Demonstration of MTab: Tabular Data Annotation with Knowledge Graphs

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 a demonstration of MTab, a tabular data annotation toolkit with knowledge graphs: Wikidata, Wikipedia, and DBpedia. MTab is the best performance system for all semantic annotation tasks at the Semantic Web Challenges on tabular data to knowledge graph matching SemTab 2019 and SemTab 2020. This paper introduces MTab's public APIs capable of structural and semantic annotations for tabular data. We also provide a graphical interface to visualize the annotation results. The tool supports multilingual tables and could process many table formats such as Excel, CSV, TSV, markdown tables, or a pasted table content. MTab's repository is publicly available at `https://github.com/phucty/mtab_tool`.

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

1 Introduction

Many valuable tabular resources have been made available on the Internet and Open Data Portals, thanks to the Open Data movement. However, the usage of the tabular data is very limited in applications due to lacking or insufficient data descriptions, various data formats, vocabulary issues. Tabular data usually do not have a description, or the description does not cover data content. Tabular data also lack specification on table structure, and layout. Moreover, many tables do not use a standard vocabulary such as expressed in non-English, abbreviation, ambiguous or contain many misspellings, encoding problems. It is crucial to have a tabular data annotation system that could provide explicit information about table content to improve tabular data usability.

Previous studies addressed many tabular data annotation tasks such as structural annotations [6], [9] or semantic annotations as the participant systems in the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching:

SemTab 2019 [3], and SemTab 2020 [4]. Unfortunately, most solutions or systems are not available to use, or require extensive configuration, setup, high computing power, or high time complexity [10].

This paper introduces MTab, a public service that generates structural and semantic annotations for tabular data. The structural annotations provide information about table headers, the table core attribute. The semantic annotations offer table elements matching knowledge graph concepts: cell-entity (CEA task), column-type (CTA task), and CPA task where the relation between core attribute to another column is annotated with a property. We also provide a graphical interface to visualize the annotation results.

The major advantages of MTab compared to other systems are as follows.

- **Effectiveness:** MTab tool is the best performance system in SemTab 2019 [5], [3] and SemTab 2020 [7], [4]. The key success of MTab is on the entity search modules with multilingual support (a keyword search with BM25 algorithm, a fuzzy search with edit distances, and an aggregation search with weighted fusion of keyword search and fuzzy search). The fuzzy search could support up to six edits (on the low-budget mac mini M1 2021), while most other systems only support two edits. As a result, MTab could address a higher level of noisiness compared to other systems. The entity search module achieves 87.98% on average of the top 1 accuracy (the top 1000 accuracy is 99.7%) [8] on Semtab 2020 [4] and Tough Tables datasets [1].
- **Efficiency:** MTab fuzzy search implementation works efficiently with candidate filtering based on entity labels and hashing with pre-calculating entity label deletes as the Symmetric Delete algorithm [2]. Moreover, the statement search also gives a tremendous efficient improvement where it could eliminate non-statements entity candidates. Additionally, we use a light way solution as the value matching to calculate the context similarity between entity candidate statements and table row values. The experiments show that our solution could improve efficiency without losing effective performance [4]. Overall, it takes only 1.52 seconds/table on average (SemTab 2020 dataset) to annotate with MTab.
- **Easy to use:** We provide public APIs, graphical interfaces so that users do not need to do intensive setup or configuration. MTab also supports multilingual and could process many table formats such as Excel, CSV, TSV, or markdown tables. According to Wang et al., they only could generate the annotations using the MTab tool, while other systems require high time complexity to process [10].
- **Privacy Policy:** MTab does not store any data from users. All users' tabular data files are completely deleted after the annotation.

MTab's repository, API documents, and other information could be accessed at `https://github.com/phucty/mtab_tool`; the demonstration video is available at `https://youtu.be/0ibTWeObWaA`.

2 MTab

2.1 Knowledge Graphs

We build a WikiGraph from the dump data of Wikidata, Wikipedia, and DBpedia as the target knowledge graph the annotation tasks. Wikidata is the central knowledge graph because it has the largest number of entities among the three graphs. 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).

2.2 Entity Search Modules

Entity Search on a Cell We introduce the search modes[1] as follows [8].

- **Keyword search with BM25 algorithm:** We use the hyper-parameters as $b = 0.75, k_1 = 1.2$.
- **Fuzzy search with edit distance:** We use Damerau–Levenshtein distance as the edit distance for fuzzy search. We also perform candidate filtering and hashing with pre-calculating entity label deletes as the Symmetric Delete algorithm [2] to reduce the number of operations on pairwise edit distance calculation. Overall, MTab could support the fuzzy search up to six edits.
- **Aggregation search:** This module is a weighted fusion of the keyword search and the fuzzy search results.

Statement Search on Two Cells This module is built on the assumption that there is a logical relation between two cells of a table row, equivalent to a knowledge graph triple. We only keep the candidates of the two cells that have equivalent statements in the WikiGraph. We implement this statement search with a sparse matrix of 91 million entities and around 500 million edges.

2.3 Table Annotation: Use Case and Demo

MTab demonstration 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 to the "Annotate" button to get the annotation results. MTab will perform the following steps.

