SportsTables: A new Corpus for Semantic Type Detection

Sven Langenecker,[1] Christoph Sturm,[2] Christian Schalles,[3] Carsten Binnig[4]

Abstract: Table corpora such as VizNet or TURL which contain annotated semantic types per column are important to build machine learning models for the task of automatic semantic type detection. However, there is a huge discrepancy between corpora that are used for training and testing since real-world data lakes contain a huge fraction of numerical data which are not present in existing corpora. Hence, in this paper, we introduce a new corpus that contains a much higher proportion of numerical columns than existing corpora. To reflect the distribution in real-world data lakes, our corpus SportsTables has on average approx. 86% numerical columns, posing new challenges to existing semantic type detection models which have mainly targeted non-numerical columns so far. To demonstrate this effect, we show the results of a first study using a state-of-the-art approach for semantic type detection on our new corpus and demonstrate significant performance differences in predicting semantic types for textual and numerical data.

Keywords: Semantic Type Detection; Column Annotated Corpora

1 Introduction

Semantic type detection is important for data lakes. Semantic type detection of table columns is an important task to exploit the large and constantly changing data collections residing in data lakes. However, manually annotating tables in data lakes comes at a high cost. Hence, in the past many approaches have been developed that automatically derive semantic types from table data [Hu19b, Zh20, De21, Su22]. Many of the recent approaches use deep learning techniques to build semantic type detection models. As such, corpora containing large amounts of table data with assigned semantic types are required for training and validating. Existing annotated table corpora (e. g. VizNet, TURL) primarily contain tables extracted from the web and therefore limit the capability to represent enterprise data lakes.

Existing corpora and models fall short on real-world data lakes. However, as we can see in Fig. 1, almost all existing corpora that provide annotated columns labeled with semantic types have a lack of table columns that contain numerical data, and tables in these datasets incorporate either only or a very high percentage of textual data. Only GitTables [HDG21] contains a more balanced ratio of textual and numerical data. Nevertheless, compared to real

---

[1] DHBW Mosbach, Germany sven.langenecker@mosbach.dhbw.de
[2] DHBW Mosbach, Germany christoph.sturm@mosbach.dhbw.de
[3] DHBW Mosbach, Germany christian.schalles@mosbach.dhbw.de
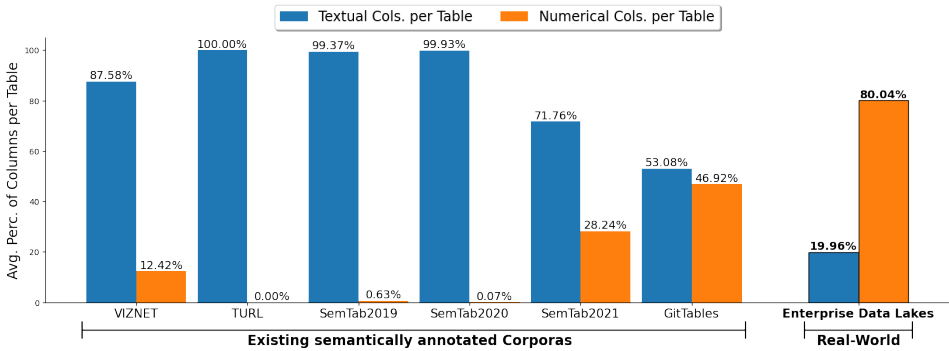[4] TU Darmstadt, Germany carsten.binnig@tu-darmstadt.de

Fig. 1: Average percentage of textual and numerical based columns per table in existing semantically annotated corpora[6] (left bars) compared to real-world data lakes (right bar). This shows the fact that there is a significant shift in the ratio of textual to numeric columns per table from existing corpora to real data lakes. Since all existing semantic type detection models were developed by using the existing corpora, shortcomings in validating the models on numerical data are present and it has not yet been studied in depth how well the models can perform on datasets containing a high proportion of numerical data.

enterprise data lakes, there is a significant discrepancy in the ratio of textual to numerical data. An inspection of a large real-world data lake at a company[5] has shown that on average approx. 20% textual data and 80% numerical data are present (see. Fig. 1 bars on the right). Moreover, semantic type detection models [Hu19b, Zh20, De21, Su22] that are trained on the available corpora also mainly target non-numerical data.

