

SYSTEM AND METHOD FOR FACILITATING SEARCH

TECHNICAL FIELD

[0001] Various embodiments relate to a system and a method for facilitating a search.

BACKGROUND

[0002] Due to development of information technology, information is created and retained in a database and is now accessible over Internet. A user may use a search engine to obtain desired information over the Internet. In order to provide more relevant information to the user, the search engine have adopted searching technology, for example Apache Lucene which is a digital document scoring technology based on frequency of a term being searched for in a digital document as well as how common the term is among digital documents. The search engine may use the digital document scoring technology, and then produce search results based on a relevance score of the digital documents.

[0003] However, the search engine may often fail to produce relevant search results in accordance with the user's level of interest. Especially, when a universal search box is used instead of separate search interface for each search scenario and/or when huge amount of potentially relevant digital documents are located by the search engine, it may be difficult for the search engine to produce relevant search results in accordance with the user's level of interest.

SUMMARY

[0004] According to various embodiments, a system for facilitating a search is provided. The system comprises an input device configured to receive an input relating to a query for the search from a user; and a processor configured to perform the search for the query to produce a set of search results from a plurality of digital documents, wherein each digital document includes a plurality of index fields. The processor is configured to predict a search intent from the query to obtain a prediction score for the search intent, boost each index field by calculating a first relevance score for the each index field using the prediction score and a boost parameter respectively assigned to the each index field, and aggregating a plurality of first relevance scores for the plurality of index fields to obtain a second relevance score for each digital document, and produce the set of search results from the plurality of digital documents based on a plurality of second relevance scores for the plurality of digital documents.

[0005] In some embodiments, the system further comprises a database storing log data including interaction information of the user for one or more previous search results, wherein the processor is further configured to use the log data to boost the each index field.

[0006] In some embodiments, the processor comprises a query understanding module including one or more machine learning models and configured to predict the search intent from the query.

[0007] In some embodiments, the query understanding module is configured to obtain the prediction score relating to a likelihood that the query relates to each predefined search intent.

[0008] In some embodiments, the processor is configured to use a boosting function with the prediction score as an input, to boost the each index field in the calculation of the first relevance score.

[0009] In some embodiments, the query understanding module is further configured to identify a likelihood that the query contains a predetermined entity, and the processor is further

configured to use the likelihood that the query contains the predetermined entity as the input of the boosting function.

[0010] In some embodiments, the boosting function includes a sigmoid function.

[0011] In some embodiments, the boosting function includes a function defined by a mathematical equation as follows:

$$f(x) = (d - c) \times \{0.5 \times \max(0, x - a) - 0.5 \times \max(0, x - b)\} + c$$

where “a” and “b” are thresholds, “c” is a boost parameter relating to a minimum boosted value, “d” is a boost parameter relating to a maximum boosted value, and “max(0, x)” is a rectifier function.

[0012] In some embodiments, the processor is further configured to use a Bayesian optimisation based on the log data collected for a predetermined time, to learn and/or tune the boost parameter for the boosting function.

[0013] According to various embodiments, a method of facilitating a search is provided. The method includes: receiving an input relating to a query for the search from a user to produce a set of search results from a plurality of digital documents, wherein each digital document includes a plurality of index fields; predicting a search intent from the query to obtain a prediction score for the search intent; boosting each index field by calculating a first relevance score for the each index field using the prediction score and a boost parameter respectively assigned to the each index field; aggregating a plurality of first relevance scores for the plurality of index fields to obtain a second relevance score for each digital document; and producing the set of search results from the plurality of digital documents based on a plurality of second relevance scores for the plurality of digital documents.

[0014] In some embodiments, the method further includes: storing log data including interaction information of the user for one or more previous search results; and using the log data to boost the each index field.

[0015] In some embodiments, predicting the search intent from the query is performed by one or more machine learning models.

[0016] In some embodiments, the prediction score relates to a likelihood that the query relates to each predefined search intent.

[0017] In some embodiments, the method further includes using a boosting function with the prediction score as an input, to boost the each index field in the calculation of the first relevance score.

[0018] In some embodiments, the method further includes: identifying a likelihood that the query contains a predetermined entity; and using the likelihood that the query contains the predetermined entity as the input of the boosting function.

[0019] In some embodiments, the boosting function includes a sigmoid function.