The annotation procedure[2] are as the following steps:

[1] Entity Search Documents: `https://mtab.app/mtabes/docs`
[2] Table Annotation Document: `https://mtab.app/mtab/docs`

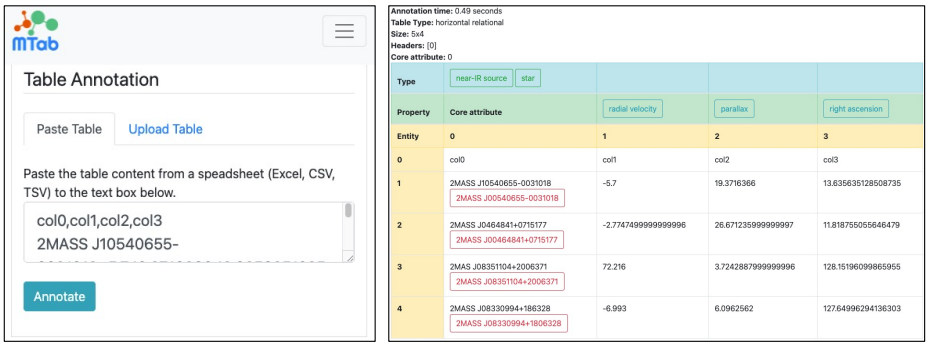

Fig. 1: Example tabular data annotation with MTab

- **Pre-processing** The input tables are pre-processed with encoding prediction, table type prediction, data type prediction for cells and columns.
- **Structural Annotations:** Then, we perform header detection based on majority voting of column data type as [6]. The core attribute detection is based on the uniqueness of cell values in a column as [6][9].
- **Semantic Annotations:** 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 strings. The CTA matching targets are columns so that the column data types are strings. The CPA matching targets are the relation between the core attribute and the remaining table columns. Then, we perform entity candidate generation for each table cell with entity search and two cells in the same row with statement search. We calculate context similarities with the value matching between statements of entity candidates in the core attributes with table row values. Finally, generate the annotations for entities, properties, and types based on majority voting of context similarities [7].

Fig. 1 illustrate an annotation example for 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 core attribute 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 core attribute and other columns are in blue and located in the property column.

3 Conclusions

This paper presents a demonstration of the MTab toolkit for table annotation with knowledge graphs of Wikidata, DBpedia, and Wikipedia. MTab is effective, efficient, and easy to use.

In the future work, we will focus on 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).

References

1. Cutrona, V., Bianchi, F., Jiménez-Ruiz, E., Palmonari, M.: Tough tables: Carefully evaluating entity linking for tabular data. In: The Semantic Web - ISWC 2020. Lecture Notes in Computer Science, vol. 12507, pp. 328–343. Springer (2020), https://doi.org/10.1007/978-3-030-62466-8_21
2. Garbe, W.: Symspell: Symmetric delete algorithm. https://github.com/wolfgarbe/SymSpell (2012)
3. Jiménez-Ruiz, E., Hassanzadeh, O., Efthymiou, V., Chen, J., Srinivas, K.: Semtab 2019: Resources to benchmark tabular data to knowledge graph matching systems. In: The Semantic Web - 17th International Conference, ESWC 2020. Lecture Notes in Computer Science, vol. 12123, pp. 514–530. Springer (2020), https://doi.org/10.1007/978-3-030-49461-2_30
4. Jimenez-Ruiz, E., Hassanzadeh, O., Efthymiou, V., Chen, J., Srinivas, K., Cutrona, V.: Results of semtab 2020. In: SemTab@ISWC. CEUR Workshop Proceedings, vol. 2775, pp. 1–8. CEUR-WS.org (2020), http://ceur-ws.org/Vol-2775/paper0.pdf
5. Nguyen, P., Kertkeidkachorn, N., Ichise, R., Takeda, H.: Mtab: Matching tabular data to knowledge graph using probability models. In: SemTab@ISWC 2019. CEUR Workshop Proceedings, vol. 2553, pp. 7–14. CEUR-WS.org (2019), http://ceur-ws.org/Vol-2553/paper2.pdf
6. Nguyen, P., Kertkeidkachorn, N., Ichise, R., Takeda, H.: Tabeano: Table to knowledge graph entity annotation. CoRR **abs/2010.01829** (2020), https://arxiv.org/abs/2010.01829
7. 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), http://ceur-ws.org/Vol-2775/paper9.pdf
8. Nguyen, P., Yamada, I., Takeda, H.: Mtabes: Entity search with keyword search, fuzzy search, and entity popularities. In: The 35th Annual Conference of the Japanese Society for Artificial Intelligence, JSAI 2021. vol. 2021. The Japanese Society for Artificial Intelligence, https://www.jstage.jst.go.jp/article/pjsai/JSAI2021/0/JSAI2021_1N4IS1a02/_pdf
9. 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), https://doi.org/10.1145/2797115.2797118
10. Wang, D., Shiralkar, P., Lockard, C., Huang, B., Dong, X.L., Jiang, M.: TCN: table convolutional network for web table interpretation. In: WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021. pp. 4020–4032. ACM / IW3C2 (2021), https://doi.org/10.1145/3442381.3450090