SemTab 2021: Tabular Data Annotation with MTab Tool

Phuc Nguyen[1], Ikuya Yamada[2], Natthawut Kertkeidkachorn[3], Ryutaro Ichise[1], and Hideaki Takeda[1]

[1] National Institute of Informatics, Japan
[2] Studio Ousia, Japan,
[3] Japan Advanced Institute of Science and Technology, Japan

Abstract. This paper presents MTab, an automatic tool for tabular data annotation with knowledge graphs. MTab tool could provide helpful information for tabular data such as structural annotations (e.g., table headers, subject column) or semantic annotations with knowledge graph concepts from Wikidata, DBpedia, and Wikipedia (e.g., cells with entities, columns with types, and column pairs with properties). The tool supports multilingual tables and could process many table formats such as Excel, CSV, TSV, markdown tables, or a pasted table content. MTab achieves impressive empirical performance on many datasets: 1st on HardTable CEA, CTA, CPA tasks, BioTable CTA, CPA tasks, and HardTablesR3 CPA task. Additionally, the system also got the 1st on usability track with advanced features: easy-to-use, generic solution, well-designed user interface. MTab's graphical interface, public APIs, documents are available at `https://github.com/phucty/mtab_tool`.

Keywords: tabular data annotation · knowledge graph · semantic annotation · structural annotation · Wikidata · Wikipedia · DBpedia

1 Introduction

The Open Data movement has made many valuable tabular resources available on the Internet and Open Data Portals. However, due to insufficient data descriptions, various data formats, and terminology issues, the use of tabular data in applications is constrained. Many tabular data lack a description, or the description is not adequately described the data. Table structure and layout are also lacking in many tabular resources. Furthermore, many tables do not employ conventional vocabularies, such as multilingual expressions, abbreviations, ambiguous or many misspellings, and encoding issues. To improve tabular data usability, it is necessary to have a tabular data annotation system capable of providing explicit information about table content.

This paper introduces MTab, an automatic tool that generates structural and semantic annotations for tabular data. MTab tool, as illustrated in Figure 1,

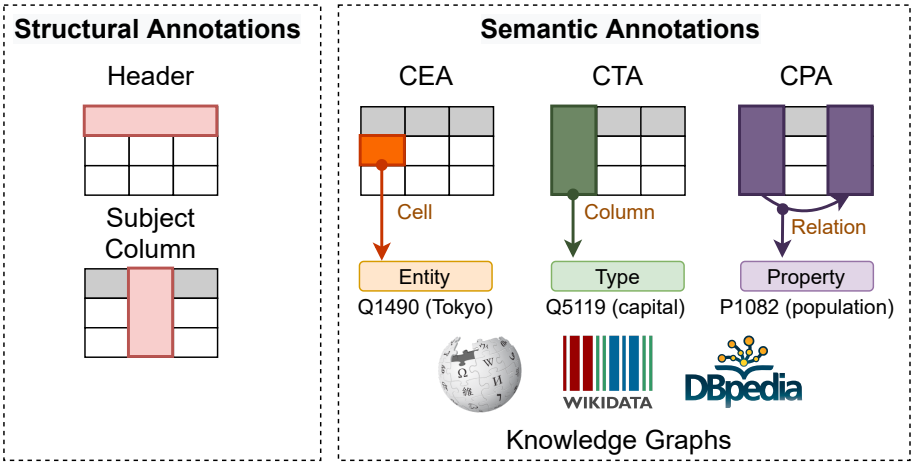

Fig. 1: Tabular data annotations with MTab Tool

could provide helpful information for tabular data such as structural annotations (e.g., table headers, subject column) or semantic annotations with knowledge graph concepts from Wikidata, DBpedia, and Wikipedia, e.g., a cell with entity annotation (CEA task), a column with type (or class) annotation (CTA task), and a column pair with property annotation (CPA task). The tool supports multilingual tables and could process many table formats such as Excel, CSV, TSV, markdown tables, or a pasted table content.

MTab archives impressive performance on many datasets: 1st on HardTable CEA, CTA, CPA tasks, BioTable CTA, CPA tasks, and HardTablesR3 CPA task. Additionally, the system also got the 1st on usability track with advanced features: easy-to-use, generic solution, well-designed user interface. The user could access MTab's graphical interface, APIs, documents at `https://github.com/phucty/mtab_tool`.

