Abstract Concept Instantiation with Context Relevance Measurement

Shengwei Gu[1,2], Xiangfeng Luo[1,5,*], Hao Wang[1,5,*], Jing Huang[4] and Subin Huang[1,3]

[1]*School of Computer Engineering and Science, Shanghai University, Shanghai, China*
[2]*School of Computer and Information Engineering, Chuzhou University, Chuzhou, China*
[3]*School of Computer and Information, Anhui Polytechnic University, Wuhu, China*
[4]*Ant Financial Services Group, Hangzhou, China*
[5]*Shanghai Institute for Advanced Communication and Data Science, Shanghai University, Shanghai, China*
E-mail: gushengwei@shu.edu.cn; luoxf@shu.edu.cn; wang-hao@shu.edu.cn; huangsubin@shu.edu.cn; tom.hj@antfin.com
[*]*Corresponding Author*

Received 02 May 2020; Accepted 20 July 2020;
Publication 17 September 2020

Abstract

In different contexts, one abstract concept (e.g., fruit) may be mapped into different concrete instance sets, which is called abstract concept instantiation. It has been widely applied in many applications, such as web search, intelligent recommendation, etc. However, in most abstract concept instantiation models have the following problems: (1) the neglect of incorrect label and label incompleteness in the category structure on which instance selection relies; (2) the subjective design of instance profile for calculating the relevance between instance and contextual constraint. The above problems lead to false prediction in terms of abstract concept instantiation. To tackle these problems, we proposed a novel model to instantiate the abstract concept. Firstly, to alleviate the incorrect label and remedy label incompleteness in

Journal of Web Engineering, Vol. 19_5–6, 575–602.
doi: 10.13052/jwe1540-9589.19562

the category structure, an improved random-walk algorithm is proposed, called InstanceRank, which not only utilize the category information, but it also exploits the association information to infer the right instances of an abstract concept. Secondly, for better measuring the relevance between instances and contextual constraint, we learn the proper instance profile from different granularity ones. They are designed based on the surrounding text of the instance. Finally, noise reduction and instance filtering are introduced to further enhance the model performance. Experiments on Chinese food abstract concept set show that the proposed model can effectively reduce false positive and false negative of instantiation results.

Keywords: Abstract concept instantiation, contextual constraint, instance ranking.

1 Introduction

Abstract concepts are the basic blocks of thought which broadly exist in natural language. They contribute descriptions of the real-world things and their mutual relationships [1, 2]. However, in many applications (e.g., web search, intelligent recommendation), it need to map an abstract concept into different concrete instance sets because of the different contextual constraints [3, 4, 5]. As shown in Figure 1, the abstract concept *"fruit"* refers to the instance set {*kiwi, apple, strawberry, grape,* $\cdots$ } in the context of *"fruit rich in vitamin E"*; but for *"fruit that are good for the eyes"*, the referred instance set is {*kiwi, lemon, orange, pineapple,* $\cdots$}. In this work, we focus on the task of abstract concept instantiation, which maps an abstract concept under contextual constraint into a set of concrete instances.

In recent years, there have been many excellent works in the research on abstract concept instantiation. Most models [6, 7, 8, 9, 10] are divided

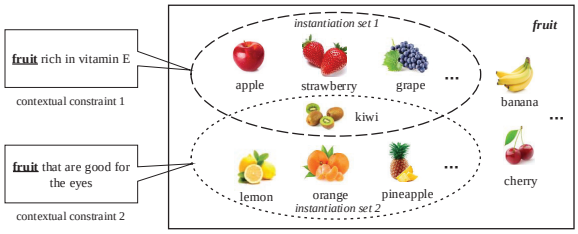

Figure 1 An example of instantiation of the abstract concept *"fruit"*. The abstract concept *"fruit"* is mapped into different instance sets under the different contextual constraints.

into two steps: candidate instance ranking and context relevance measurement. Candidate instance ranking is applied to estimate the possibility that candidate instances belong to the given abstract concept. Context relevance measurement is a method to measure the relevance between the instance and contextual constraint. For the former, encyclopedia category structure has been widely exploited [6, 7, 9, 10, 11]. However, the incorrect label and label incompleteness problems in the category structure are not taken into consideration. For example, 34% instances of the abstract concept *"liquor"* are mislabeled in HudongBaike[1]. For the latter, the instance profile plays an important role in calculating the relevance between the instance and contextual constraint. Various instance profiles have been adopted in previous works, such as sentence [6], document [8, 9, 10] or their combinations [7]. But to the best of our knowledge, few detailed comparative researches have been done in learning proper instance profile. In practice, if the instance profile is too large (e.g., a web document), it may contain more irrelevant instances and reduce the precision of abstract concept instantiation. Conversely, if the instance profile is too small (e.g., a sentence containing the instance), it will reduce the recall of abstract concept instantiation.

To remedy the problems above, we propose a novel model based on context relevance measurement to instantiate the abstract concept. Firstly, our model calculates the possibility of candidate instances belonging to the abstract concept not just by the encyclopedia category information but also by the association information in the text. Secondly, we design different instance profiles based on the surrounding text of the instance. On the basis of this, the relevance between the instance and contextual constraint is measured by the document model. Finally, noise reduction and instance filtering are performed to further improve the model performance. Experiments on Chinese food abstract concept set reveal that the proposed model effectively reduces false positive and false negative of instantiation results.

To summary, this work makes two contributions as follows.

(1) An InstanceRank algorithm is proposed to rank the candidate instances of the abstract concept. In this algorithm, category and association information complement each other, which effectively alleviate the problem of incorrect label and label incompleteness in the encyclopedia category structure.

[1]HudongBaike is the largest encyclopedia in Chinese, which can be accessed at: http://www.baike.com.

(2) To learn the proper instance profile, a systematic comparison of context relevance measurement is conducted on different granularity instance profiles. The proper instance profile contributes a lot to improve the performance of the proposed model.

The remainder of this paper is organized as follows: related work is introduced in Section 2; the proposed model of abstract concept instantiation is presented in Section 3; experimental results are shown in Section 4; the conclusion and future work are given in Section 5.

2 Related Work

There are many works related to the abstract concept instantiation, such as entity ranking, entity search, query suggestion, etc. Existing models can be divided into three categories: probability-based models, graph-based models, and machine learning based models.

Probability-based models. These models aim at modeling the relevance between the instances and contextual constraint through the bridge of documents. A series of works have explored the methods of establishing the correlation between them. Balog et al. [7, 12] represent the instances and contextual constraint as a distribution over terms and categories. The relevance between them is measured by the KL divergence. Since the category name has a short length, it is inefficient in the representation of probability distribution. To address this problem, literature [6, 9, 13] exploit the category and link structure of Wikipedia to predict the instance type. Specifically, Kaptein et al. [9] employ the category structure of Wikipedia to retrieve the instance and relevant documents. Chen et al. [6] utilize the head matching to measure category matching and achieve significant improvement over existing methods. In addition, Sun et al. [3] present a prototype system to detect and instantiate the abstract concepts in the web search query. This system takes semantic similarity, context similarity, and quality of suggesting query into consideration to calculate the instances ranking score.

