A Method of Calculating the Semantic Similarity Between English and Chinese Concepts

Jingwen Cao[1], Tiexin Wang[1,2(✉)], Wenxin Li[1], and Chuanqi Tao[1,2,3]

[1] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, 29#, Jiangjun Road, Jiangning District, Nanjing 211106, China
caojingwen1028@126.com, {tiexin.wang, freedomtot}@nuaa.edu.cn, t-chuanqi@163.com
[2] Key Laboratory of Safety-Critical Software, Nanjing University of Aeronautics and Astronautics, Ministry of Industry and Information Technology, Nanjing, China
[3] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, People's Republic of China

Abstract. In the big data era, data and information processing is a common concern of diverse fields. To achieve the two keys "efficiency" and "intelligence" to the processing process, it's necessary to search, define and build the potential links among heterogeneous data. Focusing on this issue, this paper proposes a knowledge-driven method to calculate the semantic similarity between (bilingual English-Chinese) words. This method is built on the knowledge base "HowNet", which defines and maintains the "atom taxonomy tree" and the "semantic dictionary" - a network of knowledge system describing the relationships between word concepts and attributes of the concepts. Compared to other knowledge bases, HowNet pays more attention to the connections between words based on concepts. Besides, this method is more complete in the analysis of concepts and more convenient in calculation methods. The non-relational database MongoDB is employed to improve the efficiency and fully use the rich knowledge maintained in HowNet. Considering both the structure of HowNet and characteristics of MongoDB, a certain number of equations are defined to calculate the semantic similarity.

Keywords: HowNet · MongoDB · Semantic similarity · Knowledge driven

1 Introduction

NLP (Natural Language Processing) is a science that integrates linguistics, computer science, and mathematics. The importance of a large-scale computer-available dictionary with rich information on NLP is obvious. In order to improve the efficiency of NLP technology, it is necessary to create large-scale knowledge resources, including machine-processable dictionaries [1].

X. B. Zhai et al. (Eds.): MLICOM 2019, LNICST 294, pp. 313–324, 2019.
https://doi.org/10.1007/978-3-030-32388-2_27

Currently, there are several existing large knowledge bases. Compared to other knowledge base, such as "WordNet" and "ConceptNet", "HowNet" emphasizes the relationships between concepts, the relationships between attributes and attributes of concepts.

Natural language uses words as basic units. Words can form sentences, and sentences form chapters. Therefore, the semantics of one text is synthesized by the semantics of all the sentences contained, and the semantics of one sentence is determined by the semantics of the words and certain grammars. As the basic unit of sentences and texts, the words have specific semantics and connotations. Semantic analysis is the fundamental problem of NLU (Natural Language Understanding), which has a wide range of applications in NLP, information retrieval, information filtering, information classification, and semantic mining.

In the big data era, the importance of semantic analysis is increasing. To accurately extract information, retrieve required information, tap potential information value, and provide intelligent knowledge services, semantic analysis for machine understanding is indispensable.

In order to detect the semantic similarity between the concepts of words (objects), this paper proposes a method named ***SSDH (Semantic similarity detection based on HowNet)***. SSDH is built on the HowNet knowledge base. To improve the efficiency of SSDH, MongoDB is employed.

This paper is structured as follows. The second section presents the technology foundation of this paper (i.e., MongoDB and HowNet). The third section shows an overview of SSDH. The related work is illustrated in the fourth section. Finally, the fifth section draws a conclusion.

2 Pre-work

2.1 MongoDB

MongoDB is a product between a relational database and a non-relational database [2]. It is the most versatile and most relational database among non-relational databases.

MongoDB has two basic advantages in data storage and data query. First, the data structure it supports is very loose, which is similar to JSON's BSON format. So it can store more complex data types [3]. Second, it supports a very powerful query language with a similar syntax to the object-oriented query language. It realizes almost all the functions of relational database single-table query, and also supports indexing data [4].

MongoDB owns many fine features. Four of them are: (i) easy to store data of object types, (ii) dynamic query and full index, (iii) efficient binary data storage, and (iv) supporting Python, Java, C++ and many other languages.

MongoDB has been widely used, two main application scenarios are listed below.

