From Heuristics to Language Models: A Journey Through the Universe of Semantic Table Interpretation with DAGOBAH

Viet-Phi Huynh[1,*], Yoan Chabot[1], Thomas Labbé[1], Jixiong Liu[1,2] and Raphaël Troncy[2]

[1] *Orange, France*

[2] *EURECOM, Sophia Antipolis, France*

Abstract

This paper presents DAGOBAH SL 2022, a semantic table interpretation system that has been continuously improved over the last four years when participating in the SemTab challenge. This year, we have improved the lookup coverage using external resources and we have integrated language models for better understanding the table headers. We have also implemented various system optimizations that lead to a reduction in execution time of about 30%. In this paper, we also show the relevance of using deep learning-based approaches for resolving certain ambiguities and we discuss the limitations of existing corpora and systems for maturing further this research field.

Keywords

Semantic Table Interpretation, Tabular data, DAGOBAH, SemTab

1. Introduction

The problem of semantic interpretation of tabular data is a growing topic in the scientific community spanning multiple research communities [1]. This is also a primary concern in industry since there is a growing desire to extract dormant knowledge from the internal repositories to feed enterprise knowledge graphs [2].

DAGOBAH is a mature system for performing semantic tabular interpretation that has participated in the yearly SemTab challenge series since 2019. DAGOBAH SL [3] constitutes the core of the solution: it includes a large number of heuristics enabling to pre-process tables (detecting orientation, headers, and primitive types of columns) and to produce fine-grained annotations for cells (CEA), columns (CTA) and relationships between columns (CPA) given different reference knowledge graphs. The system is available via an API for developers[1] as well as via a user-friendly web interface that offers functionalities for visualizing annotations, enriching tabular data from the knowledge graph (e.g. adding columns and filling in missing values) or enriching the knowledge graph from the tables [4]. Finally, DAGOBAH provides a

SemTab: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching, 2022

[*] Corresponding author.

vietphi.huynh@orange.com (V. Huynh)

0000-0001-7348-9650 (V. Huynh); 0000-0001-5639-1504 (Y. Chabot); 0000-0001-9295-7675 (T. Labbé); 0000-0002-8750-8637 (J. Liu); 0000-0003-0457-1436 (R. Troncy)

[1] https://developer.orange.com/apis/table-annotation

generic plug-in named Radar Station[2] for STI systems that expose multiple candidates when there are ambiguities and that use specific knowledge graph embeddings as data augmentation for resolving these ambiguities [5].

In this paper, we present the specific improvements of DAGOBAH-SL when tackling the SemTab 2022 challenge as well as motivated by industrial application cases (Section 2). We present the results of DAGOBAH-SL in Section 3. We discuss some limitations of current benchmarks and systems in Section 4. We present some preliminary results and research directions for hybridizing even more the usage of generative language models, knowledge graph embeddings and our current system (Section 5) before concluding and outlining some future work (Section 6).

2. DAGOBAH SL 2022

DAGOBAH SL is a two-stages annotation system consisting of an entity lookup step, followed by an entity scoring step (Figure 1). Given the availability of an alias table $\mathcal{A}$ which contains entities together with their associated labels or aliases (e.g. the alias table $\mathcal{A}_{wikidata}$, given the Wikidata knowledge graph, has the entry {Q317521: Elon Musk, E. Musk, CEO of Tesla, etc.}), the entity lookup involves retrieving, for a contextless mention m, a set of candidate entities from $\mathcal{A}$ in which each candidate has at least an alias similar to m lexically. Acting as the first step in the annotation pipeline, the entity lookup can significantly impact the overall quality of the system for two reasons:

- **Richness of the alias table:** an alias table with low mention-coverage per entity has limited lookup capacity. For instance, the Wikidata entity Q5544925[3] appears in $\mathcal{A}_{wikidata}$ under the sole English name: George Stroumboulopoulos Tonight. As a consequence, Q5544925 will never be returned as a relevant candidate of the mention The Hour while this mention is arguably a correct alias of Q5544925 considering Wikipedia[4].
- **Number of entity candidates (K):** ideally, the entity lookup is expected to hit the correct entity within a small ranked list of candidates. Low K helps to reduce the computation cost of the later stage in the annotation pipeline (i.e., entity scoring) as well as alleviate the influence of the noise and the ambiguity brought by other candidates.

In Section 2.1, we show how we have improved our entity lookup service by addressing the two drawbacks mentioned above. Compared with DAGOBAH SL 2021 [3], we present two other major contributions: (i) The entity scoring algorithm exploits more effectively the prior scores of candidate entities resulting from the lookup step (Section 2.2); (ii) Apart from using CTA and CPA for the CEA disambiguation, we introduce in Section 2.3 a novel disambiguation method based on column headers and entity description that leverage language models. Finally, when dealing with large tables (e.g. the ToughTables corpus has some tables that have more than

[2] https://github.com/Orange-OpenSource/radar-station

[3] https://www.wikidata.org/wiki/Q5544925

[4] https://en.wikipedia.org/wiki/George_Stroumboulopoulos_Tonight: George Stroumboulopoulos Tonight (originally known as The Hour) is a Canadian television talk show hosted by George Stroumboulopoulos that aired on CBC Television from 2005 to 2014.