Graph-based models. In graph-based models, instances are treated as nodes, and associations between them are treated as edges. Random-walk model is widely used to rank the candidate instances based on their relevance to the given type. Tsikrika et al. [14] propose a K-step random-walk model to find the related instances of expected type. The model works based on the relevance propagation between the descriptions of instances in Wikipedia. Tonon et al. [15] predict the instance type based on collection statistics and graph structure of instances and types. Gori et al. [16] present a biased

PageRank method to rank the favorite movies of users according to the correlation between movie pairs and user preferences.

Machine learning models. Based on machine learning models, related studies on abstract concept instantiation have been carried out. Fang et al. [17] propose a discriminative learning model, in which a variety of documentary evidence and document-candidate association features are integrated into a single model for the relevant experts. Uchida et al. [18] employ a fuzzy ranking support vector machine to compute the relevance between the search attributes and experience attributes. In recent years, external knowledge and entity relationship are integrated into the models to solve the problem of entity search [19]. In addition, to handle the burden of labeling large quantities of training data, Andrew et al [20]. and Xu et al. [21] use the multiple instance learning (MIL) classier to identify the most representative concept of the instances.

These models have made great improvements, but the incorrect label and label incompleteness problems in the category structure are ignored. Moreover, few detailed comparative researches have been done in learning proper instance profile, which is crucial to context relevance measurement. In this paper, we proposed a novel abstract concept instantiation model to make up the above drawbacks.

3 The Proposed Model

To solve the problems pointed out in Section 1, a novel abstract concept instantiation model is proposed. As shown in Figure 2, there are three components in the proposed model. (1) In candidate instance ranking (Section 3.1), both the category information and the association information are considered when calculating the possibility of the instance belongs to the abstract concept. (2) In context relevance measurement (Section 3.2), based on the instance profile, the relevance between the instance and contextual constraint is calculated by the language model. (3) In instances purification (Section 3.3), noise reduction and instance filtering are performed to further improve the model performance. Each component of the proposed model will be discussed in detail below.

3.1 Candidate Instance Ranking

In this section, InstanceRank algorithm is proposed to rank the candidate instances of the given abstract concept. To implement this algorithm, an

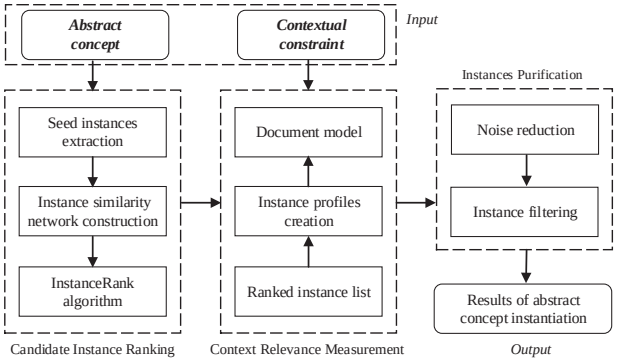

Figure 2 An overview of the proposed model. The inputs of the model are indicated in bold italics. Candidate instances of the given abstract concept are represented by the vertices of instance similarity network.

Instance Similarity Network (*ISN*) is firstly constructed. Then, based on *ISN*, we utilize InstanceRank to compute the possibility that candidate instances belong to the given abstract concept. In Appendix 1, an example is given to explain the motivation of InstanceRank algorithm.

3.1.1 Instance similarity network

The instance similarity network models the relationship between candidate instances of an abstract concept. The weight of the edge is defined as the linear combination of category similarity and association similarity of the candidate instances pairs. By utilizing the complementarity of category and association information between the *ISN* nodes, InstanceRank can rectify the incorrect label and label incompleteness problems in the category structure.

Formally, ISN can be denoted as:

$$ISN = \{E, L_E, R_E\} \tag{1}$$

where $E = \{e_i | i = 1, 2, ..., n\}$ is the vertex set (i.e., candidate instance set of the abstract concept); L_E and R_E are the weighted edges with category and association similarities, respectively. In addition, in the following, the category similarity weight matrix of L_E is represented by $L = [l(e_i, e_j)]_{n \times n}$, and the association similarity weight matrix of R_E is denoted by $R = [r(e_i, e_j)]_{n \times n}$. Category and association similarities constitute the similarity between the candidate instances. The process of ISN construction is described below.

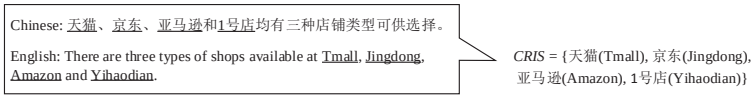

Figure 3 An example of the *CRIS*. In Chinese, items (underlined) of *CRIS* are usually separated by Chinese back-sloping comma "、", "和" (or) and "与" (and). The members of *CRIS* usually belong to the same category.

Vertex Set

The elements of vertex set E are composed of two parts. First, titles of HudongBaike pages with the category of the concept c or its sub-concepts are included. Second, due to the problem of incorrect label and label incompleteness in HudongBaike, some instances with coordination relation are selected from the text and added to E. Specifically, in a Chinese sentence, items separated by Chinese back-sloping comma ",", "和" (or) and "与" (and) usually belong to the same category.[2] We call such items have a coordination relationship, and the set they constitute is Coordination Relation Item Set (*CRIS*). Figure 3 shows an example of extracting *CRIS* form a Chinese sentence. In practice, for an abstract concept c, we select elements from the *CRISs* in which at least one seed instance of c is included. Selecting candidate instances from *CRIS*, on the one hand, can supplement unlabeled instances in the encyclopedia; on the other hand, according to the co-occurrence relationship, candidate instances with incorrect labels can be excluded.

Consequently, the vertex set E is:

$$E = \{e|e \in \cup \textit{title}, title\ with\ catgoris\ c^*\} \cup \{e|e \in \textit{CRIS}, \textit{CRIS} \cap S_c \neq \emptyset\} \tag{2}$$

where c^* denotes the abstract concept c or its sub-concepts, S_c is seed instance set of the abstract concept c. c^* is selected from the "is-a" taxonomy constructed in category similarity section. The seed instance set is obtained by using the Hearst Pattern [22, 23] because of its high accuracy (see Appendix 2 for details on seed instance selection).

Category Similarity

Category similarity reflects the relation of categories between two candidate instances. In this work, we evaluate the category similarity of two instances based on the length of the shortest path in the "is-a" taxonomy.

