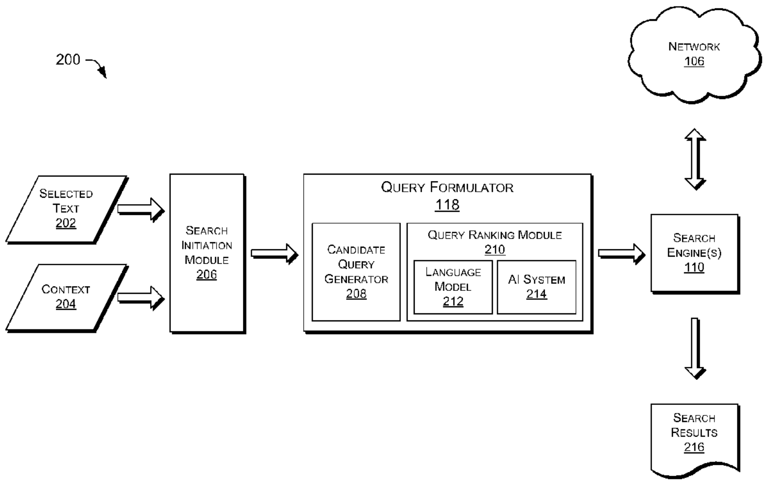


Fig. 2

(57) **Abstract**: Searching is assisted by recognizing a selection of text from a document as an indication that a user wishes to initiate a search based on the selected text. The user is provided with query suggestions based on the selected text and the query suggestions are ranked based on a context provided by the document. The user may select the text by using a mouse, drawing a circle around the text on a touch screen, or by other input techniques. The query suggestions may be based on query reformulation or query expansion techniques applied to the selected text. Context provided by the document is used by a language model and/or an artificial intelligence system to rank the query suggestions in predicted order of relevance based on the selected text and the context.

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

— *with international search report (Art. 21(3))*

CONTEXT-BASED SEARCH QUERY FORMATION

BACKGROUND

[0001] Many Internet searches are triggered by a web page that a user is browsing. That is, the user decides to initiate a search after consuming content on the web page. In order to implement the search, the user must leave the web page to access a search engine. The user may copy and paste words from the web page into a search box or manually compose a search query for entry into a search box or a search engine web page. Either technique for generating the search query may suffer from deficiencies such as lack of specificity, search terms with multiple meanings, and ambiguous relationships between the search terms.

[0002] After the search results are returned, the user may leave the searching interface and return to web browsing. This alternation between a web page and a searching interface is inefficient. Moreover, the interaction with various user interfaces (e.g., text selection, copy, paste, etc.) can become tedious particularly on small form factor devices or devices with limited ability to input text such as mobile phones, tablets computers, game consoles, televisions, etc. As an increasing number of users accesses web pages and other electronic documents through devices other than traditional computers, there will be an increasing need to smoothly integrate document consumption and searching. A system that can do so and additionally provide improved search queries will benefit users.

SUMMARY

[0003] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter.

[0004] This disclosure explains techniques for using both the area of a user's attention on a web page, or other document, as well as the surrounding context to generate and rank multiple search queries. While browsing a web page the user selects text from the web page. The selection of the text also generates a command to use that text as the starting point for generating candidate queries—search queries that may yield results relevant to the selected text. Multiple types of search query expansion or search query reformulation techniques may be applied to generate multiple candidate queries from the selected text. The user may then select one of these search queries to submit to a search engine. Thus, the act of browsing is combined with the act of searching, creating an interface that enables "Browsing to Search" by simply selecting text from the web page and then selecting one of the candidate queries.

[0005] In order to guide the user to a search query from the set of candidate queries, the context of the document is considered. Evaluation of the candidate queries in light of the context provided by the browsed web page is used to rank the respective candidate queries. Considering the surrounding context aids in ranking the candidate queries because the browsed web page may contain words which can be used (possibly with modifications) to disambiguate terms in the candidate queries and compare the candidate queries to previous search queries related to the same web page.

[0006] Ranking of the candidate queries may be performed by a language model, a classification method, or a combination of both. The language model may be implemented as a model that determines the probability of a candidate query given the selected text and the surrounding context. The classification method uses training data that contains selected text on web pages and associated queries. Human reviewers determine if the selected text of the web page likely resulted in a user making the associated search query. If so, the selected text and query pair is used by a machine learning system to learn a function that predicts a confidence level for a candidate query given the selected text and the context.

[0006a] According to one aspect of the present invention, there is provided an information-processing system comprising: one or more processing elements; a search initiation module communicatively coupled to or integrated with the one or more processing elements, the search initiation module configured to receive: an input indicating selected text from a document displayed via a device; a context including a portion of the document, the portion of the document being additional text that is relative to, and separate from, the selected text; and a command to generate a search query based at least in part on the selected text and the context; a candidate query generator coupled to or integrated with the one or more processing elements and configured to identify a plurality of candidate queries based at least in part on the selected text, the context, and a query log of queries associated with the document, the query log of queries including search queries generated by users after viewing the document; and a query ranking module coupled to or integrated with the one or more processing elements and configured to: compare each candidate query of the plurality of candidate queries to the selected text and the context; determine a value associated with each candidate query, the value representing a likelihood that the candidate query represents a meaning of the selected text and the context; and rank the plurality of candidate queries based at least in part on the value; a display to present a subset of the plurality of candidate queries in a list ordered at least partly according to the ranking; and the one or more processing elements further configured to receive a second selection of a candidate query of the subset of the plurality of candidate queries.

[0006b] According to another aspect of the present invention, there is provided a method comprising: receiving a first selection of text in a document displayed via a device; receiving a context associated with the first selection, the context including additional text from the document that is relative to, and separate from, the first selection; obtaining a plurality of candidate queries that includes queries generated at least in part by applying one or more query expansion techniques to the text and the additional text; comparing individual candidate queries of the plurality of candidate queries to the text and the context; determining values for each individual candidate query, the values representing likelihoods that respective ones of the individual candidate queries represent a meaning of the text and the context; ranking, by one or more processing elements, the plurality of candidate queries based at least in part on the values; presenting the plurality of candidate queries in a list ordered at least partly according to the ranking; receiving a second selection of a candidate query of the plurality of candidate queries; and submitting the candidate query to a search engine.

[0006c] According to still another aspect of the present invention, there is provided one or more computer storage media, wherein the one or more computer storage media is at least one device, having stored thereon computer-executable instructions which, when executed by a processor, cause a computing system to: receive a first selection of text in a document displayed via a device; receive a context associated with the text, the context including additional text from the document that is relative to, and separate from, the first selection; interpret the first selection of the text as a command to provide one or more search queries based at least in part on the text; obtain a plurality of candidate queries based at least in part on the text; compare individual candidate queries of the plurality of candidate queries to the text and the context; determine values for each individual candidate query, the values representing likelihoods that respective ones of the individual candidate queries represent a meaning of the text and the context; rank the plurality of candidate queries based at least in part on the values to determine a ranking of the plurality of candidate queries; present a subset of the plurality of candidate queries in a list ordered at least partly according to the ranking; and receive a second selection of a candidate query of the subset of the plurality of candidate queries.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] The Detailed Description is set forth with reference to the accompanying figures. In the figures, the left-most digit(s) of a reference number identifies the figure in which the reference number first appears. The use of the same reference numbers in different figures indicates similar or identical items.

[0008] Fig. 1 is an illustrative architecture showing an information-processing system including a query formulator.

[0009] Fig. 2 shows a schematic representation of illustrative data and components from the architecture of Fig. 1.

[0010] Fig. 3 shows an illustrative document with selected text.

[0011] Fig. 4 shows two illustrative user interfaces for selecting text.

[0012] Fig. 5 is an illustrative flowchart showing an illustrative method of providing a ranked listed of candidate queries in response to a user selection of text.

DETAILED DESCRIPTION

Illustrative Architecture

[0013] Fig. 1 shows an architecture 100 in which a user 102 can interact with a local computing device 104 to obtain search queries. The local computing device 104 may be any type of computing device such as a desktop computer, a notebook computer, a tablet computer, a smart phone, a game console, a television, etc. Local computing device 104 may communicate via a network 106 with one or more network-accessible computing devices 108. The network 106 may be any one or more types of data communications networks such as a local area network, wide area network, the Internet, a telephone network, a cable network, peer-to-peer network, a mesh network, and the like. The network-accessible computing devices 108 may be implemented as any type or combination of types of computing devices such as network servers, Web servers, file servers, supercomputers, desktop computers, and the like. The network-accessible computing devices 108 may include or be commutatively connected to one or more search engines 110. The search engine(s) 110 may be implemented on one or more dedicated computing devices maintained by an entity that provides the searching services.

