Building and Exploring an Enterprise Knowledge Graph for Investment Analysis*

Tong Ruan[1], Lijuan Xue[1], Haofen Wang[1], Fanghuai Hu[2], Liang Zhao[1], Jun Ding[2]

[1] East China University of Science and Technology, Shanghai, 200237, China
{ruantong,whfcarter}@ecust.edu.cn,{xuelijuanjsj,tracy_zl1993}@163.com
[2] Shanghai Hi-knowledge Information Technology Corporation, Shanghai, 200082, China
{hufh,dingjun}@hiekn.com

Abstract. Full-fledged enterprise information can be a great weapon in investment analysis. However, enterprise information is scattered in different databases and websites. The information from a single source is incomplete and also suffers from noise. It is not an easy task to integrate and utilize information from diverse sources in real business scenarios. In this paper, we present an approach to build knowledge graphs (KGs) by exploiting semantic technologies to reconcile the data from diverse sources incrementally. We build a national-wide enterprise KG which incorporates information about 40,000,000 enterprises in China. We also provide querying about enterprises and data visualization capabilities as well as novel investment analysis scenarios, including finding an enterprise's real controllers, innovative enterprise analysis, enterprise path discovery and so on. The KG and its applications are currently used by two securities companies in their investment banking and investment consulting businesses.

Keywords: knowledge graphs, D2R, information extraction, data fusion, investment analysis

1 Introduction

Full-fledged information about enterprises is useful in different areas, including analysis of regional industry distribution for the government, competitor intelligence for corporation executives, credit analysis for banks and so on. While applications of enterprise information may be similar or different, a large-scale enterprise knowledge base (KB) is always in great demand. The challenges of building such an enterprise KB are enormous. For example, the information is scattered in different databases and websites with noise. Besides, it is not realistic or necessary to get all data sources on hands at the beginning of the project. New data sources along with new pieces of data should be added incrementally on demand.

Recently, semantic technologies based on RDF representation, graph databases as well as other related technologies have been key enablers to build and explore KGs. In particular, a big KG could be constructed by integrating a few separate KGs with schema alignment and instancing matching mechanisms, and new properties or concepts as well as new instances can be easily added to the fused KG. Similarly, the ever growing KG can support more intelligent applications.

* This work was partially supported by "Action Plan for Innovation on Science and Technology " Projects of Shanghai (project No: 16511101000), and Software and Integrated Circuit Industry Development Special Funds of Shanghai Economic and Information Commission (project NO:140304)

In this paper, we build a national-wide Enterprise Knowledge Graph, called EKG, which incorporates information about 40,000,000 enterprises in China. Based on the EKG, we provide querying about enterprises and data visualization capabilities as well as novel investment analysis applications, including finding an enterprise's real controllers, innovative enterprise analysis, enterprise path discovery and so on. We encounter business challenges as well as technology challenges in constructing and deploying the KG. The major business challenge is how to provide deep and useful analysis services without violating the privacies of a company and its employees. The technology challenges include constructing problems such as transforming the databases to RDF (D2R), representing and querying difficulties when meta properties and n-ary relations are involved, and performance issues since currently the KG contains more than one billion triples.

The product name of the EKG is "Magic Mirror". We expect that the "mirror" can reflect every important aspect of a company. The product is sold as a service, and currently we target the service to securities companies. Most securities companies in China provide investment bank services and investment consulting services. With the opening of "New Three Board", small innovation companies can be listed on the "New Three Board" with the endorsement of securities companies. Therefore the securities companies serve customers from big enterprises to small and medium-sized enterprises. There are about 40M companies in China. It is difficult for the securities companies themselves to gather authentic and full-fledged company information of their customers and potential customers. We collect company information from different sources for them and represent it in easy-to-use graphs. The "Magic Mirror" targets to help the securities companies to know and to approach their target companies better and quicker.

The rest of the paper is organized as follows. Section 2 describes the challenges that we are facing. Section 3 gives a brief overview of our approach. Section 4 presents our approach for building a national-wide EKG and how we addressed these challenges. Section 5 shows how the EKG is being used in practice. Section 6 lists the related work, and we conclude the paper in Section 7.

2 Challenges

There are business challenges as well as technology challenges in constructing and deploying the EKG. The business challenges include:

- **Data Privacy** One of our data sources is from China's State Administration for Industry and Commerce[3] (CSAIC), a government agency which pays serious attention to privacy issues. In such cases, we have to be very careful to balance the information requirements from EKG users with the privacy constraints.
- **Killer Services on the Graph** Since our EKG is complex and big, users will be overloaded with too much information if we deliver the raw information directly to users.

To address the privacy problem, first we transform the original data into the rank form or the ratio form instead of using real accurate values. Secondly, we obscure critical nodes (e.g., person related information) which should not be shown when visualizing the EKG as a graph. Thirdly, we provide UI interfaces which only allow particular types of queries. We deliver services which directly meet business requirements of users. For example, the service *finding an enterprise's real controllers* tells

[3] http://www.saic.gov.cn/

the investors from investment banks who is the real owner of a company, and the service *enterprise path discovery* provides hints on how the investors could reach the enterprises they want to invest in.

Technology challenges arise from the diversity and the scale of the data sources. At the first stage of our project, we mainly utilize relational databases (RDBs) from CSAIC. Secondly, we supplement the EKG with bidding information from Chinese Government Procurement Network[4] (CGPN) and stock information from Eastern Wealth Network[5] (EWN). Then the EKG is fused with the patent information extracted from the Patent Search and Analysis Network of State Intellectual Property Office[6] (PASN-SIPO) in another project. At last, the competitor relations and acquisition events are added to the EKG. This information is extracted from encyclopedia sites, namely Wikipedia, Baidu Baike and Hudong Baike. The following challenges are encountered during the above process:

