IEICE TRANS. INF. & SYST., VOL.E102–D, NO.9 SEPTEMBER 2019

LETTER

A Knowledge Representation Based User-Driven Ontology Summarization Method

Yuehang DING[†a)], ***Member***, **Hongtao YU**[†], **Jianpeng ZHANG**[†], **Huanruo LI**[†], ***and*** **Yunjie GU**[†b)], ***Nonmembers***

SUMMARY As the superstructure of knowledge graph, ontology has been widely applied in knowledge engineering. However, it becomes increasingly difficult to be practiced and comprehended due to the growing data size and complexity of schemas. Hence, ontology summarization surfaced to enhance the comprehension and application of ontology. Existing summarization methods mainly focus on ontology's topology without taking semantic information into consideration, while human understand information based on semantics. Thus, we proposed a novel algorithm to integrate semantic information and topological information, which enables ontology to be more understandable. In our work, semantic and topological information are represented by concept vectors, a set of high-dimensional vectors. Distances between concept vectors represent concepts' similarity and we selected important concepts following these two criteria: 1) the distances from important concepts to normal concepts should be as short as possible, which indicates that important concepts could summarize normal concepts well; 2) the distances from an important concept to the others should be as long as possible which ensures that important concepts are not similar to each other. K-means++ is adopted to select important concepts. Lastly, we performed extensive evaluations to compare our algorithm with existing ones. The evaluations prove that our approach performs better than the others in most of the cases.

key words: *ontology, ontology summarization, semantic distance, K-means++*

1. Introduction

As a structural knowledge base, knowledge graph has been increasingly applied to the intelligent semantic searches, mobile personal assistants and question answering systems. Ontology is the superstructure of knowledge graph. It plays an important role in both knowledge reasoning and knowledge graph construction. With the explosive growth of data size and ontology complexity, ontology's understanding and application are becoming more and more difficult. Ontology summarization is the process to distill knowledge from ontologies. It can effectively reduce the difficulty of ontology understanding and speed up ontology application, thus it effectively alleviates the above problems.

Ontology summarization is the process of distilling knowledge from ontology to produce an abridged version for particular users and tasks [1]. Existing ontology summarization methods can be divided into user-driven and task-driven according to their application requirements [2]. The user-driven summarization aims to help users quickly understand an ontology. The task-driven summarization focuses on accelerating a specific task. Most existing user-driven summarization methods extract important concepts according to the ontologies' topology, rarely taking semantic information of concepts into consideration. However, we find that the more semantic information is used, the higher the accuracy of the summarization: Xiang Zhang et al. [1] proposed a summarization method based on the ontology structure, considering the triples as nodes, projecting ontologies to graphs. Peroni et al. [3] transferred ontologies into graphs by considering the concepts as nodes and the relations as edges. Apart from ontologies' topological information, they also chose important concepts by name simplicity, which preferred concepts labeled with simple names. Paulo et al. [4] extracted important concepts from ontologies according to their topology and their relations' labels. The accuracy of the above three algorithms is increasing in order. At the same time, their extent of utilizing semantic information is incremental. In fact, summarizations without semantic information may neglect representative concepts connecting with fewer relations, which causes a low accuracy. To further improve the accuracy of ontology summarization, we propose a knowledge representation based summarization method which utilizes concepts' semantic information.

In this paper, we transfer concepts into embeddings based on ontologies' structural and semantic information. The distance from concept A to concept B indicates the possibility that A can summarize B: the shorter the distance from A to B is, the better A can summarize B. We hope that the distances from important concepts to normal concepts can be as short as possible, which means important concepts can summarize normal concepts well. At the same time, we hope that the distances from an important concept to the others can be as long as possible, which means each important concept is not similar to the others. To satisfy the limitations above, we put the idea of K-means++ on concept embeddings to select important concepts. Specifically, our contributions are listed as follows:

A novel ontology summarization method is proposed, which takes ontologies' semantic information and topological information into consideration.