[0014] An information-processing system 112 contains one or more processing elements 114 and memory 116 distributed throughout one or more locations. The processing elements 114 may include any combination of central processing units (CPUs), graphical processing units (GPUs), single core processors, multi-core processors, application-specific integrated circuits (ASICs), and the like. One or more processing element(s) 114 may be implemented in software and/or firmware in addition to hardware implementations. Software or firmware implementations of the processing element(s) 114 may include computer- or machine-executable instructions written in any suitable programming language to perform the various functions described. Software implementations of the processing elements(s) 114 may be stored in whole or part in the memory 116.

[0015] The memory 116 may store programs of instructions that are loadable and executable on the processing element(s) 114, as well as data generated during the execution of these programs. Examples of programs and data stored on the memory 116 may include an operating system for controlling operations of hardware and software resources available to the local computing device 104, the network-accessible computing device(s) 108, drivers for interacting with hardware devices, communication protocols for sending and/or receiving data to and from the network 106 as well as other computing devices, and additional software applications. Depending on the configuration and type of local computing device 104 and/or the network-accessible computing device(s) 108, the memory 116 may be volatile (such as RAM) and/or non-volatile (such as ROM, flash memory, etc.).

[0016] The information-processing system 112 may also include additional computer-readable media such as removable storage, non-removable storage, local storage, and/or remote storage. The memory 116 and any associated computer-readable media may provide storage of computer readable instructions, data structures, program modules, and other data. Computer-readable media includes, at least, two types of computer-readable media, namely computer storage media and communications media.

[0017] Computer storage media includes volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules, or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other non-transmission medium that can be used to store information for access by a computing device.

[0018] In contrast, communication media may embody computer readable instructions, data structures, program modules, or other data in a modulated data signal, such as a carrier wave, or other transmission mechanism. As defined herein, computer storage media does not include communication media.

[0019] The information-processing system 112 may exist in whole or part on either or both of the local computing device 104 and the network-accessible computing device(s) 108. Thus, the information-processing system 112 may be a distributed system in which various physical and data components exist at one or more locations and function together to perform the role of the information-processing system 112. In some implementations all features of the information-processing system 112 may be present on the local computing device 104. In other implementations, the local computing device 104 may be a thin client that merely receives display data and transmits user input signals to another device, such as the network-accessible computing device(s) 108, which contains the information-processing system 112.

[0020] The information-processing system 112 may contain a query formulator 118 that formulates search queries for the user 102. In some implementations, the query formulator 118 may be storage in whole or part in the memory 116. In other implementations, the query formulator 118 may be implemented as part of the processing element(s) 114 such as a portion of an ASIC. Like the information-processing system 112 itself, the query formulator 118 may exist in whole or part on either or both of the local computing device 104 and the network-accessible computing device(s) 108. In implementations in which all or part of the query formulator 118 is located redundantly on multiple computing devices, selection of which computing device to use for implementing the query formulator 118 may be based on relative processing speeds, a speed of information transmission across the network 106, and/or other factors.

[0021] FIG. 2 shows information and data flow through the query formulator 118 and other portions of the architecture 100 shown in FIG. 1. When the user 102 selects text from a document this provides the inputs for the query formulator 118 to formulate queries. The selected text 202 and the context 204 are received by a search initiation module 206. The selected text 202 may be selected by the user 102 interacting with the local computing device 104 to select or indicate a passage or passages of text by any conventional mechanism for selecting text from a document. The context 204 may include other text in the document that surrounds or is located near the selected text 202. The context 204 may also include classification of the document based on intended or likely use of the document. For example, if the document is a web page and the web page is identified as a merchant web page for selling goods and services, then the context 204 may recognize that the user 102 is likely searching for a good or service to purchase. Previous actions of the user 102 before selecting the text 202 may also provide the context 204. For example, search queries recently submitted by the user 102 may provide context 204 regarding the topic or area that the user 102 is currently searching.

[0022] The search initiation module 206 may interpret a single input from the user that selects the selected text 202 as a selection of text and as a command to generate a search query based on the selected text 202. For example, if the user 102 moves a cursor to select a contiguous series of text from a document, the user 102 does not need to paste or move this text to a different interface to receive search query suggestions. Selection of the text itself may be interpreted by the search initiation module 206 as a command to generate one or more search queries. This dual role of the search initiation module 206 allows the user to both select text and request search queries with only a single input or interaction with the local computing device 104.

[0023] The search initiation module 206 passes the selected text 202, the context 204, and the command to generate search queries to the query formulator 118. The query formulator 118 may include a candidate query generator 208 that generates candidate queries from the selected text 202. The candidate query generator 208 may apply query expansion or query reformulation techniques to the selected text 202. The candidate query generator 208 may create candidate queries from the selected text 202 by including synonyms, adding alternate morphological forms of words, correct spellings of misspelled words, and/or providing alternative spellings of words. When users fail to precisely select text of interest, e.g., when the text is selected by drawing an oval around it (using finger), a word or phrase may be accidently split into two parts. The post processing work may include removing irrelevant characters or prefixing/appending relevant characters from the selected text. In some implementations, a query log of queries associated with the document is used to generate candidate queries. Query expansion techniques that use the query log may include applying a K-means algorithm to the query log, conducting a random walk on a bipartite query-document graph generated by parsing the query log, running a PageRank algorithm on a query-flow graph generated from the query log, or mining term association patterns from the query log.

[0024] The candidate query generator 208 may directly generate candidate queries or the candidate query generator 208 may pass the selected text 202 to another module or system outside of the query formulator 118 (e.g., a query reformulator module associated with a search engine). The candidate query generator 208 may effectively generate the candidate queries by passing the selected text 202 to another system or module and then receiving the candidate queries from the outside module or system. The candidate query generator 208 may generate any number of queries from the selected text 202. In some implementations, the number of candidate queries generated by the candidate query generator 208 may be limited to a predefined number such as 3 queries, 10 queries, etc.

[0025] Once a number of candidate queries are obtained, a query ranking module 210 may rank the candidate queries based on a likelihood or probability that those queries correspond to the selected text 202 and the context 204. The query formulator 118 may perform both the generation of candidate queries and the ranking of those candidate queries without submitting inquiries to the search engine 110 thereby reducing the burden on the search engine 110.

[0026] The query ranking module 210 may rank the one or more candidate queries based one or more ranking techniques. Ranking techniques that may be used include a language model 212 and an artificial intelligence (AI) system 214. Each may be used independently or in combination.

[0027] The language model 212 may create a bi-gram representation of the context 204 and the selected text 202. The context 204 may include a portion of text from the document that includes the selected text 202. Thus, the context 204 may be the selected text 202 plus additional text from the document. The language model 212 may determine relative rankings of the candidate queries from the candidate query generator 208 based on a number of words in each of the respective of candidate queries, a number of words in the selected text 202, and a number of words in the portion of text that makes up the context 204. Details of one implementation of the language model 212 are discussed below.

[0028] The artificial intelligence system 214 may be implemented as any type of artificial intelligence or machine system such as a support vector machine, neural network, expert system, Bayesian belief network, fuzzy logic engine, data fusion engine, and the like. The artificial intelligence system 214 may be created from human-labeled training data. A corpus of <document, query> tuples representing documents and queries associated with those documents obtained from past document consumption and searching behavior of one or more users may serve as all or part of the training data. In some implementations, the tuples may be obtained from search logs from the search engine 110 from users that have elected to provide their browsing and search behavior to the search engine 110. The browsing and search data may be anonymized to protect the privacy of users who choose to contribute their data. The human labelers review the tuples determine if there is a causal relationship between the document and the query. In other words, the human labelers assign a label to each tuple based on their subjective evaluation of the probability that content of a document in a tuple caused the users to submit the query in the tuple. Details of one implementation of the artificial intelligence system 214 are discussed below.

[0029] Once the query formulator 118 has formulated queries and ranked those queries, the user 102 may be presented with a ranked list of the queries. The queries with higher rankings may be listed earlier or in a more prominent position in the list than those queries with lower rankings. The user 102 may select one of the candidate queries to initiate a search on one or more search engines 110 based on the query.

[0030] The search engine(s) 110 may submit the query to the network 106 or another data store and receive search results 216 based on the search algorithm, the selected query, and the data available in the network 106. The search engine(s) 110 may use any conventional searching technique for processing the selected search query.