Referring to the methods in [24], we construct an "is-a" taxonomy using the HudongBaike category structure, named HudongBaike Taxonomy (*HBT*).

[2] https://zh.wikipedia.org/wiki/%E9%A0%93%E8%99%9F

To the best of our knowledge, there is no precise formula for calculating the category similarity of instance pairs. In order to reduce the difficulty of calculation, and inspired by [8], we define the category similarity of instance pair e_i and e_j as follows:

$$l(e_i, e_j) = \begin{cases} \frac{1}{d(e_i,c)+d(e_j,c)}, & e_i, e_j \; has \; a \; path \; in \; \textbf{HBT} \; and \; c \in C_{e_i} \cup C_{e_j} \\ 0.01, & others \end{cases} \tag{3}$$

where $d(e_i, c)$ denotes the length of the shortest path between the candidate instance e_i and the abstract concept c in the ***HBT***; $d(e_j, c)$ has the same definition as $d(e_i, c)$; C_{e_j} and C_{e_j} are category set of instance corresponding e_i to e_j, respectively. For the second case in formula (3), we assume that e_i and e_j have a common concept *"Object"* and assign $l(e_i, e_j)$ to 0.01.

Association Similarity

Association similarity indicates the co-occurrence relation between the candidate instances. First, the association similarity between candidate instance and seed instance is defined as:

$$r(e_i, e_s) = \frac{count(e_i, e_s) + 1}{\sum_{e_j} count(e_j, e_s) + 1} \tag{4}$$

where $e_i \in E$, $e_s \in S_c$; $count(e_i, e_s)$ denotes the number of ***CRISs*** which contain instance e_i and e_s. $\sum_{e_j} count(e_j, e_s)$ represents the total number of ***CRISs*** that contain instance e_s. We use Laplace smoothing to avoid assigning zero weights to unseen pairs. If e_i does not co-occur with any seed instance, $r(e_i, e_s)$ is initialized to zero. In formula (4), the purpose of using only edges between candidate instances and seed instances is to reduce the irrelevant instances.

Based on instance similarity network, it can be observed that: given a candidate instance, the higher category similarity it has with the other candidate instance, the more likely they belong to the same abstract concept; the higher association similarity it has with seed instances in the ***CRISs***, the more likely it belongs to the abstract concept of seed instances.

3.1.2 InstanceRank Algorithm

InstanceRank is an improved random-walk algorithm, which can estimate the possibility that candidate instances belong to the abstract concept based on the information expressed by ***ISN***.

The calculation of InstanceRank is an iterative process. Inspired by the definition of ItemRank [16], the iteration equation is defined as:

$$s_n = \alpha \cdot M \cdot s_{n-1} + (1 - \alpha) \cdot b \tag{5}$$

where s_n denotes the scores of at the n-th iteration; M denotes transition matrix, and its non-negative entries has to sum up to 1 for every column; b denotes the normalized vector of seed instances of the given abstract concept, that is, its non-negative entries sum up to 1; α is the decay factor. According the definition of *ISN*, transition matrix M is calculated as follows:

$$M = \beta \cdot R^* + (1 - \beta) \cdot L^* \tag{6}$$

where R^* and L^* denote the normalized R and L; β is the tuning parameter. It can be seen from formula (6) that the elements of the transition matrix M are a linear combination of category similarity and association similarity. The reason for this definition is that the two similarities can complement each other, and effectively alleviate the problem of incorrect label and label incompleteness in the encyclopedia category structure. An example is given in Appendix 1 to illustrate the motivation of transition matrix design.

Combining the equations (5) and (6), the iteration equation of InstanceRank is given by:

$$s_n = \alpha \cdot [\beta \cdot R^* + (1 - \beta) \cdot L^*] \cdot s_{n-1} + (1 - \alpha) \cdot b \tag{7}$$

After the equation (7) converges or meets a fixed number of iterations, the rank scores of instances s^* can be obtained. The InstanceRank algorithm is shown in Algorithm 1.

Lines 1–5 initialize the score vector and seed vector, and they are all assigned equal average values. In line 6, the category similarity matrix and the association similarity matrix are normalized, so that the sum of each column of the transition matrix is 1. In lines 7–9, iterative calculation is performed, and the result is recorded as s^* on line 10. Lines 11-14 of Algorithm 1 indicates that if a seed instance is not contained in E, we assign it with an average score (denotes by s_e) of the seed instances that are contained in E. We do this because seed instances belong to the abstract concept with a high probability. Finally, each instance in E and S_c is assigned a rank score. The rank score is considered the possibility that candidate instance e belongs to the abstract concept c.

Algorithm 1 InstanceRank

Input: candidate instance set E, category similarity weight matrix L, association similarity weight matrix R, seed instance set S_c, decay factor α, adjustment parameter β, number of iterations N

Output: rank scores s

$s_0 = [(\frac{1}{|E|})]_{1\times|E|}$, $b = [(0)]_{1\times|E|}$
for $e \in (S_c \cap E)$ **do**
    $b(e) = 1$
end for
$b = \frac{b}{|S_c \cap E|}$
$R^* = normalized(R)$, $L^* = normalized(L)$
for $n = 1$ *to* N **do**
    $s_n = \alpha \cdot [\beta \cdot R^* + (1-\beta) \cdot L^*] \cdot s_{n-1} + (1-\alpha) \cdot b$
end for
$s^* = s_N$
for $e \in S_c$ and $e \notin E$ **do**
    $s_e = average(S_c \cap E)$
    $s = s^* \cup s_e$
end for
return s

3.2 Context Relevance Measurement

In this section, the relevance between the instance and contextual constraint is measured with the document model [25]. However, in the document model, it is inefficient to represent the instance only using its name phrase due to poor word-wise proximity. Therefore, a key point is how to learn a proper instance profile to represent an instance in a document.

Generally, the surrounding text of instance provides the evidence how an instance is relevant to the contextual constraint [9]. In this work, the surrounding text of instance will be represented as an instance profile. We create it from three levels: sentence, paragraph, and document.

For sentence level, we design the instance profile with an instance-centric strategy that gradually expands the context by symmetrically. Based on this strategy, we built three instance profiles: *OneSent*, *ThreeSent*, *FiveSent*. *OneSent* indicates a sentence that the instance is contained. *ThreeSent* indicates three sequential sentences that the instance is contained in the middle one. *FiveSent* has a similar definition to *ThreeSent*. For paragraph level, since paragraph is an independent unit of writing that deals with a particular point or idea[3], we regard the paragraph containing the instance as a type of instance

[3] https://en.wikipedia.org/wiki/Paragraph

profile, namely *Para*. For document level, we treat a single web document as an instance profile, represented by *Text*. Summing up, five granularity instance profiles are considered in this work: *OneSent*, *ThreeSent*, *FiveSent*, *Para* and *Text*.