[0020] In some embodiments, the boosting function includes a function defined by a mathematical equation as follows:

$$f(x) = (d - c) \times \{0.5 \times \max(0, x - a) - 0.5 \times \max(0, x - b)\} + c$$

where “a” and “b” are thresholds, “c” is a boost parameter relating to a minimum boosted value, “d” is a boost parameter relating to a maximum boosted value, and “max(0, x)” is a rectifier function.

[0021] In some embodiments, the method further includes using a Bayesian optimisation based on the log data collected for a predetermined time, to learn and/or tune the boost parameter for the boosting function.

[0022] According to various embodiments, a data processing apparatus configured to perform the method of any one of the above embodiments is provided.

[0023] According to various embodiments, a computer program element including program instructions, which, when executed by one or more processors, cause the one or more processors to perform the method of any one of the above embodiments is provided.

[0024] According to various embodiments, a computer-readable medium comprising program instructions, which, when executed by one or more processors, cause the one or more processors to perform the method of any one of the above embodiments is provided. The computer-readable medium may include a non-transitory computer-readable medium.

BRIEF DESCRIPTION OF THE DRAWINGS

[0025] The invention will be better understood with reference to the detailed description when considered in conjunction with the non-limiting examples and the accompanying drawings, in which:

- FIG. 1 shows a block diagram for a system for facilitating a search according to various embodiments.
- FIG. 2 shows an exemplary flowchart for a method for facilitating a search according to various embodiments.
- FIG. 3 shows an exemplary flowchart for a method for facilitating a search according to various embodiments.
- FIG. 4 shows an exemplary flowchart for a method of dynamic query-time boosting to facilitate a search according to various embodiments.
- FIG. 5 shows exemplary graphs for dynamic boosting functions according to various embodiments.
- FIG. 6 shows exemplary graphs for Bayesian optimisation according to various embodiments.

DETAILED DESCRIPTION

[0026] The following detailed description refers to the accompanying drawings that show, by way of illustration, specific details and embodiments in which the disclosure may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the disclosure. Other embodiments may be utilized and structural, and logical changes may be made without departing from the scope of the disclosure. The various embodiments are not necessarily mutually exclusive, as some embodiments can be combined with one or more other embodiments to form new embodiments.

[0027] Embodiments described in the context of one of a system and a method are analogously valid for the other system and method. Similarly, embodiments described in the context of a system are analogously valid for a method, and vice-versa.

[0028] Features that are described in the context of an embodiment may correspondingly be applicable to the same or similar features in the other embodiments. Features that are described in the context of an embodiment may correspondingly be applicable to the other embodiments, even if not explicitly described in these other embodiments. Furthermore, additions and/or combinations and/or alternatives as described for a feature in the context of an embodiment may correspondingly be applicable to the same or similar feature in the other embodiments.

[0029] In the context of various embodiments, the articles “a”, “an” and “the” as used with regard to a feature or element include a reference to one or more of the features or elements.

[0030] As used herein, the term “and/or” includes any and all combinations of one or more of the associated listed items.

[0031] In the following, embodiments will be described in detail.

[0032] FIG. 1 shows a block diagram for a system 100 for facilitating a search according to various embodiments.

[0033] The system 100 may be a set of interacting elements. The elements may be, by way of example and not of limitation, one or more mechanical components, one or more electrical components, and/or one or more instructions, for example, encoded in a storage media.

[0034] As shown in FIG. 1, the system 100 may include an input device 110 and a processor 120. In some embodiments, the input device 110 and the processor 120 may be mounted on the same device. In some other embodiments, the input device 110 and the processor 120 may be mounted on different devices. The input device 110 and the processor 120 may be capable of data communication.

[0035] The input device 110 may receive an input relating to a query for the search from a user. The query may include string of words that the user may input. For example, the query may include the string of words that the user may type into a search box of a search engine. The query may be referred to as a search query or a search term.

[0036] In some embodiments, the input device 110 may be mounted in the client device (also referred to as a “user’s device”). In some other embodiments, the input device 110 may communicate with the client device via data communications. The client device may include, but not be limited to, a mobile phone, a tablet computer, a laptop computer, a desktop computer, a smart watch, smart glasses, a television, a navigation device, a PDA (Personal Digital Assistant) and a PMP (Portable Multimedia Player).

[0037] Although not shown, in some embodiments, the client device may include a user input device and/or a microphone to receive the input from the user.