Illustrative Language Model

[0031] The language model 212 ranks the candidate queries based on the context 204. The candidate queries are ranked by a conditional probability $p(q|s,c)$, which represents the possibility of one of the queries from the candidate queries, query *q*, to be generated given the selected text 202, represented as *s*, and the context 204, represented as *c*. The language model 212 assumes that $q = qw_1, qw_2, qw_{\mathrm{N_q}}$, $s = sw_1, sw_2, sw_{\mathrm{N_s}}$, and $c = cw_1, cw_2, cw_{\mathrm{N_c}}$where qw_i, sw_i, and cw_i represent the i^{th} word in query *q*, selected text *s*, and context *c* respectively. In the language model 212, N_q denotes the word length of query *q*, N_s denotes the word length of selected text *s*, and N_c denotes the word length of context *c*.

[0032] The language model 212 includes a further assumption that, conditioned by the selected text *s* and context *c*, each query word qw_i is only dependent on its preceding word qw_{i-1}. This assumption is similar that the assumption made for a bi-gram language model. A bi-gram representation is desirable for some implementations because a uni-gram model may not catch the term-level relationship inside a query. Conversely, n-gram (n ≥ 3) approaches may have high computational complexity that could potentially be too time-consuming for online query suggestion. However, as processing capabilities continue to increase the computational complexity of 3-gram (or higher) approaches will likely become less time consuming and it is contemplated that the language model 212 can be adapted to accommodate n-gram (n ≥ 3) approaches.

[0033] From the definitions and assumptions above, the possibility of one of the queries from the candidate queries to be generated given the selected text 202 and the context 204 may be represented as:

$$(1) \quad p(q|s,c) = p(qw_1|s,c)\prod_{i=2}^{N_q} p(qw_i\,|s,c,qw_i - 1).$$

[0034] In the above formulation that longer queries tend to have smaller probabilities. To alleviate this effect, the probability is multiplied by an additional weight and longer query is assigned a larger weight. The revised probability can be calculated by:

$$(2) \quad p(q|s,c) \propto \Lambda^{N_q} \cdot p(qw_1|s,c)\prod_{i=2}^{N_q} p(qw_i\,|s,c,qw_i - 1)$$

where Λ is a constant larger than 1.

[0035] The formulation to calculate $p(qw_i|s,c)$ is:

$$(3)\quad p(qw_i|s,c) = \frac{p(qw_i,s,c)}{p(s,c)} \propto p(qw_i,s,c) = p(qw_i)p(s,c|qw_i)$$

p(s,c) can be ignored here since each of the candidate queries being ranked is based on the same selected text *s* and context *c*.

[0036] A global query corpus can be used to estimate the value of *p(qw$_i$)*. Given a query corpus *Q*, the value of *p(qw$_i$)* can be computed by:

$$(4)\quad p(qw_i) = \frac{|\mathrm{Q}(qw_i)|}{|\mathrm{Q}|}$$

where $|\mathrm{Q}(qw_i)|$ denotes the number of queries in the query corpus which contain the word qw_i and $|\mathrm{Q}|$ stands for the total number of queries in the global query corpus.

[0037] A smoothed version of equation 4 may be used:

$$(5)\quad \tilde{p}(qw_i) = \frac{|\mathrm{Q}(qw_i)| + a \cdot (|Q| + 1)}{|\mathrm{Q}| + (|\mathrm{Q}| + 1)}$$

where *a* is a constant between 0 and 1.

[0038] Another probability in equation 3 can be derived as follows. Assuming that the selected text *s* and context *c* are independent conditioned by any query word qw_i:

$$(6)\quad p(s,c|qw_i) = p(s|qw_i)p(c|qw_i).$$

[0039] To simplify the function, the language model 212 further assumes that conditioned by any query word qw_i, the words of selected text *s* or context *c* can be generated independently. Thus,

$$(7)\quad p(s|qw_i) = \prod_{j=1}^{N_s} p(sw_j\,|qw_i)$$

$$(8)\quad p(c|qw_i) = \prod_{j=1}^{N_c} p(cw_j\,|qw_i)$$

where $p(sw_j|qw_i)$ is the probability of sw_j to appear together with qw_i when qw_i exists. This probability can be estimated using the global query corpus:

$$(9)\quad p(sw_j|qw_i) = \frac{|\mathrm{Q}(sw_j) \cap Q(qw_i)| + a \cdot (|Q(qw_i)| + 1)}{Q(qw_i) + (|\mathrm{Q}(qw_i)| + 1)}$$

where $|\mathrm{Q}(sw_j) \cap Q(qw_i)|$ is the number of queries containing sw_j and qw_i simultaneously in the global query corpus, $|Q(qw_i)|$ denotes the number of queries in the query corpus which contain the word qw_i, and $a \in (0,1)$ is used for smoothing.

[0040] The value of $p(cw_j|qw_i)$ can be computed similarly. According to equations 7 and 8, the values of $p(s|qw_i)$ and $p(c|qw_i)$ are unbalanced since N_s is always much smaller than N_c. The normalized values of $p(s|qw_i)$ and $p(c|qw_i)$ may be used to solve this unbalance.

[0041] The normalized formulation of $p(s|qw_i)$ is:

$$(10)\quad \tilde{p}(s|qw_\mathrm{i}) = \prod_{j=1}^{N_s} p(sw_j\,|qw_i))^{\frac{1}{N_s}}.$$

[0042] Similarly, the normalized value of $p(c|qw_i)$ can be calculated by:

$$(11)\quad \tilde{p}(c|qw_\mathrm{i}) = \prod_{j=1}^{N_c} p(cw_j\,|qw_i))^{\frac{1}{N_s}}.$$

[0043] The formulation for calculating $p(qw_{i-1}|s,c,qw_{i-1})$ is:

$$(12)\quad p(qw_{i-1}|s,c,qw_{i-1}) = \frac{p(qw_i,qw_{i-1}|s,c)}{p(qw_{i-1}|s,c)}$$

where $p(qw_{i-1}|s,c)$ can be calculated by equation 3. Because $p(s,c)$ takes the same value for all the candidate queries based on the same selected text 202 and the same context 204:

$$(13)\quad p(qw_i,qw_{i-1}|s,c) = \frac{p(qw_i,qw_{i-1},s,c)}{p(s,c)} \propto p(qw_i\,,qw_{i-1},s,c)$$
$$= p(qw_{i-1})p(qw_i|qw_{i-1})p(s,c|qw_i,qw_{i-1})$$

where $p(qw_{i-1})$ can be computed by equation 5. $p(qw_i|qw_{i-1})$ is the probability of qw_i to appear right after qw_{i-1} when qw_{i-1} exists. However, when calculating this probability using the global query corpus, the words qw_{i-1} and qw_i may seldom appear in succession because the global query corpus is sparse. To account for that possibility, $p(qw_i|qw_{i-1})$ may be estimated as the probability of qw_i to appear together with qw_{i-1} when qw_{i-1} exists (without requiring that qw_i and qw_{i-1} appear in immediate succession), which can be computed according to equation 9.

[0044] Finally, the formulation for calculating the probability $p(s,c|qw_i,qw_{i-1})$ is provided below. To simplify, the language model 212 assumes that the selected text s and context c are independent conditioned on the two query words qw_i and qw_{i-1}. This yields:

$$(14)\quad p(s,c|qw_i,qw_{i-1}) = p(s|qw_i,qw_{i-1})\cdot p(c|qw_1,qw_{i-1}).$$

[0045] Similar to equation 7, the language model 212 assumes that conditioned by the two query words qw_i and qw_{i-1}, the words in the selected text *s* or context *c* can be generated independently. Thus,

$$(15)\quad p(s|qw_{\mathrm{i}}, qw_{i-1}) = \prod_{j=1}^{N_s} p(sw_j\,|qw_i, qw_{i-1})$$

$$(16)\quad p(c|qw_{\mathrm{i}}, qw_{i-1}) = \prod_{j=1}^{N_c} p(cw_j\,|qw_i, qw_{i-1})$$

where $p\left(sw_j \middle| qw_i, qw_{i-1}\right)$ can be estimated by the global query corpus:

$$(17)\quad p\left(sw_j \middle| qw_i, qw_{i-1}\right) = \frac{\left|Q\left(sw_j\right) \cap Q(qw_i) \cap Q(qw_{i-1})\right| + aL}{|Q(qw_1) \cap Q(qw_{i-1})| + L}$$

$$(18)\quad L = |Q(qw_i) \cap Q(qw_{i-1})| + 1$$

where $\left|Q\left(sw_j\right) \cap Q(qw_i) \cap Q(qw_{i-1})\right|$ stands for the number of queries in the global query corpus which contain the words sw_j , qw_i, and qw_{i-1} simultaneously. $|Q(qw_i) \cap Q(qw_{i-1})|$ and *a* have similar meanings as in equation 9.