To measure the concepts' summarizing ability, concept embeddings are generated based on the word embedding algorithm, i.e., word2vec. Distance from one concept to another represents their similarity.

To select important concepts from ontologies, we pro-

Manuscript received April 3, 2019.
Manuscript revised May 8, 2019.
Manuscript publicized May 30, 2019.
[†]The authors are with the Information Engineering University, China.
a) E-mail: data_rabbit@163.com
b) E-mail: lizardwhite@163.com
DOI: 10.1587/transinf.2019EDL8069

pose a maximum-minimum iteration method using the idea of K-means++ method.

To avoid extracting similar concepts as important concepts, we set the similarity among important concepts as one of our algorithm's decision criteria. To our knowledge, existing methods only consider the similarity between important concepts and normal concepts, and do not consider that important concepts should be as dissimilar as possible.

2. Concept Distances

In this section, we transfer concepts into embeddings to calculate their similarities. Firstly, we use Google's pre-trained word vectors to represent the words in concepts. Then we define a concept distance measurement based on these vectors. By this measurement, we can calculate the semantic distances between all concept pairs. Next, we define a measurement to calculate the probability that a concept can summarize the others according to the ontology's topology. Finally, we integrate the topological probability into semantic distances to get the final distances between concepts.

2.1 Semantic Distance

Word2vec is a tool for computing continuous distributed representations of words. These representations contain words' semantic information. To transfer concepts into vectors, we use Google's pre-trained word vectors [5] to represent words in concepts. We define the semantic distance between concepts as follows.

Definition 1 (Semantic Distance) Assume word $w_i \in c_1$, $w_j \in c_2$. c_1 and c_2 are two concepts in the target ontology. Then the semantic distance from c_1 to c_2, denoted by $\underset{c_1,c_2 \in C}{D}(c_1, c_2)$, is calculated by Eq. (2):

$$dist(w_i, w_j) = L2(w2v(w_i), w2v(w_j)) \tag{1}$$

$$\underset{c_1,c_2 \in C}{D}(c_1, c_2) = \sum_{w_i \in c_1} \min_{w_j \in c_2} (dist(w_i, w_j)) \tag{2}$$

where $dist(w_i, w_j)$ is the semantic distance between w_1 and w_2. $w2v(w_i)$ is word w_i's embedding, and $L2(x, y)$ is the L2 norm of x, y. Note that $\underset{c_1,c_2 \in C}{D}(c_1, c_2) \neq \underset{c_2,c_1 \in C}{D}(c_2, c_1)$. The semantic distance from c_1 to c_2 measures the semantic similarity from c_1 to c_2. It means the more c_1 can summarize c_2 semantically, the shorter the distance is. Besides, this measurement tends to give shorter distance to concepts containing fewer words, which matches the criterion, i.e., name simplicity, proposed in [3].

2.2 Topological Probability

To integrate the topological information into semantic distance, we define the notion of the topological probability according to relations between concepts.

Definition 2 (Topological Probability) Assume a triple (c_1, r, c_2) in an ontology, where c_1, c_2 are concepts, r is the relation from c_1 to c_2. Define the weight of relation r as $weight(r)$, the topological probability that c_1 can summarize c_2 as $p(c_1, c_2)$.

$$weight(r) = \begin{cases} 0, r = subClassOf \\ 1, r = inv\_subClassOf \\ 0.5, r = other\_relations \end{cases} \tag{3}$$

$$p(c_1, c_2) = \max_i \left(\prod_{k \in path\_i} weight(r_i^k) \right) \tag{4}$$

where r_i^k is the k-th relation in relation path i. A relation path from c_1 to c_2 consists of a path of relations through which c_1 can reach c_2. If there exists $(c_1, subClassOf, c_2)$, which means c_2 is the superclass of c_1, then we can affirm that c_2 can summarize c_1, c_1 cannot summarize c_2. Thus we set $p(c_1, c_2) = 0$, $p(c_2, c_1) = 1$. For other kinds of relations, we set the probability be 0.5. The topological probability quantifies relations between concepts. Thus we can integrate the topological information into semantic distances. Finally, we recalculate the distance from c_1 to c_2 by Eq. (5):

$$D(c_1, c_2) = D(c_1, c_2)(1 - p(c_1, c_2)) \tag{5}$$

where $D(c_1, c_2)$ is the final distance from c_1 to c_2.