[0038] The user input device may allow the user to input information, for example, the string of words as the query, manually, and generate raw data associated with the user’s input relating to the query. A controller (not shown) of the client device may then receive the raw data from the user input device, and process the raw data to generate input data. The controller may send the input data relating to the query to the input device 110. The user input device may include,

but not limited to, a keypad, a keyboard, a touchpad, a touch key, a touch screen, a computer mouse, a jog switch, a jog wheel, and a pointing device.

[0039] The microphone may receive an audio input from an external environment. For example, the microphone may receive the user's audio command, for example, the string of words as the query. The microphone may generate raw data associated with the audio input. The controller may then receive the raw data from the microphone and process the raw data to generate input data. The controller may send the input data relating to the query to the input device 110. In some embodiments, the client device may include a plurality of microphones.

[0040] The client device may send the input data relating to the query for the search to the input device 110, and the input device 110 may receive the input data relating to the query from the client device. For example, the client device and the input device 110 may be capable of data communication. The input device 110 may then send the received input data relating to the query to the processor 120.

[0041] Although not shown, the client device may further include a display. The display may display a search page on an application program. The user may input the query on the search page using the user input device and/or the microphone. For example, the user may move a cursor on a search field (also referred to as a "query field") of the search page, and type the query using the user input device. As another example, the user may touch an icon or press a hard key to active a function of receiving the audio input, and speak the query so that the client device receives the audio input for the query via the microphone. In some embodiments, the application program includes a program to order food for home delivery or pick-up, and the search page is suitable for searching merchant or dish to order food for home delivery or pick-up. It may be appreciated that the application program is not limited thereto.

[0042] The processor 120 may include a microprocessor, an analogue circuit, a digital circuit, a mixed-signal circuit, a logic circuit, an integrated circuit, a Central Processing Unit (CPU), a

Graphics Processing Unit (GPU), a Digital Signal Processor (DSP), a Field Programmable Gate Array (FPGA), an Application Specific Integrated Circuit (ASIC), etc., or any combination thereof. Any other kind of implementation of the respective functions, which will be described below in further detail, may also be understood as the processor 120.

[0043] The processor 120 may receive the input data relating to the query from the input device 110. The processor 120 may process the query for the search. For example, if the user types a query of “noodle” into the search field of the search page on the application program, the client device may transmit the input data including a text of “noodle” or some other representation or indication of “noodle.” The processor 120 may receive the input data and determine that “noodle” is the query that the user desires to search.

[0044] The processor 120 may perform the search for the query to produce a set of search results from a plurality of digital documents. Each digital document may include a plurality of index fields. For example, contents (also referred to as a “document”) of the index fields may be a data-interchange format (for example, JSON) with text fields instead of a whole piece of text. For example, the index fields may include, but not be limited to, a title and contents. As another example, in the search engine to order food for home delivery or pick-up, the digital documents may include merchant information with dish information that the merchant is serving, and the index fields may include, but not be limited to, a merchant name and a dish name.

[0045] The processor 120 may perform the search for relevant digital documents from the plurality of digital documents in a search database 140. The processor 120 may score index fields of a digital document (for example, a first digital document of the plurality of digital documents) based on the query in response to the input data relating to the query received from the input device 110 and then aggregate the scores (also referred to as a “first relevance scores”) of the index fields to obtain a relevance score (also referred to as a “second relevance score”)

of the first digital document. The processor 120 may score index fields of other digital document (for example, a second digital document of the plurality of digital documents) and aggregate first relevance scores for the index fields of the second digital document to obtain a second relevance score of the second digital document. The processor 120 may then produce the set of search results based on a plurality of second relevance scores of the plurality of digital documents. The processor 120 may receive digital document data for the set of search results from the search database. The digital document data may include, but not be limited to, a URL (Universal Resource Locator) which provides a link to a digital document, a web page, or a location from which a digital document or a web page can be retrieved or accessed by the user. The display of the client device may then display the set of search results including the digital document data.

[0046] In some embodiments, the processor 120 may include a query understanding module 121. The query understanding module 121 may include one or more machine learning models. In some embodiments, the query understanding module 121 may be or include a collection of machine learning models such as an entity recognition module, a query correction and query re-write module, and an intent recognition module. For example, the entity recognition module may be a deep neural network model, in which an input is a string and an output is a list of entity name and/or score. The query correction and query re-write module may be a rule-based logic(s), in which an input is a string and an output is also a string. The intent recognition module may be a deep neural network model, in which an input is a string and an output is a list of prediction scores. As an example, for each search intent, one (1) prediction score may be output. The query correction and query re-write module may process the input query first and output the modified query. The intent recognition module and the entity recognition module may then process the modified query, make predictions, and obtain the prediction score. The prediction score may be used in the boosting function (as will be described below).