- **Data Model** In addition to the basic data types (e.g., integer, float, and string), we need to store some complex data types including sequential data (e.g., phone number list of an enterprise), range-type data (e.g., effective operating time of an enterprise), map-type data (e.g., the annual sales of an enterprise should be stored in a map structure in which the key is the particular year and the value is sales). Furthermore, the relations in EKG are not only binary relations, there exist "property of relations" and "n-ary relations". The former are called *meta property* or *property graph* in the literature[7], and the latter sometimes referred as *event*. For the meta property example in our database, if a person is employed by a company, there is an additional property "entry time" which relates to the "employ" relation. For the event example, the investment relationship contains investors, companies, investment time, investment amount, investment ratio and so on. Since relations in RDF are binary, there are many discussions on how to represent events and meta properties, such as the W3C Working Group Note "Defining N-ary Relations on the Semantic Web"[8]. However, we do not find existing mature solutions on representing and querying meta properties and events in an efficient way.
- **D2R Mapping** At first, we use existing D2R tools (e.g., D2RQ[9]) to map RDBs from CSAIC into RDF. However, we encountered the following challenges that we can not solve directly with existing tools: a) Mapping of meta property. As mentioned above, there exist meta properties in our applications, and metafacts are facts related to meta properties. b) Data in the same column of RDBs map to different classes in RDF (Referred as *conditional class mapping* later). For example, patent applicants could be natural persons as well as companies. c) Data in the same RDB tables may map to different classes having subClass relations (Referred as *conditional taxonomy mapping* later). For example, "Listed Company" is a subCalss of "Company", a record in the company table can be mapped to an instance of a parent class "Company" or an instance of a child class "Listed company".
- **Information Extraction** The "competitive", "acquisition" and other relations are extracted from encyclopedia sites in our paper. Since the related information might not only be contained in semi-structured sources like infoboxes, lists and

[4] `http://www.ccgp.gov.cn/`
[5] `http://www.eastmoney.com/`
[6] `http://www.pss-system.gov.cn/`
[7] http://www.w3.org/TR/rdf11-concepts/
[8] http://www.w3.org/TR/swbp-n-aryRelations/
[9] `http://d2rq.org/`

tables, but also be mentioned in free texts. It is not an easy task to extract them from various types of data with high accuracy. However, entity extraction becomes difficult when there are abbreviations of company names in encyclopedic sites. The same company may be written in various variations or abbreviations. For example, abbreviations like "中铝(Chalco)", "中铝公司(Chinalco)", and "中国铝业(Chinalco)" can all represent the company "中国铝业股份有限公司(Aluminum Corporation of China Limited)". Moreover, Several company names may share the same abbreviation. For example, "大连万达集团股份有限公司(Dalian Wanda Group Co., Ltd)" and "万达信息股份有限公司(Wonders Information Co., Ltd)" can all be abbreviated as "万达(Wanda)".

- **Query Performance** We encounter performance bottlenecks since the number of triples of our EGK has reached billions. Furthermore, there are more complex query patterns when the EKG usage scenarios increases: a) Query all instances of a class which is a superClass in class hierarchy. For example, each patent in the patent KG has an International Patent Classification (IPC) code as its class. The IPC is a hierarchical patent classification system used to classify the content of patents. When users query the KG on an IPC code, we should find all the subClasses of the IPC code recursively, then find the patents belong to the subClasses. b) Query all properties of an instance. The problem arises since different properties of the same instance may store as different triples in graph store. The operation usually has many I/O operations if the number of properties becomes large. For example, there are more than 100 properties of companies in our EKG, if each property of an instance is stored in a different property table (vertical partitioning method in [1]), the operation requires read operation for more than 100 times. c) The queries on meta properties and n-ary relations. There are queries containing join operations between meta properties and ordinary properties, combined with filtering and sorting criteria. For example, find investment events where investment ratio is bigger than 10 percent and investment amount is bigger than ten million. If we use the triple store method in [8] to store properties and meta properties separately, when join operations are required on these properties, we need to load all triples to the memory.

In order to solve the above challenges, we carefully select the most suitable methods and algorithms, and adapt them to our problems.

1. The following steps are designed to do D2R mapping. First, we split the original tables into atomic tables and complex tables, then we use D2RQ tools to handle mappings on atomic tables. At last, we develop programs to process ad hoc mappings on complex tables.
2. We use multi-strategy learning methods in [16] to extract competitor relations and acquisition events from various data sources of encyclopedic sites. We use HTML wrappers to extract information from semi-structured sources firstly, then we use Hearst patterns to extract information from free texts. The extracted data are fed as seeds to distantly supervise the learning process to extract more data from free texts.
3. We adopt a graph-based entity linking algorithms in [2] to accomplish the task of entity linking. First we calculate the similarity between mentions and entities in KB to find candidate entities, then we construct an undirected graph to complete disambiguation.
4. Since we do not find enough features to support meta properties or events in existing graph databases, we design our own storage structure to fully optimize the performance of miscellaneous queries in EKG. We use a hybrid storage solution composed of multiple kinds of databases. For large-scale data, we use NoSQL

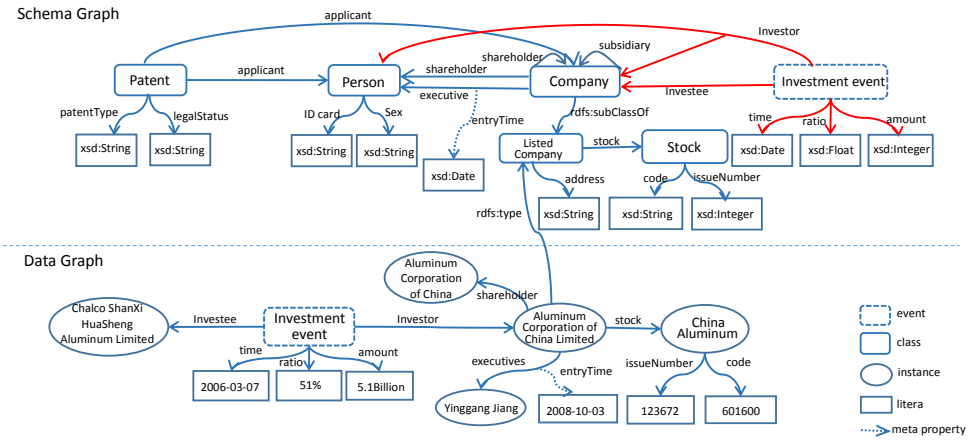


Fig. 1. Part of our Enterprise Knowledge Graph

database namely Mongodb as the underlying storage. NoSQL can store mass data and has good scalability. For high-frequency query data, we use a memory database to store data, which can greatly accelerate query speed. We also partition the main data table to reduce the number of records in a single data table. We store meta properties of a relationship in the same table so that the ordinary property and meta property can be fetched in one operation. We also build indexes on n-ary relations and meta properties.