Semantic type detection for numerical data is challenging. Detecting semantic types of numerical columns is generally harder than for textual columns. For example, for a textual column with the values {Germany, USA, Sweden, ...} a model can easily identify the semantic type *country*. Instead, for a numeric column with e. g. the values {20,22,30,34,...} it is not that straightforward and several possibilities for a matching semantic type exist such as *age, temperature, size, money*. The fundamental reason here is that numerical values can be encoded with much fewer bits than string values [Sh48], resulting in a lower overall entropy and thus providing less information content that can be used by a machine learning model to infer the underlying semantic type. Due to the existing corpora providing annotated columns that have been used to create and validate semantic type detection models, we see several essential shortcomings that could not be addressed until now because of the absence of a sufficient dataset for this purpose.

---

[5] The analyses were done at the company LÄPPLE AG

[6] Notice that for GitTables we only considered the tables and columns labeled by terms from DBpedia using the semantic annotation method as described in the GitTables paper. Therefore our reported ratios of textual and numerical data differ from those shown in the GitTables paper because they consider all data, whether annotated or not.

Contributions. In this paper, we thus contribute a new corpus containing tables with semantically annotated columns with numeric and non-numeric columns that reflect the distribution of real-world data lakes. We will make the corpus available which should stimulate research directions such as working on new model architectures that can reliably annotate types to numeric and non-numerical columns. In the following, we discuss the main contribution of this paper.

As a first contribution, we present and provide our new corpus SportsTables[7]. To the best of our knowledge, SportsTables is the first corpus with annotated table columns, which contains a significantly larger proportion of numerical data than textual data. In total, the tables in our corpus have on average about 3 textual and 18 numerical columns. Moreover, the tables in our new corpus are much larger in both the number of columns and the number of rows than in existing corpora which better reflects the characteristics of real-world tables.

As a second contribution that comes together with the corpus, we specify an ontology with semantic types for the sports baseball, basketball, football, hockey, and soccer. This ontology provides fine granular semantic types for all kinds of sports we considered to build SportsTables and allows us to semantically describe each occurring table column, which is not possible with the current ontologies (e. g. DBpedia) at this level of detail. Using a manually created dictionary, we assign a semantic type to each existing column in SportsTables.

As a third contribution, we present our initial results of using our new corpus on Sato [Zh20], a state-of-the-art semantic type detection model. Overall, we can see that when trained on our new corpora, Sato can improve the performance on numerical data types. However, one shortcoming that our analysis shows is that current model architectures are not targeting numerical columns. To be more precise, our analysis demonstrates that textual data columns are mostly correctly semantically interpreted with Sato (F1-Score of 1.0), but on numerical data columns, the model only achieves an F1-Score of about 0.55. This large difference indicates that new model architectures that take the characteristics of numerical columns into account are needed which is a direction that could be stimulated by the availability of our corpus.

Outline. In Section 2, we first provide an overview of existing corpora which was used to build and validate semantic type prediction models and discuss their characteristics and statistics. Afterward, in Section 3, we then introduce our new corpus SportsTables and describe in detail how we created the corpus and labeled the table columns with semantic types. Section 4 first demonstrates the main characteristics of our corpus before we then show the initial results of using our new corpus on Sato. Next, further research challenges are discussed in Section 5 before Section 6 concludes the paper.

---

[7] Available on `https://github.com/DHBWMosbachWI/SportsTables.git`

Corpus	#Table	#Total Columns	Avg. #Columns per Table	Avg. #Rows per Table
VIZNET	78,733	120,609	1.53	18.35
TURL	406,706	654,670	1.61	12.79
SemTab2019	13,765	21,682	1.58	35.61
SemTab2020	131,253	190,494	1.45	9,19
SemTab2021	795	3,072	3.86	874.6
GitTables	1.37M	9.3M	6.82	184.66
SportsTables	**1,187**	**24,838**	**20.93**	**246.72**

Tab. 1: Corpus statistics about the number and sizes of tables.

2 Existing Corpora with Semantic Data Types

In the following, we describe different existing corpora that contain annotated table columns and therefore can be used to build and validate semantic column type detection models. We summarized the main statistics for all corpora in Tab. 1.

