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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
15 February 2024 (15.02.2024)

WIPO | PCT

(10) International Publication Number
WO 2024/035518 A1

(51) International Patent Classification:
G06N 5/045 (2023.01) *G06N 20/00* (2019.01)
G06N 5/022 (2023.01) *G06N 3/08* (2023.01)

(21) International Application Number:
PCT/US2023/027745

(22) International Filing Date:
14 July 2023 (14.07.2023)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
202210966768.9 12 August 2022 (12.08.2022) CN

(71) Applicant: **MICROSOFT TECHNOLOGY LICENSING, LLC** [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(72) Inventors: **PANG, Bochen**; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). **ZHANG, Qi**; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). **DENG, Weiwei**; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). **SUN, Hao**; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). **XIE, Xing**; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). **LI, Chaozhuo**; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). **HAN, Weihao**; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). **LIU, Yuming**; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Redmond, Washington 98052-6399

(54) **Title:** RELEVANCE PREDICTION BASED ON HETEROGENEOUS GRAPH LEARNING

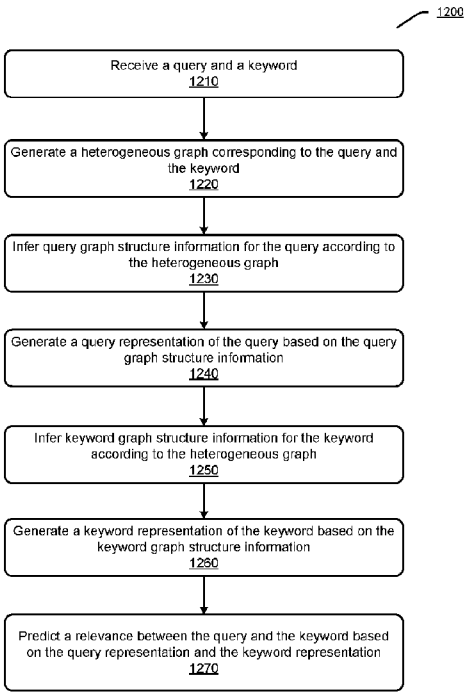

FIG 12

(57) **Abstract:** The present disclosure proposes a method, apparatus and computer program products for relevance prediction based on heterogeneous graph learning. A query and a keyword may be received. A heterogeneous graph corresponding to the query and the keyword may be generated. Query graph structure information for the query may be inferred according to the heterogeneous graph. A query representation of the query may be generated based on the query graph structure information. Keyword graph structure information for the keyword may be inferred according to the heterogeneous graph. A keyword representation of the keyword may be generated based on the keyword graph structure information. A relevance between the query and the keyword may be predicted based on the query representation and the keyword representation.

WO 2024/035518 A1

[Continued on next page]

(US). **LI, Mingzheng**; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). **ZENG, Zhengxin**; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). **WANG, Hui**; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). **LIAN, Jianxun**; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). **WANG, Yujing**; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). **ZHAO, Jianan**; MICROSOFT TECHNOLOGY LICENSING, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US).

(74) **Agent: CHATTERJEE, Aaron C.** et al.; MICROSOFT TECHNOLOGY LICENSING, LLC, 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, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, 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, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

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

RELEVANCE PREDICTION BASED ON HETEROGENEOUS GRAPH LEARNING

BACKGROUND

With the development of network technology and the growth of network information, recommendation systems are playing an increasingly important role in many online services. Based on different recommended content, there are different recommendation systems, e.g., product recommendation system, news recommendation system, video recommendation system, book recommendation system, etc. These recommendation systems may usually perform personalized recommendations for users. For example, a recommendation system usually predicts content that a user is interested in based on a query from the user and recommends the content to the user.

SUMMARY

This Summary is provided to introduce a selection of concepts that are further described below in the Detailed Description. It 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. Embodiments of the present disclosure propose a method, apparatus and computer program products for relevance prediction based on heterogeneous graph learning. A query and a keyword may be received. A heterogeneous graph corresponding to the query and the keyword may be generated. Query graph structure information for the query may be inferred according to the heterogeneous graph. A query representation of the query may be generated based on the query graph structure information. Keyword graph structure information for the keyword may be inferred according to the heterogeneous graph. A keyword representation of the keyword may be generated based on the keyword graph structure information. A relevance between the query and the keyword may be predicted based on the query representation and the keyword representation. It should be noted that the above one or more aspects comprise the features hereinafter fully described and particularly pointed out in the claims. The following description and the drawings set forth in detail certain illustrative features of the one or more aspects. These features are only indicative of the various ways in which the principles of various aspects may be employed, and this disclosure is intended to include all such aspects and their equivalents.

BRIEF DESCRIPTION OF THE DRAWINGS

The disclosed aspects will hereinafter be described in connection with the appended drawings that are provided to illustrate and not to limit the disclosed aspects.

FIG.1 illustrates an exemplary heterogeneous graph according to an embodiment of the present disclosure.

FIG.2 illustrates an exemplary process for generating a heterogeneous graph according to an

embodiment of the present disclosure.

FIG.3 illustrates an exemplary process for constructing an original heterogeneous graph according to an embodiment of the present disclosure.

FIG.4 illustrates an exemplary process for generating an augmented heterogeneous graph according to an embodiment of the present disclosure.

FIG.5 illustrates an exemplary process for pruning an augmented heterogeneous graph according to an embodiment of the present disclosure.

FIG.6 illustrates an exemplary process for relevance prediction based on heterogeneous graph learning according to an embodiment of the present disclosure.

FIG.7 illustrates an exemplary process for inferring query graph structure information for a query according to an embodiment of the present disclosure.

FIG.8 illustrates an exemplary process for generating a query representation of a query according to an embodiment of the present disclosure.

FIG.9 illustrates an exemplary process for performing a graph encoding operation according to an embodiment of the present disclosure.

FIG.10 illustrates an exemplary process for generating a query representation based on an embedding sequence according to an embodiment of the present disclosure.

FIG.11 illustrates an exemplary process for performing relevance prediction through a lightweight relevance prediction model according to an embodiment of the present disclosure.

FIG.12 is a flowchart of an exemplary method for relevance prediction based on heterogeneous graph learning according to an embodiment of the present disclosure.

FIG.13 illustrates an exemplary apparatus for relevance prediction based on heterogeneous graph learning according to an embodiment of the present disclosure.

FIG.14 illustrates an exemplary apparatus for relevance prediction based on heterogeneous graph learning according to an embodiment of the present disclosure.

DETAILED DESCRIPTION

The present disclosure will now be discussed with reference to several example implementations. It is to be understood that these implementations are discussed only for enabling those skilled in the art to better understand and thus implement the embodiments of the present disclosure, rather than suggesting any limitations on the scope of the present disclosure.

A relevance between a query from a user and a keyword for a candidate content item may be predicted through a machine learning model, and a content item to be recommended to the user may be determined based on the predicted relevance. Herein, a content item refers to an individual item with specific content. For example, a piece of news, a movie, a video, a product, etc., may be referred to as a content item. A query may be a search query from a user, which may reflect

user intent. A keyword may be a keyword for a candidate content item to be recommended to a user. Through predicting the relevance between the query and the keyword, it may be determined whether the corresponding candidate content item matches the user intent. Herein, a machine learning model used to predict a relevance between a query and a keyword is referred to as a relevance prediction model. The relevance prediction model may first generate a query representation of the query and a keyword representation of the keyword, respectively, and predict the relevance between the query and the keyword based on the generated query representation and the keyword representation. The query or the keyword often contains only a small number of words or phrases, thus they contain only limited semantic information. Historical click behaviors of users may be used as supplementary information to enrich the query representation and the keyword representation. Currently, the historical click behaviors include click behaviors between the query and the keyword. For example, for a given query, a plurality of content items may be presented, each content item having a corresponding keyword. When a content item has been clicked by the user, it is considered that there exists a click relation between the query and the keyword of the content item. A homogenous graph may be constructed based on such click relation. For example, a query and a keyword may be set as a query node and a keyword node, respectively. When there exists a click relation between the query and the keyword, an edge may exist between the corresponding query node and the corresponding keyword node. Existing relevance prediction models may model a homogeneous graph. For example, neighbor nodes of a query node and a keyword node may be identified from a homogeneous graph, and a query representation of the query and a keyword representation of the keyword may be generated accordingly with information from the neighbor nodes. When modeling a homogeneous graph, an existing relevance prediction model treat nodes in the homogeneous graph as nodes with the same attributes, and only consider one relation type. This approach has limited improvement on the relevance prediction task.

Embodiments of the present disclosure propose relevance prediction based on heterogeneous graph learning. A query and a keyword may be received. The query may be a search query from a user. The keyword may be a keyword for a candidate content item to be recommended to the user. A heterogeneous graph corresponding to the received query and keyword may be generated. Different from a homogeneous graph used in a traditional relevance prediction task, a heterogeneous graph according to the embodiments of the present disclosure may contain a node set with a plurality of node types and an edge set with a plurality of relation types. The node type may include, e.g., a query node, a keyword node, a title node, an entity node, etc. The relation type may include, e.g., query-keyword click relation, query-title click relation, query-entity extraction relation, keyword-entity extraction relation, and keyword-keyword co-occurrence

relation, etc. Compared with a homogeneous graph, a heterogeneous graph may contain more comprehensive information and richer semantics, which can more effectively improve the performance of a relevance prediction task. Query graph structure information for the query and keyword graph structure information for the keyword may be inferred according to the heterogeneous graph. A query representation of the query and a keyword representation of the keyword may be generated based on the query graph structure information and the keyword graph structure information, respectively. The generated query representation and keyword representation may be used to predict a relevance between the query and the keyword.

The embodiments of the present disclosure propose to construct a heterogeneous graph based on a search log, a knowledge graph, etc. The heterogeneous graph may be constructed with various types of historical click behaviors of users extracted from the search log, various types of entities extracted from the knowledge graph, etc. The constructed original heterogeneous graph can contain more comprehensive information and richer semantics, which helps to alleviate the graph sparsity issue. However, historical click behaviors are usually random and capricious. For example, users may generally click on some hot, popular content items, while less popular content items are rarely clicked. Such a behavior produces some long-tailed nodes with only a few graph connections, which makes the sparsity issue still exist. The embodiments of the present disclosure propose to perform a graph augmenting operation on a heterogeneous graph. The graph augmenting operation intends to generate edges related to the relevance prediction task through mining complex associations among node semantic information and a graph topology, so that the sparsity of the heterogeneous graph can be further alleviated. Additionally, users may mistakenly click on some unexpected content items. Such a behavior produces redundant graph connections that are not related to the relevance prediction task, leading to a noisy graph topology. The embodiments of the present disclosure propose to perform a graph pruning operation on the heterogeneous graph. The graph pruning operation intends to prune the heterogeneous graph based on the importance of neighbor nodes relative to a center node. This approach can remove edges that are not related to the relevance prediction task, while preserving useful edges. The process described above employs a coarse-to-fine strategy to learn task-related graph connections, in which edges related to the relevance prediction task are generated in a coarse way, and then edges that are not related to the relevance prediction task are removed from the graph in a fine way. The approach described above utilizes the semantic information and the topology of the node set of the heterogeneous graph, which can significantly alleviate the sparsity issue and the noise issue of the heterogeneous graph.

In order to reveal a diverse topology in a heterogeneous graph, the embodiments of the present disclosure define a meta-path. Herein, a path for connecting two nodes is referred to as a meta-

path. According to types of the connected nodes, there may be a plurality of meta-path types, e.g., a query-keyword type, a query-entity type, a query-title type, a query-title-query type, a query-keyword-query type, a keyword-keyword type, a keyword-entity type, a keyword-query-keyword type, etc.

The relevance between the query and the keyword may be predicted through a heterogeneous graph learning-based relevance prediction model according to the embodiments of the present disclosure. The heterogeneous graph learning-based relevance prediction model includes a graph encoder. A graph encoder can learn a heterogeneous graph through mining complex associations among node semantic information and a graph topology, and can encode the learned text semantics and structure characteristics into a query representation and a keyword representation, thereby generating an augmented query representation and an augmented keyword representation. The graph encoder may aggregate information of neighbor nodes around a node into a representation of that node based on a topology of a heterogeneous graph. The graph encoder may include an intra-meta-path encoder, an inter-meta-path encoder, a node encoder, etc. The intra-meta-path encoder enables to convey information between nodes belonging to the same meta-path. The inter-meta-path encoder enables to convey information between nodes across different meta-paths. The node encoder can aggregate information from a same meta-path and information from different meta-paths into a representation of a corresponding node, e.g., into a query representation of a query node or a keyword representation of a keyword node, thereby generating an augmented query representation or an augmented keyword representation. Predicting the relevance between the query and the keyword with the query representation and the keyword representation generated in such manner can significantly improve the relevance prediction result.

A heterogeneous graph $\mathcal{G}$ may be denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a node set, and $\mathcal{E}$ is an edge set. The heterogeneous graph $\mathcal{G}$ may be defined through two mapping functions, e.g., a node type mapping function $\phi: \mathcal{V} \rightarrow \mathcal{A}$ and a link type mapping function $\psi: \mathcal{E} \rightarrow \mathcal{R}$, where $\mathcal{A}$ is a node type set, and $\mathcal{R}$ is a relation type set.

The node type set $\mathcal{A}$ may include, e.g., a query node Q, a keyword node K, a title node T, an entity node E, etc. The query node, the keyword node, and the title node are set based on, e.g., components extracted from a search log. The search log may be a log associated with a search engine for recording search behaviors of a user. The query, the keyword, the title, etc., may be extracted from the search log. The query may be a search query from a user. The keyword may be a keyword for a content item. The content item may include news, video, movie, book, music, webpage, product information, etc. The title may be a title of a webpage searched for the query. The extracted query, keyword, and title may be set as a query node, a keyword node, and a title node, respectively. An entity node is set based on, e.g., a component extracted from the knowledge

graph. A knowledge graph is a structured semantic knowledge base that describes concepts and their interrelationships in the physical world in symbolic form. The knowledge graph contains a large number of entities and relations between them. An entity associated with a query or an entity associated with a keyword may be extracted from the knowledge graph. An extracted entity may be set as an entity node. An exemplary process for setting nodes will be described later in conjunction with FIG.3.

A relation type is also known as an edge type. A relation type set $\mathcal{R}$ may include e.g., query-keyword click relation R_{qk}, query-title click relation R_{qt}, query-entity extraction relation R_{qe}, keyword-entity extraction relation R_{ke}, keyword-keyword co-occurrence relation R_{kk}, etc.

The query-keyword click relation is used to describe a relation between a query and a keyword. A plurality of queries and a plurality of clicked keywords for each query may be obtained from a search log. Herein, a clicked keyword of a query refers to a keyword of a content item in a plurality of content items presented to the query that has been clicked by a user. The plurality of clicked keywords of the query may be ranked based on click frequencies. There may be a query-keyword click relation between the query and a predetermined number of top-ranked, e.g., top 5, clicked keywords.

The query-title click relation is used to describe a relation between a query and a title. A plurality of queries and a plurality of clicked titles for each query may be obtained from a search log. Herein, a clicked title of a query refers to a title of a webpage in a plurality of webpages searched for the query that has been clicked by a user. The plurality of clicked titles of the query may be ranked based on click frequencies. There may be a query-title click relation between the query and a predetermined number of top-ranked, e.g., top 5, clicked titles.