The machine learning models may work together to process the query that the user issues and provide insights that may be used by the processor 120 to produce the set of search results.

[0047] The query understanding module 121 may receive the input data relating to the query and predict a search intent from the query. For example, if the query is “noodle”, the query understanding module 121 may predict whether the query of “noodle” is intended to a merchant name or a dish name. This prediction may be performed based on a machine learning technology. For example, the input data may be fed into the deep neural network model which is pre-trained with a search intent dataset. The deep neural network model may obtain the one (1) prediction score for each search intent (merchant, dish, etc.) indicating how likely the query being with the intent. In some embodiments, the search intent may correspond to each index field which may be based on a type of the search engine. For example, in the search engine to order food for home delivery or pick-up, predefined search intents may include a merchant name or a dish name.

[0048] In some embodiments, the processor 120 may boost each index field based on the predicted search intent using a boost parameter respectively assigned to the each index field. For example, the boost parameter may be respectively assigned to the each index field based on an attribute of the each index field. As an example, the processor 120 may determine the boost parameter based on the attribute of the each index field and/or log data (as will be described below). In some embodiments, the each index field may be boosted with the boost parameter respectively assigned to the each index field. In some embodiments, the plurality of boost parameters for the plurality of index fields may be different from each other. As an example, a first index field (for example, a merchant name field) may have the boost parameter of 2.0 and the second index field (for example, a dish name field) may have the boost parameter of 1.0 (as a default boost parameter). As such, the first index field may have twice the weight as those of the second index field. It may be appreciated that, in some other embodiments, some

of the plurality of boost parameters may be the same. Advantageously, with the use of the different boost parameters based on the attribute of the index field (for example, the merchant name field or the dish name field), the same digital document may be up-ranked or down-ranked among other search results.

[0049] In some embodiments, the processor 120 may calculate a relevance score for the each index field (hereinafter, referred to as the “first relevance score”) based on the prediction score and the boost parameter. The processor 120 may then aggregate the plurality of first relevance scores for the plurality of index fields into one (1) score (hereinafter, referred to as the “second relevance score”) to represent a relevancy of the digital document. In this manner, the processor 120 may calculate each second relevance score of each digital document. The processor 120 may then produce the set of search results from the plurality of digital documents based on a plurality of second relevance score for the plurality of digital documents.

[0050] In some other embodiments, the system 100 may further include a database 130 storing log data of the user. The log data may include interaction information of the user for one or more previous search results. The interaction information may include queries that the user issued to the search engine, a set of search results which was provided to the user in response to each query, and/or digital document data selected from the set of search results. For example, the interaction information may be identified based on time period and saved in the database 130 accordingly. In some embodiments, the boost parameter may further be determined by the log data, in addition to the attribute of the each index field. The processor 120 may boost each index field based on the prediction score and the boost parameter which is determined by the log data, and calculate the second relevance score of each digital document using the boosted index field. The processor 120 may then produce the set of search results from the plurality of digital documents based on the calculated second relevance score.

[0051] In some embodiments, the query understanding module 121 may obtain a value relating to a likelihood that the query relates to each predefined search intent. For example, the deep neural network model may be used to process the input data and obtain one (1) prediction score for each search intent. The deep neural network model may be trained with the search intent dataset. For example, the value relating to the likelihood may be a number between “0” and “1”. If there are two predefined search intent (hereinafter, referred to as a “first search intent” and a “second search intent”) for the search engine, the query understanding module 121 may obtain a value relating to a likelihood that the query relates to the first search intent and a value relating to a likelihood that the query relates to the second search intent. The value relating to a likelihood that the query relates to each predefined search intent is referred to as the “prediction score”.

[0052] In some embodiments, the processor 120 may use a boosting function with the prediction score as an input in order to boost the index field. The boosting function may emphasize the likelihood that the query relates to each predefined search intent during a retrieval of the digital document data and calculation of the second relevance score, in order to produce more accurate set of search results in which the user intended.

