Towards an Approach based on Knowledge Graph Refinement for Tabular Data to Knowledge Graph Matching

Azanzi Jiomekong[1,*,†], Brice Foko[1,†]

[1]Department of Computer Science, University of Yaounde I, Yaounde, Cameroon

Abstract

This paper presents our contribution to the Accuracy Track of Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab). This contribution consists of the proposition of an approach based on knowledge graph refinement for tabular data annotation. Internal methods were used to predict the links between cells in the table and external methods were used to predict missing entities and relations. This approach was applied to the annotation of HardTables and ToughTables using DBpedia and Wikidata; and GitTables and BiodivTab using DBpedia and Schema.org. During Round 3 of the competition, we were ranked third and second position respectively for the annotation of GitTables and BiodivTab.

Keywords

Tabular Data, Knowledge Graph, Wikidata, DBpedia, Schema.org, Tabular data to Knowledge Graph Matching, SemTab,

1. Introduction

The addition of semantic information to tabular data[1] may enhance a large range of applications such as Web Search, Question Answering, Knowledge Graph construction and refinement, etc. For instance, adding semantic information to a food composition table can allow us to determine which ingredient can be used to substitute another ingredient in the case of allergy. On the other hand, gaining the semantic understanding of food composition tables [1] can improve the food data analysis and facilitate food data integration. However, constructing and assigning semantic tags to tabular datasets is often difficult because of incomplete data, erroneous data, incomplete metadata, and ambiguous data and metadata.

To solve the problem of tabular data to knowledge graph matching, we are proposing an approach based on KG refinement. In order to increase the utility of a graph, KG completion aims to complete the graph by adding missing knowledge such as missing entities, missing types of

Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab 2022)

*Corresponding author.

†These authors contributed equally.

fidel.jiomekong@facsciences-uy1.cm (A. Jiomekong); fokobrice3@gmail.com (B. Foko)

https://sites.google.com/facsciences-uy1.cm/azanzijiomekong (A. Jiomekong)

0000-0002-0877-7063 (A. Jiomekong)

[1] https://sem-tab-challenge.github.io/2022/

entities, and/or missing relations that exist between entities. On the other hand, error detection aims at identifying errors in the KG. These errors can be type assertions, relation between individuals, literal values and KG interlinks. To refine a KG, internal methods use knowledge in the graph and external methods use knowledge that come from external knowledge sources such as text corpora or existing knowledge graphs [2]. In this research, each tabular file is represented as a graph in which each cell is a node, labeled by the content of the cell and can be linked to another cell or column title. Our aim is to correct the misspelling of the cell's label, link the cell to its corresponding annotation in the KG and determine the type of a set of cells and the relation between cells.

The rest of the paper is structured as follows: Section 2 presents the methodology of our research, Section 3 presents the results and Section 4 presents the conclusion.

2. Research methodology

Taking advantage of our experience in empirical research in software engineering [3] and ontology learning [4], we designed the research methodology. Section 2.1 presents the research question, Section 2.2 the empirical research methods used and Section 2.3 presents the pipeline defined after Round 1, refined during Round 2 and used during Round 3 of the SemTab challenge.

2.1. Research question

To solve the tabular data to knowledge graph matching problem, one should reply to the following research question "How to annotate tabular data using knowledge graph?". To reply to this question, one should provide a system which takes as input a tabular dataset and a knowledge graph and furnishes as output the dataset annotated with entities and properties extracted from this KG. To this end, the following questions should be replied (see Fig. 1):

- Which Entity from the KG should be used to annotate a cell in the tabular data? This is the Cell Entity Annotation (CEA).
- What is the most fine-grained semantic type that should be assigned to a tabular data column? This is the Column Type Annotation (CTA) task.
- Which property from the KG should be used to link two columns that are related in the tabular data? This is the Column Property Annotation (CPA).

