Generating Conceptual Subgraph from Tabular Data for Knowledge Graph Matching†

Donguk Kim[1], Heesung Park[1], Jae Kyu Lee[1] and Wooju Kim[1*]

[1]Department of Industrial Engineering, Yonsei University, Seoul, Republic of Korea
tmdh78@yonsei.ac.kr, 2020311517@yonsei.ac.kr,
dlworb1994@yonsei.ac.kr, wkim@yonsei.ac.kr

Abstract. In this paper, we study the problem of analyzing the relationship between data given in a tabular format and a target knowledge graph, e.g., Wikidata. It is most important to find the label that indicates the correct meaning in Wikidata where data and values are annotated with each label. It is a very difficult task for a machine to correctly understand or infer its meaning. For this to happen, data must be accurately tagged. Wikidata has a label for each document. In addition, it has the characteristic of being linked to another document through these documents. These connected data can be represented as graphs. In this paper, a method is proposed to create a graph based on related elements and infer the relationship of other elements using advanced Wikidata SPARQL queries. Above all, this approach helps in interpreting clear inference relationships and provides a very suitable approach in an environment where data changes frequently such as an open access database.

Keywords: Knowledge Graph, Wikdata, SPARQL Query, Semantic Annotation.

1 Introduction

Annotating data is one of the important tasks in tabular data. Because other information can be inferred without requiring a lot of information due to accurate annotation. Therefore, putting an appropriate annotation can be considered as knowing the semantics. In that sense, it is very important to find out about the meaning in a tabular knowledge graph. Because fallacy reasoning can lead to another fallacy reasoning in a data processing pipeline. Eventually, fallacy inferences from one can spread throughout. The data we used were based on Wikidata. Wikidata is composed of several facts consisting of subject (S), predicate (P) and object (O). Each element is marked with the label in Wikidata. This makes it possible to identify the semantics in Wikidata [1].

* corresponding author

† This work is financially supported by Korea Ministry of Land, Infrastructure and Transport(MOLIT) as 「Innovative Talent Education Program for Smart City」.

1.1 Task Description

In the SemTab challenge, three tasks were given: CTA, CEA, and CPA [2]. Column Type Annotation (CTA) is assigning a semantic type to a column. Cell Entity Annotation (CEA) is matching a cell to a KG entity. This is to annotate each individual element of subject and object. Columns Property Annotation (CPA) is assigning a KG property to the relationship between two columns. This task is to find out which property the elements in the two columns are connected to on which Wikidata. In other words, this is the process of attaching annotations matching to predicate (Fig. 1.).

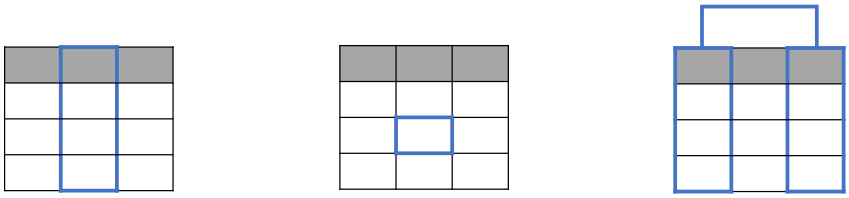

Fig. 1. Challenges in Tabular Data

1.2 Assumptions

We have made the following assumptions to solve the problem.

- Assumption 1. Every target must have a correct answer.
- Assumption 2. First column must be a key value (S) of row for making conceptual graph.
- Assumption 3. Typo cases occur only in the first column.
- Assumption 4. CTA is linked only to property called “instance of”.
- Assumption 5. Lower number labeling item represent wider range of class.

We have established Assumption 1 and assumed that there is always an answer because it is impossible to reason accurately without a clear answer. We try to approach the problem from a graph perspective and find the key part of the graph using Assumption 2. We set out the rest of the assumptions rather aggressively, considering the factors we gained empirically over the course of the round.

2 Conceptual Graph

2.1 Target Table Structure

Given tabular data to perform the task, we approached the table from a matrix perspective and redefine it as follows (Table 1). The subject cell refers to the zero column in

the target table. This cell means the title of the document in Wikidata, and it always has a document label, not a literal form. The object cell refers to the cells of all rows except the zero column in the target table. This cell means objects that exist in the document title in Wikidata, and unlike subject cells, there may be a literal form that is not tagged with a label.

Table 1. Target Table Structure

col0	…	col(j)	...	col(n)
$t_{(1,\,0)}$				$t_{(1,\,n)}$
. . .				. . .
$t_{(i,\,0)}$	…	$t_{(i,\,j)}$	…	$t_{(i,\,n)}$
. . .				. . .
$t_{(m,\,0)}$				$t_{(m,\,n)}$

— Target Table (t) : $m \times n$ matrix.