3 Approach Overview

3.1 Problem Definition

Description of the Target KG Our goal is to build a national-wide enterprise KG. Firstly we give a quasi-definition of what a KG is in our paper.

As shown in Figure 1, a knowledge graph G consists of schema graph G_s, data graph G_d and the relations R between G_s and G_d, denoted as G = <G_s, G_d, R>.

The schema graph G_s = <N_s, P_s, E_s>, where N_s is a set of nodes representing classes (concepts); P_s is a set of nodes representing properties, and P_s contains *rdfs:subClassOf*, *rdfs:equivalentClass*, and other user defined properties such as *applicant*; and E_s is a set of edges representing the relationships between classes in the graph G_s. $E_s \subseteq N_s \times P_s \times N_s$. For example, the domain of the property *applicant* in Figure 1 is *patent* and the range of the property is *company* or *person*. There are two particular situations: a) The subject of P_s is a relation instead of a class, we call the P_s *meta property*. For example, if a person is employed by a company, there is a meta property "entry time". b) If P_s links to more than one subject or object, we call it *n-ary relation*.

The data graph G_d = <N_d, P_d, E_d>, where N_d is a set of nodes representing instances and literals; P_d is a set of nodes representing properties; and E_d is a set of edges representing the relationships between nodes in the graph G_d. Actually, each edge (Subject, Predicate, Object) stands for a fact. N_d includes two disjoint parts, namely instances (N_i) and literals. If the object of a triple is an instance, we call the property *relation* or *object property* in our paper. Otherwise, we call the property *datatype property*.

The relations R between schema graph G_s and data graph G_d are the relations which link the instances in the data graph to classes in the schema graph by the *rdf:type* property. R = {(instance, *rdf:type*, class)|instance $\in N_i$, class $\in N_s$}.

Figure 1 shows a small part of our EKG. The graph has nodes representing companies, stocks, persons, patents, investment events and data extracted from multiple sources, including stock code, entry time and issue number. The figure does not show a wide array of other properties extracted from data sources. Note that the graph includes edges that represent the executives, shareholders, subsidiary and so on.

Data Sources and Related Tasks to Construct the EKG We have five major sources for constructing our EKG:

1. The KG is mainly based on structured enterprise information from CSAIC. It contains the information of 40,000,000 companies, 60,000,000 people, 8,000,000 pieces of litigation, and 1,000,000 sources of credit information. The information about a company contains company executives, registration number, address and so on. The information about a person contains entry time, ID card number, position and so on. The information about a piece of litigation contains complainant, case number, trial date and so on. A source of credit information contains performance status, court and so on. We transform RDBs into the RDF to build a basic KG[10].
2. PSAN-SIPO, one of the largest and most successful patent websites, contains a large amount of patent information. We extract 5,000,000 pieces of patent information from PSAN-SIPO including patent applicant, application number, patent name and so on to build a patent KG[11]. The basic KG and the patent KG are linked with companies and persons to form EKG.
3. We extract 3,000,000 pieces of enterprise bidding information from CGPN including investor, investee, invest time and so on. Stock information (e.g., stock code, issue number and so on) of listed companies is extract from EWN. Then the EKG is fused with the above extracted information.
4. Competitive relations as well as acquisition events extracted from encyclopedic sites are added to the EKG through the company name and person name.

An Example Here we take "中国铝业股份有限公司(Aluminum Corporation of China Limited)" as an example to show how the EKG can be constructed from different sources, as shown in figure 2. Firstly, the databases from CSAIC contain information about the directors and shareholders as well as general managers of " 中国铝业股份有限公司(Aluminum Corporation of China Limited)". We transform RDBs into RDF to form the basic KG, and we get triples such as <"中国铝业股份有限公司(Aluminum Corporation of China Limited)", director, "熊维平(Weiping Xiong)"> and <"中国铝业股份有限公司(Aluminum Corporation of China Limited)", general manager, "罗建川(Jianchuan Luo)"> from the transformed results.

Secondly, we can utilize the PSAN-SIPO website to investigate a company's technology advantages. We extract attributes and the values of attributes from the website, and also convert the attribution value pairs into triples to form a patent KG. The patent KG may contain triples such as <CN201510863837.3, applicant, "中国铝业股份有限公司(Aluminum Corporation of China Limited)">. Data fusion algorithms are expected to link the two KGs with companies and persons.

[10] `http://ent.hiekn.com:28080/`

[11] `http://kechuang365.com/charts.html`

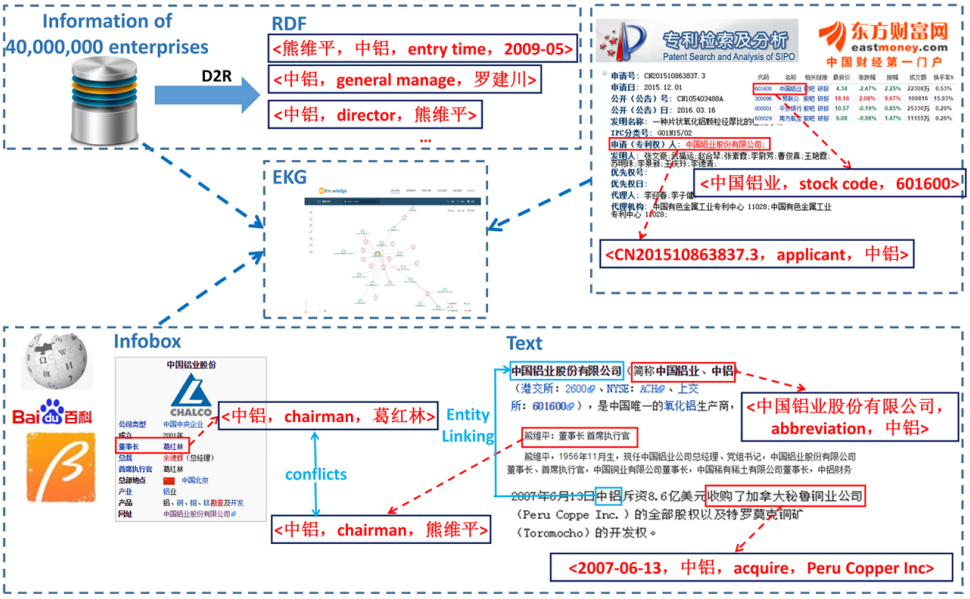


Fig. 2. Building EGK from multiple sources :*Aluminum Corporation of China Limited* Example