2.2. Empirical methods

The research methodology consists of the combination of three empirical research methods in software engineering [5]: case study research, action research and experimental research. To solve the tabular data to knowledge graph matching research problem, the SemTab organizers provided us with 8 case studies which are: (1) Annotation of HardTables [6] using Wikidata, (2) Annotation of HardTables using DBpedia, (3) Annotation of ToughTables [7, 8] using Wikidata, (4) Annotation of ToughTables using DBpedia, (5) Annotation of BiodivTab [9, 10] using DBpedia, (6) Annotation of GitTables [11, 12] using DBpedia and (8) Annotation of GitTables using Schema.org. The aim of studying these case studies is to provide a deeper understanding of the

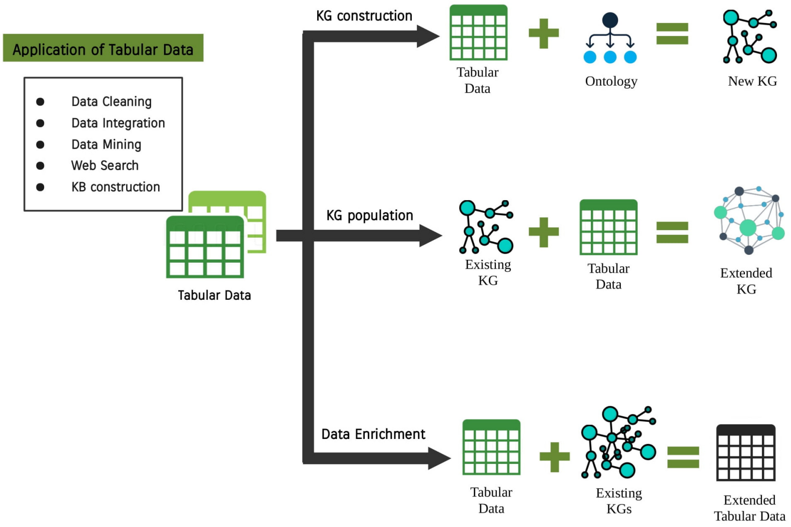


Figure 1: Tabular data to knowledge graphs matching

tabular data to the knowledge graph matching problem so that the proposed solution can be generalized to any setting.

Given that this was our first participation in the SemTab challenge, during Round 1 and Round 2, we applied action research. This consists of the exploration, testing, evaluation of possible solutions and the proposition of a reliable solution that can be used to annotate any tabular data using knowledge graphs. The proposed solution was experimented during Round 3. Given that the solution proposed was a software solution, we used the Scrum process [3]. Globally, the solution we proposed was set-up in 14 Sprints, each Sprint involving one or many iterations. The Scrum Weekly meeting was used to discuss the results obtained during the week, and how to ameliorate the approach. Ad-hoc meeting during the week were used to discuss some problems. For instance, we have had some problems when querying DBpedia and Wikidata online. This problem was solved during Ad-hoc meetings by an algorithm that allowed us to reduce the number of queries by defining links between cells so that the information obtained from a cell can be used to annotate another cell.

2.3. Pipeline

As stated in the introduction, we are using the KG refinement approach to solve the tabular data to knowledge graph matching problem. The two main refinement activities were: error correction and tabular data completion with missing entities and relations.

A deep analysis of the CEA, CTA and CPA tasks during Round 1 allowed us to define the CEA task as the core task. Solving the CEA task allowed us to improve the performance of the CTA and CPA tasks. Exploration of solutions to be used to annotate tabular dataset, applied and evaluated (by the SemTab organizers) allowed us to define the pipeline to be used for the annotation of any tabular data. This pipeline is presented in Fig. 2. It consists of: errors correction in the tabular data (cell pre-processing), and tabular data completion with missing entities and relations (information retrieval, entity discovery and link prediction).