— Subject Cell : $t_{(i,0)}$ *(i* = 1, 2 *…, m).*

— Object Cell : $t_{(i,j)}$ *(i* = 1, 2 *…, m), (j* = 1, 2 *…, n).*

— Header Row : $t_{(0,j)}$ *(j* = 0, 1, 2 *…, n).*

2.2 Generating Subgraph

If information about the target is given as follows, CTA with column id 0, CPA with head column id and tail column id 0 and 1, and CEA of 0th and 1st column cells, We can find the values in Table 2 and generate a conceptual subgraph as shown in Figure 2.

Table 2. Target Table

col0	col1	col2	col3	col4
Leesmuseum	Amsterdam	Netherlands	1800-11-17	reading museum
The Marlowe	Cambridge	United Kingdom	1907-05-01	theatrical troupe
Club Gorca	Seville	Spain	1966-01-01	organization
Pennsylvania Horticultural Society	Philadelphia	United States of America	1827-01-01	organization
College of Physicians of Philadelphia	Philadelphia	United States of America	1787-01-01	organization

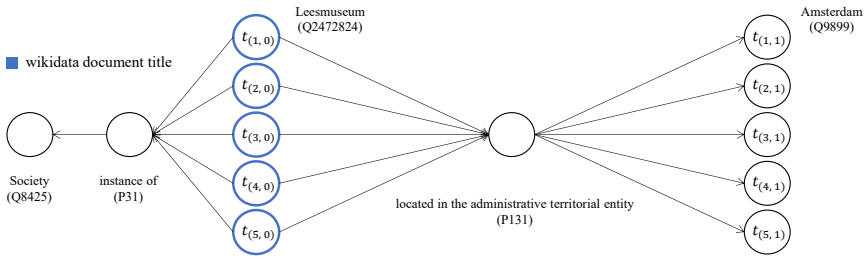

Fig. 2. Conceptual Subgraph

Tabular data can be expressed in the form of knowledge graphs in SPO relationships. Most of all, Wikidata can find all relationships in a document if it has a clear label, such as the title of a document defined as an item label. Blueline means the title of a Wikidata document. All relationships can be found in Wikidata document title label. Based on the document title, it can follow the list of targets on the right and carry out the task matching to the table. To the right of the standard, it is possible to confirm that other objects are connected to the CPA matching location in the administrative terminal entity. Similarly, it can check the CTA connected to the instance connected by Assumption 4 to the left. It is important to find Leesmuseum through these graphs. However, we have prioritized finding the labeling of Leesmuseum (Q2472824). Because if we know at least one subject, we can deduce the remaining factors or classify candidates close to the answer.

3 System Description

In this section, we present the architecture of our system, consisting of 4 stages, as illustrated in Figure 3: Stage 1. Candidate Extraction, Stage 2. Node Selection, Stage 3. Subject Crawling, Stage 4. Element Inference. Each stage is described next in more detail in a separate subsection.

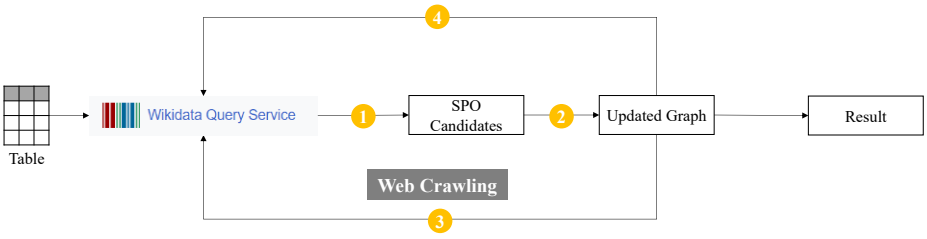

Fig. 3. System Framework

- Stage 1. Find SPO standards using advanced SPARQL query from tabular data.
- Stage 2. Select 'subject' with the high probability value among the Candidates and updated graph.

– Stage 3. Crawl the relevant subject which can't be found, and then repeat Stage 1-2.

– Stage 4. Infer to other people using updated graph, and then repeat stage 1-2.