Finally, we extract attribute value pairs of stock from EWN and also convert them into triples such as <“中国铝业(Chinalco)”, stock code, 601600>. We can extract corporate executives from the infobox of Baidu Baike (e.g., <“中铝(Chalco)”, chairman, “葛红林(Honglin Ge)”>) and Wikipedia (e.g., <“中铝(Chalco)”, chairman, “熊维平(Weiping Xiong)”>), acquisition events from free texts of encyclopedia sites (e.g., <2007-06-13, “ 中铝(Chalco)”, acquire, Peru Coppe Inc>). We find that information extracted from different sources has the problem of inconsistency. We determine which data source is correct according to the pages’ update times. Information extracted from EWN and encyclopedia sites is linked to the KG with companies and persons. As a result, the information of “中国铝业股份有限公司(Aluminum Corporation of China Limited)” is obtained, and builds a KG of “中国铝业股份有限公司(Aluminum Corporation of China Limited)”.

3.2 Data-driven KG constructing process

As shown in Figure 3, there are five major steps, namely *Schema Design*, *D2R Transformation*, *Information Extraction*, *Data Fusion with Instance Matching*, *Storage Design and Query Optimization*. When the EKG is built, we provide *Usage Scenarios* to securities companies on the EKG. The whole process is data-driven and iterative. In particular, whether the *D2R Transformation* step or the *Information Extraction* step is initiated is based on the type of data source. Whether the new iteration begins depends on the input of new data sources. Besides, if there are multiple sources in one iteration, we always use more structured data in the first place. There are two iterations in our example. For the first iteration there are two separate projects. One is an enterprise KG transformed from CSAIC; the other one is a patent KG extracted from PASN-SIPO. The two KGs serve different users. For the second iteration, we use data fusion algorithms to link the two KGs with companies and persons. There are also other data sources which could supplement the KG. We use specific HTML

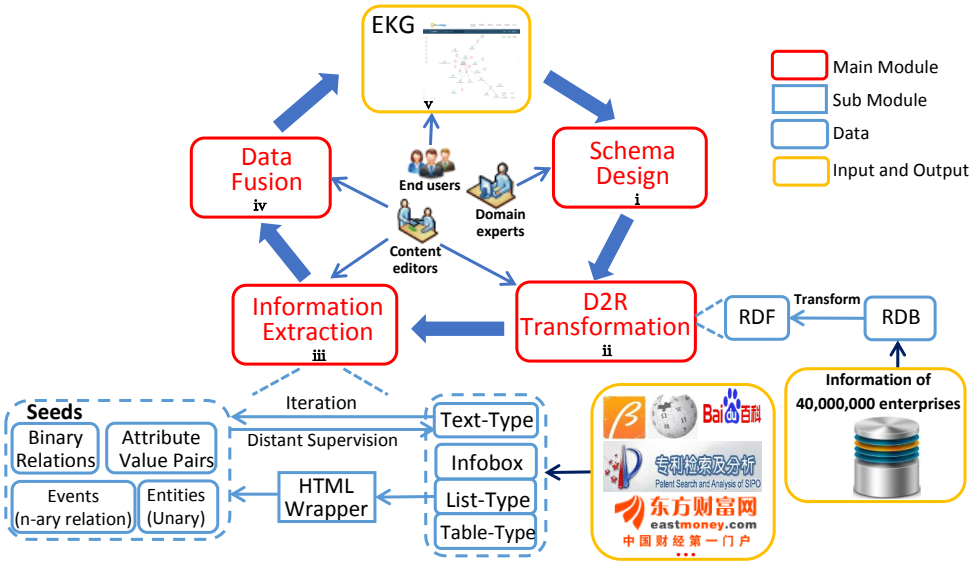


Fig. 3. Data-driven KG constructing process

wrappers to extract information from semi-structured sources. Then we use Hearst patterns and distant supervision to extract more information from free texts. At last, we use *Instance Matching* algorithms to check whether instance pairs can be aligned.

4 Building Knowledge Graphs

4.1 Schema Design

While most general KGs such as DBpedia and YAGO are built in a bottom-up manner to ensure wide coverage of cross-domain data, we adopt a top-down approach in EKG construction to ensure the data quality and stricter schema. While methods exist to automatically extract schema-level knowledge such as taxonomies and class definitions from Web sites and databases, this approach is quite useful when the domain schema is complex. We manually design or extend the schema of the EKG since the schema is subject change when new data sources are added.

At the first iteration, the EKG includes four basic concepts, namely "Company", "Person", "Credit" and "Litigation". Major relations include "subsidiary", "shareholder", and "executive". The concepts in patent KG only include "Patent". Major relation is "applicant". At the second iteration, we add "ListedCompany", "Stock", "Bidding" and "Investment" to the EKG. Besides properties of the newly added concepts such as "stock code" and "issue number", there are also new relations between the existing EKG, for example "acquisition" and"competitive" as well as "subClassof" relation between "Company" and "ListedCompany", as shown in Figure 1.