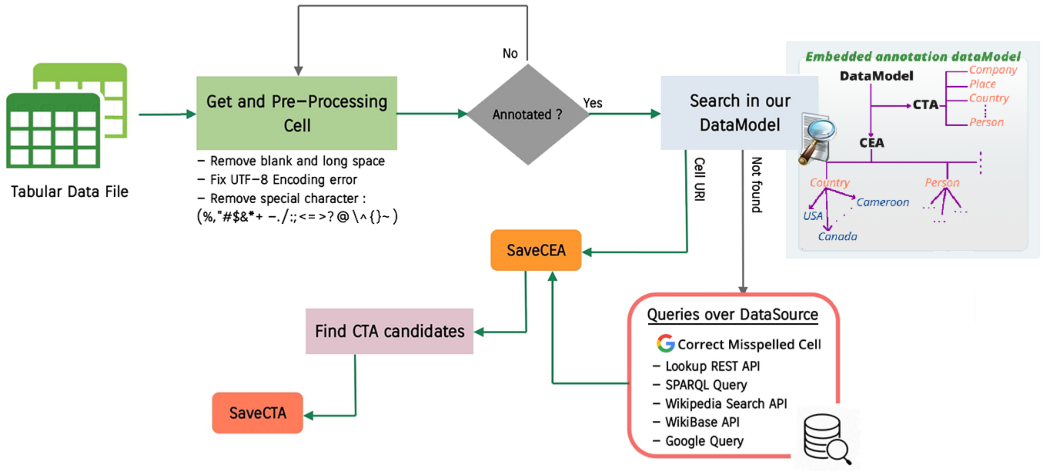
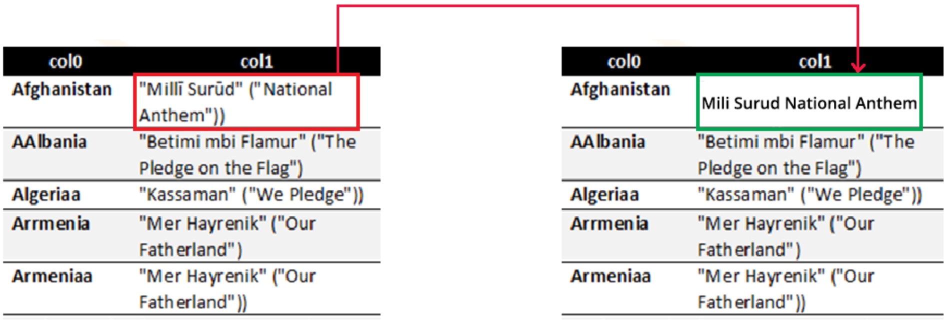


Figure 2: Pipeline of annotation process

2.3.1. Cell Pre-processing



col0	col1
Afghanistan	"Millī Surūd" ("National Anthem"))
AAlbania	"Betimi mbi Flamur" ("The Pledge on the Flag")
Algeriaa	"Kassaman" ("We Pledge"))
Armenia	"Mer Hayrenik" ("Our Fatherland")
Armeniaa	"Mer Hayrenik" ("Our Fatherland"))

col0	col1
Afghanistan	Mili Surud National Anthem
AAlbania	"Betimi mbi Flamur" ("The Pledge on the Flag")
Algeriaa	"Kassaman" ("We Pledge"))
Armenia	"Mer Hayrenik" ("Our Fatherland")
Armeniaa	"Mer Hayrenik" ("Our Fatherland"))

Figure 3: Example of a cell pre-processing

In this approach, we are using SPARQL queries to search for entities in the KG. Thus, the cell pre-processing activity consists of transforming any cell in a form that makes the SPARQL query as efficient as possible. This pre-processing consists of:

1. Removing extra spaces at the beginning, the end and between words in each cell;
2. Removing special characters such as #, (,), [,] etc.

3. Correct the Mojibake[2] errors.

Once processed, the dataset contains cleaned cells that can be used to make queries on the KG. Figure 3 presents an example of a cell that is cleaned during the pre-processing phase.

2.3.2. Completing the tabular data with missing entities and relations

This task is done using the following public endpoints: DBpedia[3], Wikidata[4], Wikibase API[5] and Lookup API[6].

The main problem during this task was the fact that there is a limited number of queries that should be done on a SPARQL endpoint. For instance, after a certain number of queries on Wikidata endpoint, we got the error *429: Too Many Requests*. To solve this problem, we suppose that any cell in a table is linked to other ones. Thus, for each SPARQL query, we extract a subgraph that is processed locally to determine the CEA, CPA and CTA of the table (Fig. 4 is an illustration).

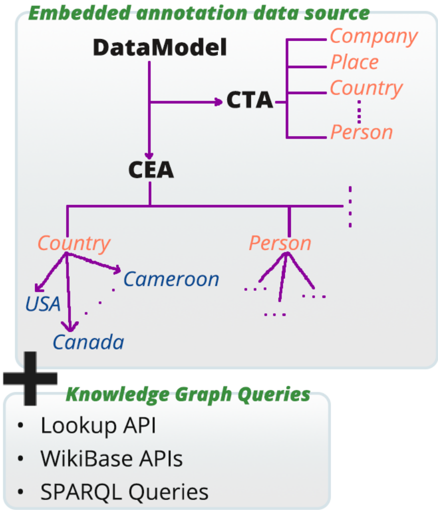


Figure 4: Searching of semantic tag

Entity search The entity searching (CEA task) process starts with the entity misspelling checking using the google service[7]. This is to check any spelling errors and to correct them. Thereafter, the cells obtained are used to define SPARQL queries. It should be noted that in many cases, a cell can have multiple annotations. For instance, the cell "Solomon", can refer to a