3. Selection of Important Concepts

Having calculated the distances between concepts, we focus on the selection of important concepts. We select important concepts based on the following two criteria: firstly, we hope that the distances from selected concepts to unselected concepts can be as short as possible. It means that the selected concepts can summarize the unselected concepts well. At the same time, we hope that the distances between selected concepts can be as long as possible. It means the selected concepts are not similar to each other.

We use the idea of K-means++ to select the important concepts. The exact algorithm is as follows:

1. Consider concepts as nodes, relations as edges, choosing initial cluster centers according to K-means++.

2. Assign each node to the cluster whose mean has the least concept distance.

3. For each cluster, select the node with the shortest sum of distances to other nodes in the cluster to be the centroid.

4. Repeat step 2 and 3 until the assignments no longer change.

With precisely selected important concepts, users can understand the target ontology clearly and quickly. Our algorithm is divided into the following three steps:

1. Calculate the semantic distances between concepts based on Google's word vectors.

2. Combine ontology's topological information with concepts' semantic distances.

3. Select important concepts with the idea of K-means++.

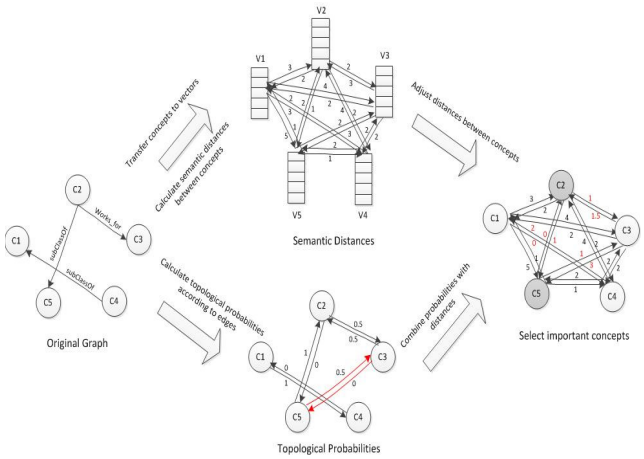

Fig. 1 The process of the proposed algorithm.

Table 1 The algorithm for selecting important concepts.

Input: network's adjacent matrix M
Output: irredundant matrix Mr

concepts ← all concepts from *Onto*
triples ← all triples from *Onto*
//calculate semantic distances between concepts
for c_1 in concepts:
    for c_2 in concepts:
        $D(c_1, c_2) = \sum_{w_1 \in c_1} \min_{w_2 \in c_2} (dist(w_1, w_2))$
//add structural information into concept distances
for (c_1, r, c_2) in *triples*:
    $p(c_1, c_2) = \max_i (\prod_{k \in path\_i} weight(r_i^k))$
    $D(c_1, c_2) = D(c_1, c_2)(1 - p(c_1, c_2))$
//select n important concepts according to distances
new_centroids = *get_ini_centroids*(D, n)
create n clusters according to *new_centroids*
for c in *new_centroids*:
    cluster(c) ← $\{c\}$
repeat:
centroids = *new_centroids*
new_centroids = {}
for c in *concepts*:
    //assign c to its nearest cluster
    cluster$(\arg\min_{c_i \in impo\_cons} (D(c_i, c))).append(c)$
for c in *centroids*:
    //save *clusters*' new centroid into *new_centroids*
    new_centroids$.append(\arg\min_{c_1 \in cluster(c)} (\sum_{c_2 \in cluster(c)} D(c_1, c_2)))$
until *centroids* = *new_centroids*
imp_con ← *centroids*

The process of our algorithm is shown in Fig. 1, where C_i represents the i-th concept, V_i represents the i-th concept's embedding.