The query-entity extraction relation is used to describe a relation between a query and an entity. A plurality of entities associated with each query may be extracted from a knowledge graph. Each entity has a corresponding confidence score. The plurality of entities of the query may be ranked based on confidence scores. There may be a query-entity extraction relation between the query and a predetermined number of top-ranked, e.g., top 3, entities.

The keyword-entity extraction relation is used to describe a relation between a keyword and an entity. A plurality of entities associated with each keyword may be extracted from a knowledge graph. Each keyword has a corresponding confidence score. The plurality of entities of the keyword may be ranked based on confidence scores. There may be a keyword-entity extraction relation between the keyword and a predetermined number of top-ranked, e.g., top 3, entities.

The keyword-keyword co-occurrence relation is used to describe a relation between a keyword and a keyword. A content item provider typically indicates a set of keywords. Co-occurrence of two keywords may mean that the two keywords are similar. For a keyword, its co-occurrencing

keywords may be ranked based on co-occurrence frequencies. There may be a keyword-keyword co-occurrence relation between the keyword and a predetermined number of top-ranked, e.g., top 5, keywords.

In order to reveal a diverse topology in a heterogeneous graph, the embodiments of the present disclosure define a meta-path. A meta-path may be defined as a path $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \dots \xrightarrow{R_l} A_{l+1}$ (abbreviated as $A_1\ A_2\ \dots A_{l+1}$), which describes a composite relation $R = R_1 \circ R_2 \circ \dots R_l$ between a node A_1 and a node A_{l+1}, where $\circ$ is a composition operator on relations. According to types of the connected nodes, there may be a plurality of meta-path types, e.g., a query-keyword QK type, a query-entity QE type, a query-title QT type, a query-title-query QTQ type, a query-keyword-query QKQ type, a keyword-keyword KK type, a keyword-entity KE type, a keyword-query-keyword KQK type, etc.

FIG.1 illustrates an exemplary heterogeneous graph 100 according to an embodiment of the present disclosure.

The heterogeneous graph 100 contains 4 node types, e.g., query node Q, keyword node K, title node T, entity node E, etc. For example, a node 102 is a query node Q, nodes 104, 112 and 114 are keyword nodes K, a node 106 is a title node T, nodes 108 and 110 are entity nodes E, etc.

Additionally, the heterogeneous graph 100 contains five relation types, e.g., query-keyword click relation R_{qk}, query-title click relation R_{qt}, query-entity extraction relation R_{qe}, keyword-entity extraction relation R_{ke}, and keyword-keyword co-occurrence relation R_{kk}. For example, the relation type between the node 102 and the node 104 is the query-keyword click relation R_{qk}, the relation type between the node 102 and the node 106 is the query-title click relation R_{qt}, the relation type between the node 102 and the node 108 is the query-entity extraction relation R_{qe}, the relation type between the node 104 and the node 110 is the keyword-entity extraction relation R_{ke}, and the relation type between the node 112 and the node 114 is the keyword-keyword co-occurrence relation R_{kk}, etc.

An exemplary process for generating a heterogeneous graph will be illustrated later in conjunction with FIG.2 to FIG.5. It should be appreciated that the heterogeneous 100 shown in FIG.1 is merely an example of the heterogeneous graph. Depending on actual application requirements, the heterogeneous graph may have any other structure and may include more or fewer elements.

FIG.2 illustrates an exemplary process 200 for generating a heterogeneous graph according to an embodiment of the present disclosure. In the process 200, a heterogeneous graph 212 may be generated based on a search log 202 and/or a knowledge graph 204 through a heterogeneous graph generating module 210.

The heterogeneous graph generating module 210 includes a graph constructing unit 220. The

graph constructing unit 220 intends to incorporate various types of nodes and edges into a graph. An original heterogeneous graph 222 may be constructed based on the search log 202 and/or the knowledge graph 204 through the graph constructing unit 220. FIG.3 illustrates an exemplary process 300 for constructing an original heterogeneous graph according to an embodiment of the present disclosure. The process 300 may be performed by the graph constructing unit 220 in FIG.2.

At 302, a component set may be extracted from a search log and/or a knowledge graph. The component set may include a historical query set, a historical keyword set and a historical title set, etc., extracted from the search log. These component sets may be extracted from various types of historical click behaviors of users contained in the search log. Preferably, the component set may further include an entity set corresponding to the historical query set, an entity set corresponding to the historical keyword set, etc., extracted from the knowledge graph.

At 304, the component set may be set as a node set. For example, a historical query may be set as a query node, a historical keyword may be set as a keyword node, and a historical title may be set as a title node, an entity may be set as a physical node, etc. Subsequently, an edge set among the node set may be determined.

At 306, for every two nodes in the node set, it may be determined whether a relation exists between two components corresponding to the two nodes. The relation may include e.g., query-keyword click relation, query-title click relation, query-entity extraction relation, keyword-entity extraction relation, and keyword-keyword co-occurrence relation, etc.

At 308, in response to determining that a relation exists between the two components, it may be determined that an edge exists between the two nodes. An edge type of an edge between two nodes may correspond to a relation type of the relation between two components corresponding to the two nodes.

The step 306 and the step 308 may be performed for every two nodes in the node set, thereby at 310, an edge set among the node set may be obtained.

At 312, the node set and the edge set may be combined into an original heterogeneous graph. Preferably, a node without any edge may be removed from the original heterogeneous graph.

Through the process 300, the original heterogeneous graph may be constructed with various types of historical click behaviors of users extracted from the search log, various types of entities extracted from the knowledge graph, etc. The constructed original heterogeneous graph can contain more comprehensive information and richer semantics, which helps to alleviate the graph sparsity issue. However, historical click behaviors are usually random and capricious. For example, users may generally click on some hot, popular content items, while less popular content items are rarely clicked. Such a behavior produces some long-tailed nodes with only a few graph connections, which makes the sparsity issue still exist. The embodiments of the present disclosure

propose to perform a graph augmenting operation on the original heterogeneous graph.

Referring back to FIG.2, after the original heterogeneous graph 222 is obtained, an augmented heterogeneous graph 232 may be generated through a graph augmenting unit 230 in the heterogeneous graph generating module 210. The graph augmenting unit 230 intends to generate potential edges related to the relevance prediction task through mining complex associations among node semantic information and a graph topology, so that the sparsity of the heterogeneous graph can be further alleviated. The augmented heterogeneous graph 232 may be generated based on the semantic information and/or a topology of a node set of the original heterogeneous graph 222 through the graph augmenting unit 230.

FIG.4 illustrates an exemplary process 400 for generating an augmented heterogeneous graph according to an embodiment of the present disclosure. The process 400 may be performed by the graph augmenting unit 230 in FIG.2. Different relation types may have their own unique meaning and amount of information. In view of this, the embodiments of the present disclosure propose an augmenting strategy based on relation types. An original heterogeneous graph may include a plurality of original partial graphs corresponding to a plurality of relation types. Herein, an original partial graph refers to a partial graph for a specific relation type in an original heterogeneous graph. The original partial graph may contain a node subset corresponding to that relation type. For example, an original partial graph corresponding to a query-keyword click relation R_{qk} may include a node subset consisting of query nodes and keyword nodes. The original partial graph for the query-keyword click relation R_{qk} may be denoted as $\mathbf{E} \in \mathbb{R}^{|\mathcal{Q}|\times|\mathcal{K}|}$, where $|\mathcal{Q}|$ is the number of query nodes in the node subset, and $|\mathcal{K}|$ is the number of keyword nodes in the node subset. In the process 400, for each relation type of a plurality of relation types, an augmented partial graph corresponding to the relation type may be generated. Subsequently, a plurality of augmented partial graphs corresponding to the plurality of relation types may be combined into an augmented heterogeneous graph. Steps 402 to 406 will take the relation type being the query-keyword click relation R_{qk} as an example to illustrate the exemplary process of generating an augmented partial graph.

At 402, a semantic partial graph corresponding to each relation type may be generated based on semantic similarity between two nodes in a node subset of an original partial graph corresponding to the relation type. The semantic partial graph may be denoted as $\mathbf{E}_s$. The semantic similarity between corresponding two nodes may be captured with the semantic partial graph. For example, the semantic partial graph for the query-keyword click relation R_{qk} may capture the semantic similarity between a query node q and a keyword node k. There may be a connection between a query node q and a keyword node k with similar semantics. An embedding $\tilde{\mathrm{h}}_q$ of the query node

q and an embedding $\tilde{\mathrm{h}}_{\mathrm{k}}$ of the keyword node k may be generated first, as shown in the following equation:

$$\tilde{\mathrm{h}}_{\mathrm{q}} = \mathcal{T}(q) \tag{1}$$

$$\tilde{\mathrm{h}}_{\mathrm{k}} = \mathcal{T}(k) \tag{2}$$

where $\mathcal{T}$ may be any pre-learned text encoder. As an example, $\mathcal{T}$ may be a TwinBert model fine-tuned through relevance labels. The embedding $\tilde{\mathrm{h}}_{\mathrm{q}}$ and the embedding $\tilde{\mathrm{h}}_{\mathrm{k}}$ are d dimension embeddings for the query node q and the keyword node k, respectively.

Subsequently, metric learning may be applied to the embedding $\tilde{\mathrm{h}}_{\mathrm{q}}$ and the embedding $\tilde{\mathrm{h}}_{\mathrm{k}}$, to generate a semantic partial graph $\mathbf{E}_s \in \mathbb{R}^{|\mathcal{Q}|\times|\mathcal{K}|}$ corresponding to the query-keyword click relation R_{qk}, as shown in the following equation:

$$\mathbf{E}_{\mathrm{s}}[q,k] = \begin{cases} \mathcal{M}_s(\tilde{\mathrm{h}}_{\mathrm{q}}, \tilde{\mathrm{h}}_{\mathrm{k}}) & \mathcal{M}_s(\tilde{\mathrm{h}}_{\mathrm{q}}, \tilde{\mathrm{h}}_{\mathrm{k}}) \geq \epsilon \\ 0 & \mathcal{M}_s(\tilde{\mathrm{h}}_{\mathrm{q}}, \tilde{\mathrm{h}}_{\mathrm{k}}) < \epsilon \end{cases} \tag{3}$$

where ϵ is a hyperparameter used to control the sparsity. $\mathcal{M}_s(\tilde{\mathrm{h}}_{\mathrm{q}}, \tilde{\mathrm{h}}_{\mathrm{k}})$ is a function used to measure the z head weighted cosine similarity between $\tilde{\mathrm{h}}_{\mathrm{q}}$ and $\tilde{\mathrm{h}}_{\mathrm{k}}$, as shown in the following equation:

$$\mathcal{M}_s(\tilde{\mathrm{h}}_{\mathrm{q}}, \tilde{\mathrm{h}}_{\mathrm{k}}) = \frac{1}{z}\sum_{i=1}^{z} \cos(\mathrm{w}_{\mathrm{i}} \odot \tilde{\mathrm{h}}_{\mathrm{q}}, \mathrm{w}_{\mathrm{i}} \odot \tilde{\mathrm{h}}_{\mathrm{k}}) \tag{4}$$

where $\odot$ is the Hadamard product, and $\mathrm{w}_{\mathrm{i}} \in \mathbb{R}^d$ is a learnable vector indicating the importance of different dimensions.

It should be appreciated that the similarity between $\tilde{\mathrm{h}}_{\mathrm{q}}$ and $\tilde{\mathrm{h}}_{\mathrm{k}}$ may also be measured in other ways. For example, for an original heterogeneous graph containing a large number of nodes, an vanilla cosine similarity between $\tilde{\mathrm{h}}_{\mathrm{q}}$ and $\tilde{\mathrm{h}}_{\mathrm{k}}$ may also be measured, which may be efficiently calculated through an Approximate Nearest Neighbors (ANN) search method.

At 404, a propagation partial graph corresponding to the relation type may be generated based on semantic similarity between two nodes of the same type in the node subset and a topology of the original partial graph. Similar nodes with similar semantics may have similar neighbor nodes. Associations between nodes and the topology may be captured through the propagation partial graph. The following takes a query node as an example to illustrate an exemplary process for generating a query propagation partial graph. First, a semantic matrix $\mathbf{S}_{\mathrm{q}} \in \mathbb{N}^{|\mathcal{Q}|\times|\mathcal{Q}|}$ between query nodes may be calculated, as shown in the following equation:

$$\mathbf{S}_{\mathrm{q}}[q_1, q_2] = \begin{cases} \mathcal{M}_{\mathrm{q}}(\tilde{\mathrm{h}}_{\mathrm{q}_1}, \tilde{\mathrm{h}}_{\mathrm{q}_2}) & \mathcal{M}_{\mathrm{q}}(\tilde{\mathrm{h}}_{\mathrm{q}_1}, \tilde{\mathrm{h}}_{\mathrm{q}_2}) \geq \epsilon \\ 0 & \mathcal{M}_{\mathrm{q}}(\tilde{\mathrm{h}}_{\mathrm{q}_1}, \tilde{\mathrm{h}}_{\mathrm{q}_2}) < \epsilon \end{cases} \tag{5}$$

where the form of $\mathcal{M}_{\mathrm{q}}$ is consistent with the equation (4), but has different w_{i}. Alternatively, $\mathcal{M}_{\mathrm{q}}$ may also be an vanilla cosine similarity measure function.

Subsequently, a query propagation graph $\mathbf{E}_q \in \mathbb{R}^{|\mathcal{Q}|\times|\mathcal{K}|}$ may be obtained, as shown in the following equation:

$$\mathbf{E}_q = \mathbf{S}_q \mathbf{E} \tag{6}$$

The original partial graph $\mathbf{E}$ may reflect original connections between nodes which are based on historical behaviors of users. Through the query propagation graph, the semantic similarity between query nodes may be propagated between nodes through such original connections.

A keyword propagation partial graph $\mathbf{E}_k \in \mathbb{R}^{|\mathcal{Q}|\times|\mathcal{K}|}$ may be generated in a manner similar to the manner in which the query propagation graph $\mathbf{E}_q$ is generated described above.

At 406, an augmented partial graph corresponding to the relation type may be generated based on the original partial graph $\mathbf{E}$, the semantic partial graph $\mathbf{E}_s$, and the propagation partial graphs $\mathbf{E}_q$ and $\mathbf{E}_k$. Continuing to take the relation type being the query-keyword click relation R_{qk} as an example, generating the augmented partial graph $\mathbf{E}_f$ corresponding to the relation type may be as shown in the following equation:

$$\mathbf{E}_f = \Upsilon(\mathbf{E}, \mathbf{E}_s, \mathbf{E}_q, \mathbf{E}_k) \tag{7}$$

where Υ is a combination function. As an example, Υ may be an efficient summation operation.

The step 402 to the step 406 may be performed for other relation types, such that at 408, a plurality of augmented partial graphs corresponding to the plurality of relation types may be obtained.

At 410, the plurality of augmented partial graphs may be combined into an augmented heterogeneous graph.

Through the process 400, the sparse original heterogeneous graph may be augmented to obtain a dense augmented heterogeneous graph, which can further alleviate the sparsity of the heterogeneous graph. However, there may be some noise in this dense augmented heterogeneous graph. For example, users may mistakenly click on some unexpected content items. Such a behavior produces redundant edges that are not related to the relevance prediction task, leading to a noisy graph topology. According to an embodiment of the present disclosure, the augmented heterogeneous graph may be pruned. The graph pruning operation intends to prune the augmented heterogeneous graphs based on the importance of neighbor nodes relative to a center node. This approach can remove edges that are not related to the relevance prediction task, while preserving useful edges.