2 Related Work

Table understanding is an important task for data integration and management. Much of the previous research on table understanding has addressed many data annotation tasks such as structural annotations, e.g., table header detection, subject column prediction as in [17], [20], [7] or semantic annotations, e.g., cell-entity annotation (CEA), column-type annotation (CTA), and column pair-property annotation (CPA) as the participant systems in the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching: SemTab 2019 [12], and SemTab 2020 [13].

SemTab 2019 is the Semantic Web challenge on tabular data to DBpedia matching. There are three annotations tasks of CEA, CTA, and CPA, and the tabular data was generated from DBpedia. MTab (the winner system) is based on

an aggregation of multiple cross-lingual lookup services and probabilistic graphical models [16]. CSV2KG (IDLab) also uses multiple lookup services to improve matching performance [24]. Tabular ISI implements the lookup part with Wikidata API, and Elastic Search on DBpedia labels and aliases [23]. ADOG [19] system also uses Elastic Search to index knowledge graph. LOD4ALL first checks whereas there is an available entity which has a similar label with table cell using ASK SPARQL, else perform DBpedia entity search [15]. DAGOBAH system performs entity linking with a lookup on Wikidata and DBpedia; the authors also used Wikidata entity embedding to estimate the entity type candidates [3]. Mantis Table provides a Web interface and API for tabular data matching [6].

In SemTab 2020, the matching target knowledge graph is Wikidata including new set of difficulties such as larger-scale of data, graph shifting, rich and complex data schema in Wikidata. Beside the generated tabular data from Wikidata, there was a new manually curated dataset (tough tables [8]). The winner system, MTab4Wikidata proposed new fuzzy entity and statement search methods to improve entity candidate generation (with 99.89% coverage) [18]. The bbw system [21] are based on contextual matching and meta-lookup with SearX metasearch engine to deal with spelling mistakes. LinkingPart [4], DAGOBAH [11], JenTab [1], MantisTable SE [5], SSL [14], AMALGAM [2] systems proposed new scoring functions to rank the matching results.

However, most solutions or systems are not available to use or require extensive configuration, setup, high computing power, or high time complexity [25]. We implement the MTab tool and release the public APIs and interfaces to address the usability issue of the current annotation systems.

3 MTab Tool

This section describes MTab tool, started with the system assumptions in Section 3.1, then the overall framework is described in Section 3.2.

3.1 Assumptions

Assumption 1 *MTab tool is built on a closed-world assumption.*

It means that the tool could return incorrect answers if table elements are not available in the knowledge graph.

Assumption 2 *We assume that the input tables are horizontal relational types.*

A horizontal relational table contains semantic knowledge graph triples in [subject, predicate, object]. The table also has a subject column containing entity names and the relation between the subject column and other columns representing the predicate relation between the entities (subject) and attribute values (object).

Assumption 3 *We assume that all the cell values of the same column have the same data type, and the entities related to cell values are of the same type.*