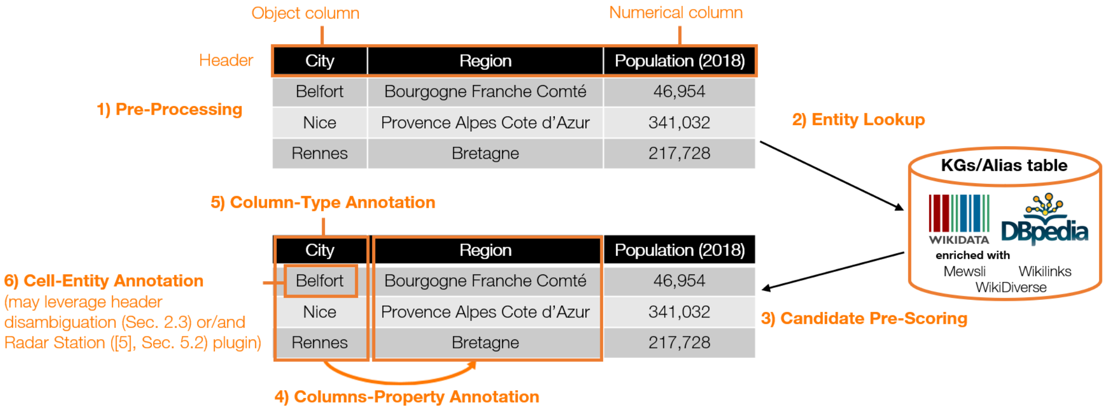


Figure 1: Pipeline of DAGOBAH SL 2022

8,000 rows and 4 columns), we propose a memory-efficient multiprocessing framework in order to accelerate the annotations (Section 2.4). The pipeline of DAGOBAH SL 2022 is presented in Figure 1 including typical steps in a STI system (Table Pre-processing, Entity Lookup, CEA, CTA, CPA) as well as novel improvements listed above.

2.1. Entity Lookup Improvement

The DAGOBAH entity lookup service provides a fuzzy search engine given a KG alias table. It is backed by Elasticsearch and it currently includes various snapshots of the Wikidata and DBpedia knowledge graphs. More details on the construction of Wikidata and DBpedia Alias Table can be found at [3].

Entity Alias Augmentation. A simple way to enrich the alias table of a KG is to supplement it with aliases found in relevant external sources. In principle, table cells annotation is close to the Entity Linking task in the NLP domain where a mention in text is linked to an entity in a KG. This naturally leads to the exploitation and the injection of Entity Linking's available labeled datasets into KG alias table. In our work, we have discovered four sources which can provide a wide diversity of entity aliases for the Wikidata KG: Wikipedia Alias[5], Mewsli[6], Wikilinks[7] and WikiDiverse[8]. We report in Table 1 some statistics illustrating the contribution of each alias source to the augmentation of Wikidata alias.

Entity Candidate Ranking. An efficient ranking mechanism $p(\hat{e}|m)$ (where m is a mention, $\hat{e}$ is a candidate of m) can improve the lookup coverage where m has more chance to match with the correct entity. It also provides more informative prior scores to the entity scoring

[5] https://dumps.wikimedia.org/enwiki/20220701/
[6] https://github.com/google-research/google-research/tree/master/dense_representations_for_entity_retrieval/mel
[7] http://www.iesl.cs.umass.edu/data/data-wiki-links
[8] https://github.com/wangxw5/wikidiverse

Table 1
Statistics for the external alias sources used for Wikidata alias augmentation.

Alias source	Wikipedia	Mewsli	Wikilinks	WikiDiverse
Total number of Wikidata entity	6,603,252	84,413	1,632,661	7,704
Number of Wikidata entity whose alias set is enriched	5,537,830	26,989	992,888	2,831
Average number of novel aliases enriched per entity	2.5	1.5	3.1	1.2

phase. Similarly to [6], we incorporate three ranking factors into the final ranking score, as following:

$$p(\hat{e}|m) = w_1 * fuzzy\_score(\hat{e}, m) + w_2 * BM25\_score(\hat{e}) + w_3 * pageRank\_score(\hat{e}) \quad (1)$$

where $fuzzy\_score(\hat{e}, m)$ is a similarity score between m and labels and aliases of $\hat{e}$ based on Levenshtein distances [3], $BM25\_score(\hat{e})$ is the normalized BM25 score calculated by the term frequency (TF) and the inverse document frequency (IDF) of word tokens in the label and aliases of $\hat{e}$. The PageRank-like popularity score of $\hat{e}$, $pageRank\_score(\hat{e})$ is calculated on Wikipedia using danker[9]. The contribution of each factor into the final ranking score is empirically defined by the associated coefficients $\{w_1, w_2, w_3\}$ = {0.7, 0.2, 0.1}. We emphasize that w_1 should be considerably higher than two others to steer the lookup towards entity labels similar to the mention.