Formally, given an instance e and corpus $D = \{d_1, d_2, ..., d_n\}$, where d_i is a document in the corpus. The five granularity instance profiles of instance e are defined as follows:

$$OneSent = \{s_{ij} | e \in s_{ij}, s_{ij} \in d_i, d_i \in D\} \tag{8}$$

$$ThreeSent = \{s_{ij-1}, s_{ij}, s_{ij+1} | e \in s_{ij}, s_{ij-1:ij+1} \in d_i, d_i \in D\} \tag{9}$$

$$FiveSent = \{s_{ij-2}, s_{ij-1}, s_{ij}, s_{ij+1} s_{ij+2} | e \in s_{ij}, s_{ij-2:ij+2} \in d_i, d_i \in D\} \tag{10}$$

$$Para = \{p_{ik} | e \in p_{ik}, p_{ik} \in d_i, d_i \in D\} \tag{11}$$

$$Text = \{d_i | e \in d_i, d_i \in D\} \tag{12}$$

where s_{ij} and p_{ik} are the $j\text{-}th$ sentence and the $k\text{-}th$ paragraph in d_i, respectively. The instance profile set of e can be expressed as follows:

$$IPS_e = \{OneSent, ThreeSent, FiveSent, Para, Text\} \tag{13}$$

On the basis of the instance profile, we then calculate the relevance between the instance and contextual constraint. In the following section, $P(e|T)$ represents the relevance between the instance e and the given contextual constraint T. Suppose that $P(e)$ is uniformly distributed, since contextual constraint T is given and thus $P(T)$ is fixed. According to the Bayes' theorem:

$$P(e|T) = \frac{P(T|e)P(e)}{P(T)} \propto P(T|e) \tag{14}$$

Based on the document model, $P(T|e)$ can be reformulated as follows:

$$P(T|e) = \sum_{IP_e \in D_e} P(T|IP_e) \times P(IP_e|e) \tag{15}$$

where D_e is a type of instance profile of instance e, that is $D_e \in IPS_e$. $P(T|IP_e)$ is the probability of generating the contextual constraint T from the IP_e. $P(IP_e|e)$ is the probability of generating the IP_e from the instance e. Following [6], the component $P(T|IP_e)$ is calculated as follows:

$$P(T|IP_e) = \prod_{t \in T} [(1-\lambda)p(t|IP_e) + \lambda p(t|D)]^{n(t,T)} \tag{16}$$

where $P(t|IP_e)$ is the probability of generating the term t from the instance profiles IP_e, and $p(t|D)$is the background probability of term t in corpus D. λ is the smoothing parameter. $n(t, T)$ is the number of times of t appears in T.

The $P(IP_e|e)$ reflects the association between e and IP_e. It can be evaluated as follows:

$$P(IP_e|e) = \frac{Count(IP_e, e)}{\sum_{w_i \in IP_e} Count(IP_e, w_i)} \tag{17}$$

where $Count(IP_e, e)$ indicates the number of instances e in instance profiles IP_e, and the denominator $\sum_{w_i \in IP_e} Count(IP_e, w_i)$ denotes the number of terms in instance profiles IP_e.

To sum up, we denote the possibility of candidate instance e belongs to abstract concept c as $P(e|c)$, and assume that c and T are independent. Under contextual constraint T, the possibility of mapping the abstract concept c to the candidate instance e is calculated as follows:

$$f_{c,T}(e) = P(e|c) \times P(e|T) \propto P(e|c) \times P(T|e) \tag{18}$$

In formula (18), we can obtain $P(e|c)$ by InstanceRank algorithm and calculate $P(e|T)$ by the formula (15). In subsection 4.3.1, we will explore the impact of different granularity instance profiles on our model performance and make a systematic comparison.

3.3 Instances Purification

Based on the formula (18), a list of ranked instances for abstract concept is obtained, but there are still irrelevant instances in it. To further improve the model performance, noise reduction and instance filtering are performed.

3.3.1 Noise Reduction

In this subsection, the distribution characteristic of instances in the semantic space is used to carry out noise reduction.

In the semantic space, the distribution of the instances that belong to the same concept is relatively concentrated [5, 26, 27]. Conversely, in the absence of the same concept constraint, the distribution of noise is relatively scattered. To characterize, instances are encoded into real value vectors using word2vec [28, 29] on the Chinese HudongBaike corpus. Then, the degree of aggregation between candidate instance and seed instances is calculated to perform noise

reduction. We define the degree of aggregation of e_i as follows:

$$d(e_i) = \frac{1}{|S_C|} \sum_{v_s} g(v_i, v_s) \tag{19}$$

where v_i and v_s are the vector representations of e_i and $e_s \in S_c$, respectively. $|S_c|$ is the size of seed instance set. In this work, we define g as the cosine similarity function. If the $d(e_i)$ is smaller than a predefined threshold δ, e_i will be removed.

In formula (19), instead of $g(v_i, c)$, we use $g(v_i, v_s)$ to calculate the degree of aggregation. In that case, the results will bias to instances with similar context to the abstract concept, because they have similar word embedding. In addition, noise reduction can also be regarded as a voting method. This method performs well under less noisy dataset [30]. Instances in formula (19) meet this requirement, because after candidate instance ranking and context relevance measurement, the top ranked instances contain only a small amount of noise.

3.3.2 Instance Filtering

After performing noise reduction, instance filtering is conducted according to synonym dictionary and manual rules.

(1) Merging synonyms. In the text, there are a variety of ways to refer to the same real-world instance. For example, "西红柿" and "番茄" (both are tomato). In this paper, CilinE[4] [31] is used to merge synonyms. For the sake of simplicity, we only reserve the instance with the highest score if there are synonym ones in the ranked instance list.

(2) Limiting instance length. Generally, for the given category, if an instance name is too long or too short, it may be an incorrect one. Some manual rules are defined to filter instances. For example, the food product names are limited to 2-8 characters, the animal and plant names are limited to 2-6 characters.[5]

4 Experiments

Experimental evaluations are designed from two perspectives: (1) compare the performance of the proposed model with baselines (Section 4.2); (2)

[4]CilinE is a well-known Chinese synonym dictionary and can be obtained from https://www.ltp-cloud.com/download.

[5]97% food product names in JD.com (https://channel.jd.com/food.html) meet the first rule; 95% animal and plant names in HudongBaike (http://www.baike.com/) meet the second rule.

Table 1 Details and roles of corpora.

Corpus Name	Data	Role in Proposed Model
HudongBaike	12,474,843 instance-category pairs	Construct "is-a" taxonomy
	5,520,343 documents and obtain word embedding	Train the word2vec model
TechFood News	663,629 documents	Extract the *CRIS*[7]

Table 2 Details of evaluation dataset.

