Graph-Based Knowledge Consolidation in Ontology Population

Pum Mo RYU†, Myung-Gil JANG†, Hyun-Ki KIM†, Nonmembers, and So-Young PARK††(a), Member

SUMMARY We propose a novel method for knowledge consolidation based on a knowledge graph as a next step in relation extraction from text. The knowledge consolidation method consists of entity consolidation and relation consolidation. During the entity consolidation process, identical entities are found and merged using both name similarity and relation similarity measures. In the relation consolidation process, incorrect relations are removed using cardinality properties, temporal information and relation weight in given graph structure. In our experiment, we could generate compact and clean knowledge graphs where number of entities and relations are reduced by 6.1% and by 17.4% respectively with increasing relation accuracy from 77.0% to 85.5%.

key words: knowledge graph, knowledge consolidation, entity consolidation, relation consolidation

1. Introduction

Much of the research into ontology population has focused mainly on relation extraction from texts. The extracted relations should be merged into graph-structured knowledge to be applied to intelligent applications such as Q&A system. An entity can be expressed in multiple notations in source texts, and it is difficult to identity semantically-equivalent entities during the relation extraction phase. For example, two relations <has_product, NC 소프트웨어, (NC so-peu-teu), e/∈/X> (ri-ni-ji) and <has_product, NCSOFT, e/∈/X> (ri-ni-ji), extracted from different documents, represent the same information, because ‘NC 소프트웨어’ and ‘NCSOFT’ are different notations for the same company. So, a process for identifying and merging equivalent entities is needed to make the knowledge graph compact. Furthermore, extracted relations can be semantically incorrect due to errors in relation extraction method or incorrect information descriptions in source text. We can identify incorrect relations using graph level constraints. Hence, we suggest a novel knowledge consolidation framework for a graph-structured knowledge base, where we identify and merge semantically-equivalent entities using lexical similarity and relation similarity metrics, and identify and remove incorrect relations using temporal cardinality restrictions and a relation weighting scheme. Our graph based approach can find equivalent entities and incorrect relations which are hard for simple relation extraction approaches to identify. Some works reported meaningful result in entity consolidation task [2], [3]. But, the works was still focusing on entity consolidation problem. Our method differs from the previous works in that we tackled the additional relation consolidation problem in a knowledge graph framework. We applied and evaluated the proposed method in computer game domain text written in Korean.

In Sects. 2 and 3, we provide basic definitions and relation extraction method. In Sect. 4, we offer our knowledge consolidation methods. In Sect. 5, we give the experiment result, and conclude in Sect. 6.

2. Definitions

In this section, we define the necessary concepts of entity type, entity, relation type, relation, and knowledge graph.

Definition 1. (Entity Type) An entity type defines a group of entities that belong together based on their common properties. For example, an entity type BOGG represents a group of business organizations such as manufacturing companies.

Definition 2. (Entity) An entity is a real world instance for an entity type and has two basic elements; name and entity type. For example, e = (NCSOFT, BOGG) is an entity for a business organization named ‘NCSOFT’.

Definition 3. (Relation Type) A relation type ri is defined as a 4-tuple, <rel_name, ETs, ETo, p>, where rel_name is the name of the relation type; ETs and ETo are the entity types for the subject and object, respectively; and optional p is a cardinality property of the relation type. Four cardinality properties are used in this paper as follows. The first two properties are defined in OWL specification [4] and additional two properties are defined by adding time constraints to the previous properties. Because information becomes outdated as time passes, time constraints are important tools to filter out the outdated relations. The cardinality properties for relation types are determined by domain experts.

- Functional property (F) is a property that can have only one object entity y for each subject entity x.
- Inverse functional property (IF) is a property where object of the property uniquely determines the subject.
- Temporal functional property (TF) is a relational property in which a relation type can have only one object entity at a time for each subject entity. While one or more consecutive CEOs may have worked for a company, their official working periods do not overlap.
- Temporal inverse functional property (TIF) is a relational...
property in which a relation type can have only one subject entity at a time for each object entity.