In Figure 2, we evaluate the hit rate at Top-K returned candidates of the Entity Lookup on three datasets: (2a) 1,000 randomly sampled mentions from the Limaye dataset [7], (2b) 3,900 randomly sampled mentions from the T2Dv2 dataset [8] and (2c) 4,000 randomly sampled mentions from the ToughTables 2021 dataset [9]. The lookup with alias enrichment and {BM25, PageRank} scores in the ranking function (namely **DAGOBAH Entity Lookup 2022**) clearly outperforms the one without alias enrichment and using only fuzzy search as ranking signal (namely **DAGOBAH Entity Lookup 2021**) by higher hit rates and approaching the upper bound on capacity faster. Within only K = 10 candidates, its performance is already competitive with other larger K.

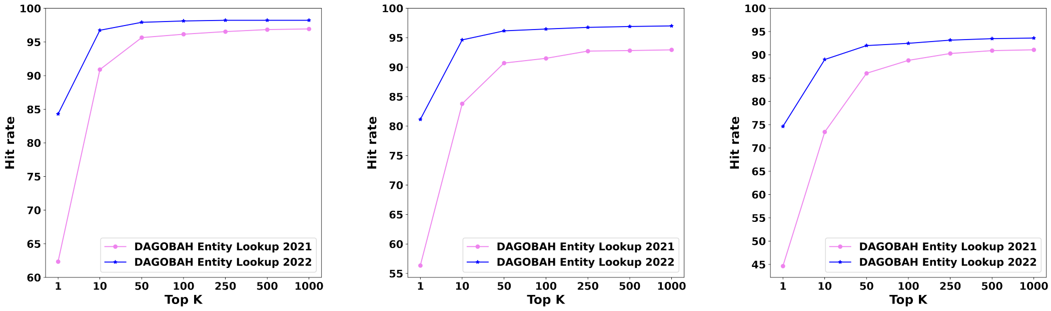
2.2. Entity Scoring Improvement

We employ DAGOBAH SL 2021 [3] as the algorithmic backbone of our 2022 annotation system. The score of an entity candidate $\hat{e}$ is given by:

$$p(\hat{e}) = p(\hat{e}|\text{table context}) \times f(p(\hat{e}|m)) \quad (2)$$

where $p(\hat{e}|\text{table context})$ is the context score of $\hat{e}$ given the table row that it lies in. More details on how the score is computed is presented in [3] (Section 2.3). $p(\hat{e}|m)$ is resulted from the entity lookup, playing as prior knowledge of $\hat{e}$ given solely mention m. Finally, f is an activation function applied on $p(\hat{e}|m)$.

[9] https://github.com/athalhammer/danker



(a) Limaye

(b) T2Dv2

(c) ToughTables 2021

Figure 2: Hit Rate@K of Entity Lookup on Limaye, T2Dv2 and ToughTables 2021 datasets.

An improvement in DAGOBAH SL 2022 comes from the choice of the activation function f. In the light of significant enhancements made in entity lookup (Section 2.1), we would like to highlight more the contribution of $p(\hat{e}|m)$ to the entity score. Intuitively, we posit that entity candidates of $p(\hat{e}|m)$ higher than 0.9 should concentrate into one cluster, while ones of $p(\hat{e}|m)$ lower than 0.7 should go into another cluster and these two clusters should be discriminative by pulling them apart. For achieving this goal, rather than selecting f as an exponential function $e^{\alpha(p(\hat{e}|m)^{\beta}-1)}$, as in DAGOBAH SL 2021 [3], we rely on a sigmoid function $\frac{1}{1+e^{-\alpha(p(\hat{e}|m)^{\beta}-1)}}$ in DAGOBAH SL 2022. The rationale is that the margin between the upper cluster ($p(\hat{e}|m)$ > 0.9) and the lower cluster ($p(\hat{e}|m)$ < 0.7) is larger when using a sigmoid function f than an exponential one. Figure 3 illustrates the different behavior of these two functions. We evaluate the role of f on two validation datasets: HardTables and ToughTables from the Round 2. Table 2 shows significant gain achieved by the sigmoid function compared to the exponential one.

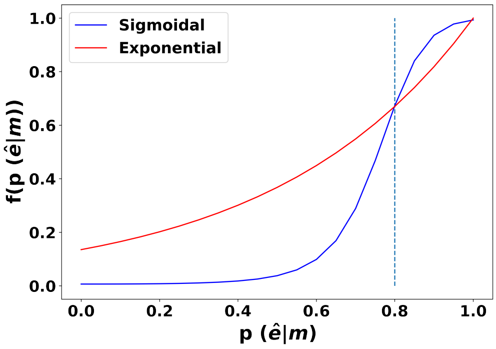


Figure 3: Behavior of different activation functions f

Table 2
F1 score on Round 2's Validation Datasets using different activation functions f.