Referring back to FIG.2, after the augmented heterogeneous graph 232 is generated through the graph augmenting unit 230, the augmented heterogeneous graph 232 may be pruned through the graph pruning unit 240 in the heterogeneous graph generating module 210, to obtain the heterogeneous graph 212. The augmented heterogeneous graph 232 may be pruned based on semantic information and/or a topology of a node set of the augmented heterogeneous graph 232.

The graph pruning unit 240 intends to remove noisy edges that are not related to the relevance prediction task from the graph, while preserving useful edges. The augmented heterogeneous graph 232 may be pruned based on the importance of neighbor nodes in the node set of the augmented heterogeneous graph 232 relative to a center node through the graph pruning unit 240.

In order to reveal a diverse topology in a heterogeneous graph, the embodiments of the present disclosure define a neighbor node set based on a meta-path. For example, given a node i and a meta-path type Φ, a neighbor node set $\mathcal{N}_i^{\Phi}$ based on the meta-path type Φ of the node i refers to a node set connected to the node i through meta-paths of the meta-path type Φ. At this point, the node i may also be referred to as a center node. A node set of an augmented heterogeneous graph may include a plurality of center nodes. Each center node may have a plurality of neighbor node sets $\{\mathcal{N}_i^{\Phi}\}$ corresponding to a plurality of meta-path types $\{\Phi\}$. Noisy or redundant neighbor nodes may be removed from respective neighbor node sets through a pruning operation. FIG.5 illustrates an exemplary process 500 for pruning an augmented heterogeneous graph according to an embodiment of the present disclosure.

At 502, a neighbor node subset that are most important to a center node may be identified from a neighbor node set of the center node that are based on a meta-path type. The following description takes the center node being a query node q and the meta-path type being query-keyword QK as an example for illustration. A neighbor node set of a query node q that are based on the meta-path type QK may be denoted as $\mathcal{N}_q^{\Phi_{QK}}$. The importance of a keyword node k in the neighbor node set $\mathcal{N}_q^{\Phi_{QK}}$ relative to the center node, i.e., the query node q, may be calculated through the following equation:

$$m_{q,k} = \Gamma(\tilde{\mathrm{h}}_{\mathrm{q}}, \tilde{\mathrm{h}}_{\mathrm{k}}, \mathbf{E}_{\mathrm{f}}[q,k]) \tag{8}$$

$$s_{q,k} = \frac{\exp(m_{q,k})}{\sum_{e \in \mathcal{N}_q^{\Phi_{QK}}} \exp(m_{q,e})} \tag{9}$$

where the function Γ may be implemented as a multi-layer feed-forward network with an output dimension as 1.

The approach described above intends to learn important neighbor nodes through incorporating the semantic information and the topology of nodes. A predetermined number, e.g., t, of neighbor nodes that are most important to the query node q may be sampled from the neighbor node set $\mathcal{N}_q^{\Phi_{QK}}$ with $s_{q,k}$.

Preferably, in order to make the sampling process differentiable, Gumbel-Softmax technique may be introduced to generate discrete samples, as shown in the following equation:

$$z_{q,k} = \frac{\exp\left((\log(s_{q,k}) + g)/\tau\right)}{\sum_{e \in \mathcal{N}_q^{\Phi_{QK}}} \exp\left((\log(s_{q,e}) + g)/\tau\right)} \tag{10}$$

where τ is the temperature used to control the closeness of the τ-softmax function to the discrete categorical distribution. At this point, a predetermined number, e.g., t, of neighbor nodes that are most important to the query node q may be sampled from the neighbor node set $\mathcal{N}_q^{\Phi_{QK}}$ with $z_{q,k}$. The sampled t neighbor nodes may be combined into a neighbor node subset that are most important to the center node. At 504, neighbor nodes in the neighbor node set that are outside the neighbor node subset may be determined. Subsequently, at 506, edges between the determined neighbor nodes and the center node may be removed. A graph with edges being removed may be referred to as a subgraph.

The graph pruning operation described above intends to prune heterogeneous graphs based on the importance of neighbor nodes relative to the center node. This approach can remove edges that are not related to the relevance prediction task, while preserving useful edges. The process described above employs a coarse-to-fine strategy to learn task-related graph connections, in which edges related to the relevance prediction task are generated in a coarse way, and then edges that are not related to the relevance prediction task are removed from the graph in a fine way. The approach described above utilizes the semantic information and the topology of the node set of the heterogeneous graph, which can significantly alleviate the sparsity issue and the noise issue of the heterogeneous graph.

It should be appreciated that the process for generating the heterogeneous graph described above in conjunction with FIG.2 to FIG.5 is merely exemplary. Depending on actual application requirements, the steps in the process for generating the heterogeneous graph may be replaced or modified in any manner, and the process may include more or fewer steps. For example, although in the process 200, both the search log 202 and the knowledge graph 204 are shown, in some embodiments, it is possible to construct the original heterogeneous graph 222 based only on the search log 202. Additionally, although in the process 200, the heterogeneous graph generating module 210 includes the graph constructing unit 220, the graph augmenting unit 230, and the graph pruning unit 240, in some embodiments, the heterogeneous graph generating module 210 may not include the graph augmenting unit 230 or the graph pruning unit 240. When the heterogeneous graph generating module 210 does not include the graph augmenting unit 230, the graph pruning unit 240 may prune the original heterogeneous graph 222 constructed by the graph constructing unit 210, to obtain the final heterogeneous graph 212. When the heterogeneous graph generating module 210 does not include the graph pruning unit 240, the augmented heterogeneous graph 232 generated by the graph augmenting unit 230 may be directly used as the final

heterogeneous graph 212. Additionally, the specific order or hierarchy of the steps in the process 200 to the process 500 is only exemplary, and the process for generating the heterogeneous graph may be performed in an order different from the described order.

FIG.6 illustrates an exemplary process 600 for relevance prediction based on heterogeneous graph learning according to an embodiment of the present disclosure. In the process 600, a query 602 and a keyword 604 may be received. The keyword 604 is a keyword for a candidate content item. The candidate content item may be a content item from a set of candidate content items that may be recommended to a user. The candidate content item may include news, video, movie, book, music, webpage, product information, etc. A relevance 612 between the query 602 and the keyword 604 may be predicted through a heterogeneous graph learning-based relevance prediction model 610.

A heterogeneous graph 622 corresponding to the query 602 and the keyword 604 may be generated first. For example, the heterogeneous graph 622 may be generated based on the search log 606 and/or the knowledge graph 608 through a heterogeneous graph generating module 620 in the heterogeneous graph learning-based relevance prediction model 610. The heterogeneous graph generating module 620 may correspond to the heterogeneous graph generating module 210 in FIG.2.

Subsequently, graph structure information 632 for the query 602 may be inferred according to the heterogeneous graph 622 through a graph structure inferring module 630 in the heterogeneous graph learning-based relevance prediction model 610. FIG.7 illustrates an exemplary process 700 for inferring query graph structure information for a query according to an embodiment of the present disclosure. The process 700 may be performed by the graph structure inferring module 630 in FIG.6. At 702, a meta-path to which a query node corresponding to a query belongs may be inferred according to a heterogeneous graph. The query node may belong to one or more meta-paths. For example, meta-paths to which the query node q belongs may be denoted as $\{\Phi_1, \Phi_2, \ldots, \Phi_O\}$, where $O \geq 1$ is the number of meta-paths. At 704, a node set which is based on the meta-path may be inferred according to the heterogeneous graph. When there are one or more meta-paths, a node set based on each meta-path may be inferred, thereby one or more node sets may be obtained. A node set based on a meta-path Φ_o may be denoted as $\mathcal{N}^{\Phi_o} = \{n_1^{\Phi_o}, n_2^{\Phi_o}, \ldots n_{k_o}^{\Phi_o}\}$, where $k_o \geq 1$ is the number of nodes included in the node set. At 706, query graph structure information may be generated based on the meta-path and the node set. The graph structure information may be denoted as $\mathcal{N} = \{\mathcal{N}^{\Phi_0}, \mathcal{N}^{\Phi_1}, \mathcal{N}^{\Phi_2}, \ldots, \mathcal{N}^{\Phi_O}\}$, where $\mathcal{N}^{\Phi_0} = \{q\}$ represents a center node, i.e., the query node q.

Referring back to FIG.6, after the graph structure information 632 for the query 602 is inferred, a query representation 646 of the query 602 may be generated based on the graph structure

information 632. FIG.8 illustrates an exemplary process 800 for generating a query representation of a query according to an embodiment of the present disclosure. The process 800 may be performed by a text encoder 640, a graph encoder 642, a pooling module 644, etc., in FIG.6.

At 802, a meta-path to which a query node corresponding to a query belongs and a neighbor node which is based on the meta-path may be obtained from query graph structure information. The query node may belong to one or more meta-paths. There may be one or more neighbor nodes on each meta-path.

At 804, a token sequence corresponding to the meta-path may be generated based on the query and the neighbor node. Herein, a token refers to a basic language unit that constitutes texts in different languages. In order to be able to convey information at different levels, the embodiments of the present disclosure propose two types of special tokens, e.g., a local token [TRA] for conveying information within a same meta-path, and a global token [TER] for conveying information across different meta-paths, etc. First, a query textual token of a query and a neighbor textual token of a neighbor node may be obtained through a tokenization operation. Next, a query local token [TRA] for the query may be added before the query textual token. Similarly, a neighbor local token [TRA] for the neighbor node may be added before the neighbor textual token. Additionally, each meta-path may have an associated global token [TER]. The global token of the meta-path, the query local token, the query textual token, the neighbor local token, and the neighbor textual token may be combined into a token sequence corresponding to the meta-path.

At 806, an initial embedding sequence corresponding to the meta-path may be generated based on the token sequence corresponding to the meta-path. For example, an initial embedding sequence may be generated through a text encoder 640.

At 808, an embedding sequence corresponding to the meta-path may be generated through iteratively performing a graph encoding operation based on the initial embedding sequence. Through performing the graph encoding operation, the previous embedding sequence corresponding to the meta-path may be updated to a current embedding sequence corresponding to the meta-path. For example, referring to FIG.6, the graph encoding operation may be performed iteratively through a plurality of, e.g., M ($M \geq 1$), graph encoders 642. The plurality of graph encoders 642 are stacked together. The graph encoder 642 includes an intra-meta-path encoder 642-1, an inter-meta-path encoder 642-2 and a node encoder 642-3, etc. The intra-meta-path encoder 642-1 can capture shared features of a same meta-path, to convey information between nodes belonging to a same meta-path. The inter-meta-path encoder 642-2 enables to convey information between nodes across different meta-paths. The node encoder 642-3 can aggregate information from a same meta-path and information and from different meta-paths into a representation of a corresponding node. An exemplary process for performing a graph encoding

operation will be described later in conjunction with FIG.9.

At 810, a query representation may be generated based on the embedding sequence. For example, the query representation may be generated based on an embedding sequence through a pooling module 644. An exemplary process for generating a query representation based on an embedding sequence will be described later in conjunction with FIG.10.

The process described above may generate the query representation of the query. For example, referring to FIG.6, a query representation 646 of the query 602 may be generated. A keyword representation 666 of the keyword 604 may be generated through a process similar to the process of generating the query representation 646. Keyword graph structure information 652 for the keyword 604 may be inferred according to the heterogeneous graph 622 through a graph structure inferring module 650. The structure and function of the graph structure inferring module 650 may be similar to the structure and function of the graph structure inferring module 630. Subsequently, a keyword representation 666 of the keyword 604 may be generated based on the keyword graph structure information 652 through a text encoder 660, M graph encoders 662 and a pooling module 664. The structure and function of the text encoder 660, the structure and function of the graph encoder 662 and the structure and function of the pooling module 664 may be similar to the structure and function of the text encoder 640, the structure and function of the graph encoder 642 and the structure and function of the pooling module 644, respectively. The graph encoder 662 includes an intra-meta-path encoder 662-1, an inter-meta-path encoder 662-2, a node encoder 662-3, etc. The structure and function of the intra-meta-path encoder 662-1, the structure and function of the inter-meta-path encoder 662-2, and the structure and function of the node encoder 662-3 may be the same as the structure and function of the intra-meta-path encoder 642-1, the structure and function of the inter-meta-path encoder 642-2, and the structure and function of the node encoder 642-3, respectively.

After the query representation 646 and the keyword representation 666 are generated, a relevance 612 between the query 602 and the keyword 604 may be predicted based on the query representation 646 and the keyword representation 666 through a prediction layer 670. The relevance 612 may be defined as cosine similarity, which may be further viewed as a probability score $p_{q,k}$. When both the query representation 646 and the keyword representation representation 666 are normalized, the cosine similarity may also be equivalent to the Euclidean distance and may be efficiently solved through an ANN search method.

A cross-entropy loss function may be selected as an objective function for training the heterogeneous graph learning-based relevance prediction model 610, as shown in the following equation:

$$\mathcal{L} = -\sum\left(y_i \log\left(p_{q_i,k_i}\right) + (1 - y_i)\log\left(1 - p_{q_i,k_i}\right)\right) \quad (11)$$

FIG.9 illustrates an exemplary process 900 for performing a graph encoding operation according to an embodiment of the present disclosure. The process 900 may be performed by a graph encoder 642 in FIG.6.

In the process 900, a query node belongs to two meta-paths, i.e., a first meta-path and a second meta-path. As an example, the type of the first meta-path is query-title QT. There is a query node and a title node on this meta-path. The type of the second meta-path is query-entity QE. There is a query node and an entity node on this meta-path. It should be appreciated that when the query node belongs to other types of meta-paths, or belongs to more meta-paths, the graph encoding operation may be performed through a process similar to the process 900. Through the process 900, a previous embedding sequence 902 corresponding to the first meta-path may be updated to a current embedding sequence 904 and a previous embedding sequence 906 corresponding to the second meta-path may be updated to a current embedding sequence 908.

A previous embedding of the jth token of a node i belonging to a meta-path Φ_o can be denoted as $\mathrm{t}_{ij}^{\Phi_o}$. $\mathrm{t}_{ia}^{\Phi_o}$, i.e., when $j = a$, is a previous embedding of a local token [TRA] of the node i. A previous embedding of a global token [TER] associated with the meta-path Φ_o may be denoted as $\mathrm{t}_{r}^{\Phi_o}$.

The previous embedding sequence 902 corresponding to the first meta-path includes previous embeddings corresponding to the query node on the meta-path, e.g., a previous local embedding 910 for the local token [TRA] of the query node and two previous textual embeddings 912 and 914 for two textual tokens of the query node. The previous embedding sequence 902 further includes a previous global embedding 916 for the global token [TER] associated with the first meta-path. The previous embedding sequence 902 further includes previous embeddings corresponding to the title node on the meta-path, e.g., a previous local embedding 918 for the local token [TRA] of the title node and two previous textual embeddings 920 and 922 for two textual tokens of the title node.