- Real-time (website) data processing. MongoDB is ideal for real-time insertions, updates, and queries. It has the replication and high scalability required to store data in real time, making it ideal for databases consisting of tens or hundreds of servers.

- Cache. Due to its high performance, MongoDB is suitable as a caching layer for the information infrastructure. After the system is restarted, the persistent cache layer built by it can avoid overloading the underlying data source.

2.2 HowNet

HowNet is a bilingual (English-Chinese) knowledge base. It provides the knowledge to design real intelligent software. The total records in HowNet are more than 120,000, which are still expanding.

Considering HowNet, two concepts "atom" and "definition" needed to be explained firstly. **"Atom" is the smallest unit of meaning that cannot be divided**. The principle of choosing atoms is that the existing atoms must be able to describe all the concepts. **"Definition" is a concept normalized in HowNet, consisting of some atoms** [5].

As a common sense knowledge base, HowNet reveals the relationships between concepts, the relationships between attributes and attributes of concepts. The basic content is a networked organic knowledge system. *The semantic dictionary is the basic file of the HowNet base, composed of many records which contain the Chinese and English translations of words and the part of speech and definitions of words*. The semantic dictionary of HowNet is not simply copying English-Chinese dictionaries, the definition of each word is also based on the current popularity.

Figure 1 shows a combination of some records in HowNet semantic dictionary.

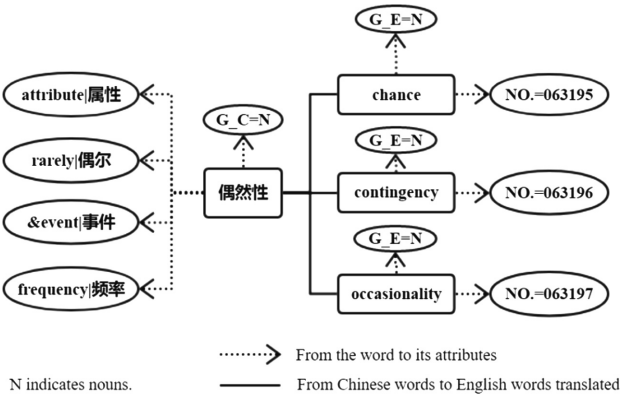

Fig. 1. An illustration of the HowNet semantic dictionary.

In Fig. 1, "*NO.*=" is followed by the serial number of the word in the dictionary. The "偶然性" in the middle of the figure is the Chinese interpretation of the word. And in English, its meaning is close to "accidentality". On the left is the definition of words made up of four atoms. The words after "*G_C* =" and "*G_E* =" are the attributes of Chinese words and the attributes of English words. In this example, they are all nouns. Due to cultural and language differences, Chinese words tend to correspond to more than one English word, such as "偶然性" to three English words "chance", "contingency", "occasionality".

Another basic file in the HowNet system is the atom taxonomy tree. Figure 2 shows an example (several layers) of the atom taxonomy tree. In this example, the closer the two atoms are (to the nearest common ancestor), the higher the similarity between the two atoms [6].

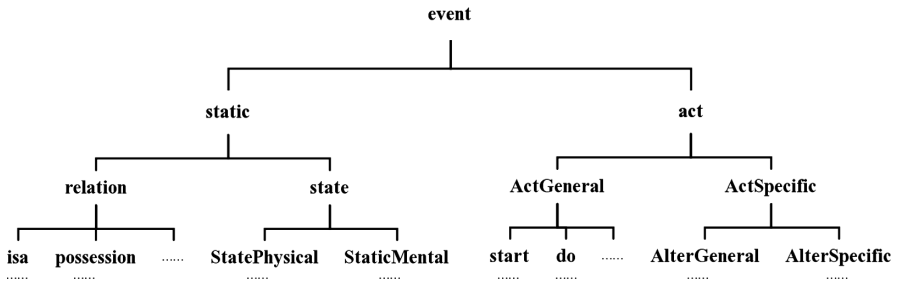

Fig. 2. An example of the atom taxonomy tree.

3 Main Work

In SSDH, the semantic similarity calculation is based on HowNet knowledge base. The data from HowNet needs to be processed and stored in MongoDB to be used.