[0046] Similar to equation 10, the probability of $p(s|qw_{\mathrm{i}}, \mathrm{qw_{i-1}})$, may be normalized:

$$\tilde{p}(s|qw_{\mathrm{i}}, qw_{i-1}) = \prod_{j=1}^{N_s} p(sw_j\,|qw_i, qw_{i-1}))^{\frac{1}{N_s}}.$$

The value of $p(c|qw_{\mathrm{i}}, qw_{i-1})$ can be calculated and normalized similarly.

Illustrative Artificial Intelligence System

[0047] The artificial intelligence system 214 may implement a classification technique for ranking candidate queries. In the classification technique, human reviewers evaluate associations between documents and queries *q* associated with those documents. Prior to labeling by the human reviewers it may be unknown whether the content of the document caused the query or if the association between the document and the query is merely coincidental or unrelated to the document.

[0048] The human labelers classify the query from one of the document-query pairs as either associated with the content of the document, not associated with content of the document, or ambiguously related to the content of the document. Thus, the human labelers review a corpus of <document, query> tuples. The tuples may be generated by actual browsing and searching behavior of users and stored in a global query corpus. This may be the same global query corpus used by the language model 212. Each document in the <document, query> tuples may be represented as selected text *s* from the document and the context *c* that includes the selected text *s*. Therefore, the work of the human labelers may be represented as labeling pairs of <*s*, *c*> and *q*, which are then used as training data for the artificial intelligence system 214. In some implementations, only pairs of <*s*, *c*> and *q* in which the query is labeled as associated with the content of the document may be used as training data.

[0049] The artificial intelligence system 214 uses the training data to learn a function $f(\langle s, c\rangle q) \rightarrow \{-1, +1\}$. The function *f* can be applied to new data such as the selected text 202, context 204, and queries candidates from Fig. 2 to predict a confidence level that the respective candidate queries are associated with the content of the document. The confidence level for various query candidates may be used to rank the query candidates by the query ranking module 210.

[0050] The manual labeling of document-query relationships may be tedious. Pseudo-training data may be used to reduce the manual labeling efforts and to obtain a larger amount of training data to improve the accuracy of the function *f*. Pseudo-training data may be created by identifying search queries in the global query corpus that were submitted by users shortly after the users viewed a document paired with the query. This close temporal relationship may suggest that the content of the document caused the user to generate the query *q*. Automatic textual comparison of similarity between search query *q* and content *c* of the document may, or may not, identify a phrase *p* in the document that is similar to the search query *q*. If such phrase *p* is identified by the automatic analysis, it is assumed that the phrase *p*, given the surrounding context *c*, may have caused or induced the search query *q*. This generates (<*p*, *c*>, *q*) pairs without manual labeling that can be added to the training data for the artificial intelligence system 214.

Illustrative User Interfaces

[0051] Fig. 3 shows an illustrative document 300 that may be displayed on the local computing device 104. The document 300 may be a web page, a text document, a word processing document, a spreadsheet document, or any other type of document containing text in any format including, but not limited to, a document written in a markup language such as hypertext market language (HTML) or extensible markup language (XML). The document 300 illustrates multiple examples of context for text selected by the user 102.

[0052] User selected text 302 is shown by a bold rectangle surrounding the word or words selected by the user 102. The user 102 may also select partial words or single characters. The selected text 302 indicates a portion of the document 300 that is receiving the use's attention. The selected text 302 exists within the context of the document 300.

The amount of the document 300 that is considered as the context by the language model 212 or the artificial intelligence system 214 may vary.

[0053] In some implementations, the entire document 300 may provide the context for the selected text 302. The entire document 300 may include multiple pages some of which are not displayed and some of which may not have not been viewed by the user. A narrower view of the context may include only the sentence 304 that includes the selected text 302. In other implementations, the context may be defined as the paragraph 306 that includes the selected text 302, a column 308 (or frame in a web page layout) that includes the selected text 302, or a page 310 of the document 300 that includes the selected text 302. For any type of document including those documents without sentences, paragraphs and/or pages, the context may be defined as a relatively larger or relatively smaller portion of the entire document 300.

[0054] The context may also be a portion of text 312 that has a predefined number of words or characters and includes the selected text 302. For example, a 60 word segment of the document 300 including the selected text 302 may be used as the context. This portion of text 312 may span multiple sentences, paragraphs, columns, or the like and begin or end in the middle of a sentence, paragraph, column, etc. The 60 word length is merely illustrative and the context may be any length such as 100 words, 20 words or alternatively be based on characters and include 20 characters, 100 characters, 500 characters, or some other number of words or characters.

[0055] In some implementations, the selected text 302 is located substantially in the middle of the portion of text 312. For example, if the selected text 302 has three words and the portion of text 312 includes 60 words, then the selected text 302 may be located about 23 or 24 words (i.e., 60 - 3 = 57; 57 ÷ 2 = 23.5) from the beginning of the portion of text 312 that makes up the context. In some implementations, the selected text 302 may be located in the middle 50% of the portion of text 312 (i.e., not in the first ¼ and not in the last ¼) or in the middle 20% of the portion of text 308 (i.e., not in the first 40% and not in the last 40%).

[0056] Calculation of the number of words, or characters, in the portion of text 308 may exclude stop words in order to base the context on words that may be most useful for ranking search queries. For example, a 20-word context centered around the selected text 302 may be of less assistance in ranking search queries if words such as “a”, “the”, “and”, “it” and other types of stop words are included in the 20 words of the context. Thus, the predetermined number of words in the portion of text 302 that makes up the context may be a predetermined number of words excluding stop words.

[0057] Fig. 3 also illustrates the location of a pre-formulated search query 314 within the document 300. The pre-formulated search query 314 may be associated with a portion of the document prior to the selection of the text by the user. For example, the pre-formulated search query 314 may be associated with a particular word, sentence, paragraph, column, page, etc. in the document 300. This example shows the pre-formulated search query 314 as associated with the sentence immediately before the selected text 302. Depending on the extent of the document 300 that is considered as context for the selected text 302, the pre-formulated search query 314 may or may not be included in the same portion of the document 300 as the selected text 302. If, for example, the sentence 304 that contains the selected text 302 is the context, then the pre-formulated search query 314 is not associated with the same part of the document 300 as the selected text 302. However, if the context is the paragraph 306, then the pre-formulated search query 314 is associated with the same part of the document 300 as the selected text 302.

[0058] The document 300 may contain zero, one, or multiple pre-formulated search queries 314. The pre-formulated search query(s) 314 may be queries that a user would be likely to conduct when consuming the associated portion of the document 300. The pre-formulated search query(s) 314 may be manually crafted by a human author for embedding in a specific portion of the document 300. Alternatively, or additionally, one or more of the pre-formulated search query(s) 314 may be determined based on analysis of query logs from other users that view the document 300 and subsequently generated a search query.

[0059] The candidate query generator 208 shown in Fig. 2 may obtain the pre-formulated search query(s) 314 together with other search queries generated from the selected text 302. In some implementations, the candidate query generator 208 may include all pre-formulated search query(s) 314 associated with the document 300 in the list of search queries presented to the user. In other implementations, the candidate query generator 208 may include only the pre-determined search query(s) 314 that is associated with the same portion, based on the definition of context, of the document 300 as the selected text 302. In yet a further implementation, only a threshold number (e.g., 1, 2, 3) of pre-determined search query(s) 314 that are associated with a location in the document 300 that is closest to the location of the selected text 302 are included in the list of search queries presented to the user.

[0060] Once the user selects a query from the list of candidate queries, that selected query may be used as pre-determined search query 314 for subsequent presentations of the document 300. That pre-determined search query 314 may be associated with the location of the selected text 302 that originally generate the search query. Thus, the number of pre-determined search queries 314 associated with the document 300 may increase as use of the system increases.

[0061] Fig. 4 shows two illustrative user interfaces 400 and 402 for selecting text on a touch-screen device. The local computing device 104 from Fig. 1 may be implemented as a device that has a touch-screen display. In the first user interface 400 the user drags his or her finger (or other pointing implement such as a stylus) across the surface of the touch screen from a point 404 at the start of the text to select to a point 406 and the end of the text he or she wishes to select. The user may draw his finger through the middle of the text, along the bottom of the text as if he or she is underlining the text, or in another motion that is generally in line with the flow of the text (e.g., left to right for English, but the direction of movement may be different for different languages). The signal for the system to formulate search queries from the selected text may be cessation of movement of the finger when it comes to rest at the end point 406, lifting of the finger from the surface of the touch screen, a tap on the touch screen at the end point 406, etc.