3.1 Candidate Extraction

Table 3. Target Table of Data Containing Typos

col0	col1	col2	col3	col4
Leesmuseum	Amsterdam	Netherlands	1800-11-17	reading museum
The Marlowe	Cambridge	United Kingdom	1907-05-01	theatrical troupe
Cl?b Gorca	Seville	Spain	1966-01-01	organization
Pen$nsylvania Horticultural Society	Philadelphia	United States of America	1827-01-01	organization
College of Physicians of Philadelphia	Philadelphia	United States of America	1787-01-01	organization

In Stage 1, the advanced SPARQL query that is a rule-based model helps us make the choice of appropriate queries for each data type, such as a constant pattern value (e.g. a date type), a numerical value and text is used to find the values in the tabular table from Wikidata. Since the tabular file can be converted to utf-8 code, preprocessing for the language type was not performed. For each table, according to the appropriate data type, a mix of query features were applied [3]. The data type was determined only by 1st row of table. Because all the correct answers exist by Assumption 1, each column must be of the same type. Assumption 1 gives an important evidence that the first column in Table 3 will have the same attributes, although it contains the typos. In this way, the data type is determined, it can reduce the effort of not having to check all cells. The data types are divided into text, number, and date. But in the case of number, number could be of non-labeled literal type and are also included in the text. As a supplementary explanation, the reason for dividing in consideration of this case is that it is more appropriate to classify numbers as text, which represent the properties of numbers such as prime numbers or even numbers. The database can always be updated, so value such as population, length and width may have a slight error. Reflecting these points, a query was created by specifying a range of $\pm 1.5\%$ based on the value. The query finds one subject and the remaining object cells in a row with a one-to-one match. This search method ensures that the answer to subject always exists among the candidates that came out through the query.

3.2 Node Selection

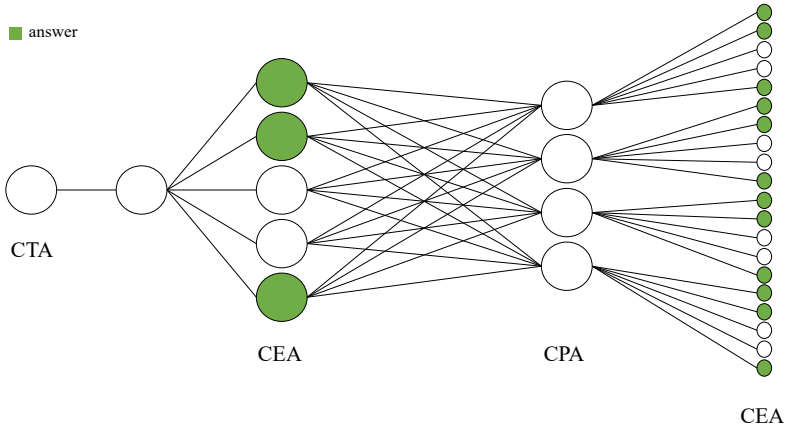

Fig. 4. Updated Graph

Selecting all the candidates using a query, we proceed with a simple preprocessing. As mentioned earlier, numbers can be either text or literal, so we use a query with two things in mind. When such a query is applied, the contents unrelated to the annotation are extracted to the predicate candidate, so the work of removing the candidates with it is proceeded. After preprocessing, the subject with the highest probability of appearance is selected among all candidates. If an equivalent probability is found, all processes for the candidates are performed the same and one is chosen randomly for the final process. When a subject is determined, then objects related to the subject are matched. However, the work of finding the predicate must continue until the rest of the Wikidata page titles are found. Through the determined subject and objects, the work of filling in the nodes in the graph was in progress (Fig. 4).

3.3 Subject Crawling

If there are no errors in the tabular data, the whole process is probably done in the previous step. However, in the assignment, there were many typos and errors in the data. There were many types of data containing incorrect values, such as misspellings, incorrect spacing, and omission of other special symbols or numbers. In order to solve the typo error, the problem was approached by crawling through a search engine. The crawl was performed through the Google search engine, but during the crawling process, several cases were classified and prioritized to perform the crawl. When using the Google search engine, Google sometimes automatically corrects typos and recommends related search terms. Crawling was performed on the top page, and the order was related to the Wikidata title, automatically correcting the typo, and finally the case related to the Wikipedia title.

3.4 Element Inference

After correcting the typo, we repeat the process in Stages 1-2. Then, the system can find a subject that is highly relevant to the subject indicated in yellow on the graph (Fig. 5a). And like subject, the candidate with the highest probability value is selected from the predicate list that has been kept so far. When an equivalent probability value comes out, we create and maintain a predicate list of only candidates with equivalent values. For CTA work, only 'instance of (P31)' is used by Assumption 4. After this process, inferring through the remaining elements in the graph is performed to find the elements corresponding to orange (Fig. 5b). If there is a predicate list composed of equivalent values, the remaining predicates are selected through the inferred elements and the final work is completed.