Abstract Concept	Contextual Constraint	Number of Related Articles
Fruit	Fruit rich in vitamin E	1,311
Fruit	Fruit that are good for the eyes	120
Liquor brand	Liquor affected by plasticizer incident	1,318
Liquor brand	Liquor affected by public consumption policy	1,122
Fish	Fish containing DHA	717
Tea	Tea of benefit to blood pressure	428

analyze the effect of components on the overall performance and validate the robustness of the proposed model (Section 4.3). In the results, the best values are in bold, and the second-best values are underlined.

4.1 Experiment Settings

Corpora. The corpora used in our experiments are crawled from HudongBaike and TechFood News[6]. The details and roles of the corpora are listed in Table 1.

Evaluation Dataset. Since there is no public evaluation dataset available for Chinese abstract concept instantiation. We select six cases randomly from search suggestions given by Baidu[8] to build the evaluation dataset. The benchmark of abstract concept is labeled by three evaluators who consult the related articles from TechFood News. The details of evaluation dataset are shown in Table 2.

Parameter Settings. The parameters are set as follows: (1) in InstanceRank algorithm, the decay factor $\alpha = 0.3$; (2) in context relevance measurement, the smoothing parameter $\lambda = 0.5$ (See Appendix 3 for tuning details); (3) in noise reduction, the threshold $\delta = 0.3$. In addition, because

[6]https://www.tech-food.com/news/.

[7]1,680,134 *CRISs* are extracted from the TechFood News dataset.

[8]https://www.baidu.com/.

Table 3 Performance on different models. Results show that our model outperforms all baselines by a large margin. Our results are obtained under β = 0.2 and *Para* instance profile.

Model	p@10	p@20	p@30	p@40	R-Pre	MAP	NDCG
M4	0.260	0.260	0.227	0.185	0.214	0.162	0.413
M5	0.360	0.290	0.233	0.195	0.255	0.181	0.432
M6	0.320	0.270	0.227	0.215	0.208	0.163	0.406
M7	0.300	0.260	0.220	0.175	0.208	0.157	0.406
LC	0.540	0.430	0.320	0.265	0.403	0.358	0.605
SC	0.220	0.280	0.207	0.170	0.270	0.156	0.360
LCR+SCR	0.300	0.320	0.260	0.205	0.253	0.313	0.469
Our model	**0.700**	**0.510**	**0.387**	**0.330**	**0.519**	**0.531**	**0.750**

the parameters β is important, we will tune it in subsection 4.3.1 to obtain the optimal value.

Evaluation Metrics. The proposed model is evaluated against four metrics: Precision@K (p@K), R-Precision (R-Pre), Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (NDCG). These metrics are widely used in information retrieval.

4.2 Performance Comparison

Baselines. We compare our model with the following baselines. Among these, M4-M7 are selected from the paper [10] because of their better performance. Long-Range Context (LC), Short-Range Context (SC), and Long-Range Context Reranking plus Short-Range Context Reranking (LCR+SCR) are chosen from [6]. In particular, the LCR+SCR model exhibits state-of-the-art results. These models are reimplemented according to the original papers since there is no publicly available source code.

Comparison Results. Experimental results are shown in Table 3. We can find that the proposed model significantly outperforms the comparison models.

As Table 3 shows, among the M4-M7 models, M5 works best in terms of overall performance, but underperforms than LC and our model. This is mainly because the strategy to generate the instance context from the short concept phrase by the generative model is not effective. Because the "head matching" method, the LC model has achieved a great improvement on overall performance, and performs best in all baselines. However, the LC model ignores the problem of incorrect label and label incompleteness of category structure, resulting in suboptimal results compared to our model.

Table 4 Effects of different β on p@K. When β = 0.2, our model achieves the best overall performance.

β	p@5	p@10	p@15	p@20	p@25	p@30	p@35	p@40
0.0	**0.776**	0.688	0.576	0.490	0.426	0.376	0.347	0.315
0.2	**0.776**	**0.696**	**0.592**	**0.502**	**0.430**	0.381	**0.349**	**0.324**
0.4	0.752	0.676	0.571	0.488	0.424	0.377	0.346	0.320
0.6	0.768	0.680	0.549	0.494	0.422	0.387	0.345	0.321
0.8	0.760	0.644	0.533	0.486	0.416	**0.388**	**0.349**	0.319
1.0	0.728	0.644	0.525	0.472	0.418	0.387	0.346	0.316

Although reranking is an effective strategy, LCR+SCR does not achieve the expected results due to the poor LC performance.

4.3 Model Analysis

To investigate the effectiveness of our model design, we analyze it from two aspects: (1) effect of components on overall performance (subsection 4.3.1); (2) robustness of the model over different numbers of seed instances (subsection 4.3.2).

4.3.1 Effect of Model Components

Effect of Category and Association Similarity. In this experiment, we investigate the effect of category and association similarity by tuning their proportions. The β value controls the proportion of category similarity and association similarity in the InstanceRank algorithm. Table 4 shows the results of p@K scores at different β values.

To eliminate the influence of instance profile, results in Table 4 are the average of overall performance on the five granularity instance profiles. As shown in Table 4, only using category similarity (β = 0) or association similarity (β = 1) cannot achieve the best performance, especially in the range of p@10$\sim$p@25. When β equals 0.2, our model performs best. It indicates that the association similarity can be effectively complementary to the category similarity.

Effect of Instance Profile. In this experiment, we investigate the effect of different instance profiles designed in Section 3.2. The comparison results are listed in Table 5.

To obtain the unbiased results, the scores in Table 5 are the average of the model performance under different β (β = 0, 0.2, 0.4, 0.6, 0.8, 1). As shown in Table 5, we can find that the proposed model with *Para* instance

Table 5 Performance with different granularity instance profiles. *Para* performs best among the five granularity instance profiles.

Instance Profile	p@5	p@10	p@15	p@20	p@25	p@30	p@35	p@40
OneSent	0.780	**0.690**	0.556	0.482	0.401	0.368	0.328	0.303
ThreeSent	0.767	0.683	0.538	0.485	0.427	0.389	**0.362**	0.327
FiveSent	0.707	0.660	0.533	0.468	0.417	0.377	0.343	0.321
Para	**0.827**	0.683	**0.596**	0.502	0.432	0.388	0.355	**0.329**
Text	0.720	0.640	0.567	**0.507**	**0.436**	**0.392**	0.348	0.317

profile performs best among the five granularity instance profiles. Although the proximity between the instance and the constraint terms is well preserved (e.g., p@10) in *OneSent*, the coverage of the instance is significantly reduced (e.g., p@25 $\sim$ p@40). The instance profile of *Text* can cover all instances in the web document and obtains the best scores of p@20 $\sim$ p@30. But the p@5 and p@10 scores are lowered due to excessive noisy instances introduced. Experimental results in Table 5 suggest that *Para* instance profile is a suitable choice to balance the precision and recall in the context relevance measurement.