The previous embedding sequence 906 corresponding to the second meta-path includes previous embeddings corresponding to the query node on the meta-path, e.g., a previous local embedding 924 for the local token [TRA] of the query node and two previous textual embeddings 926 and 928 for two textual tokens of the query node. The previous embedding sequence 906 further includes a previous global embedding 930 for the global token [TER] associated with the second meta-path. The previous embedding sequence 906 further includes previous embeddings corresponding to the entity node on the meta-path, e.g., a previous local embedding 932 for the local token [TRA] of the entity node and two previous textual embeddings 934 and 936 for two

textual tokens of the entity node.

An intermediate global embedding of a global token associated with a meta-path and a plurality of intermediate local embeddings of a plurality of local tokens associated with the meta-path may be generated, through a self-attention mechanism, based on a previous global embedding of the global token and a plurality of previous local embeddings of the plurality of local tokens. Given a meta-path Φ_o and a node set $\mathcal{N}^{\Phi_o}$ belonging to the meta-path Φ_o, a previous global embedding of a global token [TER] associated with the meta-path and a plurality of previous local embeddings of a plurality of local tokens [TRA] of a plurality of nodes may be collected, to form an intra-meta-path matrix: $\mathrm{T}_a^{\Phi_o} = [\mathrm{t}_r^{\Phi_o}, \mathrm{t}_{1a}^{\Phi_o}, \mathrm{t}_{2a}^{\Phi_o}, \ldots, \mathrm{t}_{k_o a}^{\Phi_o}] \in \mathbb{R}^{(k_o+1)\times d}$, where k_o is the number of nodes in $\mathcal{N}^{\Phi_o}$. Then, a multi-head self-attention operation may be employed for the intra-meta-path matrix $\mathrm{T}_a^{\Phi_o}$ to convey information between nodes belonging to a same meta-path. For any attention head, the intra-meta-path information conveying may be defined through the following equation:

$$\widehat{\mathrm{T}}_a^{\Phi_o} = \mathrm{softmax}\left(\frac{\widehat{\mathrm{Q}}^{\Phi_o}\widehat{\mathrm{K}}^{\Phi_o\mathrm{T}}}{\sqrt{d}}\right)\widehat{\mathrm{V}}^{\Phi_o} \tag{12}$$

where

$$\begin{cases} \widehat{\mathrm{Q}}^{\Phi_o} = \widehat{\mathrm{T}}_a^{\Phi_o}\widehat{\mathrm{W}}_Q^{\Phi_o} \\ \widehat{\mathrm{K}}^{\Phi_o} = \widehat{\mathrm{T}}_a^{\Phi_o}\widehat{\mathrm{W}}_K^{\Phi_o} \\ \widehat{\mathrm{V}}^{\Phi_o} = \widehat{\mathrm{T}}_a^{\Phi_o}\widehat{\mathrm{W}}_V^{\Phi_o} \end{cases} \tag{13}$$

where the matrix $\widehat{\mathrm{W}}_Q^{\Phi_o} \in \mathbb{R}^{d\times d}$, the matrix $\widehat{\mathrm{W}}_K^{\Phi_o} \in \mathbb{R}^{d\times d}$ and the matrix $\widehat{\mathrm{W}}_V^{\Phi_o} \in \mathbb{R}^{d\times d}$ are learnable variables.

The learned matrix may be denoted as $\widehat{\mathrm{T}}_a^{\Phi_o} = [\hat{\mathrm{t}}_r^{\Phi_o}, \hat{\mathrm{t}}_{1a}^{\Phi_o}, \hat{\mathrm{t}}_{2a}^{\Phi_o}, \ldots, \hat{\mathrm{t}}_{k_o a}^{\Phi_o}]$. This matrix contains an intermediate global embedding $\hat{\mathrm{t}}_r^{\Phi_o}$ of the global token [TER] corresponding to the meta-path Φ_o and a plurality of intermediate local embeddings $\{\hat{\mathrm{t}}_{1a}^{\Phi_o}, \hat{\mathrm{t}}_{2a}^{\Phi_o}, \ldots, \hat{\mathrm{t}}_{k_o a}^{\Phi_o}\}$ of the plurality of local tokens [TRA]. $\hat{\mathrm{t}}_{ia}^{\Phi_o}$ represents an intermediate local embedding of the local token [TRA] obtained through an intra-meta-path encoder, which may capture local heterogeneity within a single meta-path. The intermediate global embedding $\hat{\mathrm{t}}_r^{\Phi_o}$ of the global token [TER] corresponding to the meta-path Φ_o may be regarded as an embedding of the meta-path Φ_o. Through the self-attention mechanism, embeddings of other tokens may contain context information centered on different nodes.

For example, for the first meta-path, an intermediate local embedding 940, an intermediate global embedding 942 and an intermediate local embedding 944 corresponding to the first meta-path may be generated, through a self-attention-mechanism-based intra-meta-path encoder 938, based on the previous local embedding 910, the previous global embedding 916 and the previous local

embedding 918. Similarly, for the second meta-path, an intermediate local embedding 948, an intermediate global embedding 950 and an intermediate local embedding 952 corresponding to the second meta-path may be generated, through a self-attention-mechanism-based intra-meta-path encoder 946, based on the previous local embedding 924, the previous global embedding 930 and the previous local embedding 932. The intra-meta-path encoder 938 and the intra-meta-path encoder 946 may correspond to the intra-meta-path encoder 642-1 in FIG.6.

Then, a current global embedding of the global token [TER] may be generated based on the intermediate global embedding of the global token [TER] through the self-attention mechanism. For each meta-path Φ_o, the intermediate global embeddings $\hat{t}_r^{\Phi_o}$ of the global token [TER] of the meta-path Φ_o obtained through the intra-meta-path encoder may be collected, to form an inter-meta-path matrix: $\hat{T}_r = [\hat{t}_r^{\Phi_1}, \hat{t}_r^{\Phi_2}, \ldots, \hat{t}_r^{\Phi_O}]^T \in \mathbb{R}^{(O+1)\times d}$. Each row in the inter-meta-path matrix $\hat{T}_r$ represents an embedding for a corresponding meta-path. A multi-head self-attention operation may be employed for the inter-meta-path matrix $\hat{T}_r$, as shown in the following equation:

$$\tilde{T}_r = \text{softmax}\left(\frac{\tilde{Q}_r \tilde{K}_r^T}{\sqrt{d}} + \tilde{B}\right)\tilde{V}_r \tag{14}$$

where $\tilde{B} \in \mathbb{R}^{(O+1)\times(O+1)}$ is a pattern attention proposed by the embodiments of the present disclosure, which can be used to further reveal association between different meta-paths. The element B_{ij} represents an edit distance between a pattern of a meta-path Φ_i and a pattern of a meta-path Φ_j. The edit distance may be mapped to a learnable scalar, and used as a bias term in the self-attention module. Additionally, $\tilde{Q}_r$, $\tilde{K}_r$ and $\tilde{V}_r$ in the equation described above are as follows:

$$\begin{cases} \tilde{Q}_r = \hat{T}_r \tilde{W}_{Qr} \\ \tilde{K}_r = \hat{T}_r \tilde{W}_{Kr} \\ \tilde{V}_r = \hat{T}_r \tilde{W}_{Vr} \end{cases} \tag{15}$$

where the matrix $\tilde{W}_{Qr} \in \mathbb{R}^{d\times d}$, the matrix $\tilde{W}_{Kr} \in \mathbb{R}^{d\times d}$ and the matrix $\tilde{W}_{Vr} \in \mathbb{R}^{d\times d}$ are learnable variables.

The learned matrix may be denoted as $\tilde{T}_r = [\tilde{t}_r^{\Phi_0}, \tilde{t}_r^{\Phi_1}, \ldots, \tilde{t}_r^{\Phi_O}]^T$. $\tilde{t}_r^{\Phi_o}$ represents the current global embedding of the meta-path Φ_o obtained through the inter-meta-path encoder. Through the inter-meta-path encoder, information may be conveyed across different meta-paths, enabling the encoding of global heterogeneity across different meta-paths.

For example, a current global embedding 956 and a current global embedding 958 may be generated based on the intermediate global embedding 942 and the intermediate global embedding 952 through a self-attention-mechanism-based inter-meta-path encoder 954.

Subsequently, for each node in a meta-path, a current local embedding and a current textual

embedding of the node may be generated, through a self-attention mechanism, based on a current global embedding corresponding to the meta-path, an intermediate local embedding and a previous textual embedding of the node. Given a node $n_i^{\Phi_o}$, a current global embedding $\tilde{t}_r^{\Phi_o}$ corresponding to a meta-path Φ_o, an intermediate local embedding $\hat{t}_{ia}^{\Phi_o}$ and a previous textual embedding $\{t_{i1}^{\Phi_o}, t_{i2}^{\Phi_o}, \ldots, t_{iv}^{\Phi_o}\}$ of the node $n_i^{\Phi_o}$ may be combined to form a node matrix: $\overline{T}_i^{\Phi_o} = [\tilde{t}_r^{\Phi_o}, \hat{t}_{ia}^{\Phi_o}, t_{i1}^{\Phi_o}, t_{i2}^{\Phi_o}, \ldots, t_{iv}^{\Phi_o}]^T$, where v is the number of textual tokens of the node $n_i^{\Phi_o}$.

A multi-head self-attention operation may be employed for the node matrix $\overline{T}_i^{\Phi_o}$. This multi-head self-attention operation may be similar to Equation 12. The difference is that in order to ensure that different nodes on the same meta-path can share the same global embedding, the global embedding may remain unchanged after the multi-head self-attention operation.

For example, for the query node in the first meta-path, a current local embedding 962 and a current textual embeddings 964 and 966 of the query node may be generated, through a self-attention-mechanism-based node encoder 960, based on the intermediate local embedding 940, the previous textual embeddings 926 and 928 and the current global embedding 956. To ensure that different nodes on the same meta-path can share the same global embedding, the global embedding may remain unchanged after passing through the node encoder. Similarly, a current local embedding 970 and a current textual embeddings 972 and 974 of the title node may be generated through the self-attention-mechanism-based node encoder 968. Additionally, for the query node in the second meta-path, a current local embedding 978 and a current textual embeddings 980 and 982 of the query node may be generated through a self-attention-mechanism-based node encoder 976. A current local embedding 984 and a current textual embeddings 986 and 988 of the entity node may be generated through a self-attention-mechanism-based node encoder 982. The node encoder 960, the node encoder 968, the node encoder 976 and the node encoder 982 may correspond to the node encoder 642-3 in FIG.6.

Finally, a current global embedding corresponding to a meta-path and a plurality of current local embeddings and a plurality of current textual embeddings corresponding to a plurality of nodes in the meta-path may be combined into a current embedding sequence corresponding to the meta-path. For example, for the first meta path, the current global embedding 956, the current local embedding 962 and the current textual embeddings 964 and 966 corresponding to the query node, and the current local embedding 970 and the current textual embeddings 972 and 974 corresponding to the title node are combined into a current embedding sequence 904.

In the process described above, each token can pay attention to other tokens in the same meta-path via a local token [TRA], thereby implementing local heterogeneity. Additionally, each token can pay attention to other tokens in other meta-paths via a global token [TER], thereby

implementing global heterogeneity. Through the proposed local token [TRA] and global token [TER] according to the embodiments of the present disclosure, information within a same meta-path and information across different meta-paths may be fused together. In such way, hierarchical heterogeneity can be deeply integrated into semantic representations of nodes. Additionally, each textual token only needs to pay attention to other tokens within the same node. This may significantly reduce the number of learnable parameters and accelerate model training.

A current embedding sequence corresponding to each meta-path obtained through the last graph encoding operation may be regarded as an embedding sequence corresponding to the meta-path. A query representation of the query may be generated based on the embedding sequence. For example, a local embedding for the local token [TRA] of the query may be extracted from the embedding sequence. A query representation may be generated based on the extracted local embedding. When the query node belongs to only one meta-path, one embedding sequence corresponding to the meta-path may be obtained. The local embedding for the local token [TRA] extracted from this embedding sequence may be regarded as the query representation of the query. When the query node belongs to a plurality of meta-paths, a plurality of embedding sequences corresponding to the plurality of meta-paths may be obtained. A local embedding for the local token [TRA] may be extracted from each embedding sequence, thereby obtaining a plurality of local embeddings. The query representation may be generated based on the plurality of local embeddings.

FIG.10 illustrates an exemplary process 1000 for generating a query representation based on an embedding sequence according to an embodiment of the present disclosure. The process 1000 may be performed by the pooling module 644 in FIG.6. In the process 1000, a query node belongs to two meta-paths, i.e., a first meta-path and a second meta-path. As an example, the type of the first meta-path is query-title QT. There is a query node and a title node on this meta-path. The type of the second meta-path is query-entity QE. There is a query node and an entity node on this meta-path. It should be appreciated that when the query node belongs to other types of meta-paths, or belongs to more meta-paths, a query representation may be generated through a process similar to the process 1000.

An embedding sequence 1002 and an embedding sequence 1004 correspond to the first meta-path and the second meta-path, respectively. The embedding sequence 1002 and the embedding sequence 1004 may be obtained through the last graph encoder 642 in FIG.6. In other words, the embedding sequence 1002 and the embedding sequence 1004 may be obtained through performing the process 900 in FIG.9 for the last time. A local embedding 1006 for a local token [TRA] of the query node may be extracted from the embedding sequence 1002. Similarly, a local embedding 1008 for a local token [TRA] of the query node may be extracted from the embedding

sequence 1004. The local embedding 1006 and the local embedding 1008 are provided to a pooling module 1010. The pooling module 1010 may correspond to the pooling module 644 in FIG.6. A query representation 1012 may be generated by performing a pooling operation on the local embedding 1006 and the local embedding 1008 through the pooling module 1010. It should be appreciated that generating the query representation through the pooling operation is only an example of generating the query representation, and the query representation may also be generated through other operations. A keyword representation may be generated through a process similar to the process for generating the query representation.

Referring back to FIG.6, the heterogeneous graph learning-based relevance prediction model 610 described above can predict the relevance 612 between the query 602 and the keyword 604. The heterogeneous graph learning-based relevance prediction model 610 includes the graph encoders 642 and 662. The graph encoders 642 or 662 can learn the heterogeneous graph 622 through mining complex associations among node semantic information and a graph topology, and can encode the learned text semantics and structure characteristics into the query representation 646 or the keyword representation 666, thereby generating the augmented query representation 646 or the augmented keyword representation 666. The graph encoders 642 and 662 may aggregate information about neighbor nodes around a node into a representation of that node based on the topology of the heterogeneous graph 622. Each graph encoder may include an intra-meta-path encoder, an inter-meta-path encoder, a node encoder, etc. The intra-meta-path encoder enables to convey information between nodes belonging to the same meta-path. The inter-meta-path encoder enables to convey information between nodes across different meta-paths. The node encoder can aggregate information from a same meta-path and information from different meta-paths into a representation of a corresponding node, e.g., into the query representation 646 of the query node or the keyword representation 666 of the keyword node, thereby generating the augmented query representation or the augmented keyword representation. Predicting the relevance between the query 602 and the keyword 604 with the query representation 646 and the keyword representation 666 generated in such manner can significantly improve the relevance prediction result.