[0062] The user may also select text, as shown in the second user interface 402, by moving a stylus (or other pointing implement such as a finger) in a generally circular shape around the text that the user intends to select. The generally circular shape may be more ovoid than circular in shape and it may be either a closed circle in which the starting point 408 and the ending point 410 touch or an open arc in which the starting point 408 is in a different location than the ending point 410.

[0063] In this example, the circle is drawn in a clockwise direction starting at a point 408 on the lower right of the selected text moving around to a point 410 at the top right of the selected text. In some implementations, circles drawn in either clockwise or counterclockwise directions may both cause the same result. However, in other implementations initiating the generation of search queries may occur only when the circle is drawn in a clockwise (or alternatively counterclockwise) direction. The signal for the system to formulate search queries from the selected text may be cessation of movement of the stylus when it comes to rest at the end point 410, lifting of the stylus from the surface of the touch screen, closure of the circle when the stylus returns to starting point

408, a tap on the touch screen at the end point 410, or some other gesture representing the end of text selection and requesting initiation of search queries generation.

[0064] Either of the user interfaces 402 and 404 shown in Fig. 4 provides a convenient way for the user to initiate the search process without multiple commands, use of a keyboard, or switching to an interface other than the document that he or she was consuming.

Illustrative Processes

[0065] For ease of understanding, the processes discussed in this disclosure are delineated as separate operations represented as independent blocks. However, these separately delineated operations should not be construed as necessarily order dependent in their performance. The order in which the processes are described is not intended to be construed as a limitation, and any number of the described process blocks may be combined in any order to implement the process, or an alternate process. Moreover, it is also possible that one or more of the provided operations may be modified or omitted.

[0066] The processes are illustrated as a collection of blocks in logical flowcharts, which represent a sequence of operations that can be implemented in hardware, software, or a combination of hardware and software. For discussion purposes, the processes are described with reference to the architectures, systems, and user interfaces shown in Figs. 1–4. However, the processes may be performed using different architectures, systems, and/or user interfaces.

[0067] Fig. 5 illustrates a flowchart of a process 500 for identifying and presenting candidate queries to a user. At 502, a selection by a user of text in a document is received. The user may be the user 102 shown in Fig. 1 and the selection may be received by the information-processing system 112. The selected text may be a contiguous series of text such as one, two, three, four, etc. words in a row or selections of multiple words or combinations words from multiple places in the document. The document may be a web page, a text document, a word processing document, an electronic book, or any other type of document.

[0068] At 504, multiple candidate queries are obtained. The candidate queries may be obtained directly or indirectly from the candidate query generator 208. The candidate queries are generated by applying one or more query expansion techniques to the text selected at 502. The query expansion techniques may include any technique that compares the selected text with a previous query log to identify one or more queries from the previous query log based on the selected text. Illustrative techniques include applying a K-means algorithm to a query log, conducting a random walk on a bipartite query-document graph generated by parsing a query log, running a PageRank algorithm on a query-flow graph generated from a query log, or mining term association patterns from a query log.

[0069] At 506, it is determined if there are any pre-formulated queries associated with the document. The pre-formulated queries may be identified based on query logs of past searching behavior, created by a human editor, or generated by any other technique for creating search queries. The pre-formulated queries may be associated with a specific portion of the document such as a specific word, sentence, paragraph, page, etc. such as, for example the pre-formulated query 314 shown in Fig. 3. When the text selected by the user is from the same portion of the document as the pre-formulated query, process 500 proceeds along the “yes” path to 508. If, however, the document is not associated with any pre-formulated queries or if the pre-formulated queries associated with the document are not associated with the portion of the document that includes the selected text, then process 500 proceeds along the “no” path to 510.

[0070] At 508, the pre-formulated query is included in the set of candidate queries obtained at 504. The pre-formulated query may be obtained faster than the other queries obtained at 504 because it is pre-formulated and may not require processing or analysis to generate.

[0071] At 510, the candidate queries obtained at 504, including any pre-formulated queries identified at 508, are ranked. The ranking of the candidate queries provides a higher rank to those queries that are more likely to return results desired by the user based on the text selected at 502. The ranking may be based on a language model 512 that considers a context provided by the document. The context may be represented by text in the document that includes the text selected by the user at 502 and additional text (i.e., the context includes at least one additional word or character more than the text selected by the user). The ranking may additionally or alternatively be based on an artificial intelligence system 514. The artificial intelligence system 514 is trained with a set of document and query pairs (i.e., training data) that is validated by human review. The human reviewers evaluate the document and query pairs to identify those that have a query which is related to the content of the document paired with the query.

[0072] At 516, the candidate queries are presented to the user in a ranked list ordered according to the ranking. The ranked list may be shown to the user in an interface that also displays the document from which the user selected the text so that the user can view the document and selected text while choosing a search query. Alternatively, the document may no longer be shown, but instead the document may be replaced by the list (e.g., on devices with display areas too small to show both). Additional techniques for displaying the list are also contemplated such as presenting the list in a pop-up box, a drop-down menu, etc. Thus, the selection of text at 502 may cause the display of a list of recommended queries ranked in order of relevance based on the selected text and the surrounding context.

[0073] At 518, a selection by the user of one of the candidate queries from the list is received. The user may make the selection by any conventional technique for selecting an item from a list. Thus, the user is able to take the search query from the list that most closely represents his or her intention when selecting the words at 502 to search.

[0074] At 520, the query selected by the user is submitted to one or more search engines such as search engine(s) 110. The user may then receive search results from the search engine. Thus, with this method 500 the user may obtain search results based on a search query that is better designed to generate effective results than simply searching for words in selected from the document and use the user can receive those results with only minimal interactions with the document and/or search engine interface.

Conclusion

[0075] The subject matter described above can be implemented in hardware, software, or in both hardware and software. Although implementations have been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the specific features or acts described above. Rather, the specific features and acts are disclosed as illustrative forms of illustrative implementations of generating search queries.

CLAIMS:

1. An information-processing system comprising:

one or more processing elements;

a search initiation module communicatively coupled to or integrated with the one or more processing elements, the search initiation module configured to receive:

an input indicating selected text from a document displayed via a device;

a context including a portion of the document, the portion of the document being additional text that is relative to, and separate from, the selected text; and

a command to generate a search query based at least in part on the selected text and the context;

a candidate query generator coupled to or integrated with the one or more processing elements and configured to identify a plurality of candidate queries based at least in part on the selected text, the context, and a query log of queries associated with the document, the query log of queries including search queries generated by users after viewing the document; and

a query ranking module coupled to or integrated with the one or more processing elements and configured to:

compare each candidate query of the plurality of candidate queries to the selected text and the context;

determine a value associated with each candidate query, the value representing a likelihood that the candidate query represents a meaning of the selected text and the context; and

rank the plurality of candidate queries based at least in part on the value;

a display to present a subset of the plurality of candidate queries in a list ordered at least partly according to the ranking; and

the one or more processing elements further configured to receive a second selection of a candidate query of the subset of the plurality of candidate queries.

2. The information-processing system of claim 1, wherein the candidate query generator is further configured to include, in one or more candidate queries of the plurality of candidate queries, at least one of synonyms of words in the selected text, alternate morphological forms of words in the selected text, correct spellings of misspelled words in the selected text, and alternative spellings of words in the selected text.

3. The information-processing system of claim 1, wherein the value for each candidate query is determined by an artificial intelligence system,

wherein the artificial intelligence system is trained with training data comprising a history of searches originated by users when browsing a corpus of documents, the history of searches being labeled by human labelers indicating a probability that content of a document caused a respective user to submit the corresponding query in the history.

4. The information-processing system of claim 1, wherein the portion of the document spans at least one sentence, paragraph, or column of the document.

5. The information-processing system of claim 1, wherein the portion of the document includes the selected text.

6. A method comprising:

receiving a first selection of text in a document displayed via a device;

receiving a context associated with the first selection, the context including additional text from the document that is relative to, and separate from, the first selection;

obtaining a plurality of candidate queries that includes queries generated at least in part by applying one or more query expansion techniques to the text and the additional text;

comparing individual candidate queries of the plurality of candidate queries to the text and the context;

determining values for each individual candidate query, the values representing likelihoods that respective ones of the individual candidate queries represent a meaning of the text and the context;

ranking, by one or more processing elements, the plurality of candidate queries based at least in part on the values;

presenting the plurality of candidate queries in a list ordered at least partly according to the ranking;

receiving a second selection of a candidate query of the plurality of candidate queries; and

submitting the candidate query to a search engine.