[2] Errors due to UTF-8 encoding

[3] https://dbpedia.org/sparql

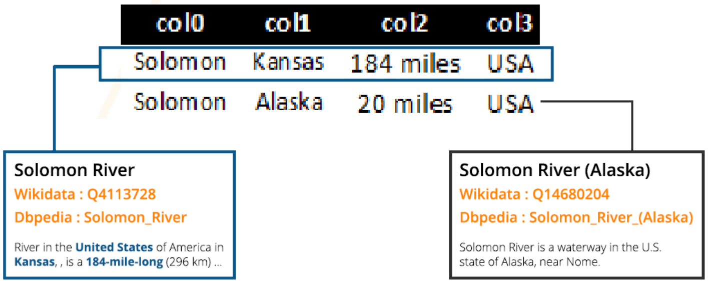
[4] https://query.wikidata.org/sparql

[5] https://en.wikipedia.org/w/api.php?action=query&format=json&prop=info|pageprops&inprop=url&ppprop=wikibase_item&redirects=1&titles=

[6] https://lookup.dbpedia.org/api/search?limit=25&format=json&query=

[7] https://www.npmjs.com/package/googlethis

piece of music (Solomon Handel, Solomon album, etc.), a person (Iser Solomon, Jack Solomon, etc.), a place or a film. To solve this problem, we defined the *cell context* as the elements in the same line of the cell in the table that can be used to lift this ambiguity. Figure 5 illustrates how we proceed for disambiguation.



col0	col1	col2	col3
Solomon	Kansas	184 miles	USA
Solomon	Alaska	20 miles	USA

Solomon River
Wikidata : Q4113728
Dbpedia : Solomon_River
River in the **United States** of America in **Kansas**, , is a **184-mile-long** (296 km) ...

Solomon River (Alaska)
Wikidata : Q14680204
Dbpedia : Solomon_River_(Alaska)
Solomon River is a waterway in the U.S. state of Alaska, near Nome.

Figure 5: Entity disambiguation

Once the entity is identified in the graph, we used cosine similarity measures to calculate the similarity between two cells. This is the prediction of the *isCloseTo* relation. The goal being to assign the annotation of $cell_i$ to $cell_j$ if the triple *($cell_i$ isCloseTo $cell_j$)* holds.

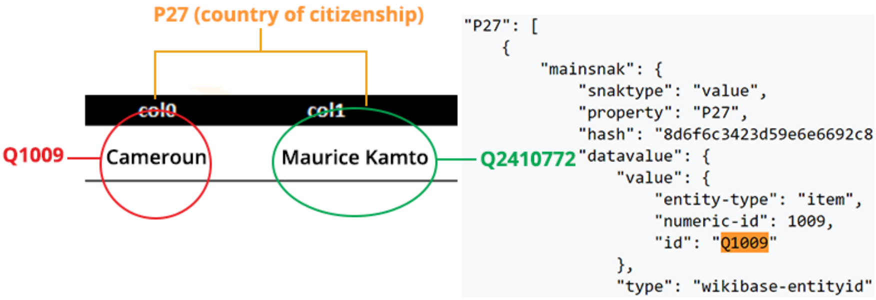
Searching for Semantic Types To search for Semantic Types of columns (CTA task), SPARQL queries are used to search for each CEA, their types. The candidates are sorted and the three most frequent are selected. If one type is the parent of the others, we validate this as the CTA; if not, the CTA having the highest frequency is selected.

Property search This task consists of using the entities to search for all the properties that may exist among these entities. Thereafter, the corresponding property is selected. Figure 6 shows an example of searching for property between two columns. It consists of: (1) Getting the QIDs of entities (Q1009 for "Maurice Kamto" $\longrightarrow$ column 0 - and Q2410772 for "Cameroon" $\longrightarrow$ column 1); (2) searching for all properties that link column 0 to column 1: getting all the properties of the entity Q2410772 and identifying from these properties, the ones that have as range entity Q1009 - in our case, we found P27 (Country of Citizenships). This process is repeated for each row in the tabular data. At its end, the property having the most occurrences is selected as the CPA

The pipeline presented in this section was experimented during Round 3 and we have obtained the second position for the annotation of BioDivTab and the third position for the annotation of GitTables. Figure 7 is an example of the execution of our program during the annotation process.