Assumption 4 *MTab tool treats input tables independently.*

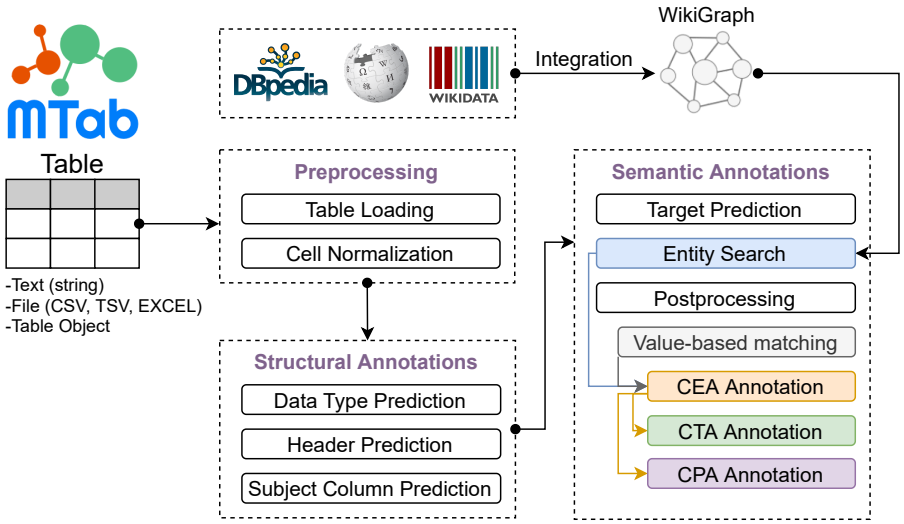

Fig. 2: MTab tool framework

3.2 Framework

In this paper, we focus on the usability factor of the annotation system. So, we implement the MTab tool to support multilingual tables and could process various table formats. The system efficiency also is an important concern of the implementation so that we optimize the annotations run time by about 1.52 sec/table on average (tested on SemTab 2020 dataset). Moreover, we also provide graphical interfaces to visualize the annotation results as in Section 4.

The overall framework of the MTab tool is described in Fig. 2. We build WikiGraph, which is an integrated knowledge graph from Wikidata, DBpedia, and Wikipedia as in Section 3.2.1. The annotation procedure is started with data preprocessing as in Section 3.2.2. Then, the system performs data type prediction, header prediction, and subject column prediction as in the structural annotations section (Section 3.2.3). Finally, MTab performs semantic annotations as in Section 3.2.4.

3.2.1 Knowledge Graph We build a WikiGraph from the dump data of Wikidata, Wikipedia, and DBpedia as the target knowledge graph for the annotation tasks. With the dump data on 1 January 2021, we extracted 91.2 million entities and 249.3 million entity labels in multilingual, including entity labels, aliases, other names, redirect entity labels, and disambiguation entities. We also extracted 3.5 billion triples in WikiGraph. Additionally, WikiGraph will be updated frequently based on the future released dumps of knowledge graphs (Wikidata, Wikipedia, and DBpedia).

3.2.2 Preprocessing

Table Loading : MTab tool supports the three types of input tables, including text (table content as a string), file object (table file such as CSV, TSV, EXCEL), and table object (matrix of rows and columns). The tool automatically predicts the encoding used in the input table and loads the table content based on the predicted encoding.

Table Cell Normalization: We remove HTML tags and non-cell-values such as -, NaN, none, null, blank, unknown, ?, #. Additionally, we use the *ftfy* tool [22] to fix all noisy cells caused by incorrect encoding during file loading.

3.2.3 Structural Annotations

Data Type Prediction The system firstly predicts a table cell's data type into either non-cell (empty cell), literal, or named-entity (NE). We use the pre-trained SpaCy models [10] (trained using the OntoNotes 5 dataset) to identify named entities (PERSON, NORP, FAC, ORG, GPE, LOC, PRODUCT, EVENT, WORK_OF_ART, LAW, LANGUAGE) and date-time and numeric entities (DATE, TIME, PERCENT, MONEY, QUANTITY, ORDINAL, CARDINAL). We associate the named entities to NE type, and date-time and numeric entities to literal types. If there is no assigned named entities of SpaCy outputs, we associate the cell type as NE because the SpaCy model could miss recognized named-entity of table cells.

Next, the system predicts a table column's data type into either a non-match column (empty column), a literal, or a named-entity column. The column data type is derived from the majority voting of all cell data types in this column.

Header Prediction We use simple heuristics to predict table headers as follows.

- Table headers could be located in some of the first rows of a table.
- If the list of data types of the header candidate row differs from most data types of the remaining rows, the candidate is the table header. For example, the list of data types of header candidate (row) is [named-entity, named-entity, named-entity], while the list of the majority data type of remaining rows is [named-entity, literal, literal].
- We also found that the length of header text is empirically shorter or longer than the remaining data rows. If the length of values of the header candidate row is less than the 0.05 quantile or larger than the 0.95 quantiles of the length of the value of remaining rows, the candidates are the table header.

Subject Column Prediction We adopt the heuristics proposed by Ritze et al. [20] as well as modify a simple heuristic to predict the subject column of a table as follows.

- A column is a subject column when its data type is a named-entity type.
- The average cell value length is from 3.5 to 200. We also add a restriction that only considers non-header cells since the length of table headers could differ from the remaining cells.