7. The method of claim 6, wherein the document comprises a mark-up language document.

8. The method of claim 6, wherein the plurality of candidate queries includes at least one pre-formulated query associated with the document.

9. The method of claim 6, wherein the one or more query expansion techniques comprise at least one of applying a K-means algorithm to a query log, conducting a random walk on a bipartite query-document graph generated by parsing a query log, running a PageRank algorithm on a query-flow graph generated from a query log, and mining term association patterns from a query log.

10. The method of claim 6, wherein the additional text comprises at least part of a

paragraph of the document, at least part of a column of the document, at least part of a sentence of the document, at least part of a cell of the document, and at least part of a frame of the document.

11. The method of claim 6, wherein:

ranking the plurality of candidate queries is further based on a language model; and

the language model is based at least in part on a number of words in the candidate query, a number of words in the text, and a number of words in the context.

12. The method of claim 6, wherein:

ranking the plurality of candidate queries is further based on a language model; and

the language model comprises a bi-gram language model in which a word in the candidate query depends on an immediately preceding word in the candidate query.

13. The method of claim 6, wherein:

ranking the plurality of candidate queries is further based on an artificial intelligence system; and

the artificial intelligence system learns a function that predicts a level of confidence in one or more candidate queries of the plurality of candidate queries given the candidate query, the text, and the context.

14. One or more computer storage media, wherein the one or more computer storage media is at least one device, having stored thereon computer-executable instructions which, when executed by a processor, cause a computing system to:

receive a first selection of text in a document displayed via a device;

receive a context associated with the text, the context including additional text from the document that is relative to, and separate from, the first selection;

interpret the first selection of the text as a command to provide one or more search queries based at least in part on the text;

obtain a plurality of candidate queries based at least in part on the text;

compare individual candidate queries of the plurality of candidate queries to the text and the context;

determine values for each individual candidate query, the values representing likelihoods that respective ones of the individual candidate queries represent a meaning of the text and the context;

rank the plurality of candidate queries based at least in part on the values to determine a ranking of the plurality of candidate queries;

present a subset of the plurality of candidate queries in a list ordered at least partly according to the ranking; and

receive a second selection of a candidate query of the subset of the plurality of candidate queries.

15. The one or more computer storage media of claim 14, wherein the computer-executable instructions, when executed by the processor, further cause the computing system to receive the first selection of the text based at least in part on a user dragging a pointing implement across the text that is displayed on a touch-screen display.

16. The one or more computer storage media of claim 14, wherein the computer-executable instructions, when executed by the processor, further cause the computing system to receive the first selection of the text based at least in part on a user moving a pointing implement in a circular or oval shape around the text that is displayed on a touch-screen display.

17. The one or more computer storage media of claim 14, wherein at least part of the plurality of candidate queries obtained comprise one or more pre-formulated queries

associated with the document that have been determined prior to the first selection of the text.

18. The one or more computer storage media of claim 17, wherein the one or more preformulated queries are associated with the additional text.

19. The one or more computer storage media of claim 14, wherein ranking the plurality of candidate queries is further based at least in part on using a bi-gram language model that ranks the plurality of candidate queries based at least in part on a number of words in candidate queries of the plurality of candidate queries and a query corpus.

20. The one or more computer storage media of claim 14, wherein ranking the plurality of candidate queries is further based at least in part on using an artificial intelligence system that uses a data set of document/query tuples, a correspondence between a document and a query in a respective one of the document/query tuples having been verified by a human reviewer.

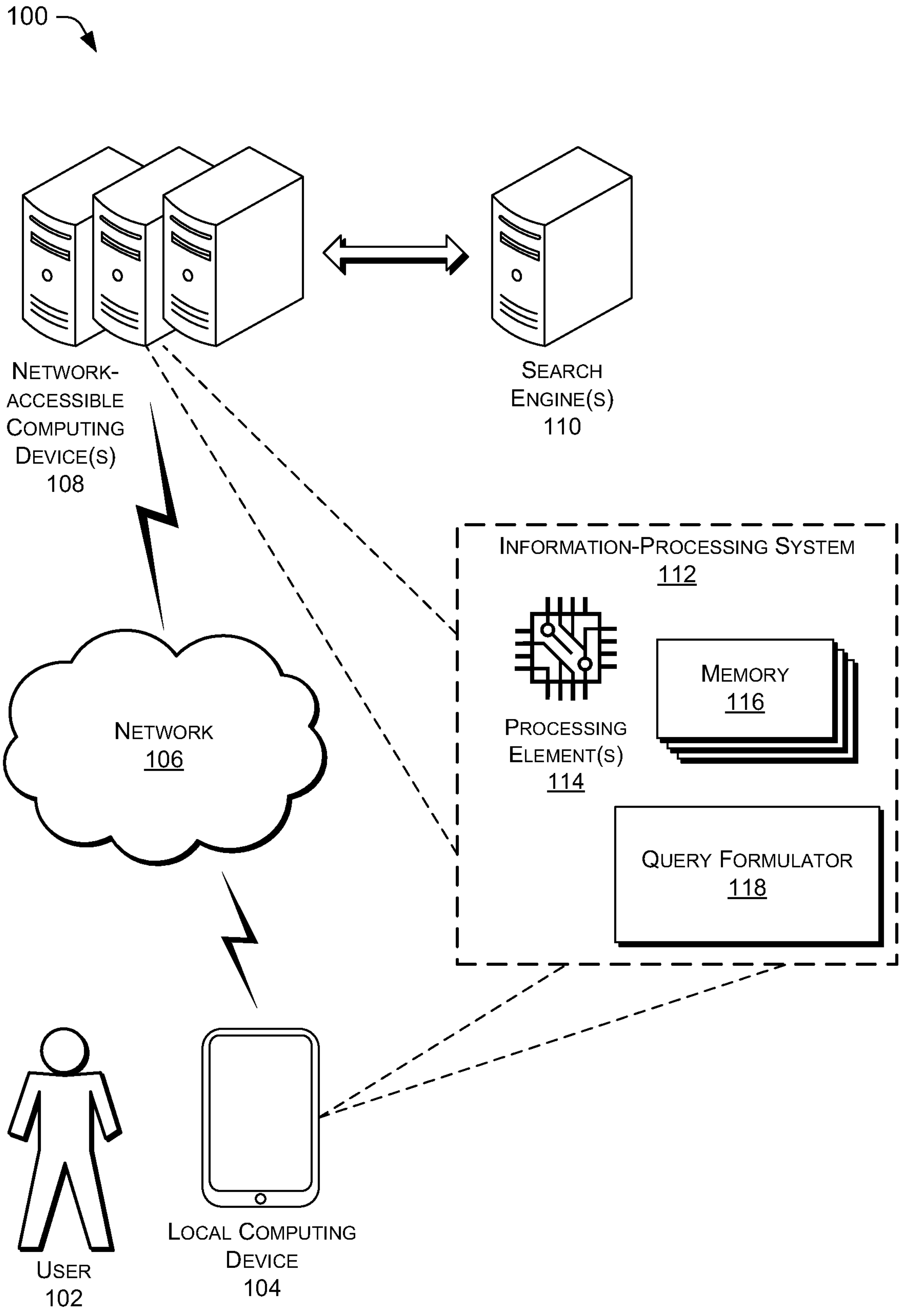

100
NETWORK-ACCESSIBLE COMPUTING DEVICE(S)
108
SEARCH ENGINE(S)
110
INFORMATION-PROCESSING SYSTEM
112
PROCESSING ELEMENT(S)
114
MEMORY
116
QUERY FORMULATOR
118
NETWORK
106
USER
102
LOCAL COMPUTING DEVICE
104