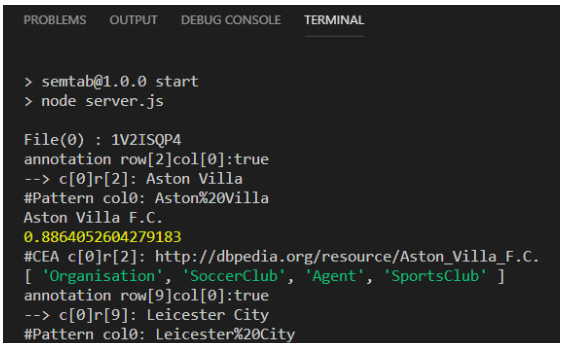
3. Results

The SemTab 2022 challenge consisted of three Rounds that lasted from June 13 to October 15, 2022. In this section, we present the results we obtained during these 3 Rounds.



```
"P27": [
    {
        "mainsnak": {
            "snaktype": "value",
            "property": "P27",
            "hash": "8d6f6c3423d59e6e6692c8
            "datavalue": {
                "value": {
                    "entity-type": "item",
                    "numeric-id": 1009,
                    "id": "Q1009"
                },
                "type": "wikibase-entityid"
```

Figure 6: Knowledge graph property research example



```
> semtab@1.0.0 start
> node server.js

File(0) : 1V2ISQP4
annotation row[2]col[0]:true
--> c[0]r[2]: Aston Villa
#Pattern col0: Aston%20Villa
Aston Villa F.C.
0.8864052604279183
#CEA c[0]r[2]: http://dbpedia.org/resource/Aston_Villa_F.C.
[ 'Organisation', 'SoccerClub', 'Agent', 'SportsClub' ]
annotation row[9]col[0]:true
--> c[0]r[9]: Leicester City
#Pattern col0: Leicester%20City
```

Figure 7: A screenshot of an annotation process made by the tool

3.1. Round 1

Round 1 consisted of the annotation of HardTables using Wikidata KG. The statistics on the dataset provided by the organizers are presented in Table 1.

Table 1
Description of the HardTables dataset for Round 1

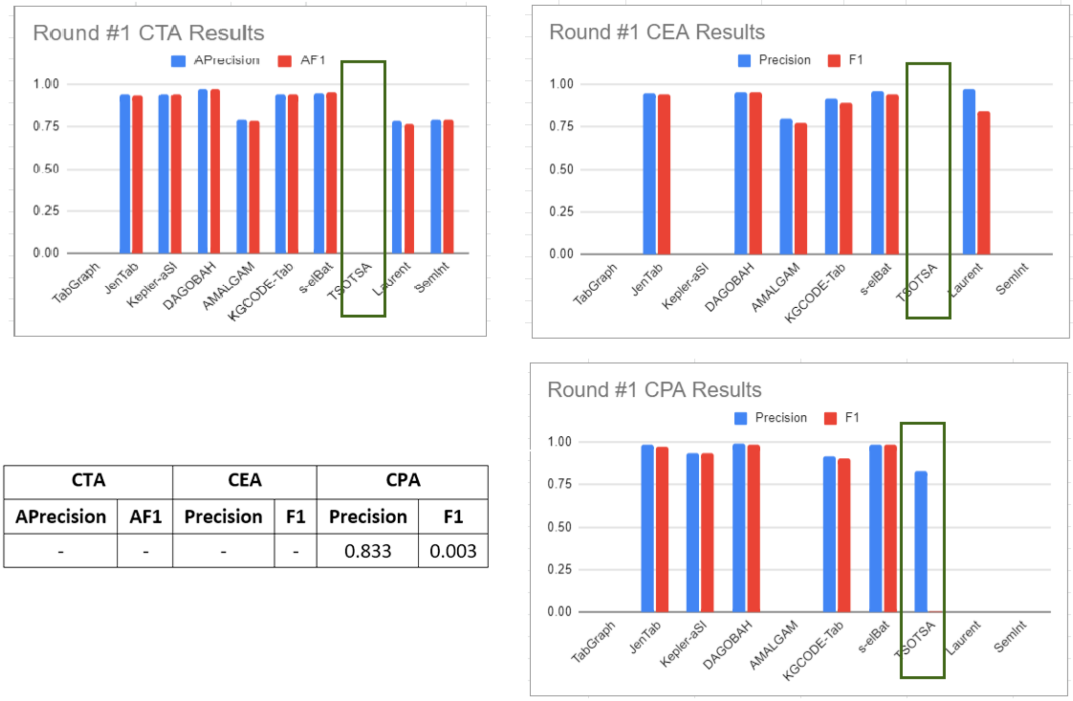
# Files	# Target for CEA task	# Targets for CTA task	# Targets for CPA task
3,691	26,189	4,511	5,745