– The subject column is determined based on the uniqueness score as an increased score for columns with many unique values and reduces the score for columns with many missing values. The subject column is the highest unique score column. If we have many columns that have the same score, the left-most column is chosen.

3.2.4 Semantic Annotations

Matching Target Prediction: MTab automatically predicts the matching targets based on data types when the input does not have matching targets. The CEA matching targets are the table cells whose data types are named entity types. The CTA matching targets are columns so that the column data types are named entity types. The CPA matching targets are the relation between the subject column and the remaining table columns.

Entity Search: We perform entity candidate generation for each table cell with the entity search modules. MTab tool provides the three entity search modules, i.e., keyword search, fuzzy search, and aggregation search[1]. We implement the keyword search using BM25 algorithm with the hyper-parameters as $b = 0.75, k_1 = 1.2$. The fuzzy search is implemented using Damerau–Levenshtein edit distance. We perform candidate filtering and hashing with pre-calculating entity label deletes as the Symmetric Delete algorithm [9] to reduce the number of operations on pairwise edit distance calculation and capable of up to six edits. In the aggregation search, we combine the results of keyword search and fuzzy search. In our experiments, we use the aggregation search as the default entity search.

Post-Processing: We calculate context similarities with the value-based matching between statements of entity candidates in the subject column with table row values. Finally, generate the annotations for entities, properties, and types based on majority voting of context similarities [18].

4 Interfaces

4.1 Entity Search

The entity search interface is available at `https://mtab.app/mtabes`. Fig. 3 depicts an example of entity search with the query of "2MASS J10540655-0031018". MTab tool supports multilingual search so that users could type entity name expressed in any language.