**Definition 4. (Relation)** Given a set of entities \( E \) and a set of relation types \( RT \), a relation is defined as a 5-tuple, \(<rt, es, eo, d, w>\), where \( es \) and \( eo \) are subject and object entities of relation type \( rt \), \( d \) is the date when the triple \(<rt, es, eo>\) is described in a text, and \( w \) is the weight of the relation assigned by the relation extraction module. Identical versions of \(<rt, es, eo>\) can be extracted from different texts written on different days.

**Definition 5. (Knowledge Graph)** Given a set of relations \( R \), a knowledge graph \( KG \) is a graph structure where identical entities are merged into one node, and identical relations are merged into one edge. Three attributes, \( DF \) (first date), \( DL \) (last date) and \( WSUM \) (sum of relation weight) are augmented to the merged relations. Figure 1 (a) shows an example of a knowledge graph.

### 3. Relation Extraction

We extracted relation from Korean text using the system developed by Lee [1]. The system consists of two modules; named entity recognition module and relation identification module. Semantic relations for all possible entity pairs are evaluated after named entities are recognized in each input sentence. Because the Named entity recognizer for Korean was mainly developed for QA system, 180 fine-grained named entity types are defined in consideration of user’s asking points for finding answer candidates. Beside common named entities, the system additionally identifies non-named entities such as data, time and quantity. For computer game domain, we added domain dependent entity types as shown in Table 1. Relation identification module identifies 37 binary relationships among the entities identified by the NER. The relation types are defined by domain experts after deep analysis of questions in famous Korean computer game FAQ sites. Table 2 shows four most frequent relation types observed in the text. For all possible entity pairs in each input sentence, the most probable semantic relation type is identified, and relation weights are assigned to the relation.

### 4. Knowledge Consolidation

Our entity consolidation method identifies groups of semantically-equivalent entities written in different notations, and merges the entities into one seminal entity. We combine two criteria for entity similarities: name similarity and relation similarity. We applied a well-known edit-distance method to determine the entity name similarity [5]. The method measures the minimum number of insertions, deletions, and substitutions required to transform one string into another. Relation similarity is based on

---

**Table 1** Entity types for computer game domain.

| Entity type   | Description                          |
|---------------|---------------------------------------|
| TMIG_HW       | Hardware for computer games, such as PlayStation, Xbox 360 |
| TMIG_SW       | Software for computer games, such as Winning Eleven |
| TMIG_GENRE    | Genres of computer games such as Role-playing game (RPG) |

**Table 2** Relation types for computer game domain.

| Relation     | Subject       | Object        | Constraint |
|--------------|---------------|---------------|------------|
| has_product  | BOGG          | TMIG_SW       | IF         |
| has_person   | BOGG          | PERSON        | TF         |
| has_lw       | TMIG_SW       | TMIG_HW       | TF         |

---

**Fig. 1** Entity and relation consolidation steps.
the assumption that if two entities are connected to common entities using the same relation type, the two entities have common information. For example, in Fig. 1 (a), the two entities, ‘NCSOFT:BORG’ and ‘NC <LINEAGE> (NC so-peu-teu):BORG’, share two relations ‘<has_product, *’, LINEAGE> and ‘<has_ceo, *, KIM>’. The similarities between the two entities, $e_1$ and $e_2$, are calculated based on each relation type $r$, and then merged as in Eq. (1).

$$\text{sim}_r(e_1, e_2) = \frac{\max(|R(e_1)|, |R(e_2)|)}{|R(e_1) \cup R(e_2)|} \sum_{r \in R(e_1) \cup R(e_2)} |E(e_1, r) \cap E(e_2, r)| / |E(e_1, r) \cup E(e_2, r)|$$  \hspace{1cm} (1)

where $R(e)$ is a set of relation types where $e$ is involved, $E(e, r)$ is a set of entities to which entity $e$ is connected by relation type $r$, and $|S|$ is the number of elements in set $S$. $\max(|R(e_1)|, |R(e_2)|)$ is multiplied so as not to penalize entities that participate in many different relationships. The final entity similarity metric between two entities is shown in Eq. (2).