It should be appreciated that the process for relevance prediction based on heterogeneous graph learning described above in conjunction with FIG.6 to FIG.10 is merely exemplary. Depending on actual application requirements, the steps in the process for relevance prediction based on heterogeneous graph learning may be replaced or modified in any manner, and the process may include more or fewer steps. Additionally, the specific order or hierarchy of the steps in the process 600 to process 1000 is only exemplary, and the process for relevance prediction based on heterogeneous graph learning may be performed in an order different from the described order. Additionally, the heterogeneous graph learning-based relevance prediction model 610 shown in

FIG.6 is only an example of the heterogeneous graph learning-based relevance prediction model. Depending on actual application requirements, the heterogeneous graph learning-based relevance prediction model may have any other structure and may include more or fewer layers.

A heterogeneous graph learning-based relevance prediction model according to the embodiments of the present disclosure, e.g., the heterogeneous graph learning-based relevance prediction model 610 in FIG.6, may generate a query representation and a keyword representation through fusing rich context information related to a query and a keyword. Since the heterogeneous graph learning-based relevance prediction model includes multiple layers of graph encoders, there may be large prediction latency when performing relevance prediction, especially for a new query. A Knowledge Distillation strategy may be employed to treat the heterogeneous graph learning-based relevance prediction model as a teacher model, and learn, through the teacher model, a student model, e.g., a lightweight relevance prediction model. A large number of query-keyword pairs may be sampled from a search log and fed to the teacher model. Each query-keyword pair includes a query and a keyword. The teacher model may predict a relevance between the query and the keyword. The predicted relevance may be used as pseudo-label, which may be used with corresponding query and keyword to generate a training sample for training the lightweight relevance prediction model. The trained lightweight relevance prediction model may be deployed to perform the relevance prediction task online.

FIG.11 illustrates an exemplary process 1100 for performing relevance prediction through a lightweight relevance prediction model according to an embodiment of the present disclosure. In the process 1100, a relevance 1112 between a query 1102 and a keyword 1104 is predicted through a lightweight relevance prediction model 1110. The lightweight relevance prediction model 1110 includes an embedding layer 1120 and N ($N \geq 1$) transformer layers 1130 for the query 1102, an embedding layer 1140 and N transformer layers 1150 for the keyword 1104, and a prediction layer 1160. It should be appreciated that the lightweight relevance prediction model 1110 shown in FIG.11 is only an example of the lightweight relevance prediction model that can be deployed to perform a relevance prediction task online. Depending on actual application requirements, the lightweight relevance prediction model may have any other structure and may include more or fewer layers.

FIG.12 is a flowchart of an exemplary method 1200 for relevance prediction based on heterogeneous graph learning according to an embodiment of the present disclosure.

At 1210, a query and a keyword may be received.

At 1220, a heterogeneous graph corresponding to the query and the keyword may be generated.

At 1230, query graph structure information for the query may be inferred according to the heterogeneous graph.

At 1240, a query representation of the query may be generated based on the query graph structure information.

At 1250, keyword graph structure information for the keyword may be inferred according to the heterogeneous graph.

At 1260, a keyword representation of the keyword may be generated based on the keyword graph structure information.

At 1270, a relevance between the query and the keyword may be predicted based on the query representation and the keyword representation.

In an implementation, the keyword may be a keyword for a candidate content item. The candidate content item may include at least one of news, video, movie, book, music, web page, and product information.

In an implementation, the generating a heterogeneous graph may comprise: constructing an original heterogeneous graph based on a search log and/or a knowledge graph; generating an augmented heterogeneous graph based on semantic information and/or a topology of a node set of the original heterogeneous graph; and pruning the augmented heterogeneous graph based on semantic information and/or a topology of a node set of the augmented heterogeneous graph, to obtain the heterogeneous graph.

In an implementation, the constructing an original heterogeneous graph may comprise: extracting a component set from the search log and/or the knowledge graph; setting the component set as a node set; determining an edge set among the node set; and combining the node set and the edge set into the original heterogeneous graph.

The component set may include: at least one of a historical query set, a historical keyword set and a historical title set extracted from the search log; and/or an entity set corresponding to the historical query set and/or an entity set corresponding to the historical keyword set extracted from the knowledge graph.

The determining an edge set may comprise, for every two nodes in the node set: determining whether a relation exists between two components corresponding to the two nodes; and in response to determining that a relation exists between the two components, determining that an edge exists between the two nodes.

The relation may comprise at least one of: query-keyword click relation, query-title click relation, query-entity extraction relation, keyword-entity extraction relation, and keyword-keyword co-occurrence relation.

The original heterogeneous graph may comprise a plurality of original partial graphs corresponding to a plurality of relation types. The generating an augmented heterogeneous graph may comprise: for an original partial graph corresponding to each relation type, generating an

augmented partial graph corresponding to the relation type based on semantic information and/or topology of a node subset of the original partial graph; and combining a plurality of augmented partial graphs corresponding to the plurality of relation types into the augmented heterogeneous graph.

The generating an augmented partial graph may comprise: generating a semantic partial graph corresponding to the relation type based on semantic similarity between two nodes in the node subset; generating a propagation partial graph corresponding to the relation type based on semantic similarity between two nodes of the same type in the node subset and a topology of the original partial graph; and generating the augmented partial graph corresponding to the relation type based on the original partial graph, the semantic partial graph, and the propagation partial graph.

A node set of the augmented heterogeneous graph may include a plurality of center nodes. Each center node has a plurality of neighbor node sets based on a plurality of meta-path types. The pruning the augmented heterogeneous graph may comprise, for each neighbor node set in the plurality of neighbor node sets: identifying a neighbor node subset that are most important to the center node from the neighbor node set; determining neighbor nodes in the neighbor node set that are outside the neighbor node subset; and removing edges between the determined neighbor nodes and the center node.

In an implementation, the inferring query graph structure information may comprise: inferring, according to the heterogeneous graph, a meta-path to which a query node corresponding to the query belongs; inferring, according to the heterogeneous graph, a node set which is based on the meta-path; and generating the query graph structure information based on the meta-path and the node set.

In an implementation, the generating a query representation may comprise: obtaining, from the query graph structure information, a meta-path to which a query node corresponding to the query belongs and a neighbor node which is based on the meta-path; generating a token sequence corresponding to the meta-path based on the query and the neighbor node; generating an initial embedding sequence corresponding to the meta-path based on the token sequence; generating, based on the initial embedding sequence, an embedding sequence corresponding to the meta-path through iteratively performing a graph encoding operation; and generating the query representation based on the embedding sequence.

The generating a token sequence may comprise: obtaining a query textual token of the query and a neighbor textual token of the neighbor node through a tokenization operation; adding a query local token for the query before the query textual token; adding a neighbor local token for the neighbor node before the neighbor textual token; and combining a global token of the meta-path, the query local token, the query textual token, the neighbor local token, and the neighbor textual

token into the token sequence corresponding to the meta-path.

The graph encoding operation may comprise: generating, through a self-attention mechanism, an intermediate global embedding of a global token associated with the meta-path and a plurality of intermediate local embeddings of a plurality of local tokens associated with the meta-path based on a previous global embedding of the global token and a plurality of previous local embeddings of the plurality of local tokens; generating, through a self-attention mechanism, a current global embedding based on the intermediate global embedding; for each node in the meta-path, generating, through a self-attention mechanism, a current local embedding and a current textual embedding of the node based on the current global embedding and an intermediate local embedding and a previous textual embedding of the node; and combining the current global embedding and a plurality of current local embeddings and a plurality of current textual embeddings of a plurality of nodes in the meta-path into a current embedding sequence corresponding to the meta-path.

The generating the query representation based on the embedding sequence may comprise: extracting a local embedding of the query from the embedding sequence; and generating the query representation based on the local embedding.

In an implementation, the method 1200 may further comprise: generating a training sample for training a lightweight relevance prediction model based on the query, the keyword, and the relevance.

It should be appreciated that the method 1200 may further comprise any step/process for relevance prediction based on heterogeneous graph learning according to the embodiments of the present disclosure as mentioned above.

FIG.13 illustrates an exemplary apparatus 1300 for relevance prediction based on heterogeneous graph learning according to an embodiment of the present disclosure.

The apparatus 1300 may comprise: a query and keyword receiving module 1310, for receiving a query and a keyword; a heterogeneous graph generating module 1320, for generating a heterogeneous graph corresponding to the query and the keyword; a query graph structure information inferring module 1330, for inferring query graph structure information for the query according to the heterogeneous graph; a query representation generating module 1340, for generating a query representation of the query based on the query graph structure information; a keyword graph structure information inferring module 1350, for inferring keyword graph structure information for the keyword according to the heterogeneous graph; a keyword representation generating module 1360, for generating a keyword representation of the keyword based on the keyword graph structure information; and a relevance predicting module 1370, for predicting a relevance between the query and the keyword based on the query representation and the keyword

representation. Moreover, the apparatus 1300 may further comprise any other modules configured for relevance prediction based on heterogeneous graph learning according to the embodiments of the present disclosure as mentioned above.

FIG.14 illustrates an exemplary apparatus 1400 for relevance prediction based on heterogeneous graph learning according to an embodiment of the present disclosure.

The apparatus 1400 may comprise a processor 1410; and a memory 1420 storing computer-executable instructions. The computer-executable instructions, when executed, may cause the processor 1410 to: receive a query and a keyword, generate a heterogeneous graph corresponding to the query and the keyword, infer query graph structure information for the query according to the heterogeneous graph, generate a query representation of the query based on the query graph structure information, infer keyword graph structure information for the keyword according to the heterogeneous graph, generate a keyword representation of the keyword based on the keyword graph structure information, and predict a relevance between the query and the keyword based on the query representation and the keyword representation.

In an implementation, the generating a heterogeneous graph may comprise: constructing an original heterogeneous graph based on a search log and/or a knowledge graph; generating an augmented heterogeneous graph based on semantic information and/or a topology of a node set of the original heterogeneous graph; and pruning the augmented heterogeneous graph based on semantic information and/or a topology of a node set of the augmented heterogeneous graph, to obtain the heterogeneous graph.

In an implementation, the generating a query representation may comprise: obtaining, from the query graph structure information, a meta-path to which a query node corresponding to the query belongs and a neighbor node which is based on the meta-path; generating a token sequence corresponding to the meta-path based on the query and the neighbor node; generating an initial embedding sequence corresponding to the meta-path based on the token sequence; generating, based on the initial embedding sequence, an embedding sequence corresponding to the meta-path through iteratively performing a graph encoding operation; and generating the query representation based on the embedding sequence.

It should be appreciated that the processor 1410 may further perform any other step/process for relevance prediction based on heterogeneous graph learning according to the embodiments of the present disclosure as mentioned above.

The embodiments of the present disclosure propose a computer program product for relevance prediction based on heterogeneous graph learning, comprising a computer program that is executed by at least one processor for: receiving a query and a keyword; generating a heterogeneous graph corresponding to the query and the keyword; inferring query graph structure

information for the query according to the heterogeneous graph; generating a query representation of the query based on the query graph structure information; inferring keyword graph structure information for the keyword according to the heterogeneous graph; generating a keyword representation of the keyword based on the keyword graph structure information; and predicting a relevance between the query and the keyword based on the query representation and the keyword representation. Additionally, the computer program may further be performed for implementing any other steps/processes for relevance prediction based on heterogeneous graph learning according to the embodiments of the present disclosure as mentioned above.

The embodiments of the present disclosure may be embodied in a non-transitory computer-readable medium. The non-transitory computer readable medium may comprise instructions that, when executed, cause one or more processors to perform any operation of the method for relevance prediction based on heterogeneous graph learning according to the embodiments of the present disclosure as mentioned above.

It should be appreciated that all the operations in the methods described above are merely exemplary, and the present disclosure is not limited to any operations in the methods or sequence orders of these operations, and should cover all other equivalents under the same or similar concepts. In addition, the articles "a" and "an" as used in this specification and the appended claims should generally be construed to mean "one" or "one or more" unless specified otherwise or clear from the context to be directed to a singular form.

It should also be appreciated that all the modules in the apparatuses described above may be implemented in various approaches. These modules may be implemented as hardware, software, or a combination thereof. Moreover, any of these modules may be further functionally divided into sub-modules or combined together.

Processors have been described in connection with various apparatuses and methods. These processors may be implemented using electronic hardware, computer software, or any combination thereof. Whether such processors are implemented as hardware or software will depend upon the particular application and overall design constraints imposed on the system. By way of example, a processor, any portion of a processor, or any combination of processors presented in the present disclosure may be implemented with a microprocessor, microcontroller, digital signal processor (DSP), a field-programmable gate array (FPGA), a programmable logic device (PLD), a state machine, gated logic, discrete hardware circuits, and other suitable processing components configured for performing the various functions described throughout the present disclosure. The functionality of a processor, any portion of a processor, or any combination of processors presented in the present disclosure may be implemented with software being executed by a microprocessor, microcontroller, DSP, or other suitable platform.

Software shall be construed broadly to mean instructions, instruction sets, code, code segments, program code, programs, subprograms, software modules, applications, software applications, software packages, routines, subroutines, objects, threads of execution, procedures, functions, etc. The software may reside on a computer-readable medium. A computer-readable medium may include, by way of example, memory such as a magnetic storage device (e.g., hard disk, floppy disk, magnetic strip), an optical disk, a smart card, a flash memory device, random access memory (RAM), read only memory (ROM), programmable ROM (PROM), erasable PROM (EPROM), electrically erasable PROM (EEPROM), a register, or a removable disk. Although memory is shown separate from the processors in the various aspects presented throughout the present disclosure, the memory may be internal to the processors, e.g., cache or register.

The previous description is provided to enable any person skilled in the art to practice the various aspects described herein. Various modifications to these aspects will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other aspects. Thus, the claims are not intended to be limited to the aspects shown herein. All structural and functional equivalents to the elements of the various aspects described throughout the present disclosure that are known or later come to be known to those of ordinary skilled in the art are expressly incorporated herein and intended to be encompassed by the claims.

CLAIMS

1. A method for relevance prediction based on heterogeneous graph learning, comprising:

receiving a query and a keyword;

generating a heterogeneous graph corresponding to the query and the keyword;

inferring query graph structure information for the query according to the heterogeneous graph;

generating a query representation of the query based on the query graph structure information;

inferring keyword graph structure information for the keyword according to the heterogeneous graph;

generating a keyword representation of the keyword based on the keyword graph structure information; and

predicting a relevance between the query and the keyword based on the query representation and the keyword representation.

2. The method of claim 1, wherein the generating a heterogeneous graph comprises:

constructing an original heterogeneous graph based on a search log and/or a knowledge graph;

generating an augmented heterogeneous graph based on semantic information and/or a topology of a node set of the original heterogeneous graph; and

pruning the augmented heterogeneous graph based on semantic information and/or a topology of a node set of the augmented heterogeneous graph, to obtain the heterogeneous graph.

3. The method of claim 2, wherein the constructing an original heterogeneous graph comprises:

extracting a component set from the search log and/or the knowledge graph;

setting the component set as a node set;

determining an edge set among the node set; and

combining the node set and the edge set into the original heterogeneous graph.

4. The method of claim 3, wherein the determining an edge set comprises, for every two nodes in the node set:

determining whether a relation exists between two components corresponding to the two nodes; and

in response to determining that a relation exists between the two components, determining that an edge exists between the two nodes.

5. The method of claim 4, wherein the relation comprises at least one of: query-keyword click relation, query-title click relation, query-entity extraction relation, keyword-entity extraction

relation, and keyword-keyword co-occurrence relation.