Fig. 1

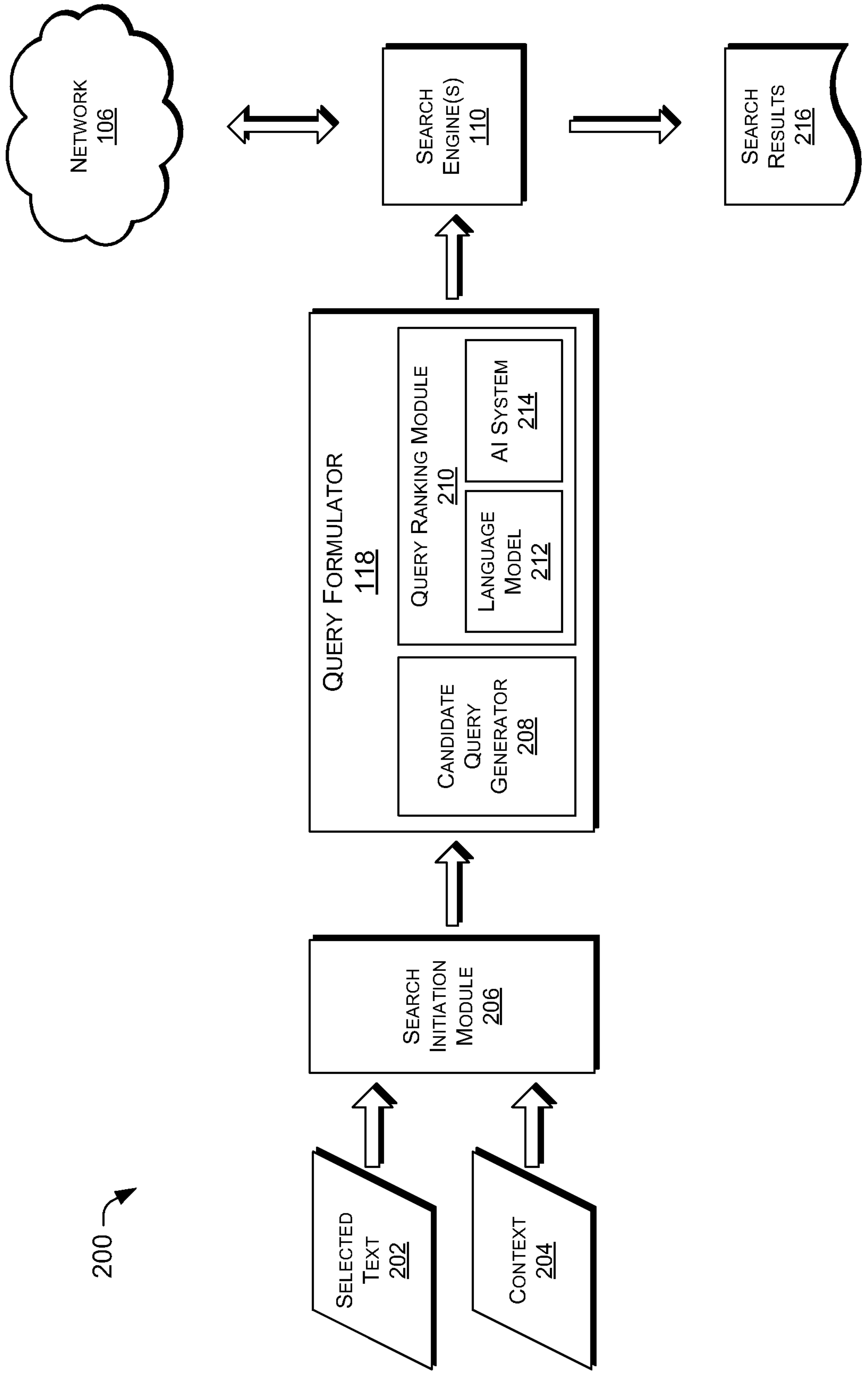

200
Selected Text 202
Context 204
Search Initiation Module 206
Query Formulator 118
Candidate Query Generator 208
Query Ranking Module 210
Language Model 212
AI System 214
Search Engine(s) 110
Network 106
Search Results 216