[0053] In some embodiments, a graph of the boosting function may include an inverted S-curve which can cover a desired output range with one input range that is sensitive and two input ranges that are saturated. The processor 120 may obtain an output value (hereinafter referred to as a “boosting value” or the “first relevance score”) of the boosting function based on the prediction score as the input. If there are two prediction scores corresponding to two predefined search intents respectively, the processor 120 may obtain two first relevance scores for each index field using the boosting function.

[0054] In some embodiments, the query understanding module 121 may identify a likelihood that the query contains a predetermined entity. For example, when a user searches a term of

“mcd hollywood”, the query understanding module 121 may identify a term of “hollywood” as a point of interest (POI) and extract the term of “hollywood” as the query. In some embodiments, the query understanding module 121 may use the identified likelihood that the query contains the predetermined entity as an additional input of the boosting function. In some embodiments, the query understanding module 121 may obtain an entity score relating to the likelihood that the query contains the predetermined entity. The processor 120 may use the entity score as the additional input of the boosting function.

[0055] In some embodiments, the boost parameter for the boosting function may be determined by the processor 120. The processor 120 may further use a Bayesian optimisation based on the log data collected for a predetermined time, in order to learn and/or tune the boost parameter for the boosting function. The processor 120 may use the boosting function with the determined boost parameter to obtain the first relevance score for the each index field.

[0056] After the plurality of first relevance scores for the plurality of index fields are obtained, the processor 120 may aggregate the plurality of first relevance scores for the plurality of index fields to obtain the second relevance score for each digital document. The processor 120 may then produce the set of search results from the plurality of digital documents based on the plurality of second relevance scores for the plurality of digital documents. The client device may receive the set of search results, and display of the client device may display the set of search results. According to various embodiments, the set of search results may be tailored for each query for search, depending on the user’s search intention.

[0057] FIG. 2 shows an exemplary flowchart for a method 200 for facilitating a search according to various embodiments. According to various embodiments, the method 200 of facilitating the search may be provided.

[0058] In some embodiments, the method 200 may include a step 201 of receiving an input relating to a query for the search from a user to produce a set of search results from a plurality of digital documents. Each digital document may include a plurality of index fields.

[0059] In some embodiments, the method 200 may include a step 202 of predicting a search intent from the query to obtain a prediction score for the search intent. For example, the prediction score may relate to a likelihood that the query relates to each predefined search intent.

[0060] In some embodiments, the method 200 may include a step 203 of boosting each index field by calculating a first relevance score for the each index field using the prediction score and a boost parameter respectively assigned to the each index field. In some embodiments, a boosting function is used with the prediction score as an input, and the first relevance score for each index field is output.

[0061] In some embodiments, the method 200 may include a step 205 of aggregating a plurality of first relevance scores for the plurality of index fields to obtain a second relevance score for each digital document.

[0062] In some embodiments, the method 200 may include a step 205 of producing the set of search results from the plurality of digital documents based on a plurality of second relevance scores for the plurality of digital documents.

[0063] FIG. 3 shows an exemplary flowchart 300 for a method for facilitating a search according to various embodiments. According to various embodiments, the method 300 of facilitating the search may be provided. As shown in FIG. 3, blocks 301, 303, 305, 307, 309, 311 and 313 may represent intermediary results, and steps 302, 304, 306, 308, 310 and 312 may represent actions and/or process.

[0064] In some embodiments, as shown in a block 301, an input relating to a query for the search from a user may be fed.

[0065] In some embodiments, the method 300 may include a step 302 of understanding the query. A search intent from the query may be predicted.

[0066] In some embodiments, as shown in a block 303, a prediction score relating to a likelihood that the query relates to each predicted search intent may be output.

[0067] In some embodiments, the method 300 may include a step 304 of boosting each index field based on the prediction score and data boost parameter. The boost parameter may be respectively assigned to the each index field. A log data may include interaction information of the user for one or more previous search results. For example, the boosting function may be used, with the prediction score as an input and the boost parameter determined based on the log data and/or an attribute of the each index field. In some embodiments, the step 304 may further include obtaining a first relevance score for the each index field by boosting the each index field, and aggregating a plurality of first relevance scores for the plurality of index fields to obtain a second relevance score for each digital document.

[0068] As shown in FIG. 3, in some embodiments, the steps 302 and 304 and the block 303 may be grouped as query-time dynamic boosting steps (a).