6. The method of claim 2, wherein the original heterogeneous graph comprises a plurality of original partial graphs corresponding to a plurality of relation types, and the generating an augmented heterogeneous graph comprises:

for an original partial graph corresponding to each relation type, generating an augmented partial graph corresponding to the relation type based on semantic information and/or topology of a node subset of the original partial graph; and

combining a plurality of augmented partial graphs corresponding to the plurality of relation types into the augmented heterogeneous graph.

7. The method of claim 6, wherein the generating an augmented partial graph comprises:

generating a semantic partial graph corresponding to the relation type based on semantic similarity between two nodes in the node subset;

generating a propagation partial graph corresponding to the relation type based on semantic similarity between two nodes of the same type in the node subset and a topology of the original partial graph; and

generating the augmented partial graph corresponding to the relation type based on the original partial graph, the semantic partial graph, and the propagation partial graph.

8. The method of claim 2, wherein a node set of the augmented heterogeneous graph includes a plurality of center nodes, each center node having a plurality of neighbor node sets based on a plurality of meta-path types, and the pruning the augmented heterogeneous graph comprises, for each neighbor node set in the plurality of neighbor node sets:

identifying a neighbor node subset that are most important to the center node from the neighbor node set;

determining neighbor nodes in the neighbor node set that are outside the neighbor node subset; and

removing edges between the determined neighbor nodes and the center node.

9. The method of claim 1, wherein the inferring query graph structure information comprises:

inferring, according to the heterogeneous graph, a meta-path to which a query node corresponding to the query belongs;

inferring, according to the heterogeneous graph, a node set which is based on the meta-path; and

generating the query graph structure information based on the meta-path and the node set.

10. The method of claim 1, wherein the generating a query representation comprises:

obtaining, from the query graph structure information, a meta-path to which a query node corresponding to the query belongs and a neighbor node which is based on the meta-path;

generating a token sequence corresponding to the meta-path based on the query and the neighbor node;

generating an initial embedding sequence corresponding to the meta-path based on the token sequence;

generating, based on the initial embedding sequence, an embedding sequence corresponding to the meta-path through iteratively performing a graph encoding operation; and

generating the query representation based on the embedding sequence.

11. The method of claim 10, wherein the generating a token sequence comprises:

obtaining a query textual token of the query and a neighbor textual token of the neighbor node through a tokenization operation;

adding a query local token for the query before the query textual token;

adding a neighbor local token for the neighbor node before the neighbor textual token; and

combining a global token of the meta-path, the query local token, the query textual token, the neighbor local token, and the neighbor textual token into the token sequence corresponding to the meta-path.

12. The method of claim 10, wherein the graph encoding operation comprises:

generating, through a self-attention mechanism, an intermediate global embedding of a global token associated with the meta-path and a plurality of intermediate local embeddings of a plurality of local tokens associated with the meta-path based on a previous global embedding of the global token and a plurality of previous local embeddings of the plurality of local tokens;

generating, through a self-attention mechanism, a current global embedding based on the intermediate global embedding;

for each node in the meta-path, generating, through a self-attention mechanism, a current local embedding and a current textual embedding of the node based on the current global embedding and an intermediate local embedding and a previous textual embedding of the node; and

combining the current global embedding and a plurality of current local embeddings and a plurality of current textual embeddings of a plurality of nodes in the meta-path into a current embedding sequence corresponding to the meta-path.

13. The method of claim 10, wherein the generating the query representation based on the embedding sequence comprises:

extracting a local embedding of the query from the embedding sequence; and

generating the query representation based on the local embedding.

14. An apparatus for relevance prediction based on heterogeneous graph learning, comprising:

a processor; and

a memory storing computer-executable instructions that, when executed, cause the processor to:

receive a query and a keyword,

generate a heterogeneous graph corresponding to the query and the keyword,

infer query graph structure information for the query according to the heterogeneous graph,

generate a query representation of the query based on the query graph structure information,

infer keyword graph structure information for the keyword according to the heterogeneous graph,

generate a keyword representation of the keyword based on the keyword graph structure information, and

predict a relevance between the query and the keyword based on the query representation and the keyword representation.

15. A computer program product for relevance prediction based on heterogeneous graph learning, comprising a computer program that is executed by a processor for:

receiving a query and a keyword;

generating a heterogeneous graph corresponding to the query and the keyword;

inferring query graph structure information for the query according to the heterogeneous graph;

generating a query representation of the query based on the query graph structure information;

inferring keyword graph structure information for the keyword according to the heterogeneous graph;

generating a keyword representation of the keyword based on the keyword graph structure information; and

predicting a relevance between the query and the keyword based on the query representation and the keyword representation.

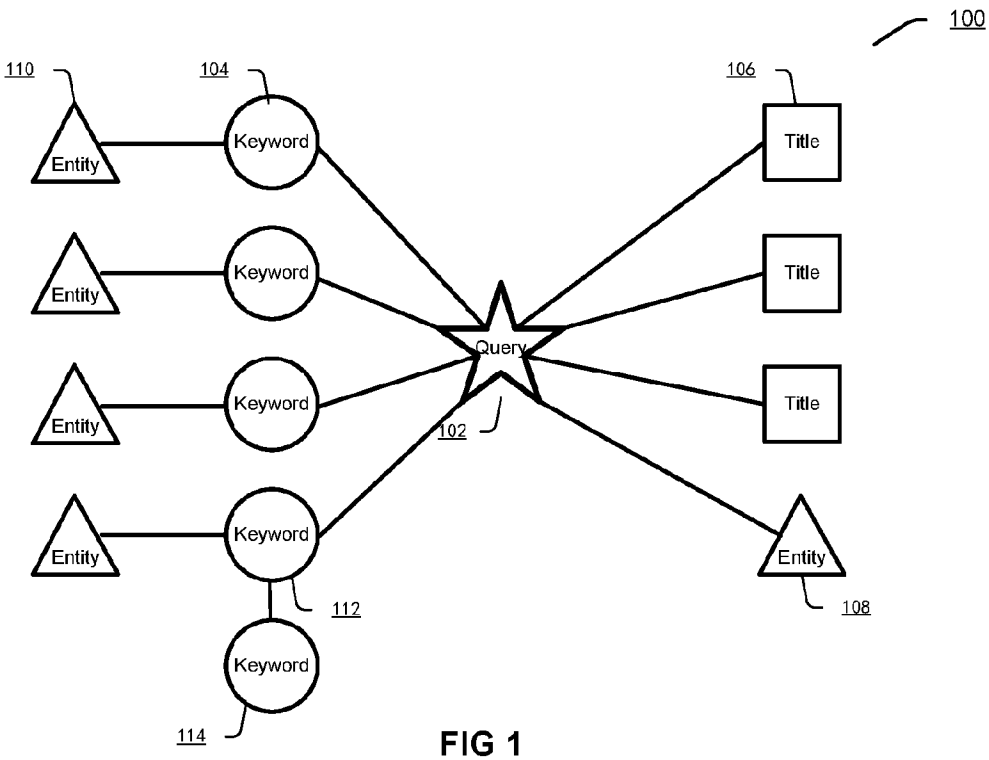
100
110
104
106
Entity
Keyword
Title
Entity
Keyword
Title
Query
Entity
Keyword
Title
102
Entity
Keyword
Entity
112
108
Keyword
114

FIG 1

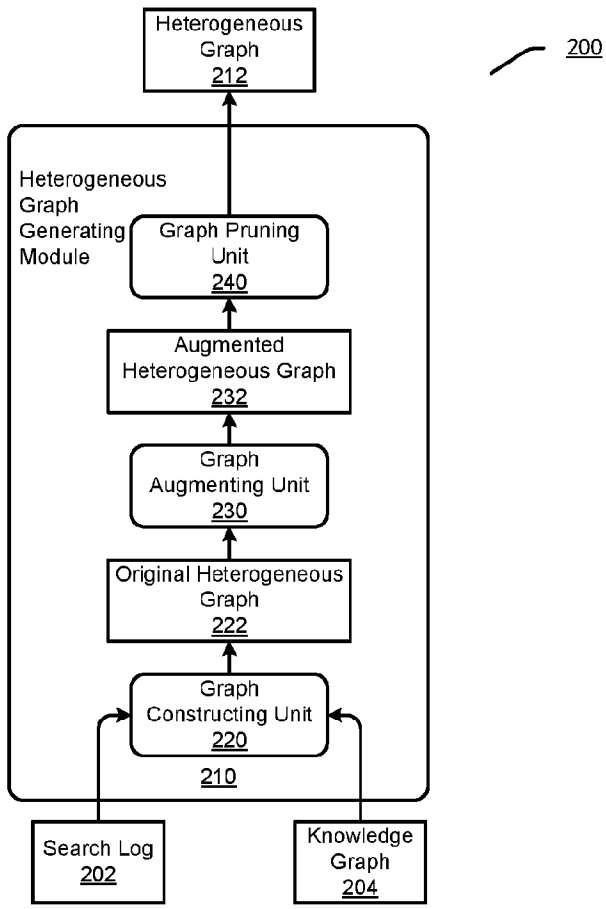
Heterogeneous Graph 212
200
Heterogeneous Graph Generating Module
Graph Pruning Unit 240
Augmented Heterogeneous Graph 232
Graph Augmenting Unit 230
Original Heterogeneous Graph 222
Graph Constructing Unit 220
210
Search Log 202
Knowledge Graph 204

FIG 2

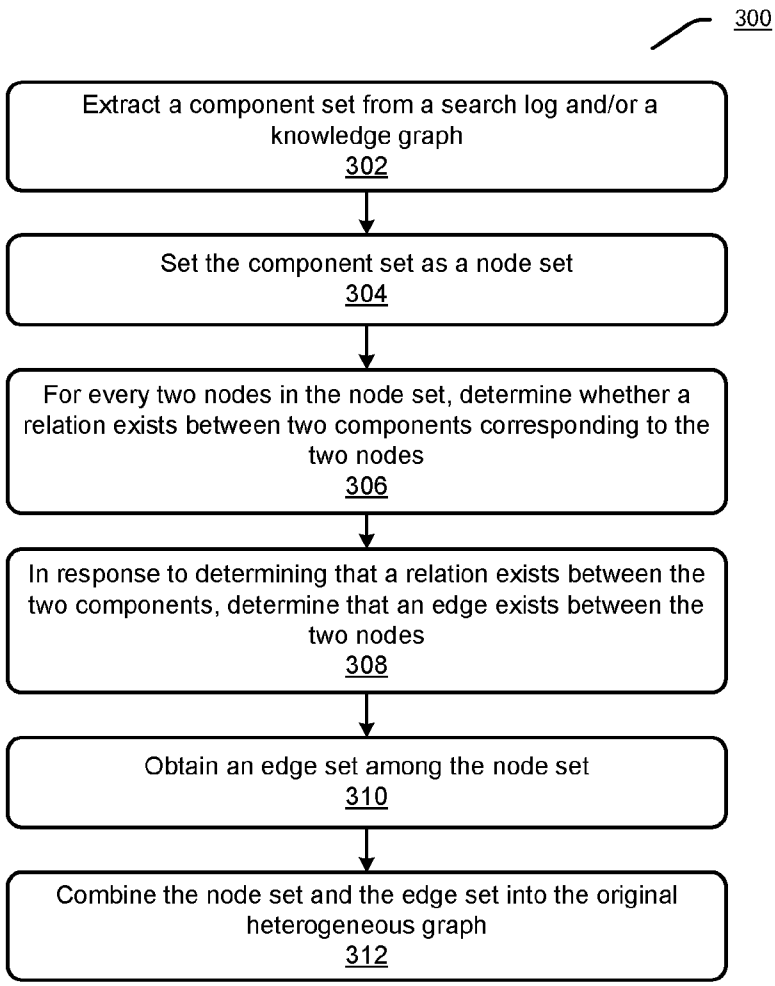
300
Extract a component set from a search log and/or a knowledge graph
302
Set the component set as a node set
304
For every two nodes in the node set, determine whether a relation exists between two components corresponding to the two nodes
306
In response to determining that a relation exists between the two components, determine that an edge exists between the two nodes
308
Obtain an edge set among the node set
310
Combine the node set and the edge set into the original heterogeneous graph
312

FIG 3

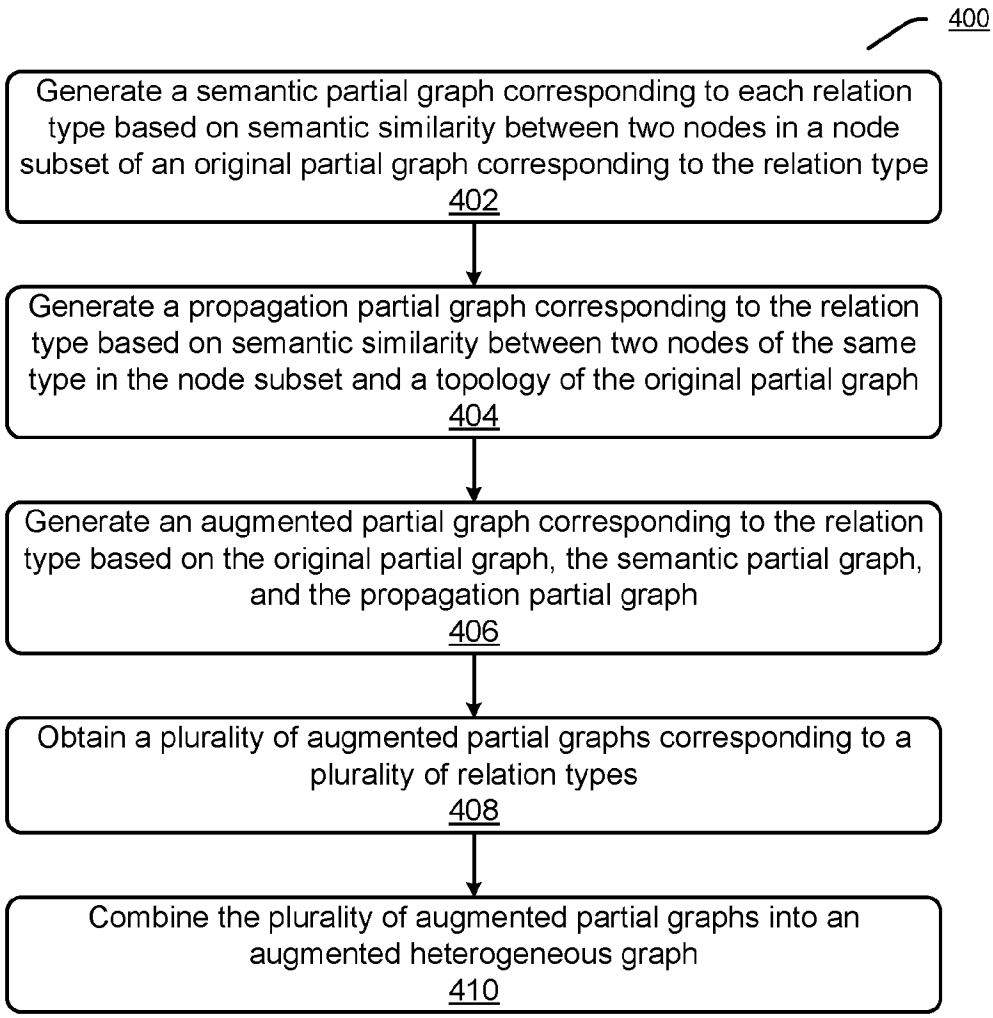
400
Generate a semantic partial graph corresponding to each relation type based on semantic similarity between two nodes in a node subset of an original partial graph corresponding to the relation type
402
Generate a propagation partial graph corresponding to the relation type based on semantic similarity between two nodes of the same type in the node subset and a topology of the original partial graph
404
Generate an augmented partial graph corresponding to the relation type based on the original partial graph, the semantic partial graph, and the propagation partial graph
406
Obtain a plurality of augmented partial graphs corresponding to a plurality of relation types
408
Combine the plurality of augmented partial graphs into an augmented heterogeneous graph
410

