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 each search intent of a plurality of search intents from the query to obtain a prediction score for the each search intent, wherein each of the plurality of search intents corresponds to each of the plurality of index fields,

boost each index field of the plurality of index fields, by calculating a first relevance score for the each index field using the prediction score for the each search intent and a boost parameter respectively determined based on an attribute of the each index field and assigned to the each index field,

aggregate 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 configured to determine the boost parameter further based on the log data.

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 each 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 the each 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 each search intent of a plurality of search intents from the query to obtain a prediction score for the each search intent, wherein each of the plurality of search intents corresponds to each of the plurality of index fields;

boosting each index field of the plurality of index fields, by calculating a first relevance score for the each index field using the prediction score for the each search intent and a boost parameter respectively determined based on an attribute of the each index field and 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

determining the boost parameter further based on the log data.

12. The method according to claim 11, wherein predicting the each 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 the each 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.