1N3-OS-39b-5

An Automatic Knowledge Graph Creation Framework from Unstructured Text

Natthawut Kertkeidkachorn[*1] Ryutaro Ichise[*1*2]

[*1] Department of Informatics, Sokendai (The Graduate University for Advanced Studies)
[*2] National Institute of Informatics, Tokyo, Japan

Knowledge Graph creation from unstructured text plays a crucial role in the semantic web community. Consequently, there are many approaches proposing to create Knowledge Graph from unstructured text. However, Knowledge Graph integration is omitted. Knowledge Graph integration is an essential procedure because it could reduce the heterogeneous problem and could increase searchability over Knowledge Graphs. In our previous work, we proposed the T2KG framework, an automatic framework for Knowledge Graph creation from unstructured text, with keeping the integration issue in mind. Although we could achieve better results to create a Knowledge Graph than the previous approaches, the reasonable precision is still not reached. In this paper, we therefore propose T2KG_Ext: an extension of the T2KG framework. In the T2KG_Ext framework, we re-organize the T2KG framework to increase the ability to generate candidate triples and introduce the extension component, namely candidate selection, to the T2KG framework. In the preliminarily experiments, we reported the problem of the T2KG framework and showed some evidences that the T2KG_Ext could deal with such problems.

1. Introduction

Knowledge Graph (KG) is a structure knowledge base, which stores real-world entities and their relationships. Such entities and their relationships are represented by Linked Data. Linked Data defines the standard of publishing data as follows: 1) the data must be published under Resource Description Framework, 2) entities must be represented by Uniform Resource Identifier (URI) and 3) the representation of the data should be a triple (Subject Predicate, Object). Currently, there are many available KGs such as DBpedia, Freebase and YAGO. Such KGs play an important role in many applications. However, new knowledge emerges every day and most of new knowledge comes in the form of unstructured text. Consequently, it is necessary to populate new knowledge from unstructured text to existing KGs in order to keep the existing KGs up to date.

Recently, many approaches have been proposed the methods for extracting knowledge from unstructured text and populating knowledge to existing KGs. Although those studies performed well for extracting triples from unstructured text, they still have a limitation regarding mapping a predicate of a triple extracted from unstructured text to its identical predicate in the KG. Generally, many studies focus on mapping only an entity, which is usually a subject or an object of a triple, to its identical entity in a KG. Mapping a whole predicate to its identical predicate is usually ignored. Mapping a predicate to its identical predicate in a KG is an essential procedure because it can reduce the heterogeneity problem and can increase the searchability over a KG. Although some studies introduced mapping a predicate of a triple extracted from unstructured text to an identical predicate in a KG, the approach uses the simple rule-based approach. As a result, it cannot efficiently deal with the limitation of rule generation due to the sparsity of unstructured text. Therefore, we propose T2KG: an end-to-end system for creating a KG from unstructured text [1] to overcome the problems in the previous approach. Although T2KG provides the improvement over the previous approaches, we still could not achieve the reasonable precision. In T2KG, the pre-defined ontology had not been used as the prior knowledge to verify whether the extracted triples are correct or not.

In this paper, we present T2KG_Ext: an automatic knowledge graph creation framework from unstructured text. In T2KG_Ext, we re-organize the T2KG framework to increase the ability to generate candidate triples and introduce the extension component, namely candidate selection, to the T2KG framework. The candidate selection component is to select the suitable candidate triple and verify whether the suitable candidate triple is valid by using the constraint from the pre-defined ontology of the existing KGs.

2. Knowledge Graph Creation

The design of the T2KG_Ext framework is based on T2KG framework [1]. Similar to the T2KG framework, the T2KG_Ext framework is to take unstructured text as an input and produce a KG as an output. As shown in Figure 1, T2KG_Ext has five components: 1) Coreference Resolution, 2) Triple Extraction, 3) Triple Integration, 4) Candidate Generation and 5) Candidate Selection. The coreference resolution component detects coreferring chains of entities in unstructured text. The triple extraction component extracts a relation triple from unstructured text by using the open information extraction technique. The triple integration component generates a text triple by integrating the results from the coreference resolution component and the triple extraction component. The candidate generation uses