[1] Entity Search Documents: `https://mtab.app/mtabes/docs`

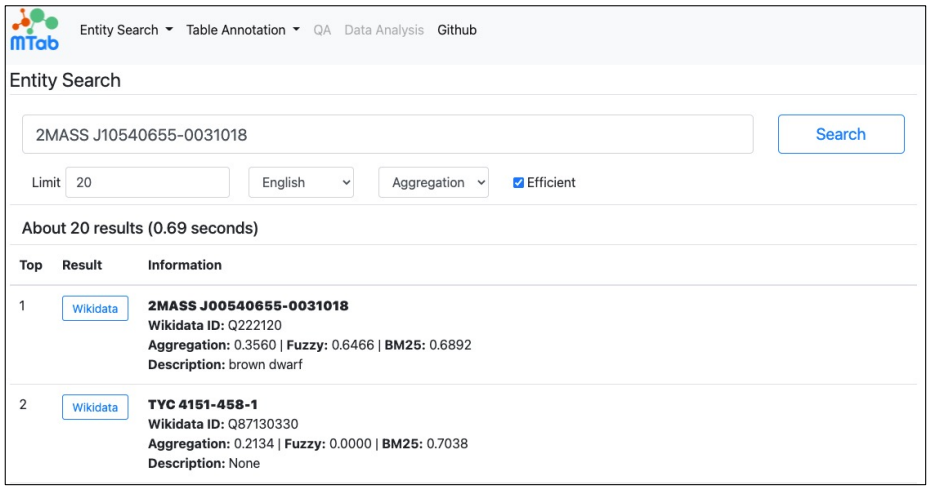

Fig. 3: Example of entity search with MTab

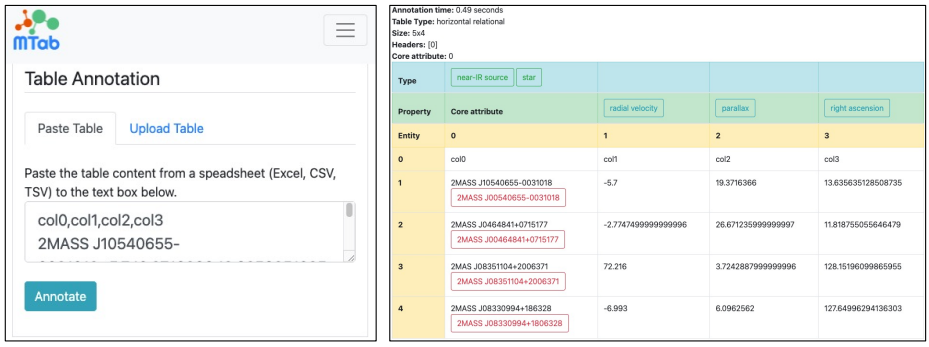

Fig. 4: Example of tabular data annotation with MTab

4.2 Table Annotation

The table annotation interface is available at `https://mtab.app`. Users could submit table files in various table formats, expressed in any language to MTab API, or copy data content and paste it to the interface. Then, users could tap the "Annotate" button to get the annotation results.

Fig. 4 illustrates an annotation example of a SemTab dataset's table. MTab took 0.49 seconds to annotate a pasted table from the text box (left picture). The photo on the right is the annotation results. The table header is in the first row, and the subject column is in the first column. Entity annotations are in red and located below the table cell value. The type annotation is in green and located in the "Type" column. Finally, the relations between the subject column and other columns are in blue and located in the property column.

Table 1: Overall result of MTab tool on HardTable and BioTable Datasets at SemTab 2021

Dataset	CEA		CTA		CPA	
	F1	Rank	AF1	Rank	F1	Rank
HardTable	0.985	1	0.977	1	0.998	1
BioTable	0.964	2	0.956	1	0.947	1
BioDivTab	0.522	2	0.123	3	-	-
HardTablesR3	0.968	2	0.984	2	0.993	1

5 SemTab 2021 Results

Table 1 reports the overall results of the MTab tool for three matching tasks (CEA, CTA, and CPA) of HardTable, BioTable, BioDivTab, and HardTalesR3 Datasets. Overall, these results show that MTab tool achieves impressive performances on many datasets: 1st on HardTable CEA, CTA, CPA tasks, BioTable CTA, CPA tasks, and HardTablesR3 CPA task. MTab tool consistently archive the best performance in CPA task on many dataset. The detail of results of all SemTab 2021 participants are available in AICrowd[2].

Additionally, we also release public APIs and graphical interfaces that enable users access annotations without doing many intensive setup or configuration. At the end, MTab tool also got the first rank in the usability track with advanced features: easy-to-use, generic solution, well-designed user interface.

6 Conclusions

This paper presents the MTab tool for table annotation with Wikidata, DBpedia, and Wikipedia knowledge graphs. MTab tool achieves promising performance on many datasets of SemTab 2021. Moreover, the system also got the first rank of usability track.

In the future work, we will focus on efficiency improvement of the MTab tool by processing only small parts of table content and continues expanding until there is no difference in the annotation results. Another direction is building downstream applications based on MTab's annotations, such as question answering and data analysis.

Acknowledgements

The research was supported by the Cross-ministerial Strategic Innovation Promotion Program (SIP) Second Phase, "Big-data and AI-enabled Cyberspace Technologies" by the New Energy and Industrial Technology Development Organization (NEDO).

[2] SemTab 2021 Leaderboards: `https://www.aicrowd.com/challenges/semtab-2021/leaderboards`