4.2 D2R Transformation

We take three steps to transform RDBs to RDF, namely *table splitting*, *basic D2R transformation by D2RQ* and *post processing*. The original data tables from CSAIC are integrated from multiple databases of provincial bureaus. These tables do not follow

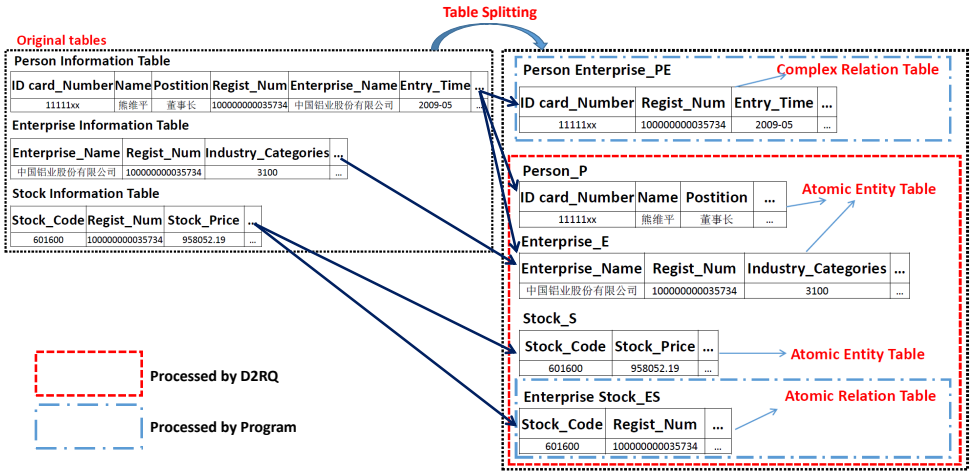


Fig. 4. An example of table splitting

the basic design principles of databases (e.g., BNCF[12]). Some tables may contain multiple entities and relations. Furthermore, the table may contain n-ary relations or the same table column may refers to different entity types, as mentioned in Section 2. In order to make the tables easier to understand and to handle, we split the original tables virtually into smaller ones, namely atomic entity tables, atomic relation tables, complex entity tables (e.g., tables require *conditional class mapping* mentioned in section 2) and complex relation tables (e.g., relation table with meta properties). An atomic entity table corresponds to a class, and an atomic relation table corresponds to relation instances where the domain and the range are two classes. We use D2RQ to transform the atomic entity tables and the atomic relation tables into RDF. We also write special programs to deal with complex relation tables which may require *meta property mapping*, *conditional taxonomy mapping* and so on, which have been mentioned in Section 2.

- *Table Splitting:* As shown in Figure 4, the original table *Person Information* also contains enterprise information. We divide the table into *Person_P*, *Enterprise_E* and *Person Enterprise_PE*. The *Enterprise_E* table is furthered merged with the original *Enterprise Information* table, because the two tables share the similar information about enterprises.
- *Basic D2R Transformation by D2RQ:* We write a customized mapping file in D2RQ to map fields related to atomic entity tables and atomic relation tables into RDF format. We map table names into classes, columns of the table into properties, and cell values of each record as the corresponding property values for a given entity.
- *Post Processing:* a) Meta property Mapping. The program gives a self-increasing ID annotation to the fact which has meta properties. The meta properties will then be properties of this n-ary relation identify by this ID. Thus we get some new triples(e.g., <ID, meta property, value>). b) *Conditional taxonomy mapping.* Our program determines whether the entity maps to the subClass according to whether the entity appears in the table related to the subClass. For example, if a company exists in the relation table of company and stock, it implies that the company is a listed company, so we add a triple <registration number, rdf:type,

[12] BCNF: Boyce and Codd Normal Form

Listed Company> to illustrate the fact, otherwise we add a triple <registration number, rdf:type, Company>. c) *Conditional class mapping.* In our example, in the entity table of applicant, there is a column called "applicant type" which indicates whether the applicant is a or a natural person. Our program uses this field as the condition of the mapping choice.

4.3 Information Extraction

The EKG extracts information from various data sources, including HTML websites like PSAN-SIPO, EWN, and CGPN, and encyclopedic sites like Wikipedia, Baidu Baike, and Hudong Baike. Furthermore, while most information extraction research work focuses on extracting one particular kind of target such as entities or relations between entities, we have to extract different types of entities(e.g., companies), binary relations(e.g., competitors), and attribute value pairs(e.g., CEO of a company). Our tasks also include event(n-ary relation) extraction (e.g., company acquiring) and synonym extraction (e.g., abbreviation of companies). The extraction strategy varies according to the data sources and extraction targets. We adopt a multi-strategy learning method to extract multiple types of data from various data sources. The whole process is as follows:

1. Entities and attribute value pairs of patent, stock and bidding information are extracted from PSAN-SIPO, EWN and CGPN respectively by using HTML wrappers.
2. Attribute value pairs (e.g., the chairman of an enterprise) of enterprises are extracted from infoboxes of encyclopedic sites by using HTML wrappers. Information extracted from different sources has the problem of inconsistency. We evaluate this information according to the pages' update time to determine which data source is correct. This extracted information can supplement the null value of databases.
3. Binary relations, events and synonyms identification on free texts require seeds annotation in sentences to learn patterns. These patterns are further used in other sentences to extract information. The quality of the extracted information heavily depends on the number of annotated sentences, whereas manual annotation costs too much human effort. Thus, for binary relation, event and synonym, we define a set of Hearst patterns to extract data from free texts of encyclopedic sites. For example, leveraging the Hearst pattern "X收购 (acquire) Y" can extract triples such as <中国铝业(Chinalco), acquire, 永晖焦煤股份有限公司(Winsway Coking Coal Holdings)>, and <中铝(Chalco), acquire, 秘鲁铜业公司(Peru Coppe Inc.)> from free texts. Then the extracted triples are fed as seeds automatically label free texts. This kind of distant supervision can significantly reduce the effort of manually labeling sentences. We first collect sentences that contain seeds and label these sentences. Then we generate extraction patterns from the annotated sentences. A good pattern should be generated by several sentences, thus we compute the support of each pattern. The pattern with a score greater than the threshold is selected as the extraction pattern. Finally, we use the generated extraction patterns to extract new information from other free texts. The newly extracted information is added to the seeds for bootstrapping. The whole process is iterative until no new information can be extracted.