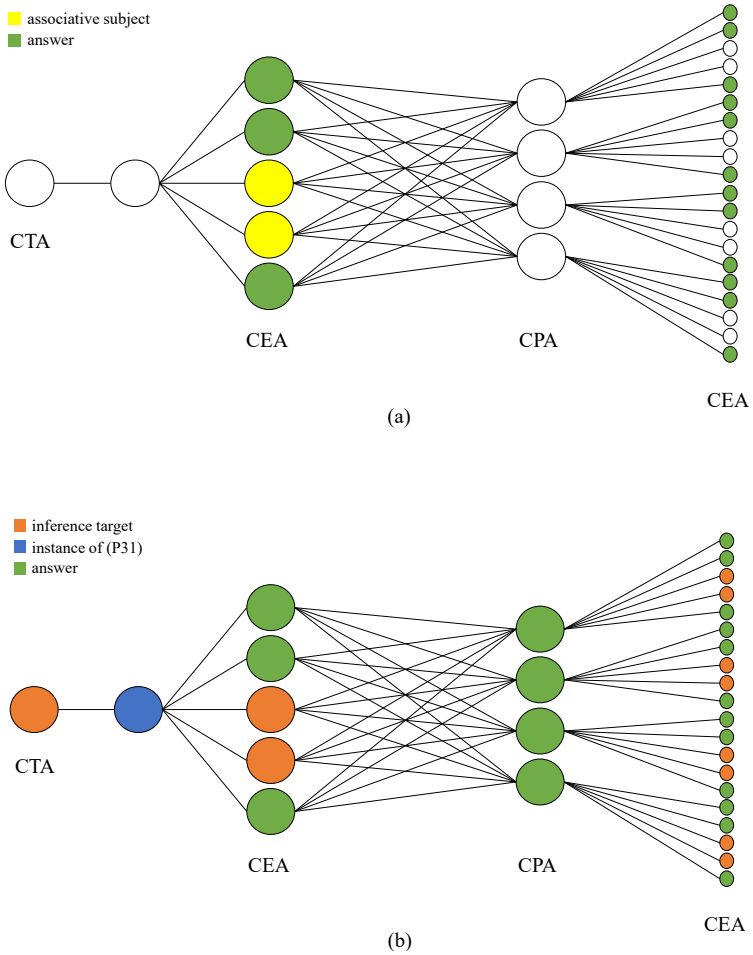

Fig. 5. Update Graph Process. The graph on the upper is the graph after Stage 3.

4 Results

We continued to update our system as we performed round by round. In Round 1 and Round 2, results were good and had few problems. However, in Round 3, the CPA results were particularly bad, including all rounds in Table 4. This result came as the number of rows in the table decreased and the number of errors increased. It was confirmed that if the number of rows in the table is large, incorrect reasoning rarely occurs, but in the opposite case, many errors occur.

Table 4. Challenge Results

	CTA		CEA		CPA	
	AF1-Score	A-Precision	F1-Score	Precision	F1-Score	Precision
Round 1	0.861	0.860	0.936	0.936	0.943	0.943
Round 2	0.966	0.966	0.961	0.961	0.973	0.973
Round 3	0.913	0.913	0.906	0.906	0.815	0.815
Round 4	0.655	0.655	0.617	0.819	0.924	0.924

5 Conclusion

This system presents a method of approaching the semantic table annotation tasks by creating SPRAQL queries and graphs. Accessing Wikidata using queries is very simple and much lighter than downloading a database dump directly. Especially, in the case of small sized data, this advantage is clear. In addition, this approach is well suited to the nature of Wikidata, which has the potential to modify data at any time. There are several improvements to this system. It is a method that can only be applied within the closed world called Wikidata. Additionally, if terms with many comprehensive meanings exist, it takes a lot of time to work. Although many assumptions were set up to solve the problem above, if data problems occur in other cells, a more advanced system is needed rather than a crawl method. This problem can show better performance if we apply learning about pattern sequence in characters.

References

1. Adrian Bielefeldt, Julius Gonsior, and Markus Krötzsch.: Practical Linked Data Access via SPARQL: The Case of Wikidata. In: LDOW@ WWW, (2018).
2. "Ernesto Jiménez-Ruiz, Oktie Hassanzadeh, Vasilis Efthymiou, Jiaoyan Chen, Kavitha Srinivas. SemTab 2019: Tabular Data to Knowledge Graph Matching Challenge. ESWC 2020" or the challenge website: https://www.cs.ox.ac.uk/isg/challenges/sem-tab/
3. Daniel Hernández, Aidan Hogan, Cristian Riveros, Carlos Rojas, and Enzo Zerega.: Querying wikidata: Comparing sparql, relational and graph databases. In: International Semantic Web Conference, pp. 88–103. Springer, (2016).