We joined the challenge at the end of the first Round. Thus, during the last week of the first Round, we annotated 25 files manually in order to understand the tasks. Thus, the following annotations were assigned: 16 CTA over 4,511 targets, 79 CEA over 26,189 targets and 14 CPA over 5,745 targets. Evaluated by the organizers, we obtained the results presented in Fig. 8.

3.2. Round 2

Round 2 consisted of annotating HardTables (see Table 1) and ToughTables (see Table 3 for Wikidata annotation and table 4 for DBpedia annotation) datasets.

During Round 2, we used an automatic annotation approach. From this approach, we have



CTA		CEA		CPA	
APrecision	AF1	Precision	F1	Precision	F1
-	-	-	-	0.833	0.003

Figure 8: Round 1 CTA, CEA and CPA results for HardTables

Table 2
Description of the HardTables dataset for Round 2

# Files	# Target for CEA task	# Targets for CTA task	# Targets for CPA task
4,649	22,009	4,534	3,954

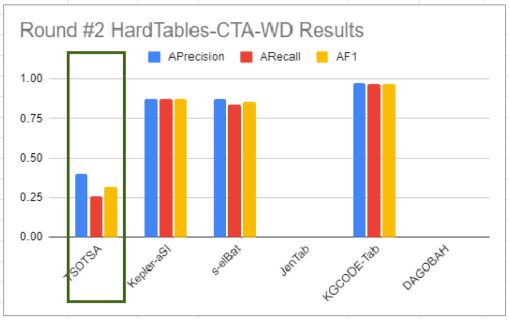
Table 3
Description of the ToughTableWD dataset for Round 2

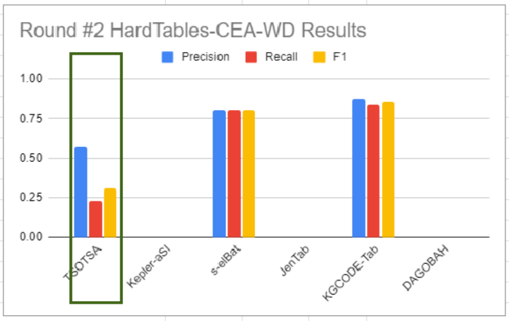
# Files	# Target for CEA task	# Targets for CTA task	# Targets for CPA task
144	586,118	443	-

Table 4
Description of the ToughTableDB dataset for Round 2

# Files	# Target for CEA task	# Targets for CTA task	# Targets for CPA task
144	486,203	429	-

obtained better results than during Round 1. The results from the SemTab organizers are presented in Fig. 9 for HardTables and Fig. 10 for ToughTables.





CTA		CEA		CPA	
APrecision	AF1	Precision	F1	Precision	F1
0.404	0.318	0.571	0.311	0.8	0.002

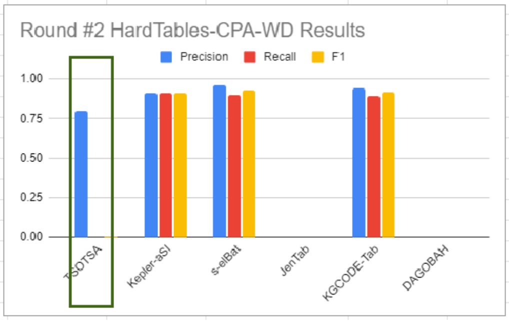
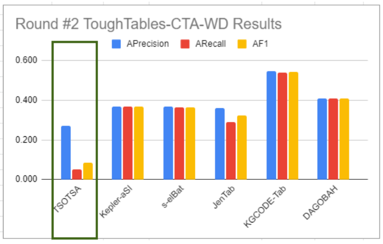
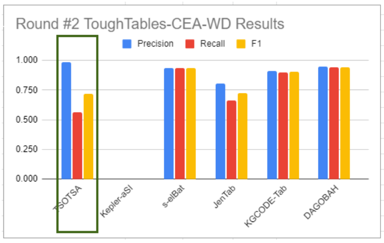