In the process of information extraction, there are many abbreviations of company names in encyclopedic sites, as mentioned in Section 2. Here we use entity linking algorithms to link a company mentioned in text to companies in the basic EKG. We

adopt a graph-based method to accomplish the task of entity linking in two steps: a)*candidate detection*, that is, finding candidate entities in the KB that are referred by each mention. We first normalize company names, including company names extracted from multiple data sources and company names in the KB. More specifically, for a company name, if it contains any suffix (e.g., "股份有限公司(Corp.)", " 有限公司(Co.,Ltd)", "集团(Group)", and so on), the suffix is deleted. The purpose of this step is to be able to calculate the similarity between the core word of the mention and the core word of the entity in KB. Then, we use *Context Similarity* to calculate the similarity between the mentions and the entities in KB which are normalized to find candidates. Context similarity is to compute cosine similarity between the sentence containing the mention and the textual description (the first section of the Wikipedia article) of the entity in the KB. b)*disambiguation*, selecting the most possible candidate to link. Here, we use the disambiguation algorithm proposed in the literature [2].

4.4 Data Fusion with Instance Matching

Information from different sources should be fused into EKG. For example, if the value of the "applicant" property of a patent is a company name, it should link to the instance of the company in the EKG. The problem is simple for instance matching of companies. In China, as requested by China's State Administration for Industry and Commerce, the full company name should be unique. The company names on patent and bidding web sites are also full names. Therefore it is very easy to link the patent KG with the basic KG. However, the problem is tough for instance matching of persons. While there are personal ID numbers for every person, there are no such IDs in the patent data sources. Currently, we use a simple heuristic rule to match the person in the patent KG to that person in the basic KG. If the name of the patent inventor and the applicant equals the name of the person and company in the basic KG respectively, we say the patent inventor matches the person name in the basic KG.

4.5 Storage Design and Query Optimization

We design our own triple store on the top of existing NoSQL databases. In particular, we use MongoDB as main storage for its large install bases, good query performance, mass data storage, and scalability with clustering support. We implement varies data types on top of MongoDB including List type, Range type and Map type. The corresponding query interfaces for each type are also implemented. For example, the Map type implements interfaces such as accessing all values, a key of specific values and the maximum or minimal value in the map.

Query performance is improved in varies ways: a) Design a storage structure which supports efficient querying on meta properties and n-ary relations. We store meta properties and their values in different columns of the same table as original SPO triples. In the similar way, we store n-ary relations in the same row of a table. The consequence of such design is promising. The property and meta property are be retrieved together with one operation. Furthermore, when filtering and sorting operations were performed on n-ary relations, query can be completed on the database level by using the indexes we built. There are no additional in-memory operation required. b) Use in-memory database Redis to store the data which are heavily accessed. Redis supports abundant data structures, which is very useful in our application context. The data stored in Redis includes the schema definition table and the class hierarchy relation table. c) Construct sufficient indexes. Besides commonly used indexes such

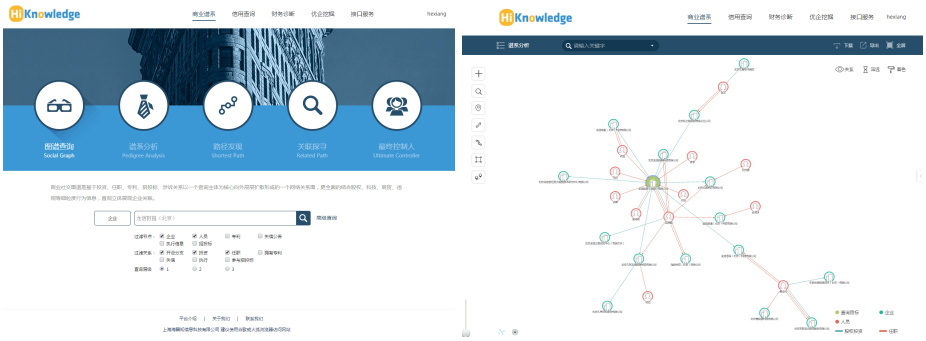


Fig. 5. Screenshot of EKG query interface showing results of a query on the *Long Credit Data (Beijing) CO.,Ltd*

as SP, SO, PS and so on, we also build indexes on meta properties and n-ary relations based on application requirements. For example, the indexes on investment amount and investment ratio. d) Data sharding. We partition triples into different tables according to the data type of the property value. For example, basic types are divided into Integer, Float, DateTime, String and Text, and each is stored in separate table.

5 Deployment and Usage Scenarios

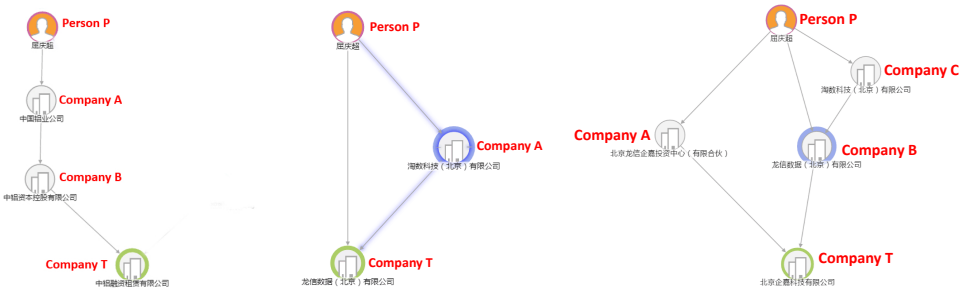
The EKG platform is deployed on the Internet with access control. There are about two hundred million entities, one billion attribute value pairs, and two hundred million relations in EKG. It takes an hour to extract entities, three hours to extract attribute value pairs, and three hours to extract relations from various sources. It is rebuilt once a month to incorporate newly added enterprise data.

We sell the whole solution as services instead of software. Securities companies have customized the EKG portal and have integrated it into their own applications. We provide general querying and graph visualization capabilities, including browsing shareholders, subsidiaries, patents and executives of a particular company. We also provide in-depth analyzing services dedicated to investing requirements in securities companies, including finding an enterprise's real controllers, innovative enterprise analysis, enterprise path discovery, multidimensional relationships discovery of enterprises and so on, as shown in Figure 5.

A typical example to query an entity is shown in Figure 5. A recorded demo about this example can be accessed from YouTube[13]. Firstly users can select the types of entities in our EKG, including Company, Person, Patent, Litigation and so on. Users can further select the number of relation levels which shown in the results. The level is limited to be less than three, otherwise the graph would become too big. Users may further design the filtering criteria based on what kinds of nodes and relations are included. The result of the query is shown in the right panel of Figure 5. The target entity and its relationships as well as related entities are shown in different shapes and colors depending on the types of relations and entities.