$$\text{sim}_N(e_1, e_2) = \alpha \cdot \text{sim}_N(e_1, e_2) + (1 - \alpha) \cdot \text{sim}_r(e_1, e_2)$$  \hspace{1cm} (2)

where $\text{sim}_N(e_1, e_2)$ is the name similarity based on edit distance of two entities, $\text{sim}_r(e_1, e_2)$ is the relation similarity, and $\alpha$ is the weighting scheme for each of them.

The entity similarity is calculated for all entity pairs of the same entity type in a knowledge graph. Once two entities are determined as being identical, a less seminal entity is merged into a more seminal one. A seminal entity is selected based on the subsumption ratio as shown in Eq. (3). If $\text{subsumed}(e_1, e_2)$ is greater than $\text{subsumed}(e_2, e_1)$, $e_2$ is more seminal than $e_1$. Figure 1(b) shows the graph after ‘NC <LINEAGE> (NC so-peu-teu)’ is merged to ‘NCSOFT’.

$$\text{subsumed}(e_1, e_2) = |R(e_1) \cap R(e_2)| / |R(e_1)|$$  \hspace{1cm} (3)

We identify incorrect relations using the following algorithm. This algorithm is applied to the graph where all the equivalent entities are merged.

**Input:** Knowledge graph (KG)

**Output:** Cleaned knowledge graph (CKG)

For each entity $e$ in KG,  
1.1 for each relation group $r_g$ of same relation type of $e$,  
1.1.1 Sort the relations in $r_g$ in decreasing order of the relation weight.  
1.1.2 If the relations in $r_g$ violate F or IF properties, then select the top relation in $r_g$ and remove the remaining relations from KG  
1.1.3 Else if the relations in $r_g$ violate the TF or ITF properties, then select a series of relations not violating the properties of the prior relations, and remove the remaining relations from the KG.

For example, two business organizations are connected to ‘LINEAGE’ with ‘has_product’ in Fig. 1(b). Because ‘has_product’ is under IF property, ‘LINEAGE’ must be connected to one business organization. The relation between ‘A3Security’ and ‘LINEAGE’ is removed because ‘NCSOFT’ is more strongly connected to ‘LINEAGE’ than ‘A3Security’ as in step 1.1.2. ‘NCSOFT’ is connected to two persons with a ‘has_ceo’ relation type. Because ‘has_ceo’ is under TF property, ‘NCSOFT’ cannot have two CEOs at the same time. The duration of ‘LEE’ (2004/07–2004/07) overlaps the duration of ‘KIM’ (2004/04–2007/12). The algorithm removes the relation to ‘LEE’ using $W_{SUM}$ as described in step 1.1.3. The final knowledge graph is shown in Fig. 1(c).

## 5. Experiment and Evaluation

We collected 1,500 news articles on computer game domain from Korean online news sites, and automatically extracted the relations from the collection, and then merged them into an initial knowledge graph. The graph has 3,577 entities and 4,622 relations. Three domain experts manually tagged the equivalent entities and incorrect relations to make a clean reference knowledge graph for evaluation. Final decision was made by majority when they showed different decisions. Table 3 shows statistics of equivalent entity groups which contain one to four equivalent entities. A group of size two has one equivalent entity pair, a group of size three has three equivalent entity pairs and a group of size four has six equivalent entity pairs. Because our entity consolidation method is defined as finding equivalent entity pairs in knowledge graph, we should find 473 equivalent entity pairs in the initial knowledge graph. Table 4 shows manual evaluation result of the extracted relations. Relations of F, TF, IF and ITF properties account for 56.1% (= 2,594/4,622) of all relations. Our relation consolidation method should identify all incorrect relations among F, TF, IF, ITF property relations minimizing false positive.

