T2KG : A Demonstration of Knowledge Graph Population from Text and Its Challenges

Natthawut Kertkeidkachorn[1,2] and Ryutaro Ichise[2,1]

[1] National Institute of Advanced Industrial Science and Technology, Tokyo 135-0064, Japan
n.kertkeidkachorn@aist.go.jp
[2] National Institute of Informatics, Tokyo 101-8430, Japan
ichise@nii.ac.jp

Abstract. Knowledge Graphs play an important role in many AI applications as prior knowledge. In recent years, there are many existing Knowledge Graphs such as DBpedia, Freebase, YAGO. Nevertheless, massive amounts of knowledge are being produced every day. Consequently, Knowledge Graphs become more obsolete over time. It is therefore necessary to populate new knowledge into Knowledge Graphs in order to keep them useable. In this study, we present our end-to-end system for populating knowledge graph from natural language text, namely T2KG. Also, we demonstrate use-cases, achievements, challenges, and lessons learned of the system in practice.

1 Introduction

Knowledge Graphs (KGs) store real-world facts in a form of a triple (subject, predicate, object). A triple expresses a relationship (predicate) between entities (subject and object). Recently, KGs have been widely used in many artificial intelligence (AI) related tasks. Examples of such uses include question and answering systems, entity resolution systems, and information retrieval systems. Consequently, immense efforts have been put to construct wide-ranging KGs to support the usages. As a result, there are many available KGs such as DBpedia, Freebase, and GeoNames. However, new knowledge is produced every day. As discussed by Kriz et al. [5], most of the new knowledge generally have been published as natural language text on the web and the rate of publishing natural language text is dramatically growing faster than the growth of KGs. Therefore, it is necessary to automatically populate new knowledge from natural language text to KGs in order to keep existing KGs up to date.

To populate KG, many studies have been proposed, e.g. Reverb, OLLIE, Knowledge Vault. However, most of the approaches focus on extracting knowledge from natural language text without considering existing KGs. Without considering the existing KGs, new knowledge is isolated and the existing KGs could not be updated with such new knowledge. The knowledge integration becomes an essential step to enrich the existing KGs with new knowledge.

In this study, we present our solution for populating knowledge graph from natural language text, namely T2KG. T2KG is the knowledge graph population

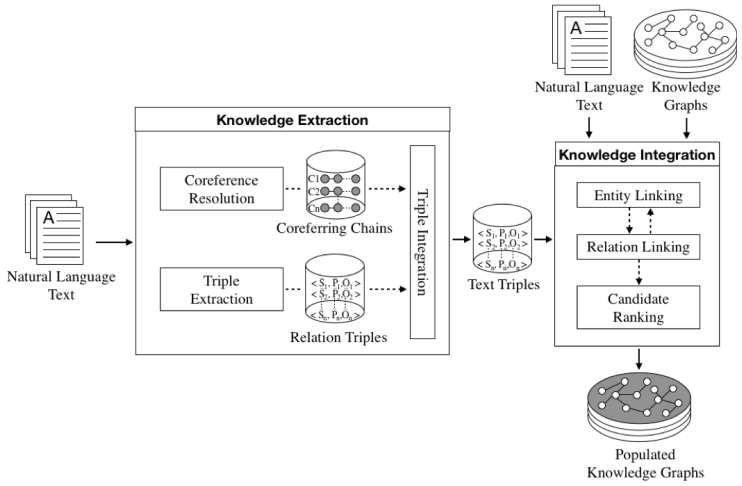

Fig. 1. The Architecture of T2KG

system, which combines the knowledge extraction and the knowledge integration technologies to build an end-to-end solution for populating KGs with or without ontologies. The details of T2KG [2, 3] and the technical details of knowledge integration of T2KG [1, 4] have been reported. Therefore, the position of this paper is the complementary study of T2KG from the studies [2, 3, 1, 4]. Here, we demonstrate the use-case, achievement and lesson learned of T2KG. Also, the video demonstrations and the supplementary are provided.

2 T2KG : Knowledge Graph Population

As shown in Figure 1, T2KG consists of two main components: 1) Knowledge Extraction and 2) Knowledge Integration. The knowledge extraction component deals with extracting new knowledge from text, while the knowledge integration component aims to unify new knowledge to existing KGs. The details of each component and their implementation are as follows.

Knowledge Extraction is a component to retrieve relationships between entities as triples from text. There are three modules: 1) Coreference Resolution, 2) Triple Extraction and 3) Triple Integration. Coreference Resolution resolves surface forms of an entity in natural language text into co-referring chains. Triple Extraction searches and extracts the relation between two entities as a relation triple. Triple Integrate aggregates the co-referring chains and relation triples in order to create text triples. Currently, Coreference Resolution, Triple Extraction and Triple Integration are implemented by using Stanford Core NLP. The configuration of this component is similar to the studies [2, 3].