[0069] In some embodiments, as shown in a block 305, a second relevance score for each digital document may be output.

[0070] In some embodiments, the method 300 may include a step 306 of inputting the query and the second relevance score for the each digital document to a search engine for the search.

[0071] In some embodiments, as shown in a block 307, a set of search results may be output.

[0072] In some embodiments, the method 300 may include a step 308 of monitoring log data relating to the user's browsing behaviour. For example, the log data may relate to the user's browsing behaviour on the produced set of the search results.

[0073] In some embodiments, as shown in a block 309, the log data may be output. In some embodiments, the log data relating to the user's browsing behaviour may be stored as

interaction information of the user. For example, the log data including the query that the user issued to the search engine in the block 301, the set of search results output in the block 307, and/or digital document data selected from the set of search results by the user may be stored. It may be appreciated that time data for the search may additionally be stored in relation to the log data. In some embodiments, the log data may be collected from both frontend application(s) and backend application programming interface (API) servers. The log data may be stored in the form of a compressed text file on an object storage (for example, AWS S3) or stored as Hive tables.

[0074] In some embodiments, the method 300 may include a step 310 of calculating business metrics in relation to the log data. For example, the metrics may include, but not be limited to, gross merchandise value (GMV), revenue and conversion rate.

[0075] In some embodiments, as shown in a block 311, an objective value for the calculated metrics may be output. The object value may be a number obtained after plugging in the GMV, the revenue and the conversion rate. For example, a mathematical equation for calculating the object value is as follows:

$$\text{Object Value} = x \times \text{GMV} + y \times \text{revenue} + z \times \text{conversion rate}$$

Where x , y , and z are scalar values which may be decided based on a current business strategy.

[0076] In some embodiments, the method 300 may include a step 312 of using a Bayesian optimisation. The Bayesian optimisation may be used based on the log data collected for a predetermined time, in order to learn and/or tune the boost parameter for the boosting function. Advantageously, an optimal parameter for the boosting function may be obtained by using the Bayesian optimisation.

[0077] As shown in FIG. 3, in some embodiments, the steps 310 and 312 and the block 311 may be grouped as iterative parameter tuning steps (b).

[0078] In some embodiments, as shown in a block 313, the determined boost parameter may be output. In the step 304 for the next search, the boost parameter output in the block 313 may be used. In the block 307 for the next search after operating the step 306, a set of search results for the next search may be produced based on the predicted search intent (i.e. the prediction score) and the determined boost parameter. Although not shown, in some embodiments, the set of search results may then return to a frontend to be shown to users.

[0079] FIG. 4 shows an exemplary flowchart for a method of dynamic query-time boosting to facilitate a search according to various embodiments. According to various embodiments, the method 400 of dynamic query-time boosting may be provided. The method 400 may include some of the same steps as the method 300. As shown in FIG. 4, blocks 401, 403 and 405 may represent intermediary results, and steps 402, 404 and 406 may represent actions and/or process.

[0080] In some embodiments, as shown in a block 401, an input relating to a query for the search from a user may be fed. For example, if the user types a query of “noodle” into a search field of a search page on an application program for ordering food, input data including a text of “noodle” or some other representation or indication of “noodle” may be fed.

[0081] In some embodiments, the method 400 may include a step 402 of understanding the query. A search intent from the query may be predicted.

[0082] In some embodiments, as shown in a block 403, a prediction score relating to a likelihood that the query relates to each predicted search intent may be output. For example, whether the query of “noodle” is intended to the merchant name or the dish name is predicted. A prediction score relating to a likelihood that the query relates to the merchant name and a prediction score relating to a likelihood that the query relates to the dish name may be output, for example, as follows:

- Prediction Score for Merchant Intent: 0.6

- Prediction Score for Dish Intent: 0.9

[0083] In some embodiments, the method 400 may include a step 404 of boosting each index field based on the prediction score. For example, a boosting function may be used, with the prediction score for the merchant intent and the prediction score for the dish intent as an input.

[0084] In some embodiments, as shown in a block 405, a first relevance score for each index field may be output. An example of the first relevance score is as follows:

- First Relevance Score for Merchant Name: 3.5
- First Relevance Score for Dish Names: 10.0

[0085] In some embodiments, the method 400 may include a step 406 of inputting the query of the step 401 and the first relevance score for each index field of the step 405 to a search index for the search. An exemplary search index table 406a may include a plurality of digital documents each including two index fields, for example, the merchant name and the dish names.