The main work of this paper contains three parts: (i) processing HowNet data and storing it in MongoDB, (ii) querying the data stored in MongoDB for atom distance calculation and atom similarity calculation, (iii) comparing the atom similarity of each pair of definitions to calculate the semantic similarity between two words.

3.1 Processing HowNet Data and Store into MongoDB

Java is the main developing language of this work, and Eclipse is selected as the IDE. The detail of developing process of SSDH is out of the scope of this paper.

Since this work involves the usage of the atom taxonomy tree and semantic dictionary of HowNet, two collections in MongoDB were created to store these data: "atomtree" and "semanticdictionary".

MongoDB can store five kinds of tree structures: parent link structure, sub-link structure, ancestor queue structure, materialized path structure, and collection model. In this work, the "parent link structure" is employed.

MongoDB stores data as a document, and the data structure consists of key-value pairs. A MongoDB document is similar to a JSON object. Field values may contain other documents, arrays, and document arrays [7].

Figure 3 is a text file of the atom taxonomy tree stored in the parent node format. On each line, the serial number, the English name, the Chinese name of the atom, and the serial number of its parent node are listed sequentially.

```
0,event,事件,0
1,static,静态,0
2,relation,关系,1
3,isa,是非关系,2
4,be,是,3
5,become,成为,4
6,mean,指代,4
7,BeNot,非,3
8,possession,领属关系,2
9,own,有,8
10,obtain,得到,9
11,receive,收受,9
12,BelongTo,属于,8
```

Fig. 3. The text file of an atom taxonomy tree.

If an atom is the root node in a tree, serial number of its parent node will be its own serial number. After searching, these atoms form a total of nine trees. Taking the first line record as an example, the format stored in collection "atomtree" is as follows:

Document {{ID=0, EnglishName="event", ChineseName="事件", parent="0"}}

A record in a processed semantic dictionary consists of eight lines. As a document is stored in another collection, some definitions may not be currently used, but may be useful for future secondary development.

Take the number "2" as an example, the import process from HowNet to MongoDB is as follows.

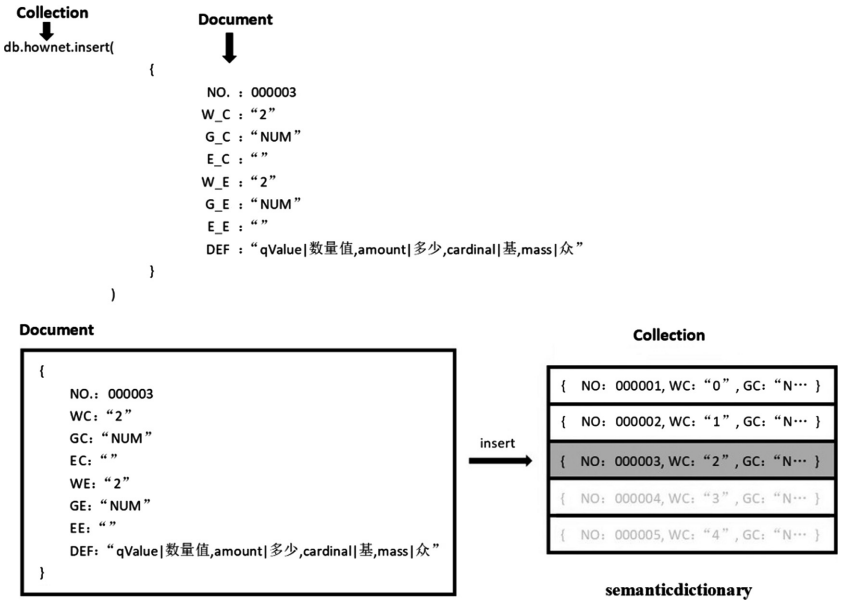

Fig. 4. Examples of importing data into MongoDB.

The format stored in collection "semanticdictionary" is presented below (Fig. 4).

Document {{NO=000003, WC=2, GC=NUM, EC=, WE=2, GE=NUM, EE=, DEF=qValue|数量值, amount|多少, cardinal|基, mass|众}}

After storing the data in MongoDB, the related key value is used for data indexing to select required documents for calculation.