Function f	Valid HardTables R2	Valid ToughTables R2
Exponential	0.888	0.941
Sigmoid	0.907	0.959

2.3. Entity Disambiguation by Reading Entity Descriptions

Table headers, if appropriately given, are useful sources of information for the disambiguation of cell entity annotation (CEA). Effectively, a relevant header often represents attributes related to the type of a cell entity. We develop **DAGOBAH SL + Header Disambiguation**, an hybrid model incorporating DAGOBAH SL and a BERT [10]-based cross encoder for CEA disambiguation by evaluating the semantic correlation between the cell entity and the headers. This means investigating the strength of the connection between the headers (when provided) and the entity descriptions (given in a particular knowledge graph).

Model. This problem can be reformulated as a binary text classification task [11]:

$$f : [\textit{headers } H, \textit{entity description } d_e] \rightarrow \{0,1\} \; i.e., \; f(H,e) = \mathrm{P}(\text{Matched}|H, d_e) \qquad (3)$$

We experiment with a pre-trained ELECTRA-based [12] Cross Encoder[10] for modeling f. It takes as input the concatenation of the headers h_{ls}, h, h_{rs} and the entity description d_e: ([CLS] h_{ls} [H$_l$] h [H$_r$] h_{rs} [SEP] d_e) where h is the header of the column associated with the entity e, h_{ls} and h_{rs} are respectively left-side and right-side headers of h. Two special tokens [H$_l$], [H$_r$] are added to signal the position of the target header h. The embedding at the last hidden layer of [CLS] token is fed into a softmax layer to yield an output value between 0 and 1 indicating the likelihood $f(H,e)$ of e w.r.t. h_{ls}, h, h_{rs}.

Dataset. For fine-tuning the Cross Encoder, we construct a dataset consisting of $\sim$ 700K positive {headers h, entity description d_e} pairs from the Wikipedia Table [14] and the ToughTables 2021[11] [9] datasets. For each positive sample {h, d_e}, we generate 4 negative samples {h, $d_{e'}$} where entity e' is not semantically relevant for h. Instead of a random sampling which may not guarantee that a negative sample is actually not related to h, we propose two negative sampling strategies:

- A sentence transformer[12] (bi-encoder) is leveraged to score the cosine similarity between the descriptions of e and e'. We consider e' as a negative sample if cosine$(d_e, d_{e'})$ is smaller than 0.

[10] We rely on https://github.com/UKPLab/sentence-transformers for the implementation of the Cross Encoder [13]

[11] Tables from the ToughTables 2021 corpus do actually not contain meaningful headers, as they use Col0, Col1.... We take advantage of the column type annotation (CTA) as possible headers. Furthermore, since ToughTables is used for fine-tuning, we do not use our model on its 2022 version in the Round 2 of SemTab 2022.

[12] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

- In a table, entities coming from other columns (with different headers) can be used as hard negative samples for the target column that can help the model to learn better discriminative capacity. For example, assuming headers: [CLS] Human Settlement [H$_l$] Sovereign State [H$_r$] [SEP], entity **Wilkesboro** ("Wilkesboro is a town in and the county seat of Wilkes County, North Carolina, **United States**. The population was 3,687 at the 2020 census.") from column Human Settlement is seen as a negative sample for column Sovereign State. A poor model may rely on the information **United States** in the description of **Wilkesboro** or may favor the left header **Human Settlement** than target header **Sovereign State** to conclude wrongly that **Wilkesboro** belongs to column Sovereign State.

DAGOBAH SL + Header Disambiguation. Without assuming the existence of other information than the table itself, we do not know if the column headers can provide non-trivial evidence for the annotation of table components. Cases in which the columns are artificially labelled (e.g. Col0, Col1) or meaningless (e.g. Name) are frequent. Therefore, in order to avoid the spurious contribution of a trivial header, we follow a simple method to quantify its relevance, and thus determine whether it should be taken into account for the disambiguation of CEA. Considering a column c_i with associated header H_i in a table that has N rows, the compatibility of H_i w.r.t. c_i is defined as:

$$relevance(H_i, c_i) = \frac{\sum_{n=2}^{N} \mathbf{1}_{s_n > \alpha}(s_n)}{N-1} \quad s.t. \;\; s_n = \text{Max}_{k=1..K}(f(H_i, e_{nk})) \tag{4}$$

where s_n is a partial compatibility score between H_i and the table mention m located at n_{th} row and column c_i. It is calculated as the maximal likelihood of any entity e_{nk} among K entity candidates of m in view of H_i. H_i is said to be compatible to m if s_n is higher than α. By averaging the Heaviside step function[13], $\mathbf{1}_{s_n > \alpha}$ of s_n over rows, $relevance(H_i, c_i)$ indicates how well the header H_i can represent the column c_i. Given this, from Eq. 2, the score of an entity candidate $\hat{e}$ in column c_i is updated accordingly, as following:

$$p(\hat{e}) = \frac{p(\hat{e}) + \mathbf{1}_{relevance(H_i,c_i) > \beta} \times relevance(H_i, c_i) \times \mathbf{1}_{f(H_i,\hat{e}) > \alpha} \times f(H_i, \hat{e})}{1 + \mathbf{1}_{relevance(H_i,c_i) > \beta} \times relevance(H_i, c_i)} \tag{5}$$

where β defines a threshold at which the information flow from the header to entity is activated.