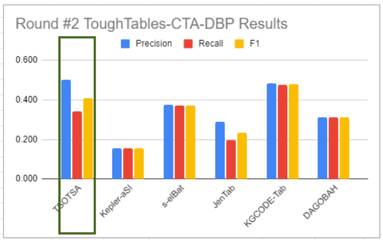
Figure 9: Round 2 CTA, CEA and CPA results for HardTables(EXTRA)

CTA		CEA	
APrecision	AF1	Precision	F1
0.272	0.084	0.986	0.717





CTA		CEA	
APrecision	AF1	Precision	F1
0.502	0.407	0.958	0.693



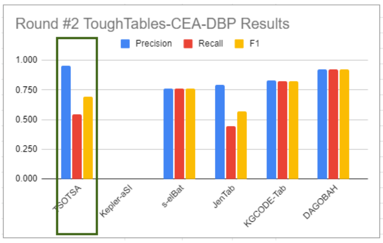


Figure 10: Round 2 CTA and CEA results for ToughTables

3.3. Round 3

Our main goal during Round 3 was to experiment the approach that was defined from Round 1 and Round 2. Round 03 consisted of annotating the BiodivTab (see Table 5) and GitTable (see Table 6 for DBpedia annotation and Table 7 for Schema.org annotation) datasets.

Table 5
Description of the BiodivTab dataset for Round 3

# Files	# Target for CEA task	# Targets for CTA task	# Targets for CPA task
45	31,942	526	-

Table 6
Description of the GitTableDBP dataset for Round 3

# Files	# Target for CEA task	# Targets for CTA task	# Targets for CPA task
6,892	-	6,228	-

Table 7
Description of the GitTableSCH dataset for Round 3

# Files	# Target for CEA task	# Targets for CTA task	# Targets for CPA task
6,892	-	5,411 (4411 properties and 1000 classes)	-

The approach defined after Round 2 and presented in Section 2.3 was evaluated on the datasets of Round 3. The analysis of the results by the SemTab organizers positioned us in the second position for the annotation of the BiodivTab and the third position for the annotation of GitTables (see Fig. 11).