Fig. 2

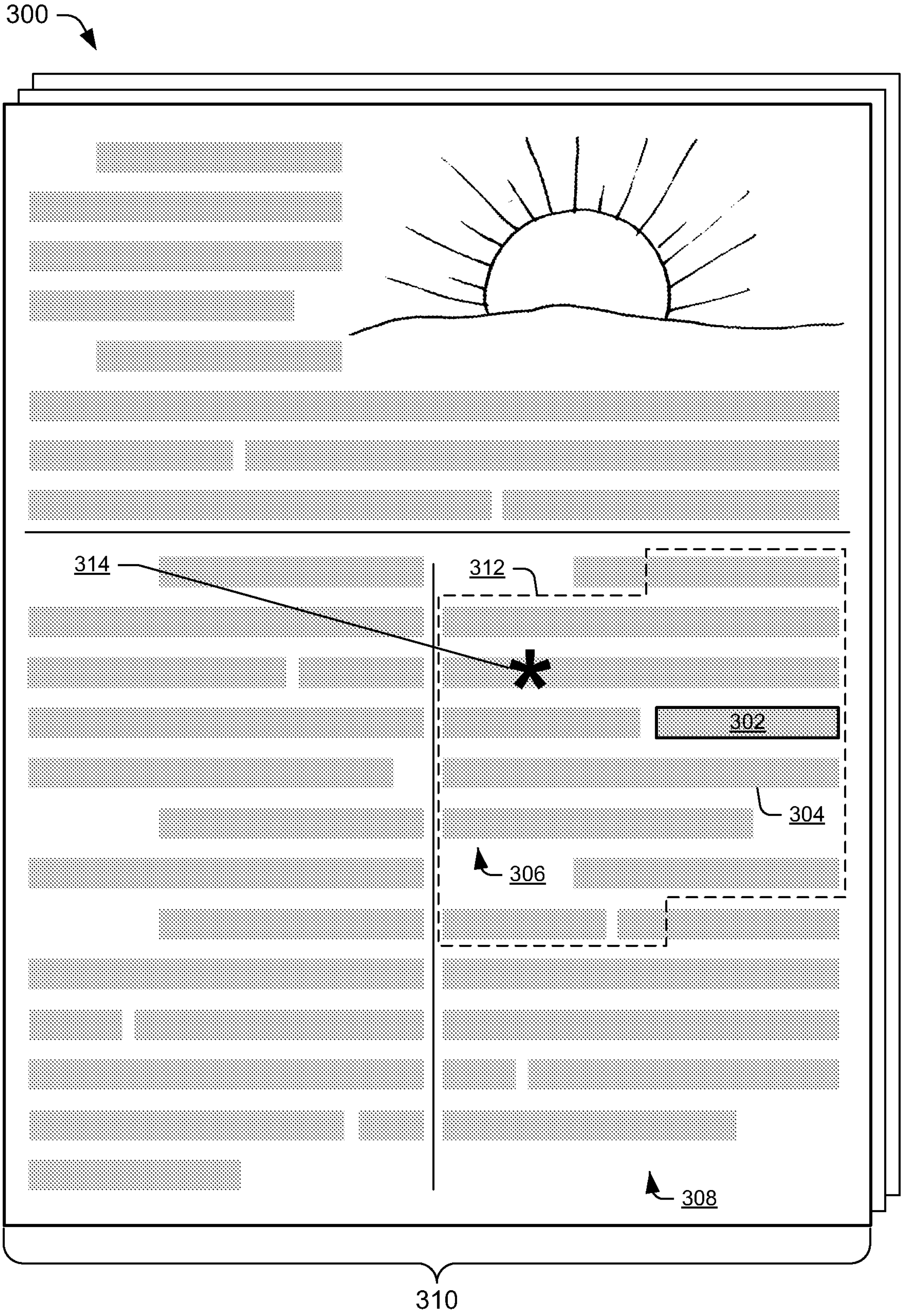


Fig. 3

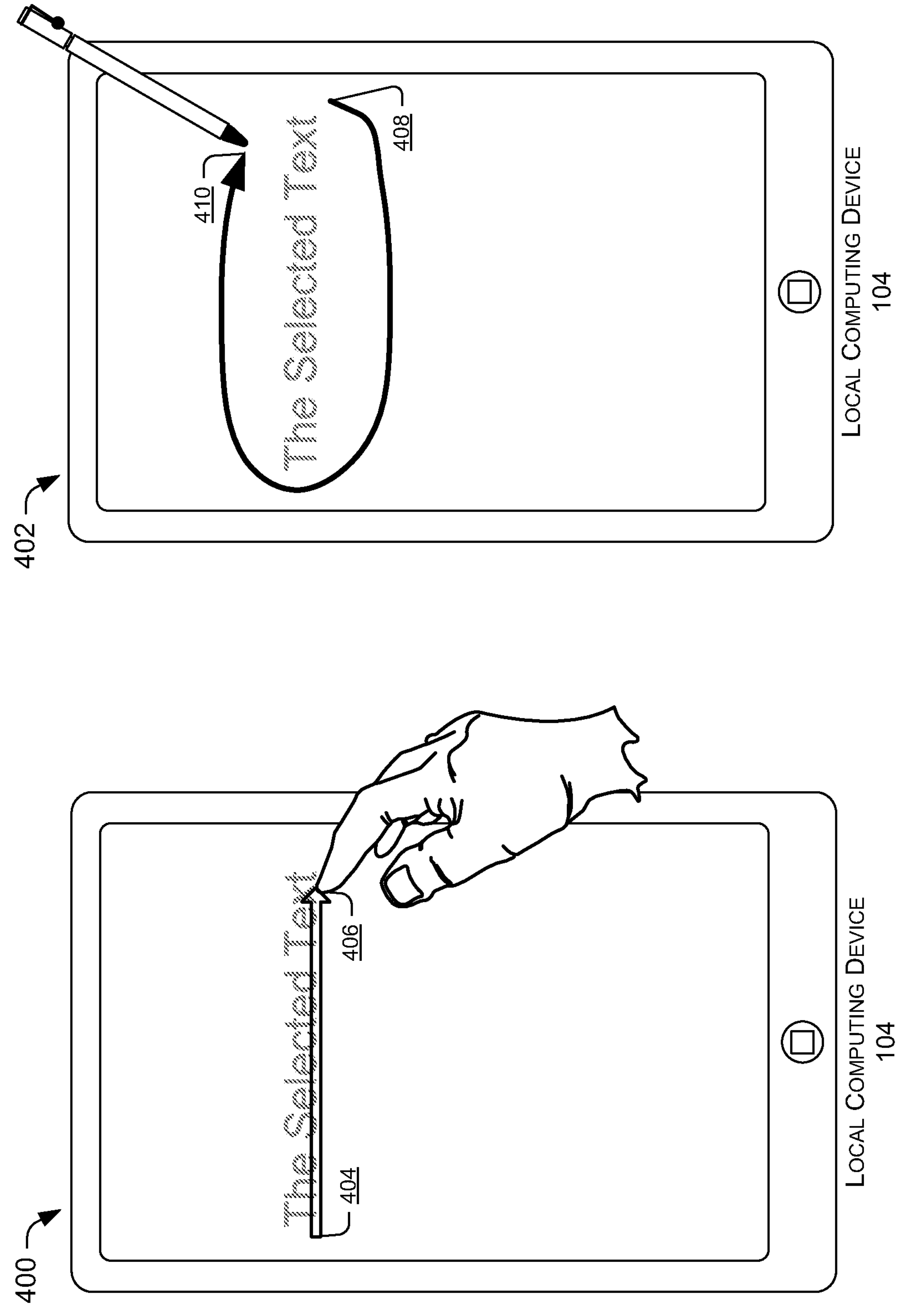

400
The Selected Text
404
406
Local Computing Device
104
402
The Selected Text
408
410
Local Computing Device
104

Fig. 4

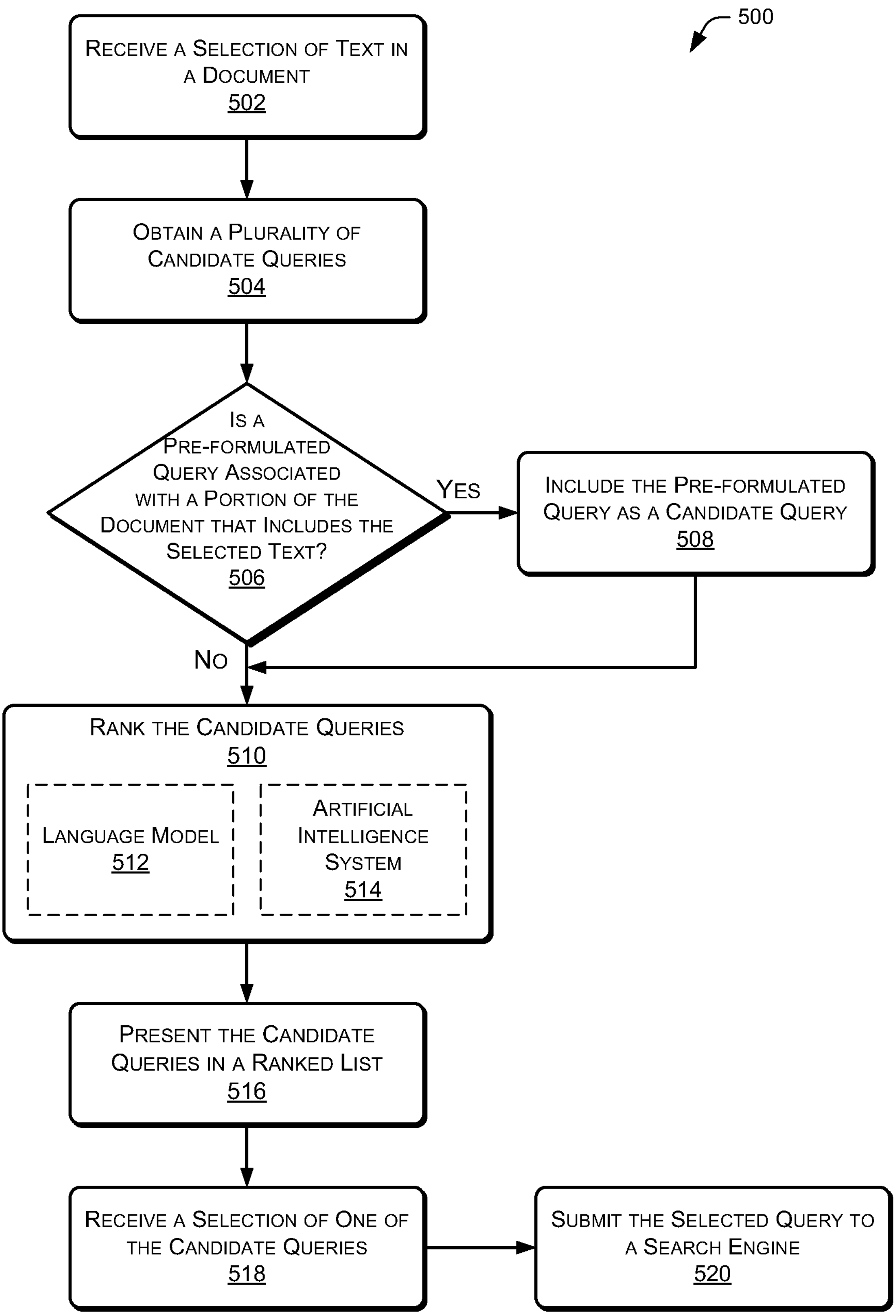
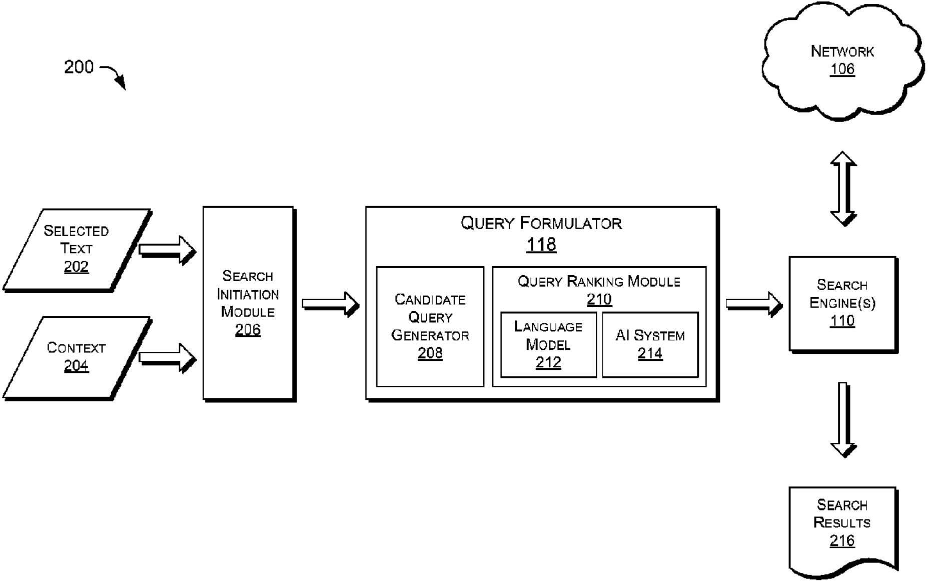

500
Receive a Selection of Text in a Document
502
Obtain a Plurality of Candidate Queries
504
Is a Pre-formulated Query Associated with a Portion of the Document that Includes the Selected Text?
506
Yes
Include the Pre-formulated Query as a Candidate Query
508
No
Rank the Candidate Queries
510
Language Model
512
Artificial Intelligence System
514
Present the Candidate Queries in a Ranked List
516
Receive a Selection of One of the Candidate Queries
518
Submit the Selected Query to a Search Engine
520

Fig. 5


200
Network
106
Selected Text
202
Context
204
Search Initiation Module
206
Query Formulator
118
Candidate Query Generator
208
Query Ranking Module
210
Language Model
212
AI System
214
Search Engine(s)
110
Search Results
216