Effect of Noise Reduction. In this experiment, we investigate the effect of noise reduction on the overall performance. Figure 4 shows the comparison results of the MAP and NDCG scores with and without the noise reduction. MAP-NR and NDCG-NR represent the results without the noise reduction process. In contrast, MAP+NR and NDCG+NR indicate that noise reduction has been performed. As shown in Figure 4, the noise reduction is beneficial for improving the MAP and NDCG scores of our model.

4.3.2 Model Robustness Analysis

In subsection 3.1.1, Hearst Pattern method is used for obtaining the seed instances of the abstract concept. Literature [32] shows that the method of Hearst Pattern can achieve higher accuracy, but suffers from the lower coverage. The question is raised whether the proposed model is robust to the number of seed instances.

For different abstract concepts, the number of seed instances obtained by Hearst Pattern is different. It is not appropriate to set a uniform number of seed instances to evaluate the proposed model. In this work, we replace the *uniform number* with the *same percentage*. Specifically, we randomly select the same percentage (denoted by p) of instances from each original seed instance set to form a corresponding new subset. After that, the new subset replaces the original one in subsequent experimental steps. In this

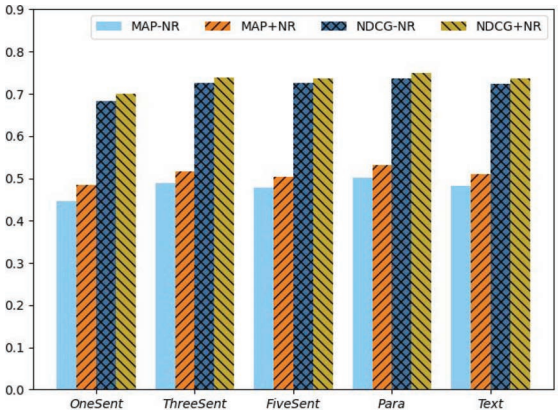

Figure 4 Effect of noise reduction on the performance of our model. Comparisons show that noise reduction can improve MAP and NDCG scores. MAP-NR and NDCG-NR indicate the scores without using noise reduction, and MAP+NR and NDCG+NR represent that the noise reduction has been performed.

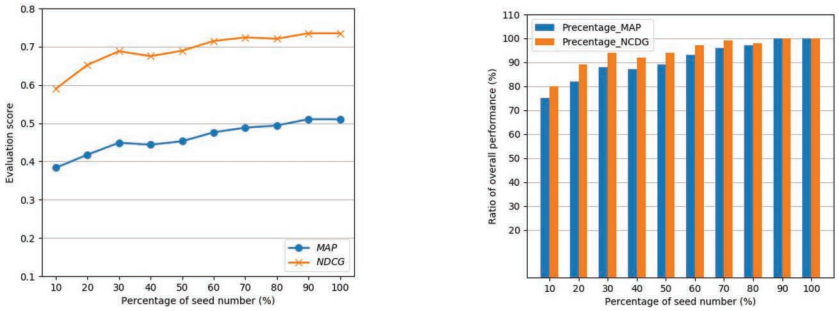

Figure 5 Effect of different percentage seed instances on the overall performance. Our model is substantially robust to the number of seed instance. (a) MAP and NDCG scores of the proposed model for different percentages of seed instances. (b) Ratios of MAP and NDCG values between the model performance of different percentage seed instances and the overall performance.

work, we adopt p from the range 0.1 to 1.0 with the step 0.1. To obtain unbiased results, three runs are performed for each p value, and then the average value is calculated as the corresponding result. Note that, we set the minimum number of seed instances per concept to 3 in the experiments.

Figure 5 presents the effect of different percentage seed instances on the overall performance. Specifically, Figure 5(a) shows MAP and NDCG values of our model at different the percentage of seed number. For the

convenience of comparison, the ratios between the model performance of different percentage seed instances and the overall performance are given in Figure 5(b).

As can be observed in Figure 5(a), as the seed ratio increases, the scores of MAP and NDCG are gradually improved. Furthermore, Figure 5(b) shows that even 20% seed instances are used, our model still can achieve over 80% of the overall performance, and 90% model performance is obtained when the proportion of seeds reaches 30%. It can be inferred that the proposed model has better adaptability for the number of seed instances.

5 Conclusions and Future Work

We have investigated the problem of abstract concept instantiation, which maps one abstract concept into a concrete instance set under contextual constraint. Previous models exist the following problems: (1) incorrect label and label incompleteness are ignored in the category structure on which instance selection depends; (2) the instance profile, which is used for calculating the relevance between candidate instance and contextual constraint, is usually determined by one's own supposition. To address these problems, we have proposed a novel abstract concept instantiation model. In summary, we have made the following two contributions:

(1) An InstanceRank algorithm has been proposed to rank the possibility that candidate instances belong to the abstract concept. In InstanceRank, the combination of category and association similarity has effectively remedied the problem of incorrect label and label incompleteness in the category structure.
(2) Five different granularity instance profiles are designed based on the surrounding text of instance, and then we have performed a systematic performance comparison on them. A proper instance profile will be beneficial to the proposed model.

Experimental results on Chinese food abstract concept set show that our model is effective on the task of abstract concept instantiation. Compared with the baseline models, our model has made a great improvement that MAP increased from 0.358 to 0.531, and NDCG increased from 0.605 to 0.750.

In future work, we will apply our method to event inference. With the help of the method, the paths between the abstract concept and its concrete instances are established, and we expect to produce more accurate reasoning results.

Appendix

Appendix 1: An example of explaining the motivation of the InstanceRank algorithm

We use an example to illustrate that the association of instances can rectify the incorrect and incompleteness problems in the category structure. For instance, in the *HudongBaike*, the instance "*Jinguoyuan Group*"[9] is incorrectly labeled as "*fruit*", but the instance "*pomegranate*"[10] is not labeled as "*fruit*". In InstanceRank, suppose we select "*pear*" as a seed instance. For the instance "*pomegranate*", even though it doesn't have the category of "*fruit*", there are 35 occurrences with "*pear*" in the coordination relation item set (*CRIS*, defined in subsection 3.1.1) in the experimental corpus. According to the definition of *CRIS*, this implies that the instance "*pomegranate*" are likely to share the same category label with the seed instance "*pear*". As for "*Jinguoyuan Group*" which even though has the category of "*fruit*", it does not co-occur with any fruit instance in the coordination relation item set in the experimental corpus. This indicates that the co-occurrence relationship between the "*Jinguoyuan Group*" and the seed instance "*pear*" does not support "*Jinguoyuan Group*" as a fruit instance.