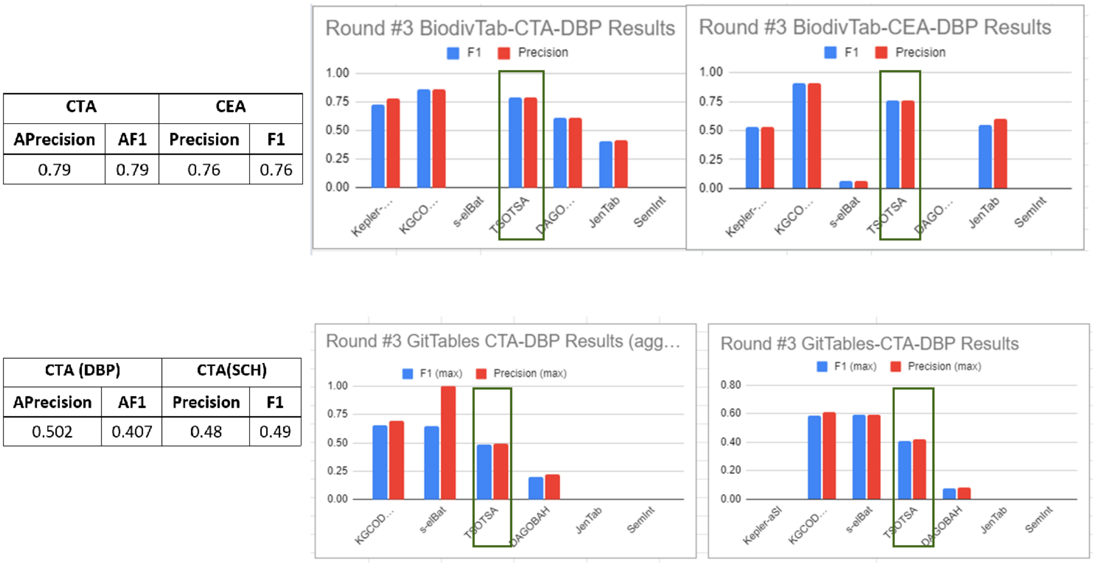
4. Conclusion

This paper presents the approach we proposed for the annotation of tabular data using knowledge graphs. This approach is based on knowledge graph refinement. Error correction aims to put the cells in the table in a form that can be used to make SPARQL queries and to solve disambiguation of cells. Tabular data completion aims to complete the table with missing entities and relations. To add the context and solve the ambiguity problem encountered during the CEA task, we are exploring language models such as BERT.

Online resource

The source code we produced during this work is available on GitHub[8].

[8] https://github.com/jiofidelus/tsotsa/tree/SemTab_22



CTA		CEA	
APrecision	AF1	Precision	F1
0.79	0.79	0.76	0.76

CTA (DBP)		CTA(SCH)	
APrecision	AF1	Precision	F1
0.502	0.407	0.48	0.49

Figure 11: Round 3 CTA and CEA results for BiodivTables and GitTables

Acknowledgment

We are grateful to SemTab organizers for having given us the opportunity to share this work with the community. We are also grateful to Vinsight and neuralearn.ai for the training support.

References

[1] A. Jiomekong, Comparison of food composition tables/databases, 2022. URL: https://orkg.org/comparison/R206121/.

[2] P. Cimiano, H. Paulheim, Knowledge graph refinement: A survey of approaches and evaluation methods, Semant. Web 8 (2017) 489–508. URL: https://doi.org/10.3233/SW-160218. doi:10.3233/SW-160218.

[3] A. Jiomekong, H. Tapamo, G. Camara, Combining Scrum and Model Driven Architecture for the development of the EPICAM platform, in: CARI 2022, Yaounde, Cameroon, 2022. URL: https://hal.archives-ouvertes.fr/hal-03712484.

[4] A. Jiomekong, G. Camara, M. Tchuente, Extracting ontological knowledge from java source code using hidden markov models, Open Computer Science 9 (2019) 181–199.

[5] A. S. I. G. on Software Engineering, Empirical standards, 2020. URL: https://github.com/acmsigsoft/EmpiricalStandards.

[6] O. Hassanzadeh, V. Efthymiou, J. Chen, E. Jiménez-Ruiz, K. Srinivas, SemTab 2021: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching Data Sets, 2021. URL: https://doi.org/10.5281/zenodo.6154708. doi:10.5281/zenodo.6154708.

[7] V. Cutrona, F. Bianchi, E. Jiménez-Ruiz, M. Palmonari, Tough Tables: Carefully Evaluating

Entity Linking for Tabular Data, 2020. URL: https://doi.org/10.5281/zenodo.4246370. doi:10.5281/zenodo.4246370.

[8] V. Cutrona, F. Bianchi, E. Jiménez-Ruiz, M. Palmonari, Tough tables: Carefully evaluating entity linking for tabular data, in: J. Z. Pan, V. Tamma, C. d'Amato, K. Janowicz, B. Fu, A. Polleres, O. Seneviratne, L. Kagal (Eds.), The Semantic Web – ISWC 2020, Springer International Publishing, Cham, 2020, pp. 328–343.

[9] N. Abdelmageed, S. Schindler, B. König-Ries, BiodivTab: A Tabular Benchmark based on Biodiversity Research Data, in: SemTab@ISWC, submitted, 2021.

[10] N. Abdelmageed, S. Schindler, B. König-Ries, fusion-jena/biodivtab, 2021. URL: https://doi.org/10.5281/zenodo.5584180. doi:10.5281/zenodo.5584180.

[11] M. Hulsebos, Ç. Demiralp, P. Groth, Gittables: A large-scale corpus of relational tables, arXiv preprint arXiv:2106.07258 (2021).

[12] M. Hulsebos, Çağatay Demiralp, P. Demiralp, Gittables benchmark - column type detection, 2021. URL: https://doi.org/10.5281/zenodo.5706316. doi:10.5281/zenodo.5706316.