Since each word has multiple attributes, this will increase the workload of the index, so two interfaces are created to provide arbitrary key value lookups for the documents in the two collections.

3.2 Calculating Atom Distance

HowNet defines and maintains the atom taxonomy tree, and the similarity of atom can be calculated by the relative distance on the atom taxonomy tree.

The nearest common ancestor of the two comparing atoms has to be located first. Then use the *upward recursive algorithm* to find the distance between the two atoms and the common ancestor, that is, the height difference between the layers, and add them to get the relative distance between the two atoms. If the two atoms are not on a tree, the default atom distance is 100.

Equation (1) is defined to calculate this.

$$AtomDistance(a,b) = Distance(a, com(a,b)) + Distance(b, com(a,b)) \quad (1)$$

"*AtomDistance(a,b)*" is the distance between "atom *a*" and "atom *b*". "*com(a,b)*" is the nearest common ancestor to "atom *a*" and "atom *b*". "*Distance(a,com(a,b))*" is the distance between "atom *a*" and the nearest common ancestor (of the two atoms).

Then calculate the similarity between the two atoms by employing Eq. (2).

$$AtomSim(a,b) = (1 - \frac{AtomDistance(a,b)}{2 \times TreeHigh_i}) \times (\frac{TreeHigh_i}{TreeHigh_i + TreeHigh_i - Deep}) \quad (2)$$

"*AtomSim(a,b)*" is the similarity between "atom *a*" and "atom *b*". "$TreeHigh_i$" is the height of the classification tree where the "atom *a*" and "atom *b*" are located. "*Deep*" is the depth of the root to the common ancestor (of "atom *a*" and "atom *b*").

For a branch node, the nodes of its first child are equidistant from all nodes of any other children in the same layer, and thus the longest distance of the two nodes can be roughly estimated to be twice the height of the tree. Different atom taxonomy trees have different "*TreeHigh(s)*", and it is necessary to determine which tree the atom is on and then find the corresponding "*TreeHigh(s)*".

Considering the usage of Eqs. (1) and (2), Table 1 shows a simple use case of comparing the atom distance and atom similarity between four pairs of words (i.e. male-female, male-young, Animal Human-human, royal-family). The testing results are also shown in Table 1.

Table 1. Experimental results on the similarity between two atom similarities.

No	Atom 1	Atom 2	Atom Distance	Atom Similarity
1	男|male	女|female	2	0.5
2	男|male	幼|young	4	0.2
3	动物|AnimalHuman	人|human	1	0.69
4	皇|royal	家|family	2	0.3

3.3 Computing Semantic Similarity

Since some Chinese words may correspond to multiple English words or have different meanings, it is necessary to compare all the definitions while calculating the similarity between two words. To find same definitions, Eq. (3) is defined.

$$comdef(A,B) = \{def | def \in DEF_A \wedge def \in DEF_B\} \tag{3}$$

"*comdef (A,B)*" is a collection of the same "*def(s)*". "DEF_A" is a collection storing all the definitions of word A, while "*def*" is one of the definitions of word A.

Then calculate the proportion of the same definition in all "*def(s)*", employing Eq. (4).

$$defRatio = \frac{|comdef|^2}{|DEF_A| \times |DEF_B|} \tag{4}$$

"*def Ratio*" is the same "*def*" rows account for the ratio in the "*DEF*" collection. "$|DEF_A|$" is the number of the "*def(s)*" in the "DEF_A" collection.

Some Chinese words correspond to different English words, but the definitions are the same, so there are cases where multiple definitions of a word are the same, and repeated definitions are counted as one in all calculations. If all the definitions of the two words are the same, then the similarity between the two words is judged to be 1, and the comparison is no longer continued.

$$WordSim(A,B) = 1 \tag{5}$$

"*WordSim(A,B)*" is the similarity between word *A* and word *B*.

Otherwise, the similarity between the two words needs to be continuously calculated, and the same definitions in the "def(s)" of the two definition items are respectively removed, only the definitions of the two words different are left, and then the similarity calculation of the atoms is performed for each pair of definitions of the two words.