To model the above idea, in Section 3.1, we define the category relation and the co-occurrence relation of the instances by the metrics of category similarity and association similarity, respectively. In the InstanceRank algorithm, the two similarities can complement each other, and effectively alleviate the problem of incorrect label and label incompleteness in the encyclopedia category structure.

Appendix 2: Seed Instance Selection

Seed instance selection is employed to identify desirable instances of an abstract concept. The best way to obtain the seed instances is to choose from the manually constructed taxonomy (e.g., Hownet[11]). However, there are two problems in these taxonomies. Firstly, they are mainly composed of commonsense knowledge, while the domain-specific instances are relatively few. Secondly, the granularity of the "is-a" relation is relatively coarse. For

[9] http://www.baike.com/wiki/%E9%87%91%E6%9E%9C%E6%BA%90%E9%9B%86%E5%9B%A2.

[10] http://www.baike.com/wiki/%E7%9F%B3%E6%A6%B4&prd=button_doc_entry.

[11] Hownet is a well-known knowledge base of taxonomy in Chinese, it can be obtained from www.keenage.com/html/c_index.html.

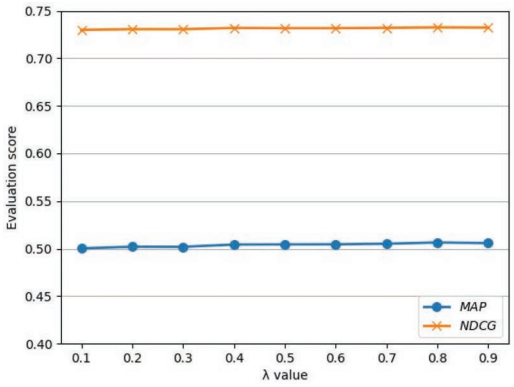

Figure 6 Effect of the λ value for experimental results. In order to explore the impact of λ, noise reduction is not performed in this experiment. The λ value has little effect on the MAP and NDCG score.

example, in Hownet, both "*shrimp*" and "*tortoise*" belong to the category "*fish*".

For these reasons, the Hearst Pattern method [23] is utilized to obtain the seed instances, mainly considering its high accuracy. In Hearst Pattern, the more times the pattern is matched, the more likely the instance belongs to the concept. The possibility is evaluated by [22]:

$$p(e) = 1 - 0.5^{n(e)} \tag{20}$$

where $n(e)$is the number of times that the instance e is matched by the "is-a" Hearst patterns.

In this work, the seed instances are selected according to the possibility values arranged in descending order, and the number of choices depends on the proportion of the sum of possibilities. More formally:

$$S_c = \left\{ e_1, e_2, \ldots, e_k | \sum_{i=1}^{k} p(e_i) = \theta * \sum_{i=1}^{n} p(e_i), p(e_1) \geq p(e_2) \geq \ldots \geq p(e_n), 1 \geq k \geq n \right\} \tag{21}$$

where $0 < \theta \leq 1$ determines the proportion of selected instances that is going to be placed in the seed instance set. In this work, we set $\theta = 0.7$ for empirical.

Appendix 3: Adjustment of λ Value

The λ value controls the proportion of the background probability in our model. Figure 6 shows the MAP and NDCG scores of the proposed model at different λ values. Results show that the λ value has little effect on the MAP and NDCG score. The reason for this phenomenon is that the value of $p(t|IP_e)$ would be much greater than $p(t|D)$ in web document corpus. Therefore, we set λ = 0.5 in the rest of the experiments.

Acknowledgments

The research reported in this paper was supported in part by the National Natural Science Foundation of China under the grant No.91746203. This work was jointly supported by a grant from Ant Financial Services Group, the Natural Science Foundation of the Anhui Higher Education Institutions under grant the No.KJ2017B18 and Shanghai Sailing Program 20YF1413800.

References

[1] Marcel Adam Just, Jing Wang, and Vladimir Cherkassky. Neural representations of the concepts in simple sentences: Concept activation prediction and context effects. *NeuroImage*, 157:511–520, 2017.

[2] Subin Huang, Xiangfeng Luo, Jing Huang, Yike Guo, and Shengwei Gu. An unsupervised approach for learning a chinese IS-A taxonomy from an unstructured corpus. *Knowl. Based Syst.*, 182, 2019.

[3] Jack Sun, Franky, Kenny Q. Zhu, and Haixun Wang. Query suggestion by concept instantiation. In *Proceedings of the ISWC 2013 Posters & Demonstrations Track*, volume 1035 of *CEUR Workshop Proceedings*, pages 181–184, Sydney, Australia, 2013. CEUR-WS.org.

[4] Yue Wang, Hongsong Li, Haixun Wang, and Kenny Qili Zhu. Concept-based web search. In *Conceptual Modeling - 31st International Conference ER*, volume 7532 of *Lecture Notes in Computer Science*, pages 449–462, Florence, Italy, 2012. Springer.

[5] Sheng-Jun Huang, Wei Gao, and Zhi-Hua Zhou. Fast multi-instance multi-label learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(11):2614–2627, 2019.

[6] Yueguo Chen, Lexi Gao, Shuming Shi, Xiaoyong Du, and Ji-Rong Wen. Improving context and category matching for entity search. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 16–22, Québec, Canada, 2014. AAAI Press.

[7] Krisztian Balog, Marc Bron, Maarten de Rijke, and Wouter Weerkamp. Combining term-based and category-based representations for entity search. In *Focused Retrieval and Evaluation, 8th International Workshop of the Initiative for the Evaluation of XML Retrieval*, volume 6203 of *Lecture Notes in Computer Science*, pages 265–272, Brisbane, Australia, 2009. Springer.

[8] Yi Fang and Luo Si. Related entity finding by unified probabilistic models. *World Wide Web*, 18(3):521–543, 2015.

[9] Rianne Kaptein and Jaap Kamps. Exploiting the category structure of wikipedia for entity ranking. *Artif. Intell.*, 194:111–129, 2013.

[10] Krisztian Balog, Marc Bron, and Maarten de Rijke. Query modeling for entity search based on terms, categories, and examples. *ACM Trans. Inf. Syst.*, 29(4):22:1–22:31, 2011.

[11] Denghao Ma, Yueguo Chen, Kevin Chen-Chuan Chang, Xiaoyong Du, Chuanfei Xu, and Yi Chang. Leveraging fine-grained wikipedia categories for entity search. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 1623–1632, Lyon, France, 2018. ACM.

[12] Krisztian Balog, Marc Bron, and Maarten de Rijke. Category-based query modeling for entity search. In *Advances in Information Retrieval, 32nd European Conference on IR Research*, volume 5993 of *Lecture Notes in Computer Science*, pages 319–331, Milton Keynes, UK, 2010. Springer.