FIG 4

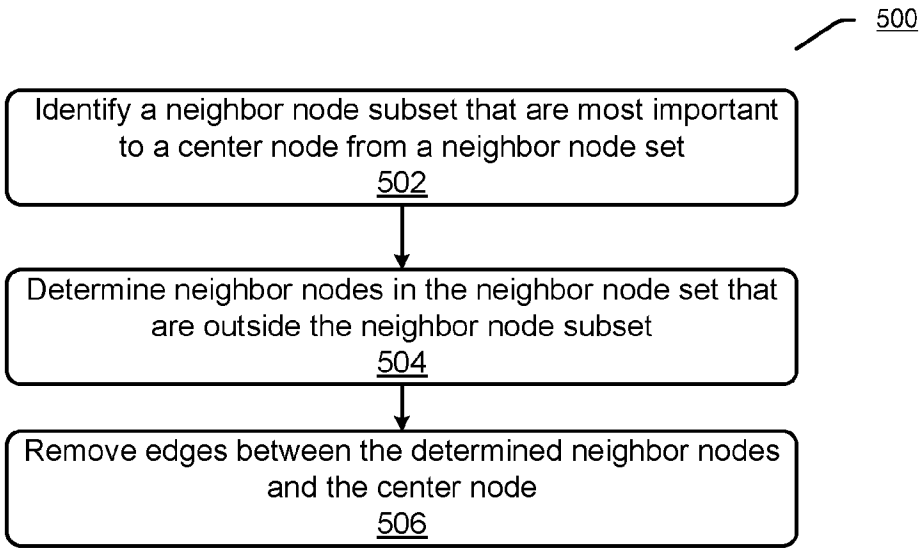
500
Identify a neighbor node subset that are most important to a center node from a neighbor node set
502
Determine neighbor nodes in the neighbor node set that are outside the neighbor node subset
504
Remove edges between the determined neighbor nodes and the center node
506

FIG 5

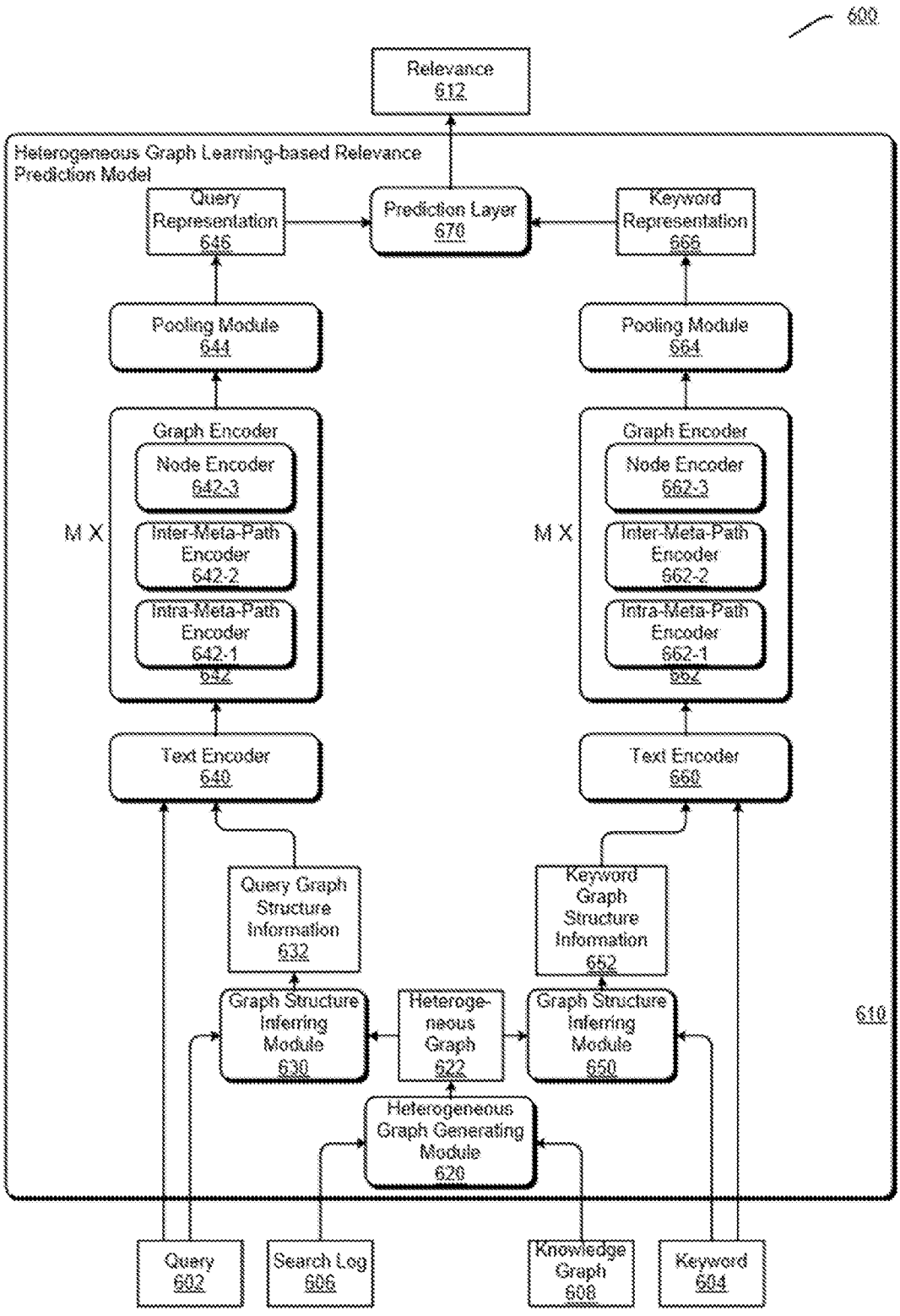
600
Relevance
612
Heterogeneous Graph Learning-based Relevance Prediction Model
Query Representation
646
Prediction Layer
676
Keyword Representation
666
Pooling Module
644
Pooling Module
664
Graph Encoder
Node Encoder
642-3
Inter-Meta-Path Encoder
642-2
Intra-Meta-Path Encoder
642-1
M X
642
Graph Encoder
Node Encoder
662-3
Inter-Meta-Path Encoder
662-2
Intra-Meta-Path Encoder
662-1
M X
662
Text Encoder
640
Text Encoder
660
Query Graph Structure Information
632
Keyword Graph Structure Information
652
Graph Structure Inferring Module
630
Heterogeneous Graph
622
Graph Structure Inferring Module
650
610
Heterogeneous Graph Generating Module
626
Query
602
Search Log
606
Knowledge Graph
608
Keyword
604

FIG 6

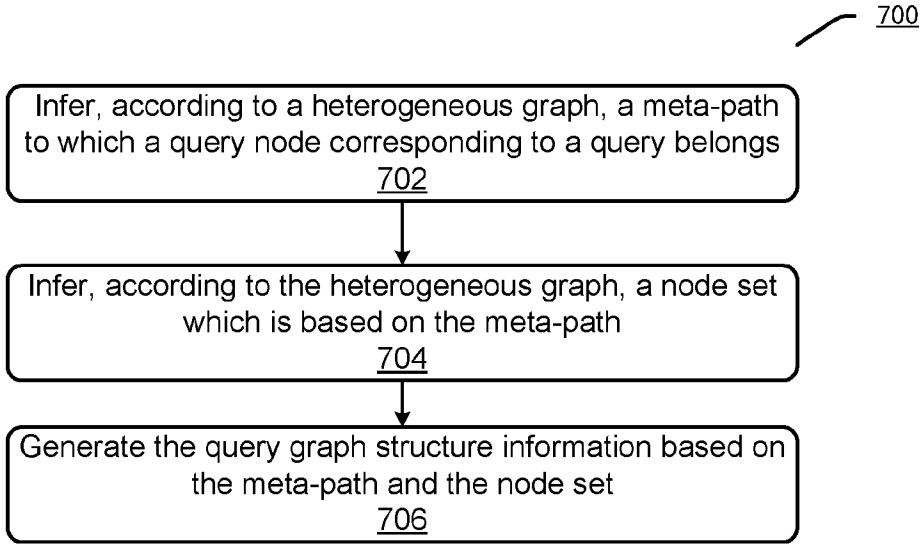
700
Infer, according to a heterogeneous graph, a meta-path to which a query node corresponding to a query belongs
702
Infer, according to the heterogeneous graph, a node set which is based on the meta-path
704
Generate the query graph structure information based on the meta-path and the node set
706

FIG 7

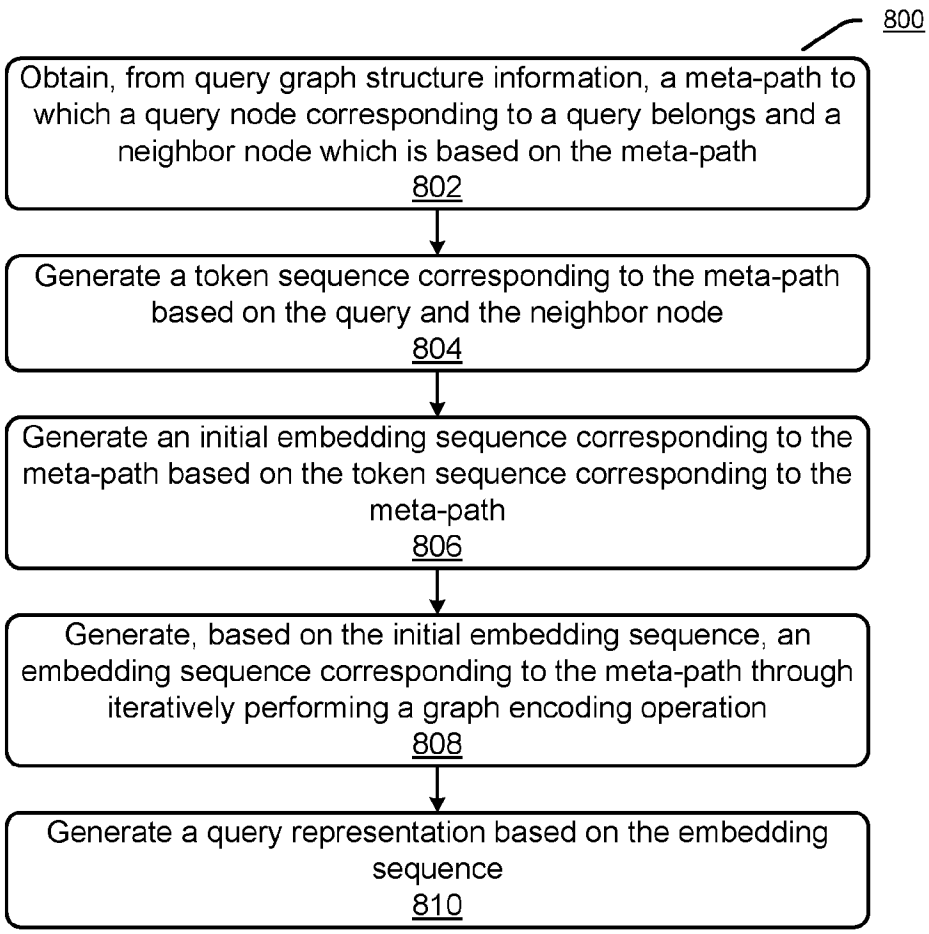
800
Obtain, from query graph structure information, a meta-path to which a query node corresponding to a query belongs and a neighbor node which is based on the meta-path
802
Generate a token sequence corresponding to the meta-path based on the query and the neighbor node
804
Generate an initial embedding sequence corresponding to the meta-path based on the token sequence corresponding to the meta-path
806
Generate, based on the initial embedding sequence, an embedding sequence corresponding to the meta-path through iteratively performing a graph encoding operation
808
Generate a query representation based on the embedding sequence
810

FIG 8

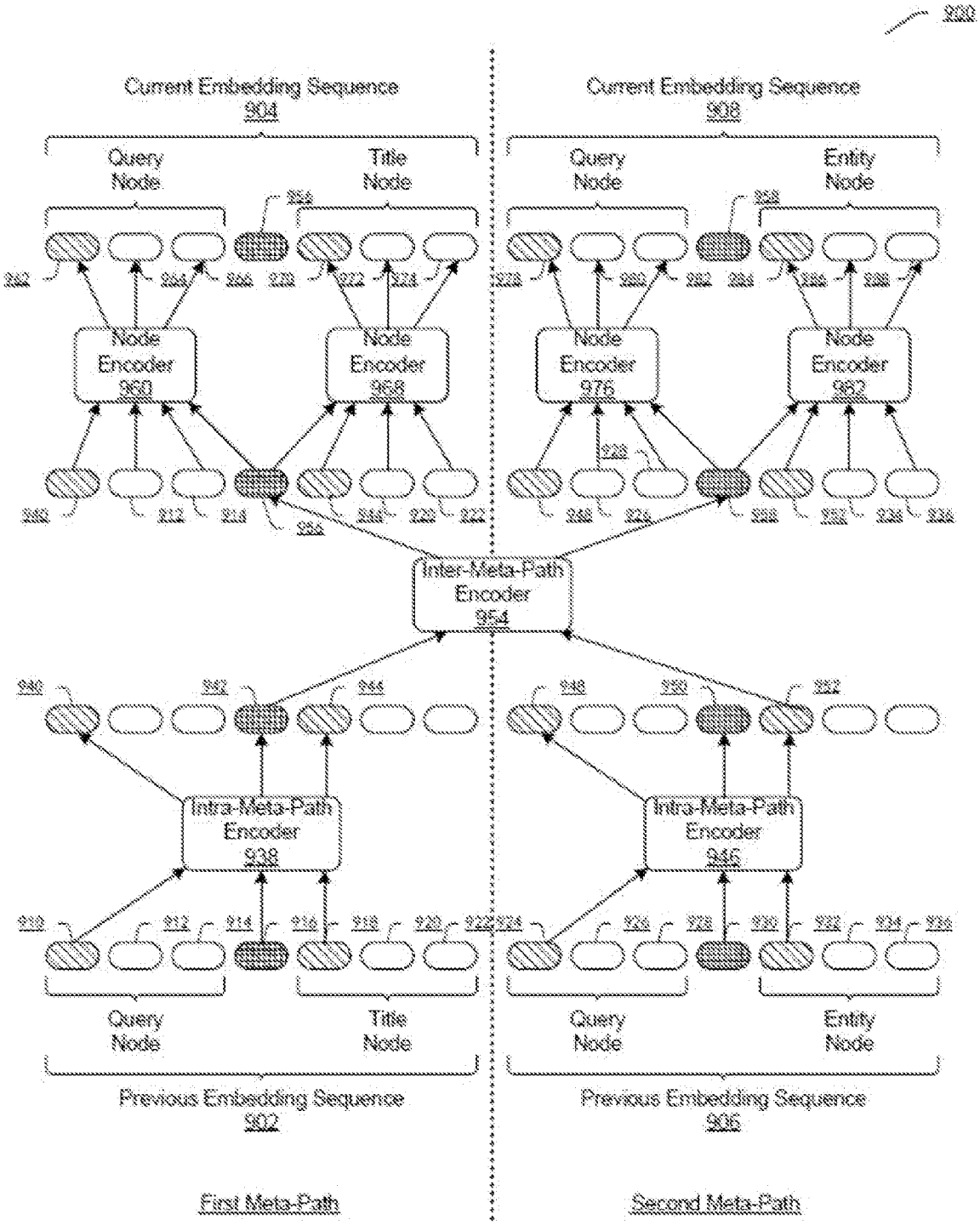
900
Current Embedding Sequence
904
Query Node
Title Node
Node Encoder
960
Node Encoder
968
Current Embedding Sequence
908
Query Node
Entity Node
Node Encoder
976
Node Encoder
982
Inter-Meta-Path Encoder
954
Intra-Meta-Path Encoder
938
Intra-Meta-Path Encoder
946
Query Node
Title Node
Previous Embedding Sequence
902
Query Node
Entity Node
Previous Embedding Sequence
906
First Meta-Path
Second Meta-Path