First need to compare the first attribute of the two "def(s)", that is, whether the atoms are the same, if they are the same, let "*mainatom=1*", otherwise let "*mainatom=0*". And count the sum of the "*mainatom*" before Eq. (10).

$$allmainatom = \sum_{\substack{\forall def \in DEF_A \\ \forall def \in DEF_B}} mainatom \tag{6}$$

Then compare the remaining atoms, if a same atom exists in a pair of definitions setting to "def_{Ai}" and "def_{Bj}", put it into a collection "$Common(def_{Ai},def_{Bj})$", and count the ratio of common atoms to all atoms in this pair of definitions.

$$sameatomRatio(def_{A_i}, def_{B_j}) = \frac{\left|Common\left(def_{A_i}, def_{B_j}\right)\right|^2}{(|def_{A_i}| - 1) \times \left(\left|def_{B_j}\right| - 1\right)} \tag{7}$$

"def_{Ai}" is the "i th" def in the collection "DEF_A". In each pair of "def(s)" between word A and word B, "$atomRatio$" is the common atoms account for the ratio of all atoms in the two "def" collections.

Sum the repetition rates of all common atoms as shown in Eq. (8) before Eq. (10).

$$allsameatom = \sum_{\substack{\forall def_A \in DEF_A \\ \forall def_B \in DEF_B}} \sum_{\forall atom \in Common(def_{A_i}, def_{B_j})} sameatomRatio(atom) \tag{8}$$

Then remove the same atoms in the two definitions, and sum the atom similarity between the remaining atoms and the sum is set to "$alldiffatomsim$" before Eq. (10).

$$alldiffatomism = \sum_{\substack{\forall def_{A_i} \in DEF_A \\ \forall def_{B_j} \in DEF_B}} \frac{(1 - sameatomRation(def_{A_i}, def_{B_j})) \sum AtomSim(a, b)}{(\left|def_{A_j}\right| - 1 - |comatom|) \times (\left|def_{B_j}\right| - 1 - |comatom|)} \tag{9}$$

"a" belongs only to the remaining atoms in the collection "def_{Ai}" (not the intersection of "def_{Ai}" and collection "def_{Bj}").

Because there are many relationships in HowNet, such as component-total relationship (%), attribute-host relationship (&), material-finished relationship (?), incident event relationship (*), etc., these relationships will be reflected in adding the corresponding symbols "%", "&" etc. before atoms, and for these "$atom$"s, you need to compare them separately and compare the "$atom$"s with the same symbol. The comparison methods are the same as the above-mentioned same atoms processing methods and different atoms processing methods.

Finally define semantic similarity of two word.

$$\begin{aligned} &WordSim(A, B) = defRatio \\ &+ (1 - defRatio) \times \frac{\alpha \times allmainatom + \beta \times (allsameatom + alldiffatomsim)}{(|DEF_A| - |comdef|) \times (|DEF_B| - |comdef|)} \end{aligned} \tag{10}$$

The parameters "α" and "β" do not only limit the similarity range between 0 and 1, but also set the importance of different levels of first atom and other atoms. "α" defaults to 0.6, "β" defaults to 0.4, but it can be changed as needed.

Table 2 shows the use case and testing results of applying the above equations. Four pairs of words are contained in this use case.

Table 2. Experimental result about semantic similarity calculation between two words.

No	Word 1	Word 2	Semantic Similarity
1	医生(doctor)	人(human)	0.30
2	男人(male)	女人(female)	0.76
3	男人(male)	girl	0.71
4	commute a sentence	reduce a penalty	1.0

4 Related Work

This section introduces a classical corpus called WordNet and makes a distinction between WordNet and HowNet, then presents some latest research works employing HowNet.

4.1 WordNet

WordNet [WordNet: a lexical database for English] [8] is an on-line lexical reference system whose design is inspired by current psycho linguistic theories of human lexical memory.

WordNet and HowNet [HowNet - a hybrid language and knowledge resource] [9] have the same semantic concepts, and both of them believe that semantics is the interpretation of the conceptual world in the human's brain. However, they have different methods to characterize the conceptual structures and the relationships. WordNet uses a different approach to express the semantics of verbs, nouns, adjectives, and adverbs with the interrelation between synonym and Synset. HowNet uses constructive conceptual representations to explain various relationships between concepts by using "Sememe".