[0086] FIG. 5 shows exemplary graphs for dynamic boosting functions according to various embodiments. FIG. 5(a) shows a graph of an inverted S-curve and FIG. 5(b) shows a graph of a piecewise linear S-curve. As shown in FIG. 5, the x-axis of the graph may represent the predicted search intent, for example, the prediction score, and the y-axis of the graph may represent a boosting value (i.e. the first relevance score) for each index field.

[0087] In some embodiments, the processor 120 may make each index field boost a function of the predicted search intent, for example, the prediction score, from the query understanding module 121, instead of a static value.

[0088] As an example, the function may be bounded and monotonic such as the inverted S-curve as shown in FIG. 5(a). The inverted S-curve may cover a desired output range with one input range that is sensitive and two input ranges that are saturated. An exemplary boosting function for the inverted S-curve may be a sigmoid function.

[0089] As shown in FIG. 5(b), the piecewise linear S-curve may be described with the boost parameter, with the help of a rectifier function “ $\max(0, x)$ ”. For example, the rectifier function may provide one sensitive range and one saturated range. Two such functions may be used to synthesize an S-curve function. An exemplary boosting function for the piecewise linear S-curve defined by a mathematical equation is as follows:

$$f(x) = (d - c) \times \{0.5 \times \max(0, x - a) - 0.5 \times \max(0, x - b)\} + c$$

where “a” and “b” are thresholds, “c” is a boost parameter relating to a minimum boosted value, and “d” is a boost parameter relating to a maximum boosted value, and “ $\max(0, x)$ ” is the rectifier function. Here, “x” value outside of “[a, b]” leads to the minimum constant boosted value and the maximum constant boosted value. In some embodiments, the exemplary boosting function may be extended to smoother functions by replacing the rectifier function with a function such as a SoftPlus function which is a smooth approximation to the ReLU function.

[0090] FIG. 6 shows exemplary graphs for Bayesian optimisation according to various embodiments. FIG. 6(a) shows a graph for the Bayesian optimisation for a certain period of time (for example, $t = 3$), and FIG. 6(b) shows a graph for the Bayesian optimisation for another certain period of time (for example, $t = 4$). As shown in FIG. 6, the x-axis of the graph may represent the boost parameter for the boosting function, and the y-axis of the graph may represent a business metric.

[0091] As a relationship between the boost parameter for the boosting function and a result business metrics is unknown, in order to find an optimal parameter for the boosting function, the Bayesian optimisation may be used to efficiently explore potential configurations while optimising the business metrics. For example, a conversion rate over a period of time may be taken as a proxy of the user’s satisfactions with the previous search results, in order to optimise the boost parameter for the boosting function. As an example, the conversion rate may be the number of checkouts and/or the number of views.

[0092] In some embodiments, for each period of time, a data point of the boost parameter for the boosting function and the resulting business performance may be collected. For example, the data point may be a data point for the Bayesian optimisation in a pair of an input and an output. As an example, if the Bayesian optimisation is to optimise $y = f(x)$, the data point is (x, y) . Given a set of observed data points, the Bayesian optimisation may suggest the next boost parameter for the boosting function to experiment.

[0093] In some embodiments, an iteration speed may depend on capability of the metric aggregation pipeline and stability of the metrics being monitored. In some embodiments, the boost parameter for the boosting function may be optimised in real time.

[0094] In some embodiments, if multiple business metrics are being considered simultaneously, such as the conversion rate and revenue, the Bayesian optimisation may be made to optimise for the weighted sum of the multiple business metrics.

[0095] As described above, according to various embodiments, the index field may be boosted in an adaptive and dynamic manner, based on query-time signals, for example, the predicted search intent, from the query understanding module 121 and a real-time feedback, for example, the user's log data for the previous search queries.

[0096] According to various embodiments, a parameterised boosting function may be used to translate the predicted search intent into the boosting value (i.e. the first relevance score) for each index field.

[0097] According to various embodiments, the Bayesian optimisation may be used to automatically tune the boosting function with the real-time feedback, for example, the user's log data for the previous search queries.

[0098] While the disclosure has been particularly shown and described with reference to specific embodiments, it should be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the

invention as defined by the appended claims. The scope of the invention is thus indicated by the appended claims and all changes which come within the meaning and range of equivalency of the claims are therefore intended to be embraced.