2.4. Performance Optimisation

The annotation algorithm in DAGOBAH SL is decomposed into several consecutive stages: context scoring → [CEA → CTA → CPA]$_3$ (where [...]$_3$ means that the pipeline is repeated 3 times for iterative disambiguation). Each stage encompasses row-independent calculations. This system design enables us to leverage multiprocessing-based parallelism in order to speed up the annotation. Specifically, at each annotation stage, a batch of table rows is sent to a worker and is executed independently of other batches. It is important to notice that the processing deals with numerous large objects (entity graph, score components of all entity candidates, cached

[13] https://en.wikipedia.org/wiki/Heaviside_step_function

relations between possible pairs of candidates). Hence, the use of parallel processing should not invoke an explosion in memory usage. To this end, we make use of Ray[14], a distributed execution framework with a very effective memory management that has recently attracted a lot of attention from the machine and deep learning community [15]. We compare sequential annotation and parallel annotation (with 6 workers, each worker has 1 CPU) in terms of average execution time (Table 3). The comparison is performed on the Round 2's validation ToughTables. In order to see the advantage of parallel annotation for large tables[15], we focus on 11 tables that have more than 200 rows, among 36 tough tables. With parallel setting, we achieved remarkable gains from 40% (for K=50) to 58% (for K=150).

Table 3

Comparison of the execution time (s) w.r.t. number of entity candidate per mention (K) of Sequential Annotation and Parallel Annotation on 11 large tables (i.e. #rows > 200) of Round 2's Validation ToughTables.

K	Avg. Sequential Annotation Time (s)	Avg. Parallel Annotation Time (s)	Speed up (times)
50	131	78	1.68x
100	328	141	2.33x
150	497	209	2.37x

3. Experiments and Evaluation

We report the performance of DAGOBAH SL 2022 for the SemTab 2022 challenge in Table 4. We achieve very high scores (in terms of precision and F1) for the HardTables corpus (CTA, CEA, CPA tasks) and for the ToughTables corpora (CEA task). We observe a much lower performance for the CTA score on ToughTables despite the excellent CEA score. We argue that the way the CTA gold standard has been generated is controversial and challenging since a column can often be tagged with a variety of correct types. Furthermore, we have also questioned the quality of the ground truth in some occasions. For example, we believe that the first column in table 8QA9EYPI of ToughTables should be annotated as Q5 (Human) or Q82955 (politician) but not with Q11028 (information) as currently declared in the ground truth. We believe that a proper adjudication phase would be necessary to further enhance the quality of the gold standard.

For the annotation of the BiodivTab corpus, given that the tables contain meaningful headers, we exploit this information via an hybrid model DAGOBAH SL + Header Disambiguation as introduced in Section 2.3. It is worth noting that the Header Disambiguation module will give no gain for HardTables and ToughTables since they contain artificial headers such as Col0 that will not match with any entity description (i.e. $f(\text{Col0}, e) <<, \forall e$). We propose a rough comparison with our last year's annotation system (DAGOBAH SL 2021, in gray color) conducted with Wikidata KG, and we observe that DAGOBAH SL 2022 + Header Disambiguation obtains considerable improvement on both CTA and CEA tasks.

[14] https://github.com/ray-project/ray

[15] Regarding the annotation of small tables, we favor sequential setting over parallel setting as the serialization/deserialization of in-function objects/output objects occupies a significant proportion of total execution time

In dealing with the GitTables dataset, we perform two operations to figure out a column representative: (i) we perform entity lookup (Section 2.1) with the Wikidata KG on all column cells. The type (CTA) that appears most frequently among entity candidates is retained. (ii) A pre-processing step is also applied to each column to find a primitive type such as ORG (organization), LOC (location), MONEY (currency), etc. which is typical for the Named Entity Recognition task in NLP. The types resulting from (i) and (ii) are finally manually mapped to Schema.org and DBpedia classes and properties to provide annotations using the target Schema.org vocabulary.

Table 4
Performance of the DAGOBAH system in Rounds 1, 2, and 3 of the SemTab 2022 challenge. "F1" stands for F1-score, "P" stands for Precision.