We measured the similarity of all possible entity pairs of same entity types using Eq.(2) with the threshold 0.4

| Table 3 | Equivalent entity groups in the initial knowledge graph. |
|----------|-----------------------------------------------------------|
| Group size | Equivalent entity groups | Total |
| Entities | 2,845 | 562 | 126 | 44 | 3,577 |
| Eq. entity groups | 2,845 | 281 | 42 | 11 | 3,179 |
| Eq. entity pairs | 0 | 281 | 126 | 66 | 473 |

| Table 4 | Correct/Incorrect relations in the initial knowledge graph. |
|----------|-----------------------------------------------------------|
| Relations | Correct | Incorrect | Total | Correct Ratio |
| Non-property | 1,802 | 426 | 4,622 | 79.0% |
| Property | 1,956 | 638 | 2,594 | 75.4% |
| All relations | 3,558 | 1,064 | 4,622 | 77.0% |
based on repeated experiments. We assumed that relation-based similarity is efficient when entities are tightly connected to each other entities. Thus, we repeated the experiment with two entity groups: those connected to other entities with one or more relations ($G_1$), and those connected to other entities with two or more relations ($G_2$). $G_2$ is a subset of $G_1$. The comparison was repeated 11 times, changing $\alpha$ from 0.0 to 1.0 with 0.1 step. When $\alpha$ is 0.3 or 0.4, the precisions are higher than those in other experiments, as shown in Fig. 2. We can say that our similarity metric makes up for the limitations of the two methods. Precision of $G_2$ is higher than that of $G_1$ as we expected. The range of recall was between 0.2 and 0.3, which is relatively lower than the precision value.

We applied the relation consolidation method to the knowledge graph where all identical entities are merged by domain experts. Table 5 shows the contingency matrix of relation consolidation. We obtained precision 91.8% ($= (1,917 + 465)/2,594$) for all relations, and recall 72.9% ($= 465/638$) for incorrect relations. We can filter out the incorrect relations with high accuracy, although it is hard to identify all incorrect relations. We applied the entity and the relation consolidation methods in sequence to show the validity of the suggested methods in real application. Accuracy for property relations sharply increased from 75.4% to 92.3% and accuracy for all relations increased from 77.0% to 85.5% which is a promising result for consolidating knowledge graph with ontological constraints (Fig. 3). We applied Hoser’s ontology analysis method to clarify the degree of consolidation (Table 6) [6]. Number of entities and number of relations in knowledge graph reduced 6.1% and 17.4% respectively. Diameter is the maximum shortest distance of all entity pairs in graph. Shorter diameter means that the graph is more compact. Density is average degree of connectedness of all entities. High density means that entities are tightly connected to other entities.

6. Conclusion

We proposed an integrated graph-based method for entity and relation consolidation. Our entity consolidation process is more efficient than the simple string matching method. In the relation consolidation process, cardinality constraints combined with temporal information are applied to find incorrect relations. We will apply the consolidated graph as a knowledge source for Q&A systems in large scale.

References

[1] C. Lee, Y. Hwang, and M. Jang, “Fine-grained named entity recognition and relation extraction for question answering,” Proc. SIGIR 2007, pp.799–800.
[2] A. Hogan, A. Harth, and S. Decker, “Performing object consolidation on the semantic web data graph,” Proc. 1st I3 Workshop at WWW2007, 2007.
[3] H. Alani, S. Kim, D.E. Millard, et al., “Web-based knowledge extraction and consolidation for automatic ontology instantiation,” Proc. K-CAP ’03, 2003.
[4] OWL Web Ontology Language Reference, W3C Recommendation, http://www.w3.org/TR/owl-ref/
[5] E.S. Ristad and P.N. Yianilos, “Learning-string edit distance,” IEEE Trans. Pattern Anal. Mach. Intell., vol.20, no.5, pp.522–532, 1998.
[6] B. Hoser et al., “Semantic Network Analysis of Ontologies,” ESWC 2006, LNCS 4011, pp.514–529, 2006.