[13] Anne-Marie Vercoustre, Jovan Pehcevski, and James A. Thom. Using wikipedia categories and links in entity ranking. In *Focused Access to XML Documents, 6th International Workshop of the Initiative for the Evaluation of XML Retrieval*, volume 4862 of *Lecture Notes in Computer Science*, pages 321–335, Dagstuhl Castle, Germany, 2007. Springer.

[14] Theodora Tsikrika, Pavel Serdyukov, Henning Rode, Thijs Westerveld, Robin Aly, Djoerd Hiemstra, and Arjen P. de Vries. Structured document retrieval, multimedia retrieval, and entity ranking using pf/tijah. In *Focused Access to XML Documents, 6th International Workshop of the Initiative for the Evaluation of XML Retrieval*, volume 4862 of *Lecture Notes in Computer Science*, pages 306–320, Dagstuhl Castle, Germany, 2007. Springer.

[15] Alberto Tonon, Michele Catasta, Gianluca Demartini, Philippe Cudré-Mauroux, and Karl Aberer. Trank: Ranking entity types using the web of data. In *The Semantic Web - ISWC 2013 - 12th International Semantic*

Web Conference, volume 8218 of *Lecture Notes in Computer Science*, pages 640–656, Sydney, Australia, 2013. Springer.

[16] Marco Gori and Augusto Pucci. Itemrank: A random-walk based scoring algorithm for recommender engines. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 2766–2771, Hyderabad, India, 2007.

[17] Yi Fang, Luo Si, and Aditya P. Mathur. Discriminative models of integrating document evidence and document-candidate associations for expert search. In *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 683–690, Geneva, Switzerland, 2010. ACM.

[18] Shinryo Uchida, Takehiro Yamamoto, Makoto P. Kato, Hiroaki Ohshima, and Katsumi Tanaka. Entity ranking by learning and inferring pairwise preferences from user reviews. In *Information Retrieval Technology - 13th Asia Information Retrieval Societies Conference*, volume 10648 of *Lecture Notes in Computer Science*, pages 141–153, Jeju Island, South Korea, 2017. Springer.

[19] Le Li, Junyi Xu, Weidong Xiao, Shengze Hu, and Haiming Tong. Exploiting external knowledge and entity relationship for entity search. In *Natural Language Understanding and Intelligent Applications - 5th CCF Conference on Natural Language Processing and Chinese Computing, and 24th International Conference on Computer Processing of Oriental Languages*, volume 10102 of *Lecture Notes in Computer Science*, pages 689–700, Kunming, China, 2016. Springer.

[20] Andrew Karem and Hichem Frigui. Multiple instance learning with multiple positive and negative target concepts. In *23rd International Conference on Pattern Recognition*, pages 474–479, Cancún, Mexico, 2016. IEEE.

[21] Tao Xu, Iker Gondra, and David K. Y. Chiu. A maximum partial entropy-based method for multiple-instance concept learning. *Appl. Intell.*, 46(4):865–875, 2017.

[22] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr., and Tom M. Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, Atlanta, USA, 2010. AAAI Press.

[23] Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of 14th International Conference on Computational Linguistics*, pages 539–545, Nantes, France, 1992.

[24] Mohamed Ben Aouicha, Mohamed Ali Hadj Taieb, and Malek Ezzeddine. Derivation of "is a" taxonomy from wikipedia category graph. *Eng. Appl. Artif. Intell.*, 50:265–286, 2016.

[25] Krisztian Balog, Leif Azzopardi, and Maarten de Rijke. Formal models for expert finding in enterprise corpora. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 43–50, Washington, USA, 2006. ACM.

[26] Gemma Boleda, Abhijeet Gupta, and Sebastian Padó. Instances and concepts in distributional space. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 79–85, Valencia, Spain, 2017. Association for Computational Linguistics.

[27] Xin Lv, Lei Hou, Juanzi Li, and Zhiyuan Liu. Differentiating concepts and instances for knowledge graph embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1971–1979, Brussels, Belgium, 2018. Association for Computational Linguistics.

[28] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems*, pages 3111–3119, Lake Tahoe, United States, 2013.

[29] Subin Huang, Xiangfeng Luo, Jing Huang, Hao Wang, Shengwei Gu, and Yike Guo. Improving taxonomic relation learning via incorporating relation descriptions into word embeddings. *Concurrency and Computation: Practice and Experience*, 2020.

[30] Michael R. Smith and Tony R. Martinez. The robustness of majority voting compared to filtering misclassified instances in supervised classification tasks. *Artif. Intell. Rev.*, 49(1):105–130, 2018.

[31] Wanxiang Che, Zhenghua Li, and Ting Liu. LTP: A chinese language technology platform. In *23rd International Conference on Computational Linguistics*, pages 13–16, Beijing, China, 2010. Demonstrations Volume.

[32] Stephen Roller, Douwe Kiela, and Maximilian Nickel. Hearst patterns revisited: Automatic hypernym detection from large text corpora. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 358–363, Melbourne, Australia, 2018. Association for Computational Linguistics.

Biographies

Shengwei Gu received the master's degree in School of Mathematics and Computer Science from Nanjing Normal University in 2008, China. Currently, he is pursuing his PhD degree in the School of Computer Engineering and Science, Shanghai University, China. His main research interests include information retrieval and question answering systems.

Xiangfeng Luo is a professor in the School of Computer Engineering and Science, Shanghai University, China. He received the master's and PhD degrees from the Hefei University of Technology in 2000 and 2003, respectively. He was a postdoctoral researcher with the China Knowledge Grid Research Group, Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), from 2003 to 2005. His main research interests include Web Wisdom, Cognitive Informatics, and Text Understanding. He has authored or co-authored more than 50 publications and his publications have appeared in IEEE Trans. on Automation Science and

Engineering, IEEE Trans. on Systems, Man, and Cybernetics-Part C, and IEEE Trans. on Learning Technology, Concurrency and Computation: Practice and Experience, etc. He has served as the Guest Editor of ACM Transactions on Intelligent Systems and Technology, as well as more than 40 PC members of conferences and workshops.

Hao Wang received the PhD degree from Waseda University in 2019, partly supported by Oversea Graduate Student Project of the China Scholarship Council. He is currently an assistant professor of Shanghai University. His research interests include natural language processing, especially machine translation.

Jing Huang received his master's degree from Nankai University and Boston University in 1995 and 1998, respectively. He is currently working at Ant Financial Services Group, Hangzhou, China. His research interests include data mining and knowledge graph.

Subin Huang received the master's degree in School of Computer and Information from Anhui Polytechnic University in 2012, China. Currently, he is pursuing his PhD degree in the School of Computer Engineering and Science, Shanghai University, China. His main research interests include information retrieval, data mining, and knowledge graph.