– *Finding an enterprise's real controllers.* When a securities company wants to approach a new customer, they should know the real decision makers of the potential

[13] `https://youtu.be/y3ZCMNrisGM`



(a) Linear structure (b) Triangle structure (c) Complex triangle structure

Fig. 6. Different structures for a person to control an enterprise

customer. The person who owns the biggest equity share is the real decision maker. However, a person may also control a company indirectly. For example, he or she can control a company by holding equity shares of companies which in turn are shareholders of the target company. We develop an algorithm to traverse the KG to find the real controllers of a company on the EKG. The shareholders of a company can be roughly divided into two types: enterprises and natural persons. The real controllers here refer to natural persons. To find the persons, we calculate the equity shares of all shareholders of the target company recursively until the shareholders are natural persons. When the shareholders are natural persons, we multiply all the equity shares on the recursion path as the nature persons' equity share, and add equity shares to the same person. Ultimately, the persons who have the largest equity shares are chosen as real controllers of the target company. The results of querying includes the information of real controllers and control paths. We find many different investment structures that a person can use to control a company. Figure 6(a) is a simple linear structure. A person P controls a company T through company A plus company B. Figure 6(b) is a typical triangle structure. It shows that a person P controls a company T by simultaneously investing in the company T and a shareholder of the company T. Figure 6(c) is more complex. The person P controls the target company T in three ways, namely from company A, from company B, and from company C plus company B.

- *Innovative enterprise analysis.* Securities companies want to find new and innovative companies that are worthy of investment. We connect the notion of innovativeness with the number of patents a company holds. In general, securities companies provide the field they are interested in, for example, robotics or remote healthcare, and the EKG system returns a list of companies which owns the largest number of patents in this field. The fusion of the patent KG into basic KG gives customers benefits that they can use to further investigate other information about the target company.
- *Enterprise path discovery.* Securities companies would like to know whether there are paths to reach their new customers, and they also want to know whether their potential customers have paths to their competitors, namely other securities companies. We use path discovery to find the path between any two companies or persons. As shown in Figure 7, in practice, we found most targeting companies can be approached by our securities company in less than four hops, since both securities companies and their targeting companies are fairly big and their investors are very famous.
- *Multidimensional relationships discovery.* Given two companies, there might be varies relationships between them (e.g., competitive relationship, patent transfer-

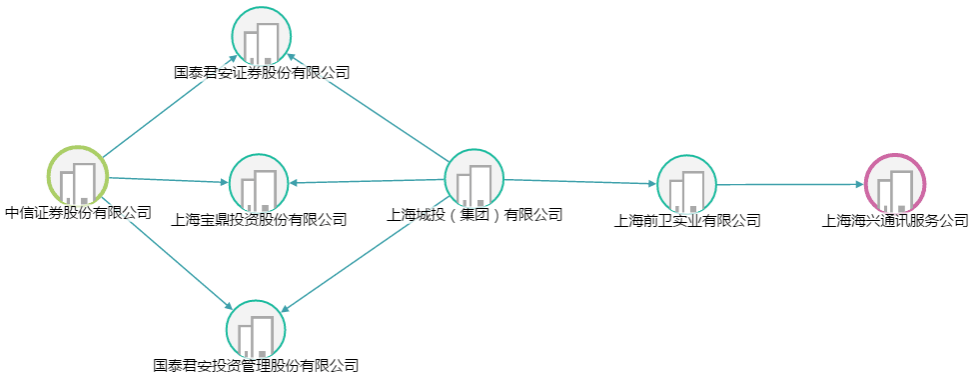


Fig. 7. Enterprise path discovery

ring relationship, acquisition relationship or investment relationship). Currently, we provide visualized graphs to help securities companies investigate different relationships between companies. Securities companies use these graphs to find new customers they are interested in. For example, if they find there are difficulties in investing in their target companies, they can approach the competitors of their target companies.

6 Related Work

6.1 Knowledge graphs and their applications

Knowledge graphs have attracted more and more attention from both academia and industry. Linked Open Datasets such as DBpedia and YAGO can be viewed as cross-domain knowledge graphs. Several Internet search engine companies have made an effort to build knowledge graphs in order to improve their search engine capabilities, including Google Knowledge Graph, Baidu Intimate, and Sogou Cubic Know. Nguyen et al. [12] analyzed the fitness for use of two encyclopedic datasets, namely DBpedia and Freebase, in music recommendations. Pirrò et al. [13] provided a tool called RECAP to generate explanations of relatedness on entities in encyclopedic datasets such as DBpedia and Freebase.

In contrast to the general-purposed KGs such as DBpedia and YAGO, Szekely et al. [18] presented a system called *DIG* and discussed how it can be used to build a knowledge graph for combating human trafficking. We also present our solution on building marine-oriented knowledge graphs in [15]. The notion of enterprise knowledge graph is also used by IBM [7] and other information technology companies when they adopt knowledge graph technology to particular enterprises.

While all the above work shows the benefit of semantic technologies, the large-scale enterprise knowledge graph built in this paper have a wider business perspective since EKG can further be used in more business scenarios such as competitor intelligence and credit analysis. The incrementally construction model proposed in this paper reduces the investment risk in early stages of the KG project.

6.2 Technologies in constructing and storing KGs

D2R In the process of transforming databases to RDF formats, we use the D2RQ tool. While the tool in general is to provide a virtual RDF layer on top of the relational

database [3] [17], it can also export RDF triples from RDB. D2RQ mainly consists of the D2R Server, D2RQ Engine and D2RQ Mapping. D2RQ Engine [5] use D2RQ Mapping to convert the data in relational database into RDF format. The D2RQ Engine does not convert the relational database into real RDF data, but uses D2RQ Mapping file to map the database into virtual RDF format. In this paper we use D2RQ to export RDF triples from simple tables generated from the table-splitting step, and then post-process the exported RDF triples for metafacts, n-ary relations and other complex D2R mapping situations.