Dataset	System	CTA		CEA		CPA	
		F1	P	F1	P	F1	P
Round 1 – HardTables WD	DAGOBAH SL	0.975	0.975	0.954	0.955	0.984	0.99
Round 2 – HardTables WD	DAGOBAH SL	0.96	0.96	0.904	0.905	0.931	0.97
Round 2 – ToughTables WD	DAGOBAH SL	0.409	0.409	0.945	0.946	-	-
Round 2 – ToughTables DBP	DAGOBAH SL	0.312	0.312	0.926	0.926	-	-
Round 3 – BiodivTab DBP	DAGOBAH SL + Header Disambiguation ($\alpha = 0.8, \beta = 0.0$)	0.616	0.616	0.736	0.736	-	-
	DAGOBAH 2021	0.344	0.345	0.62	0.62	-	-
Round 3 – GitTables SCH class	Preprocessing + Mapping	0.312	0.342	-	-	-	-
Round 3 – GitTables SCH property	Preprocessing + Mapping	0.087	0.095	-	-	-	-
Round 3 – GitTables DBP property	Preprocessing + Mapping	0.075	0.082	-	-	-	-

4. Limits of Existing Corpora and Systems

We noticed several issues and limitations regarding the dataset proposed in the SemTab challenge. First of all, annotations provided in the ground truth are not always correct, or are subject to debate. For instance, disambiguation pages are sometimes proposed as annotation for the CEA task whereas a correct entity exists in the exact snapshot of the knowledge graph that should be used. Incorrect (e.g. Q142 (France) while Q159 (Russia) should be the correct entity in table PRDTMM8A) or arguably not the best (e.g. Q11028 (information) instead of Q35657 (U.S. state) in table 1C9LFOKN) annotations also exist from time to time. In addition, the same type of mentions is not consistently annotated: e.g. postal addresses in GitTable are sometimes typed as `schema:email` and sometimes as `schema:address`.

We argue that some other issues are related to the very nature of the data. Hence, tables that are artificially and synthetically generated may not reflect what is actually found in the wild. Tables often serve a specific purpose for the creator, and the attributes are selected accordingly. For example, one might want to use a table for presenting all books within the topic Star Wars, but not all entities from the type literary work (Q7725634). At the same time, the creator of this table might also want to focus on the publication dates without other attributes of books (e.g. the authors) in the table to emphasize that the Star Wars series are continuously updating[16]. Tables can be grouped into collection with a common theme but at the moment, STI system annotate tables very independently as if they were no notion of collection. This context may typically not be made explicit in a corpora but could be detected using topic modeling algorithms adapted to tables. Last, structure of real tables are in general much more complex than the one proposed in SemTab2022. We propose to consider more variety of tables types (e.g. entity tables) and to increase the complexity of the table structure (e.g. merging cells) [1]. Integrating these new challenges will allow to cover a wider range of real world scenario, hence will benefit to the community.

Finally, we think that the current challenge workflow composed of rounds (targeting different knowledge graphs) encourages team participants to over-tune their systems on specific rounds to the detriment of genericity. To overcome this issue, we recommend to evaluate the final system on all rounds in order to highlight the most generic solution.

5. Hybridization with Language Models and Knowledge Graph Embeddings

Heuristics methods have proven their capabilities to handle with high accuracy the majority of datasets provided by the SemTab challenges over the past four years. However, more challenging datasets introduced gradually highlighted the limits of these methods with a clear performance drop. To cover these limitations without sacrificing the genericity of the solution, we believe deep learning based approaches, aiming at modeling different kinds of objects through embeddings, shall be investigated.

5.1. Text Modeling

Inspired by emerging zero-shot entity linking approaches (e.g. ZESHEL [16], BLINK [17], ReFinED [18]) for textual data in which the proposed methods only rely on the description[17] of the entity to link it to a mention detected in the text, we are convinced that entity disambiguation can be solved by reading its textual description with a powerful natural language understanding model. We believe this could also be a right direction for the STI field. Our initial promising results on exploring the semantic correlation between an entity and a column header via reading entity description (Section 2.3) pave the way for future dedicated works. We plan to evaluate the feasibility of building a reading-comprehension model on entity description and table context (e.g. the table row that contains the entity), similarly to the core principle of zero-shot entity linkers mentioned above, except that the input textual data will be replaced by tabular data. Last but not least, the fact that the model leverages only entity description for zero-shot entity linking makes it appealing for long-tail entities, long-tail domains or early-stage knowledge graphs in which entities are often reduced to a short text as opposed to a rich set of attribute values.