In terms of relationship, "Sememe" can be regarded as a more economic expression of conceptual relations, and it can be used to explain conceptual relations. Therefore, the relationship between WordNet and HowNet can be regarded as a phenomenon and a corresponding interpretation, and use the "Sememe" in HowNet to make a general explanation of the semantic relationships in WordNet. In this way, the relationship between the two knowledge networks can be built.

4.2 Research Works Concerning HowNet

Semantic similarity detecting is one of the key technologies of natural language processing, which has been widely used in many fields such as information extraction and

text classification. Certain of the research works about semantic similarity computation are based on HowNet.

In Ref. [10], Bai et al. presented an improved algorithm for detecting the semantic similarity based on HowNet. The method is built on the atom axonomy tree, and calculates the semantic similarity based on the atom distance. It considers the influence of the children's node density under the common parent node, and the similarity calculation between words of polysemy terms.

In Ref. [11], Zhu et al. proposed a word semantic similarity computation method based on the HowNet. They made full use of the semantic information of words in different knowledge networks to obtain a more accurate and reasonable similarity.

In Ref. [12], Nie et al. proposed a new semantic similarity detecting method based on HowNet. They consider the ordering of the weights of the "Sememe classes" and set a function to make the weights change moderately. Moreover, the authors proposed the element matching method for the similarity between two texts.

In Ref. [13], Zhang et al. proposed a word semantic similarity calculation method combining HowNet and search engine by making full use of rich network knowledge. They used double correlation detection algorithm and pointed mutual information method based on search engines to improve the match degree of the semantic description of the specific word and subjective cognition of vocabulary.

In Ref. [14], it presented a new depth & path-based semantic similarity method to improve the existing meaning-based approaches in HowNet. The authors construct a complete concept tree according to the concept definitions in HowNet and use the improved depth & path algorithm to compute the depth and path of concept in the concept tree.

A brief survey of related works proposed above is listed in Table 3. Two ways to improve the semantic similarity detecting algorithms: (i) an improved algorithm is often proposed according to the structure or the internal feature of the HowNet, for example [10, 12, 14], and (2) researchers introduce external information to optimize the algorithm. For example, other knowledge bases [11] or related information like search engine [13].

Table 3. A brief survey of the related work.

Research Works	Main contributions
Bai et al. [10]	Introduce the influence of sememe density; Reconsider the multi-meaning words
Zhu et al. [11]	Base on the HowNet CiLin; Propose a dynamic weighting strategy to calculate semantic similarity
Nie et al. [12]	Define a new semantic similarity; Propose an element matching method
Zhang et al. [13]	Combine HowNet and search engine algorithms
Guo et al. [14]	Construct a concept tree; Propose an improved depth & path algorithm

This paper introduces a novel method to measure the semantic similarity between two words. Different from existing research works, SSDH further supports to compare the semantic similarity between two paragraphs of text.

5 Conclusion

This paper proposes SSDH method, which is built on HowNet and implemented with MongoDB, to calculate the semantic similarity between words. Since HowNet is a bilingual knowledge base, SSDH supports language-cross (English - Chinese) semantic similarity detecting among concepts.

There are many potential relationships maintained in HowNet, such as the relevance of words. As mentioned above, HowNet is a networked organic knowledge system. In the HowNet knowledge system, in addition to the basic semantic relationship between words and words, there is also the relationship between possession, event implementer and event enforcer.

Based on these relationships, Knowledge Network can also detect word relevance. Figure 5 shows an example between word "community" and word "home-owner". Although they all have "#house" in their concept, their similarity is not high because the first attribute of "community" is "house" while the first attribute of "home-owner" is "human", and their relevance is high because the houses in the community are owned by the home-owners.