Knowledge Integration is a component to unify triples and integrate triples into existing KGs. There are three modules: 1) Entity Linking, 2) Predicate Linking and 3) Candidate Ranking. Entity Linking involves linking a subject or an object of a text triple to an existing entity in any KGs. Predicate Linking

Table 1. The error analysis of T2KG on Knowledge Graph Population

Component	All Error proportion	Error Type I	Error Type II
Coreference Resolution	21.60%	45.90%	54.10%
Triple Extraction	35.21%	48.45%	51.55%
Entity Linking	20.19%	61.02%	38.98%
Predicate Linking	23.00%	65.75%	34.25%
All	100.00%	59.86%	40.14%

is to identify similar predicate in existing KGs for a predicate of a text triple. Candidate Ranking is to rank all possible candidates from Entity Linking and Predicate Linking and selects the best candidate to populate to existing KGs. We implemented Entity Linking as presented in the studies [2, 4], Predicate Linking as described in the study [1] and Candidate Ranking as in the studies [2, 3].

3 Use Cases and Demonstration

To demonstrate the use-case and achievements of T2KG, two use cases: KG population and KG construction, are presented as follows. Also, the demonstration videos of KG construction and KG population use cases are available at http://ri-www.nii.ac.jp/T2KG/ .

KG Population: The first T2KG use-case focuses on populating knowledge to the existing KGs. In this use case, the knowledge extraction component creates text triples from texts and the knowledge integration component successfully integrates those text triples into KGs.

KG Construction: The second use-case of T2KG deal with populating a new entity and new relation to KG. In this use case, the knowledge extraction component can extract text triples from texts; however, the knowledge integration component cannot find similar entity or relation. Therefore, a new entity or relations is populated.

As presented in the study [3], the current T2KG system can achieve approximately the F1 score at 50% for KG population and KG construction tasks in open domains.

4 Discussion

In this section, we discuss the challenges and lesson learned from T2KG. To date, we have used T2KG to transform more than 100,000 text articles into text triples and have populated those triples into KGs. To better understand T2KG, we randomly selected 100 sentences from various text articles and analyzed the types of error caused by each module. There are two types of errors: 1) Error Type I and 2) Error Type II. Comparing with the standard, Error Type I occurs when the results are mismatched, while Error Type II is missing result. We also

break down the analysis of these errors for each component as listed in Table 1. Note that, Triple Integration and Candidate Selecting are not considered in the analysis because Triple Integration does not produce any errors and Candidate Selecting currently selects only the best candidate for KG population.

The results show that the majority of error is from Triple Extraction. Consequently, improving the performance of the triple extraction contribute for the overall performance of T2KG. To understand the big picture, we investigate the error at the component level. Coreference Resolution and Relation Extraction are in Knowledge Extraction, while Entity Linking and Predicate Linking are in Knowledge Integration. As shown by the Type II error in Table1, Knowledge Extraction encounters the missing knowledge problem, where the existing knowledge in natural language text could not be extracted due to the complex structure of language. Unlike Knowledge Extraction, it turns out that Knowledge Integration provides Error Type II, which is a mismatch error due to the heterogeneous problem. Therefore, this analysis of T2KG shows that the challenge in Knowledge Extraction is the language complexity, while the challenge in Knowledge Integration is the heterogeneous problem.

5 Conclusion

In this paper, we demonstrated an end-to-end knowledge graph population system from the text, namely T2KG, and discussed the use cases, achievements and lessons learned from it. We are currently planning to release the framework to the public in the future. Our current demonstration video and the supplementary are available on http://ri-www.nii.ac.jp/T2KG/.

Acknowledgment

This work was partially supported by the New Energy and Industrial Technology Development Organization (NEDO).

References

1. N. Kertkeidkachorn and R. Ichise. Leveraging distributed representations of elements in triples for predicate linking. In *Proceedings of International Conference on Hybrid Artificial Intelligence Systems*, pages 75–87, 2017.
2. N. Kertkeidkachorn and R. Ichise. T2KG: An end-to-end system for creating knowledge graph from unstructured text. In *AAAI Workshop on Knowledge-based Techniques for Problem Solving and Reasoning*, pages 743–750, 2017.
3. N. Kertkeidkachorn and R. Ichise. An automatic knowledge graph creation framework from natural language text. *IEICE TRANSACTIONS on Information and Systems*, 101(1):90–98, 2018.
4. N. Kertkeidkachorn, R. Ichise, A. Suchato, and P. Punyabukkana. An automatic instance expansion framework for mapping instances to linked data resources. In *Proceedings of Joint International Semantic Technology Conference*, pages 380–395, 2013.
5. V. Kríž, B. Hladká, M. Nečaskỳ, and T. Knap. Data Extraction Using NLP Techniques and its Transformation to Linked Data. In *Proceedings of Mexican International Conference on Artificial Intelligence*, pages 113–124, 2014.