[16] This example is actually modeled in the file 'file405599 0 cols1 rows23.csv' from the Limaye dataset [7]

[17] ReFinED also makes use of entity type in addition to entity description

5.2. Knowledge Graph Modeling

We can take more advantage of the target KGs and the richness of the entity descriptions to improve the disambiguation of cell mentions. Currently, DAGOBAH-SL's performance relies on overlapping the table components with the labels, relations, types, and descriptions of a given entity in a knowledge graph. However, this entity-wised focus does not take into account other possible relatedness between entities that are typically captured in knowledge graph embeddings [19]. Entities from the same table are generally related to each other, especially entities from the same column. [20] build a weighted correlation subgraph in which each node represents a CEA candidate. The edges are weighted by the cosine similarity between two related nodes. The best candidates are the ones whose accumulated weights over all incoming and outcoming edges are the highest. Our first approach, DAGOBAH-Embeddings [21] aims to apply clustering over the candidates' embeddings for the disambiguation by choosing the right cluster. However, we have achieved negative results on the CEA tasks since some correct entity candidates are not in our chosen cluster. Recently, we propose Radar Station [5], a plugin for a STI system that takes as input the multiple candidates with their scores for a cell mention, and leverages the distance between candidates in the embedding space to increase the coherency of the annotations.

5.3. Table Modeling

We can finally apply language modeling approaches to learn enriched table representations. We propose to consider tables as an alternative language structure, with latent relationships between mentions, not necessarily following a formal grammar. In the past few years, several works have been released trying to tackle this kind of latent representations [22, 23, 24, 25], putting tabular data in the foreground of deep learning approaches. With the success of BERT-like language models, most recent papers focused on modifying the way a neural network can learn such representation through the implementation of dedicated Transformer's attention mechanisms [26, 27, 28]. The general idea is to learn deep contextualized representations of tabular data in an unsupervised or self-supervised way, and then apply transfer learning with fine-tuning on target downstream tasks. However, except for TURL [28], most of these works cover tasks such as question answering or table-as-a-whole understanding, and only partly address the tabular data semantic annotations such as defined in SemTab, namely CEA, CTA and CPA. We believe that SemTab data from 2019 are interesting table corpora to train or fine-tune tabular language models (TLM), even if the associated GTs do not cover all table elements. In this spirit, the DAGOBAH team has already leveraged these data to generate consistent contextual embeddings associated to mentions pairs and corresponding target triples which is the first step towards an end-to-end vectorial annotation processing through TLM. Moreover, the fact that more specific knowledge can be injected into language model through the verbalization of KG [29] reinforces our conviction that the future of tabular data annotation will be in that direction. Nonetheless, a generic TLM might not be suitable for all target domains, and it might be more realistic to think about several adapted models to handle per-vertical use cases.

6. Conclusion and Future Work

In this paper, we have presented the DAGOBAH 2022 system for semantic table interpretation. We have emphasized several key improvements: (i) an entity lookup with a richer alias table and more powerful intrinsic ranking function facilitates the entity retrieval for more variants of input mentions; (ii) a high-performance entity scoring algorithm characterizes more thoroughly the behaviors of entity candidates; (iii) a first effort, yet promising result on the application of language model to better understand table components (e.g. headers vs. cell entity description). While we believe this approach is a step in the right direction, our future work will continue to dive more into the emerging research area for table understanding based on language models, as discussed in Section 5.

References

[1] J. Liu, Y. Chabot, R. Troncy, V.-P. Huynh, T. Labbé, P. Monnin, From Tabular Data to Knowledge Graphs: A Survey of Semantic Table Interpretation Tasks and Methods, Journal of Web Semantics (2022). Under revision.

[2] Y. Chabot, P. Monnin, F. Deuzé, V. Huynh, T. Labbé, J. Liu, R. Troncy, A Framework for Automatically Interpreting Tabular Data at Orange, in: 20^{th} International Semantic Web Conference (ISWC), Posters, Demos and Industry Tracks, volume 2980 of *CEUR Workshop Proceedings*, 2021.

[3] V.-P. Huynh, J. Liu, Y. Chabot, F. Deuzé, T. Labbé, P. Monnin, R. Troncy, DAGOBAH: Table and Graph Contexts for Efficient Semantic Annotation of Tabular Data, in: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab), 2021.

[4] C. Sarthou-Camy, G. Jourdain, Y. Chabot, P. Monnin, Deuzé, V.-P. Huynh, J. Liu, T. Labbé, R. Troncy, DAGOBAH UI: A New Hope For Semantic Table Interpretation, in: 19^{th} European Semantic Web Conference (ESWC), Poster and Demo Track, Springer, 2022.

[5] J. Liu, V.-P. Huynh, Y. Chabot, R. Troncy, Radar Station: Using KG Embeddings for Semantic Table Interpretation and Entity Disambiguation, in: 21^{st} International Semantic Web Conference (ISWC), 2022.

[6] P. Nguyen, I. Yamada, H. Takeda, MTabES: Entity Search with Keyword Search, Fuzzy Search, and Entity Popularities, in: 35^{th} Annual Conference of the Japanese Society for Artificial Intelligence (JSAI), 2021.

[7] G. Limaye, S. Sarawagi, S. Chakrabarti, Annotating and searching web tables using entities, types and relationships, VLDB Endowment 3 (2010) 1338–1347.