Information Extraction Information extraction has been studied for a considerable amount of time. *Wrapper induction* is a sort of information extraction, which extracts knowledge from semi-structured data. Dalvi et al. [4] presented a generic framework to learn wrappers across websites. Gentile et al. [6] presented a methodology called multi-strategy learning, which combines text mining with wrapper induction to extract knowledge from lists, tables, and web pages. Distant supervision is an effective method to leverage redundancies among different sources, which has been used in [11] [14]. In this paper, we combine multi-strategy learning with distant supervision to extract information from various data sources. Entity Linking is the task of linking named entity mentions in text with their referent entities in a specific KB. Alhelbawy et al. [2] proposed a collective disambiguation method using a graph model. He zhengyan et al. [9] proposed an entity disambiguation model based on Deep Neural Networks. [10] provides an overview of recent instance linking approaches. In this paper, we use entity linking algorithms proposed by Alhelbawy et al. to help determine abbreviated company names in the text of encyclopedic websites.

Graph Database As we have mentioned, our EKG requires representation and querying on meta properties and n-ary relations. We have evaluated a few graph databases including Jena[14], Blazegraph[15], sesame[16], AllegroGraph[17] and Hexastore[18]. Portion of graph databases such as Hexastore support named graph or 4-ary relation. Weiss et al. [19] also built an all permutation indexing of (S,P,O,C) to speed the structural query with data source constraints in Hexastore. However, these databases do not directly support more than 4-ary relations which frequently take place in our usage scenarios.

7 Conclusion and Future Work

We find real-world enterprise information could be represented by the KG in a very natural way, and the KG also provides an easy way to integrate new data sources even after the basic KG has been built. The application requirements can conveniently be transformed to graph traversing and graph mining algorithms. The analyzing results visualized in the graph can be easily understood by non-IT customers.

We sell the EKG as services, and it can be easily integrated into different applications. For example, one of our customers, China Securities, who is one of the top ten securities companies in China, integrates the EKG into their customer relation management system. Thus the enterprise information provided by the EKG can not

[14] http://jena.apache.org/
[15] https://www.blazegraph.com/
[16] http://rdf4j.org/
[17] http://franz.com/
[18] https://www.npmjs.com/package/hexastore

only be used by the investment banking sector, but also be used by other sectors such as asset management sector or investment consulting sector.

In the future, we plan to add more data sources to the KG, such as tax and invoice information per month. With the applications of such information, investors can analyze the status of enterprises' business operations. Furthermore, we will also try to monitor the change of shareholders as well as share ratios. As we know, if investors want to control a company, they may not directly buy the share of the target company; instead, they can buy shares of companies who are shareholders of the company. In that case, we could develop interesting applications such as "Control intention recognition" to warn the current controller of the company.

References

1. D. J. Abadi, A. Marcus, S. R. Madden, and K. Hollenbach. Scalable semantic web data management using vertical partitioning. In *Proceedings of the 33rd international conference on Very large data bases*, pages 411–422. VLDB Endowment, 2007.
2. A. Alhelbawy and R. J. Gaizauskas. Graph ranking for collective named entity disambiguation. In *ACL (2)*, pages 75–80, 2014.
3. B. H. L. Bing. Semantic pattern mapping between rdbms and linked data based on open source software. *New Technology of Library and Information Service*, 2011.
4. N. Dalvi, R. Kumar, and M. Soliman. Automatic wrappers for large scale web extraction. *Proceedings of the VLDB Endowment*, 4(4):219–230, 2011.
5. V. Eisenberg and Y. Kanza. D2rq/update: updating relational data via virtual rdf. In *Proceedings of WWW*, pages 497–498. ACM, 2012.
6. A. L. Gentile, Z. Zhang, and F. Ciravegna. Web scale information extraction with lodie. In *2013 AAAI Fall Symposium Series*, pages 197–212, 2013.
7. I. Guy. Mining and analyzing the enterprise knowledge graph. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 497–498. ACM, 2013.
8. S. Harris and N. Gibbins. 3store: Efficient bulk rdf storage. 2003.
9. Z. He, S. Liu, M. Li, M. Zhou, L. Zhang, and H. Wang. Learning entity representation for entity disambiguation. In *ACL (2)*, pages 30–34, 2013.
10. J. Heflin and D. Song. Ontology instance linking: Towards interlinked knowledge graphs. In *Proceedings of AAAI 2016*, pages 4163–4169, 2016.
11. M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. ACL, 2009.
12. P. T. Nguyen, P. Tomeo, T. Di Noia, and E. Di Sciascio. Content-based recommendations via dbpedia and freebase: a case study in the music domain. In *International Semantic Web Conference*, pages 605–621. Springer, 2015.
13. G. Pirrò. Explaining and suggesting relatedness in knowledge graphs. In *International Semantic Web Conference*, pages 622–639. Springer, 2015.
14. B. Roth, T. Barth, M. Wiegand, M. Singh, and D. Klakow. Effective slot filling based on shallow distant supervision methods. *arXiv preprint arXiv:1401.1158*, 2014.
15. T. Ruan, H. Wang, F. Hu, J. Ding, and K. Lu. Building and exploring marine oriented knowledge graph for zhoushan library. In *Proceedings of ISWC 2014*, 2014.
16. T. Ruan, L. Xue, H. Wang, and J. Z. Pan. Bootstrapping yahoo! finance by wikipedia for competitor mining. In *JIST*, pages 108–126. Springer, 2015.
17. S. S. Sahoo, W. Halb, S. Hellmann, K. Idehen, T. Thibodeau Jr, S. Auer, J. Sequeda, and A. Ezzat. A survey of current approaches for mapping of relational databases to rdf. *W3C RDB2RDF Incubator Group Report*, 2009.
18. P. Szekely, C. A. Knoblock, J. Slepicka, A. Philpot, A. Singh, C. Yin, D. Kapoor, P. Natarajan, D. Marcu, K. Knight, et al. Building and using a knowledge graph to combat human trafficking. In *ISWC*, pages 205–221. Springer, 2015.
19. C. Weiss, P. Karras, and A. Bernstein. Hexastore: sextuple indexing for semantic web data management. *Proceedings of the VLDB Endowment*, 1(1):1008–1019, 2008.