The pseudo-code of our algorithm is shown in Table 1.

4. Experiments and Evaluations

4.1 Evaluation Measures

To evaluate our algorithm, we use in total three ontologies, whose information is shown in Table 2. The three ontologies are available in [6].

Table 2 Synthetic network parameter.

Ontology	Characteristics	Description
Biosphere	87 classes, 3 properties	Models information in the domain of bioinformatics
Financial	188 classes, 4 properties	Describes information onthe financial domain
Aktors Portal	247 classes, 167 properties	Describes an academic computer science community

We perform an extensive evaluation to assess the effectiveness of our algorithm. The above three ontologies are used to compare our algorithm with the algorithms proposed by Peroni et al. [3], Queiroz-Sousa et al. [4] and Troullinou et al. [6]. To compare these algorithms, we use the reference summaries and the results published in [3] and [4]. Peroni et al. [3] combined cognitive science, network topology, and lexical statistics to automatically select important concepts. Queiroz-Sousa et al. [4] proposed an algorithm that selected important concepts through centrality measures or user indication. Troullinou et al. [6] used the information from the data layer to calculate the relative cardinality of relations. Then they combined relative cardinality with node centrality to select important concepts. The reference summaries used in this paper were generated by Peroni et al. and were also used by Queiroz- Sousa and Troullinou in their evaluations [6]. The reference summaries were generated by eight human experts, who were requested to select up to 20 concepts [3].

We evaluate the degree of agreement between a generated summary and a reference summary by the measurement proposed in [6]. Assuming $Sim(G_S, G_R)$ is the similarity between two summaries, $G_S = (V_S, E_S)$ is the generated summary, $G_R = (V_R, E_R)$ is the reference summary. Assuming $\{c_k, \ldots, c_p\}$ are the classes in V_R that are subclasses of the classes $\{c_k', \ldots, c_p'\}$ of V_s. Assuming $\{c_m, \ldots, c_n\}$ are the classes in V_R that are superclasses of the classes $\{c_m', \ldots, c_n'\}$ of V_s. $Sim(G_S, G_R)$ is defined as follows:

$$Sim(G_s, G_R) = \frac{|V_s \cap V_R| + 0.6 * \sum_{i=k}^{p} \frac{1}{d_{p(c_i \to c_i')}} + 0.3 * \sum_{i=m}^{n} \frac{1}{d_{p(c_i \to c_i')}}}{|V_R|} \quad (6)$$

Consequently, the summarization accuracy of an algorithm is calculated by the average of the similarities between the generated summary and expert selected summaries [6].

4.2 Experiments

To evaluate the effectiveness of our algorithm, we put our algorithm to the three ontologies introduced in Table 2. We calculate the similarities between summaries produced by our algorithm and those expert-selected ones. The summaries produced by our algorithm are shown in Table 3.

We compared the four algorithms by their similarities with reference summaries. The result is shown in Fig. 2.

We can observe that our algorithm performs better in

Table 3 Synthetic network parameter.

Ontology	Algorithm choices
Aktors Portal	'information-transfer-event', 'intangible-thing', 'generic-area-of-interest', 'method', 'abstract-information', 'composite-publication-reference', 'tangible-thing', 'information-bearing-object', 'person', 'affiliated-person', 'employee', 'temporal-thing', 'event', 'generic-agent', 'geopolitical-entity', 'legal-agent', 'organization', 'address', 'organization-unit', 'technology'
Bank	'bond', 'market', 'stock', 'government', 'contract', 'card', 'asset', 'organization', 'broker', 'bear', 'certificate', 'order', 'lender', 'holder', 'payment', 'tax-exempt-bond', 'agency-bond', 'security', 'floating-rate-security', 'bank'
Biosphere	'vegetation', 'animal', 'plant', 'living-thing', 'mold', 'crown', 'microbiota', 'microbiota-taxonomy', 'flagellate', 'marine-animal', 'litter', 'fungi', 'canopy', 'human', 'mammal', 'mushroom', 'algae', 'crop', 'marine-plant', 'dairy'