CLAIM

1. A system for facilitating a search comprising:

an input device configured to receive an input relating to a query for the search from a user; and

a processor configured to perform the search for the query to produce a set of search results from a plurality of digital documents, wherein each digital document includes a plurality of index fields,

wherein the processor is configured to predict a search intent from the query to obtain a prediction score for the search intent, boost each index field by calculating a first relevance score for the each index field using the prediction score and a boost parameter respectively assigned to the each index field, and aggregating a plurality of first relevance scores for the plurality of index fields to obtain a second relevance score for each digital document, and produce the set of search results from the plurality of digital documents based on a plurality of second relevance scores for the plurality of digital documents.
2. The system according to claim 1 further comprising a database storing log data including interaction information of the user for one or more previous search results,

wherein the processor is further configured to use the log data to boost the each index field.
3. The system according to claim 2, wherein the processor comprises a query understanding module including one or more machine learning models and configured to predict the search intent from the query.

4. The system according to claim 3, wherein the query understanding module is configured to obtain the prediction score relating to a likelihood that the query relates to each predefined search intent.

5. The system according to claim 4, wherein the processor is configured to use a boosting function with the prediction score as an input, to boost the each index field in the calculation of the first relevance score.

6. The system according to claim 5, wherein the query understanding module is further configured to identify a likelihood that the query contains a predetermined entity, and the processor is further configured to use the likelihood that the query contains the predetermined entity as the input of the boosting function.

7. The system according to claim 5 or claim 6, wherein the boosting function includes a sigmoid function.

8. The system according to claim 5 or claim 6, wherein the boosting function includes a function as follows:

$$f(x) = (d - c) \times \{0.5 \times \max(0, x - a) - 0.5 \times \max(0, x - b)\} + c$$

where “a” and “b” are thresholds, “c” is a boost parameter relating to a minimum boosted value, “d” is a boost parameter relating to a maximum boosted value, and “max(0, x)” is a rectifier function.

9. The system according to any one of claims 5 to 8, wherein the processor is further configured to use a Bayesian optimisation based on the log data collected for a predetermined time, to learn and/or tune the boost parameter for the boosting function.

10. A method of facilitating a search, the method comprising:

receiving an input relating to a query for the search from a user to produce a set of search results from a plurality of digital documents, wherein each digital document includes a plurality of index fields;

predicting a search intent from the query to obtain a prediction score for the search intent;

boosting each index field by calculating a first relevance score for the each index field using the prediction score and a boost parameter respectively assigned to the each index field;

aggregating a plurality of first relevance scores for the plurality of index fields to obtain a second relevance score for each digital document; and

producing the set of search results from the plurality of digital documents based on a plurality of second relevance scores for the plurality of digital documents.

11. The method according to claim 10 further comprising:

storing log data including interaction information of the user for one or more previous search results; and

using the log data to boost the each index field.

12. The method according to claim 11, wherein predicting the search intent from the query is performed by one or more machine learning models.

13. The method according to claim 12, wherein the prediction score relates to a likelihood that the query relates to each predefined search intent.

14. The method according to claim 13 further comprising using a boosting function with the prediction score as an input, to boost the each index field in the calculation of the first relevance score.

15. The method according to claim 14 further comprising:
identifying a likelihood that the query contains a predetermined entity; and
using the likelihood that the query contains the predetermined entity as the input of the boosting function.

16. The method according to claim 14 or claim 15, wherein the boosting function includes a sigmoid function.

17. The method according to claim 14 or claim 15, wherein the boosting function includes a function defined by a mathematical equation as follows:

$$f(x) = (d - c) \times \{0.5 \times \max(0, x - a) - 0.5 \times \max(0, x - b)\} + c$$

where “a” and “b” are thresholds, “c” is a boost parameter relating to a minimum boosted value, “d” is a boost parameter relating to a maximum boosted value, and “max(0, x)” is a rectifier function.

18. The method according to any one of claims 14 to 17 further comprising using a Bayesian optimisation based on the log data collected for a predetermined time, to learn and/or tune the boost parameter for the boosting function.

19. A data processing apparatus configured to perform the method of any one of claims 10 to 18.

20. A computer program element comprising program instructions, which, when executed by one or more processors, cause the one or more processors to perform the method of any one of claims 10 to 18.