FIG 9

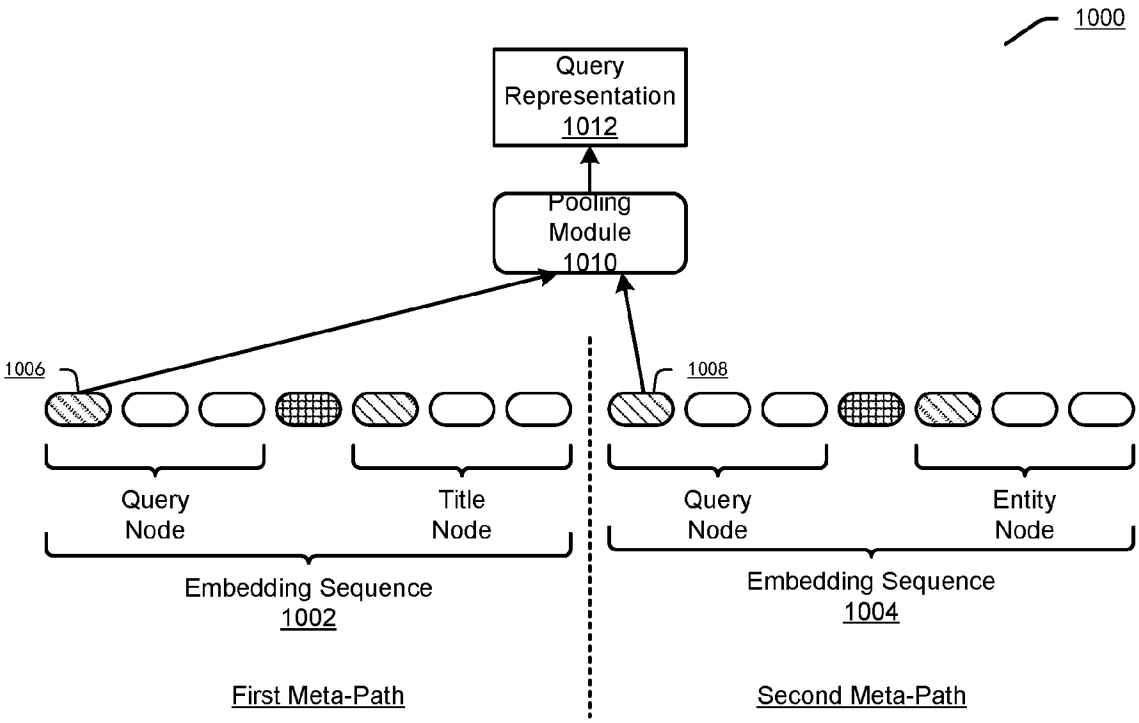
1000
Query Representation 1012
Pooling Module 1010
1006
1008
Query Node
Title Node
Query Node
Entity Node
Embedding Sequence 1002
Embedding Sequence 1004
First Meta-Path
Second Meta-Path

FIG 10

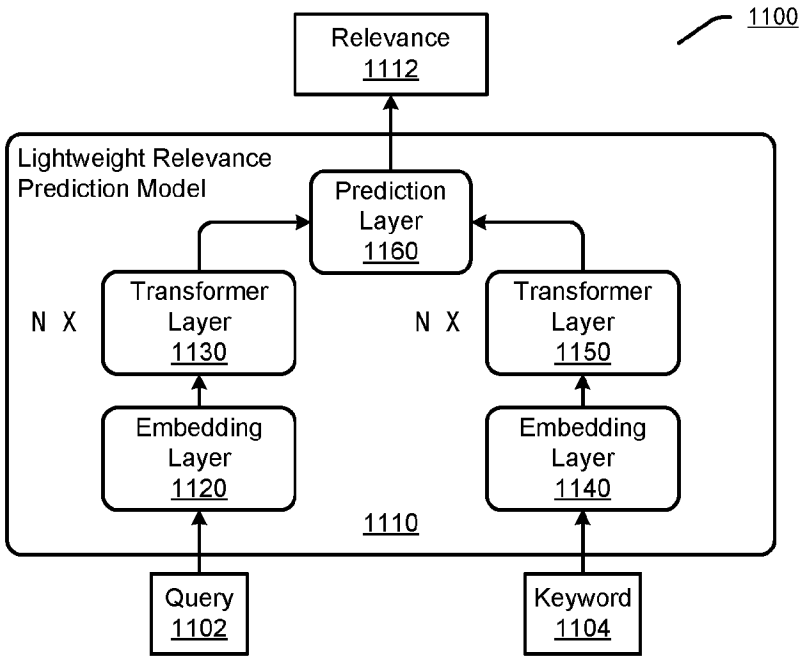
1100
Relevance 1112
Lightweight Relevance Prediction Model
Prediction Layer 1160
N X
Transformer Layer 1130
N X
Transformer Layer 1150
Embedding Layer 1120
Embedding Layer 1140
1110
Query 1102
Keyword 1104

FIG 11

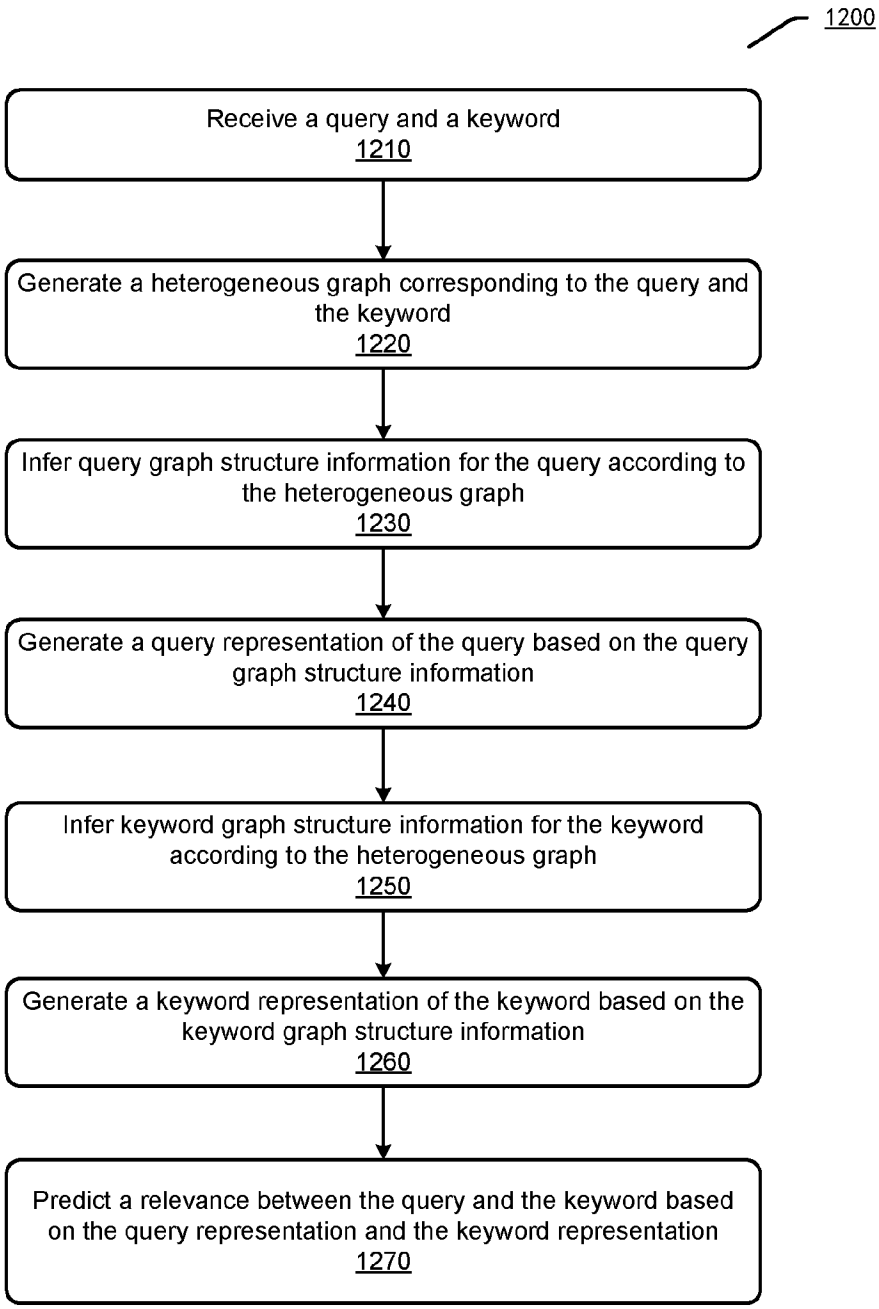
1200
Receive a query and a keyword
1210
Generate a heterogeneous graph corresponding to the query and the keyword
1220
Infer query graph structure information for the query according to the heterogeneous graph
1230
Generate a query representation of the query based on the query graph structure information
1240
Infer keyword graph structure information for the keyword according to the heterogeneous graph
1250
Generate a keyword representation of the keyword based on the keyword graph structure information
1260
Predict a relevance between the query and the keyword based on the query representation and the keyword representation
1270

FIG 12

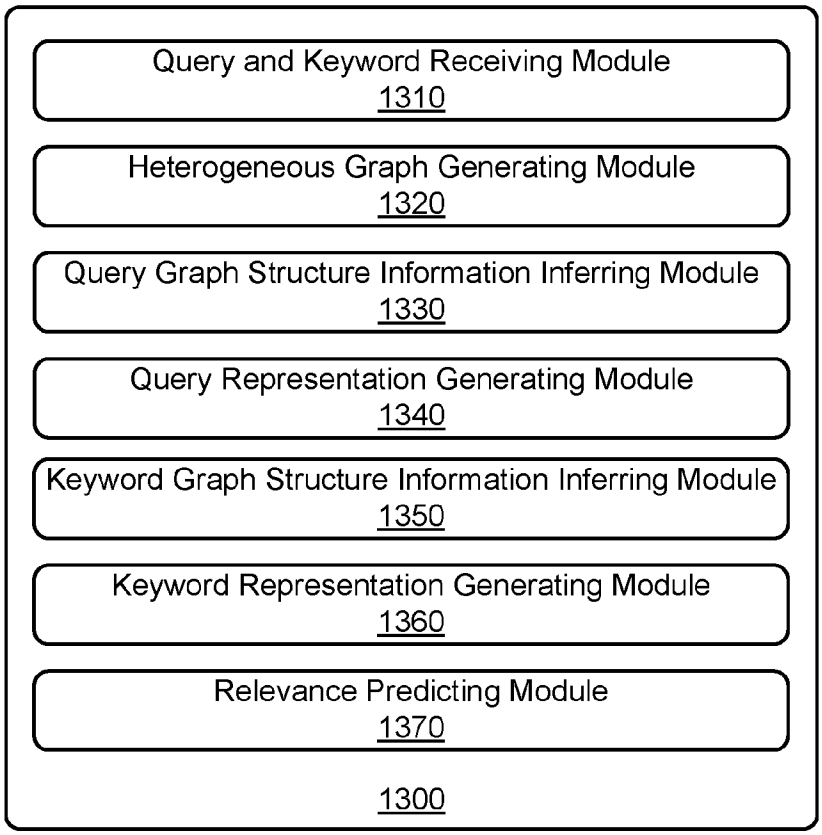
Query and Keyword Receiving Module
1310
Heterogeneous Graph Generating Module
1320
Query Graph Structure Information Inferring Module
1330
Query Representation Generating Module
1340
Keyword Graph Structure Information Inferring Module
1350
Keyword Representation Generating Module
1360
Relevance Predicting Module
1370
1300

FIG 13

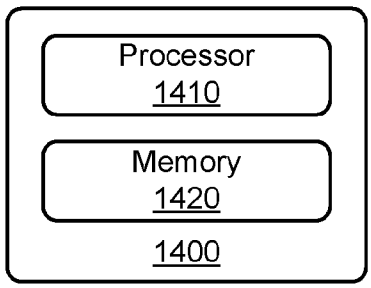
Processor
1410
Memory
1420
1400

FIG 14

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2023/027745

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06N5/045 G06N5/022 G06N20/00 G06N3/08
ADD.

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06N

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

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	LIU SHANG SHANG LIU@NTU EDU SG ET AL: "Structural Relationship Representation Learning with Graph Embedding for Personalized Product Search", PROCEEDINGS OF THE 29TH ACM INTERNATIONAL CONFERENCE ON INFORMATION & KNOWLEDGE MANAGEMENT, ACMPUB27, NEW YORK, NY, USA, 19 October 2020 (2020-10-19), pages 915-924, XP058708798, DOI: 10.1145/3340531.3411936 ISBN: 978-1-4503-6859-9 the whole document ----- -/--	1-15

[X] Further documents are listed in the continuation of Box C.

[X] See patent family annex.

* Special categories of cited documents :

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
4 October 2023	12/10/2023

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Manfrin, Max

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2023/027745

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	LE YU ET AL: "Heterogeneous Graph Representation Learning with Relation Awareness", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 17 April 2022 (2022-04-17), XP091193517, DOI: 10.1109/TKDE.2022.3160208 the whole document	1-15
A	WANG CHI CHIWANG1@ILLINOIS EDU ET AL: "Learning relevance from heterogeneous social network and its application in online targeting", COMPUTERS AND ACCESSIBILITY, ACM, 2 PENN PLAZA, SUITE 701 NEW YORK NY 10121-0701 USA, 24 July 2011 (2011-07-24), pages 655-664, XP058499358, DOI: 10.1145/2009916.2010004 ISBN: 978-1-4503-0920-2 the whole document	1-15
A	US 2022/058714 A1 (CHEN YONGJUN [US] ET AL) 24 February 2022 (2022-02-24) the whole document	1-15

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2023/027745

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2022058714 A1	24-02-2022	CN 115917577 A	04-04-2023
		EP 4200761 A1	28-06-2023
		JP 2023537879 A	06-09-2023
		US 2022058714 A1	24-02-2022
		WO 2022039902 A1	24-02-2022