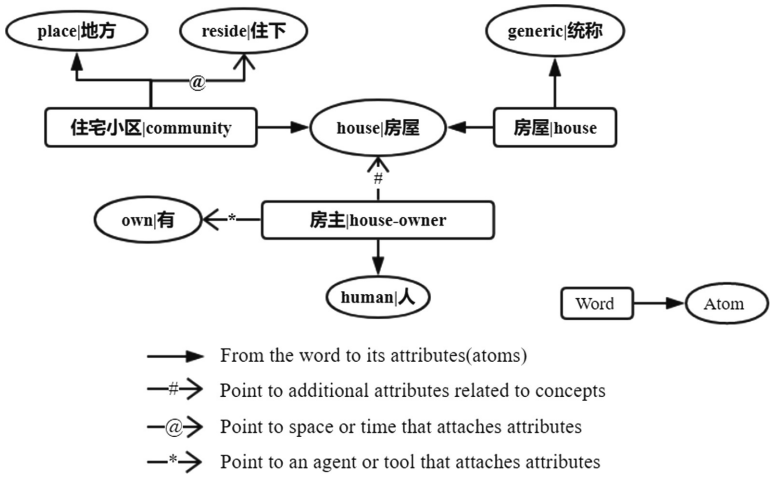

Fig. 5. The connection among "community", "house" and "house-owner".

The future research direction of SSDH will focus on word correlation detection to achieve more accurate requirements document analysis and a wider range of related content retrieval.

Acknowledgement. This work was supported by the "Fundamental Research Funds for the Central Universities Nos. 3082018NS2018057", the National Natural Science Foundation of China (61872182), the National Natural Science Foundation of China under Grant No.61402229 and No.61602267, the Open Fund of the State Key Laboratory for Novel Software Technology (KFKT2018B19) and the Open Fund of the Ministry Key Laboratory for Safety-Critical Software Development and Verification (1015-XCA1816401).

References

1. Banker, K.: MongoDB in Action. Manning Publications Co., Shelter Island (2011)
2. Brad, D.: The Definitive Guide to MongoDB: The Nosql Database for Cloud and Desktop Computing. Springer, Heidelberg (2010). https://doi.org/10.1007/978-1-4302-3052-6
3. Vohra, D.: Migrating mongodb (2015)
4. Boicea, A., Radulescu, F., Agapin, L.I.: MongoDB vs oracle – database comparison. In: Emerging Intelligent Data and Web Technologies (EIDWT), 2012 Third International Conference. IEEE (2012)
5. Dong, Z., Dong, Q.: HowNet - a hybrid language and knowledge resource. In: Natural Language Processing and Knowledge Engineering, 2003. Proceedings. 2003 International Conference. IEEE (2003)
6. Xu, Y., Fan, X.Z., Zhang, F.: Semantic relevancy computing based on hownet. J. Beijing Inst. Technol. **25**(5), 411–414 (2005)
7. Hows, D., Membrey, P., Plugge, E., Hawkins, T.: Installing mongodb. In: Definitive Guide to Mongodb, pp. 17–31 (2013). http://doi.org/10.1007/978-1-4302-5822-3_2
8. Miller, G.A.: Wordnet: a lexical database for English. Commun. Assoc. Comput. Mach. **38**(11), 39–41 (1995)
9. Diao, L., Yan, H., Fuxue, L.I., Xiumin, L.I., Lei, G.: An improved HowNet-based algorithm for semantic similarity computation (2017)
10. Bai, J., Bu, Y.: An improved algorithm for semantic similarity based on HowNet. In: 2018 2nd International Conference on Data Science and Business Analytics (ICDSBA), pp. 65–70 (2019)
11. Zhu, X.H., Ma, R.C., Sun, L., Chen, H.C.: Word semantic similarity computation based on HowNet and CiLin. J. Chin. Inf. Process. **30**(4), 29–36 (2016)
12. Nie, H., Zhou, J., Guo, Q., Huang, Z.: Improved semantic similarity method based on HowNet for text clustering. In: 2018 5th International Conference on Information Science and Control Engineering (ICISCE), pp. 266–269. IEEE (2018)
13. Zhang, S., Ouyang, C., Yang, X., Liu, Y., Liu, Z., et al.: Word semantic similarity computation based on integrating HowNet and search engines. J. Comput. Appl. **37**, 1056–1060 (2017)
14. Guo, X., Zhu, X., Li, F., Li, Q.: A new semantic similarity measurement based on HowNet concept tree. In: 2016 International Forum on Management, Education and Information Technology Application. Atlantis Press (2016)