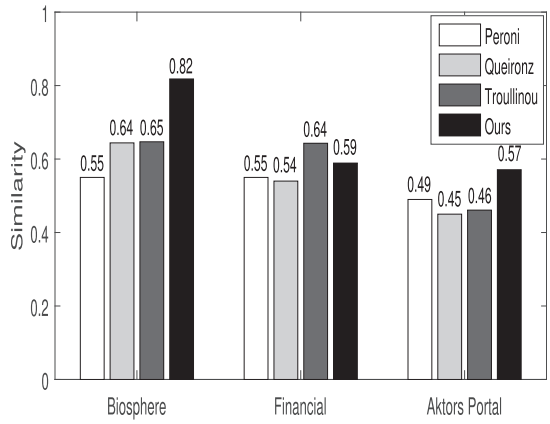

Fig. 2 Similarity results.

Biosphere and Aktors Portal ontology. Although our algorithm performs slightly worse than Troullinou's algorithm in Financial ontology, it performs better than others in other cases.

The reason that our algorithm performs worse than Troullinou's in Financial is that the meaning of a concept label may different from the words in it. In Financial ontology, many concept labels consist of several words. The meaning of these words may irrelevant with the meaning of the entire label. However, our algorithm calculates semantic similarity between two labels according to the words in them. Thus the accuracy in Financial is at a lower level.

5. Conclusion

In this paper, we proposed an improved-novel ontology summarization algorithm which made use of concept label's semantic information. At first we transfer concepts into high-dimensional vectors according to their labels. Then we adjust distances between concept vectors based on the relationship between concepts. Experiments reveal that our algorithm outweighs some previous studies merely concerning topological information as we take semantic and topological information into consideration.

From the experiments, we have proved that integrating semantic information into ontology summarization can improve summarization accuracy. Combined with other algorithms, we draw the following conclusion: more semantic information integration leads to more accurate summarization.

Thus, in the future, we will find methods to integrate more semantic information into summarization process. For instance, we will crawl corpora according to concepts, and have concept vectors trained corresponding to those corpora, or jointly training concept vectors on the basis of both corpora and ontology's topology. Moreover, since experts' cognition vary, their summaries should be weighted depending on reliability, where weights could be decided by the similarity of their summaries.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61521003).

References

[1] X. Zhang, G. Cheng, and Y. Qu, "Ontology summarization based on RDF sentence graph," C.L. Williamson, M.E. Zurko, P.F. Patel-Schneider, and P.J. Shenoy, eds., Proc. 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, pp.707–716, 2007. doi: 10.1145/1242572.1242668.

[2] N. Li and E. Motta, "Evaluations of user-driven ontology summarization," Knowledge Engineering and Management by the Masses, EKAW 2010, Lecture Notes in Computer Science, vol.6317, pp.544–553, Springer, Berlin, Heidelberg, 2010.

[3] S. Peroni, E. Motta, and M. d'Aquin, "Identifying key concepts in an ontology, through the integration of cognitive principles with statistical and topological measures," The Semantic Web, ASWC 2008, Lecture Notes in Computer Science, vol.5367, pp.242–256, Springer, Berlin, Heidelberg, 2008.

[4] P.O. Queiroz-Sousa, A.C. Salgado, and C.E. Pires, "A method for building personalized ontology summaries," J. Information and Data Management, vol.4, no.3, p.236, 2013.

[5] https://code.google.com/archive/p/word2vec/

[6] G. Troullinou, H. Kondylakis, E. Daskalaki, and D. Plexousakis, "Ontology understanding without tears: The summarization approach," Semantic Web, no.8, pp.1–19, 2017.