References

1. Abdelmageed, N., Schindler, S.: Jentab: Matching tabular data to knowledge graphs. In: SemTab@ ISWC. pp. 40–49 (2020)
2. Azzi, R., Diallo, G.: Amalgam: A matching approach to fairfy tabular data with knowledge graph model. arXiv preprint arXiv:2101.06637 (2021)
3. Chabot, Y., Labbe, T., Liu, J., Troncy, R.: Dagobah: an end-to-end context-free tabular data semantic annotation system. In: SemTab@ ISWC. pp. 41–48 (2019)
4. Chen, S., Karaoglu, A., Negreanu, C., Ma, T., Yao, J.G., Williams, J., Gordon, A., Lin, C.Y.: Linkingpark: An integrated approach for semantic table interpretation. In: SemTab@ ISWC. pp. 65–74 (2020)
5. Cremaschi, M., Avogadro, R., Barazzetti, A., Chieregato, D.: Mantistable se: an efficient approach for the semantic table interpretation. In: SemTab@ ISWC. pp. 75–85 (2020)
6. Cremaschi, M., Avogadro, R., Chieregato, D.: Mantistable: an automatic approach for the semantic table interpretation. In: SemTab@ ISWC. pp. 15–24 (2019)
7. Cremaschi, M., De Paoli, F., Rula, A., Spahiu, B.: A fully automated approach to a complete semantic table interpretation. Future Generation Computer Systems **112**, 478–500 (2020)
8. Cutrona, V., Bianchi, F., Jiménez-Ruiz, E., Palmonari, M.: Tough tables: Carefully evaluating entity linking for tabular data. In: ISWC. pp. 328–343. Springer (2020)
9. Garbe, W.: Symspell: Symmetric delete algorithm. `https://github.com/wolfgarbe/SymSpell` (2012)
10. Honnibal, M., Montani, I.: spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing (2017), `https://spacy.io/`, to appear
11. Huynh, V.P., Liu, J., Chabot, Y., Labbé, T., Monnin, P., Troncy, R.: Dagobah: Enhanced scoring algorithms for scalable annotations of tabular data. In: SemTab@ ISWC. pp. 27–39 (2020)
12. Jiménez-Ruiz, E., Hassanzadeh, O., Efthymiou, V., Chen, J., Srinivas, K.: Semtab 2019: Resources to benchmark tabular data to knowledge graph matching systems. In: ESWC. vol. 12123, pp. 514–530. Springer (2020)
13. Jimenez-Ruiz, E., Hassanzadeh, O., Efthymiou, V., Chen, J., Srinivas, K., Cutrona, V.: Results of semtab 2020. In: SemTab@ISWC. vol. 2775, pp. 1–8 (2020)
14. Kim, D., Park, H., Lee, J.K., Kim, W.: Generating conceptual subgraph from tabular data for knowledge graph matching. In: SemTab@ ISWC. pp. 96–103 (2020)
15. Morikawa, H.: Semantic table interpretation using lod4all. In: SemTab@ ISWC. pp. 49–56 (2019)
16. Nguyen, P., Kertkeidkachorn, N., Ichise, R., Takeda, H.: Mtab: Matching tabular data to knowledge graph using probability models. In: SemTab@ISWC 2019. vol. 2553, pp. 7–14 (2019)
17. Nguyen, P., Kertkeidkachorn, N., Ichise, R., Takeda, H.: Tabeano: Table to knowledge graph entity annotation. CoRR **abs/2010.01829** (2020)
18. Nguyen, P., Yamada, I., Kertkeidkachorn, N., Ichise, R., Takeda, H.: Mtab4wikidata at semtab 2020: Tabular data annotation with wikidata. In: SemTab@ISWC. vol. 2775, pp. 86–95 (2020)
19. Oliveira, D., d'Aquin, M.: Adog-annotating data with ontologies and graphs. In: SemTab@ ISWC. pp. 1–6 (2019)
20. Ritze, D., Lehmberg, O., Bizer, C.: Matching html tables to dbpedia. In: Proceedings of the 5th International Conference on Web Intelligence, Mining and Semantics, WIMS 2015. pp. 10:1–10:6. ACM (2015)

21. Shigapov, R., Zumstein, P., Kamlah, J., Oberländer, L., Mechnich, J., Schumm, I.: bbw: Matching csv to wikidata via meta-lookup. vol. 2775, pp. 17–26 (2020)
22. Speer, R.: ftfy. Zenodo (2019), `https://github.com/LuminosoInsight/python-ftfy`, version 5.5
23. Thawani, A., Hu, M., Hu, E., Zafar, H., Divvala, N.T., Singh, A., Qasemi, E., Szekely, P.A., Pujara, J.: Entity linking to knowledge graphs to infer column types and properties. In: SemTab@ ISWC. pp. 25–32 (2019)
24. Vandewiele, G., Steenwinckel, B., De Turck, F., Ongenae, F.: Cvs2kg: Transforming tabular data into semantic knowledge. In: SemTab@ ISWC. pp. 33–40 (2019)
25. Wang, D., Shiralkar, P., Lockard, C., Huang, B., Dong, X.L., Jiang, M.: TCN: table convolutional network for web table interpretation. In: WWW '21. pp. 4020–4032. ACM / IW3C2 (2021)