Contact: Natthawut Kertkeidkachorn, natthawut@nii.ac.jp

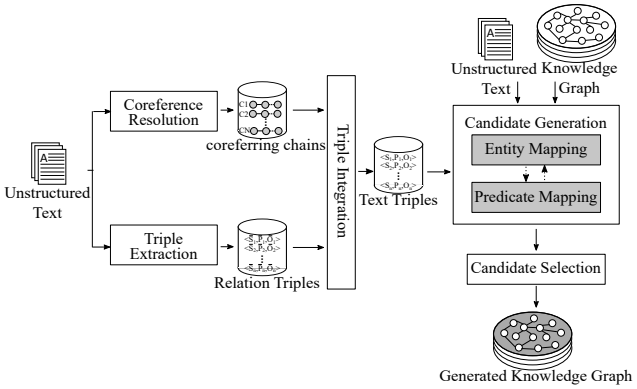

Figure 1: Architecture of the T2KG_Ext Framework

text triples to generate candidate triples, which link an entity of a triple, which usually is subject or object of a triple, and a predicate of a triple to a predefined terms in existing KGs. The candidate selection selects the most suitable candidate triple for each text triple as the populated triple. In this paper, we mainly focus on the candidate generation and the candidate selection components. For details of other components, please refer to the T2KG framework [1].

Candidate Generation. The candidate generation component is to generate candidate triples for a text triple. In the candidate generation component, two modules, entity mapping and predicate mapping, are installed. The entity mapping module is to find candidates for an entity in the text triple, while the predicate mapping is to find candidates for the predicates. For example, given text triple (Ise Shrine, be located in, Ise), the candidate list for entities and predicates is generated as follows.

S: Ise Shrine = { dbr*1:Ise_Grand_Shrine}

P: be located in = { dbr*2:location, dbr:city, dbr:country}

O: Ise = {dbr:Ise_Grand_Shrine, dbr:Ise,_Mie}

All possible combinations of S,P and O are generated as *candidate triples*. Based on this strategy, it enables the framework to generate more candidate triples that could not be obtained by the T2KG framework. Specifically, considering dbr:Ise,_Mie it is not listed as the first candidate. Therefore, the T2KG framework discards this possibility when generating a new triple.

Candidate Selection. The candidate selection is to select the most suitable candidate for a text triple. To select the candidate, we first filter candidate triples that their domain and range are not comply with the pre-defined ontology. For example, (dbr:Ise_Grand_Shrine, dbo:country, dbr:Ise,_Mie) is filtered because the dbo:country property defined its range as dbo:country. However, the type of dbr:Ise,_Mie is not dbo:country. Secondly, we compute a score for each candidate triple by using the rank obtained by the entity mapping and the predicate mapping module as shown in Eq 1.

$$Score(S_i, P_j, O_k) = \frac{R(S_i) + R(P_j) + R(O_k)}{3} \quad (1)$$

∗1 dbr : http://dbpedia.org/resource

∗2 dbo : http://dbpedia.org/ontology

where S_i, O_k and P_j are candidates generated by entity mapping and predicate mapping modules and $R(x)$ is a function returning the rank of x in the candidate list. To select the suitable candidate the candidate triple that has highest score is selected as the correct candidate.

3. Preliminary Experiment

The preliminary experiment is designed to investigate the problems in the T2KG framework and to show some evidences where the T2KG_Ext could deal with such problems. The setup in this experiment follows Experiment 2 [1]. Based upon the results, the generated triples reached precision of 49.39%, recall of 52.26% and F-measure of 50.78%. To further investigate the result, we analysis the ratio of errors caused by the components in the T2KG framework. The result shows that 46.78% of the error caused by the entity mapping and predicate mapping. This means that the first candidate might not be correct when populating the knowledge. Therefore, it is necessary to generate more possible candidate triples so that we could select the suitable candidate triple.

Furthermore, when investigating the results provided by T2KG, we found that some triples are not valid due to the corruption of domain and range. For example, T2KG generates the triple (dbr:Sandbach, dbo:country, dbr:Cheshire). Considering the type of dbr:Cheshire, its type is dbo:location. However, dbo:country requires the range, which is dbo:Country. Consequently, the T2KG framework could not reach the reasonable precision.

To show the possibility, where T2KG_Ext could avoid such problems, we generate the list of entity candidates and predicate candidates. Then, we manually identify whether in the entity and predicate lists the correct entity and predicate are existed or not. We found that more than 50% of correct entity and predicate mapping can be discovered in the candidate list. Therefore, we suppose that we can get the possible candidates. This possibility allows T2KG_Ext framework to select better candidate so that we can reach their reasonable precision.

4. Conclusion

In this paper, we reported the T2KG_Ext framework, which extended the T2KG framework. The advantages of the T2KG_Ext framework over the T2KG framework is that the pre-defined ontology has been used to control populated knowledge as the constraints. Also, the T2KG_Ext framework allows the entity mapping to produce more possible candidates so that the most suitable entity with the suitable predicate will be listed as candidates. In the future, we will propose more sophisticated heuristic score to compute the suitable candidate when populating the knowledge.

References

[1] N. Kertkeidkachorn and R. Ichise. T2KG: An end-to-end system for creating knowledge graph from unstructured text. In *AAAI Technical Report*, 2017.