[8] O. Lehmberg, D. Ritze, R. Meusel, C. Bizer, A large public corpus of web tables containing time and context metadata, in: 25^{th} International Conference on World Wide Web (WWW), Companion Volume, 2016, pp. 75–76.

[9] V. Cutrona, F. Bianchi, E. Jiménez-Ruiz, M. Palmonari, Tough tables: Carefully evaluating entity linking for tabular data, in: 19^{th} International Semantic Web Conference, Springer, 2020, pp. 328–343.

[10] J. D. M.-W. C. Kenton, L. K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: NAACL-HLT, 2019, pp. 4171–4186.

[11] K. Halder, A. Akbik, J. Krapac, R. Vollgraf, Task-aware representation of sentences for generic text classification, in: 28^{th} International Conference on Computational Linguistics, 2020, pp. 3202–3213.

[12] K. Clark, M.-T. Luong, Q. V. Le, C. D. Manning, Electra: Pre-training text encoders as discriminators rather than generators, arXiv:2003.10555, 2020.

[13] N. Reimers, I. Gurevych, Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks, in: Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, 2019.

[14] C. S. Bhagavatula, T. Noraset, D. Downey, Tabel: Entity linking in web tables, in: 14^{th} International Semantic Web Conference, Springer, 2015, pp. 425–441.

[15] P. Moritz, R. Nishihara, S. Wang, A. Tumanov, R. Liaw, E. Liang, M. Elibol, Z. Yang, W. Paul, M. I. Jordan, et al., Ray: A distributed framework for emerging $\{$AI$\}$ applications, in: 13^{th} USENIX Symposium on Operating Systems Design and Implementation (OSDI), 2018, pp. 561–577.

[16] L. Logeswaran, M.-W. Chang, K. Lee, K. Toutanova, J. Devlin, H. Lee, Zero-shot entity linking by reading entity descriptions, in: 57^{th} Annual Meeting of the Association for Computational Linguistics, 2019, pp. 3449–3460.

[17] L. Wu, F. Petroni, M. Josifoski, S. Riedel, L. Zettlemoyer, Scalable zero-shot entity linking with dense entity retrieval, in: Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 6397–6407.

[18] T. Ayoola, S. Tyagi, J. Fisher, C. Christodoulopoulos, A. Pierleoni, ReFinED: An Efficient Zero-shot-capable Approach to End-to-End Entity Linking, in: NAACL, 2022.

[19] R. Biswas, R. Türker, F. B. Moghaddam, M. Koutraki, H. Sack, Wikipedia Infobox Type Prediction Using Embeddings, in: International Workshop on Deep Learning for Knowledge Graphs and Semantic Technologies (DL4KGS), 2018, pp. 46–55.

[20] V. Efthymiou, O. Hassanzadeh, M. Rodriguez-Muro, V. Christophides, Matching web tables with knowledge base entities: from entity lookups to entity embeddings, in: 16^{th} International Semantic Web Conference (ISWC), Springer, 2017, pp. 260–277.

[21] Y. Chabot, T. Labbe, J. Liu, R. Troncy, DAGOBAH: an end-to-end context-free tabular data semantic annotation system, in: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching, 2019, pp. 41–48.

[22] P. Yin, G. Neubig, W.-t. Yih, S. Riedel, Tabert: Pretraining for joint understanding of textual and tabular data, in: The 58^{th} Annual Meeting of the Association for Computational Linguistics, 2020, pp. 8413–8426.

[23] J. Herzig, P. K. Nowak, T. Mueller, F. Piccinno, J. Eisenschlos, Tapas: Weakly supervised table parsing via pre-training, in: The 58^{th} Annual Meeting of the Association for Computational Linguistics, 2020, pp. 4320–4333.

[24] S. Ö. Arik, T. Pfister, Tabnet: Attentive interpretable tabular learning, in: AAAI Conference on Artificial Intelligence, volume 35, 2021, pp. 6679–6687.

[25] H. Iida, D. Thai, V. Manjunatha, M. Iyyer, Tabbie: Pretrained representations of tabular data, in: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, pp. 3446–3456.

[26] J. Eisenschlos, M. Gor, T. Mueller, W. Cohen, Mate: Multi-view attention for table transformer efficiency, in: Conference on Empirical Methods in Natural Language Processing,

2021, pp. 7606–7619.

[27] Z. Wang, H. Dong, R. Jia, J. Li, Z. Fu, S. Han, D. Zhang, TUTA: tree-based transformers for generally structured table pre-training, in: 27^{th} ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 1780–1790.

[28] X. Deng, H. Sun, A. Lees, Y. Wu, C. Yu, Turl: Table understanding through representation learning, ACM SIGMOD Record 51 (2022) 33–40.

[29] Y. Lu, H. Lu, G. Fu, Q. Liu, Kelm: Knowledge enhanced pre-trained language representations with message passing on hierarchical relational graphs, in: ICLR Workshop on Deep Learning on Graphs for Natural Language Processing, 2022.