VizNet [Hu19a]. The original VizNet corpus [Hu19a] is a collection of data tables from diverse web sources ([Ca08, Vi07, Pl18, NUP16]) which initially do not contain any semantic label annotation. The corpus we consider in this paper is a subset of the original VizNet corpus, which was annotated by a set of mapping rules from column headers to semantic types and then used to build and validate the Sherlock [Hu19b] and Sato[Zh20] prediction model. The corpus contains in total 78,733 tables and 120,609 columns annotated with 78 unique semantic types. Overall, the tables in the corpus contain only 1.53 columns and 18.35 rows on average. Furthermore, the distribution of the column data types is 87.58% textual and 12.42% numerical and thus leads to the shortcomings as described before.

TURL [De21]. The TURL corpus uses the WikiTable corpus [BND15] as basis. To label each column they refer to the semantic types defined in the Freebase ontology [Go22] with a total number of 255 different semantic types. What distinguishes TURL from other corpora is that columns can have multiple semantic types assigned. In total, there are 406,706 tables resulting in 654,670 columns, and on average a table consists of 1.61 columns and 12.79 rows. Again, these are rather small dimensions. In addition, the Turl corpus includes no numerical data at all, which leads to the shortcomings mentioned above when using the corpora.

SemTab. SemTab is a yearly challenge with the goal of benchmarking systems that match tabular data to knowledge graphs since 2019. The Challenge includes the tasks of assigning a semantic type to a column, matching a cell to an entity, and assigning a property to the relationship between columns. Every year, the challenge provides different datasets to validate the participating systems against each other. In this paper we observed the provided corpora for the years 2019 [Ha19], 2020 [Ha20, Cu20], and 2021 [Ha21, Cu20, OP21, ASKR21, HDD21]. Statistic details of the corpora are shown in Tab. 1. In case more than one dataset was provided per year, we aggregated the statistics over all datasets included in the challenge. While SemTab2019 consists of 13,765 tables and 21,682 columns in

total, there are 131,253 tables and 190,494 columns in SemTab2020. In both corpora, the dimensions of the included tables are rather small (on average 1.58 columns and 35.61 rows in 2019 and 1.45 columns and 9.19 rows in 2020). In SemTab2021, the contained tables are the largest in terms of rows with almost 875 on average. However, the number of columns (3.86 on average) is only moderate and the corpus in general is the smallest with a total of 795 tables and 3,072 columns. Numerical data is almost nonexistent in the first two years (0.63% in 2019 / 0.07% in 2020), increasing to 28.24% numeric columns per table on average in 2021, which is still not comparable to the number of numeric data in real world data lakes.

GitTables [HDG21]. GitTables is a large-scale corpus of relational tables created by extracting CSV files from GitHub repositories. Table columns are labeled with semantic types from Schema.org [GBM16] and DBpedia [Au07] using two different automated annotation methods (syntactically/semantically similarity matching from semantic type to column header). In this paper, we have focused on the annotations origin from DBpedia and the results of the semantic annotations method as described in the GitTables paper [HDG21]. This leads to a corpus containing over 1.37M tables and 9.3M columns in total. Although this is by far the largest collection of data tables, the dimensions of the tables are on average only moderate with 6.82 columns and 184,66 rows. Overall, GitTables incorporates the most numeric data with an almost balanced ratio of 53.08% textual and 46.92% numerical columns per table.

Discussion. The overview in Tab. 1 and the discussion before shows that most existing corpora contain no or only a minimal fraction of numerical data types which is very different from real-world data lakes. An exception is GitTables which has a much higher ratio of numerical columns. However, as we show in Sect. 4, GitTables still lacks a good coverage of different numeric semantic types which is one important aspect that we tackle with our new corpus SportsTables which covers a wide variety of different numerical semantic types. Moreover, another important (but orthogonal) aspect is that existing corpora include a large number of tables. However, on average the tables are very small in terms of the number of columns and the number of rows. Instead, our new corpus SportsTables contains fewer tables, but on average a significantly higher number of columns and rows per table to better reflect the characteristics of real-world data lakes

3 The SportsTables Corpus

In the following, we will introduce our new corpus and describe in detail the implemented construction pipeline to build SportsTables.

Methodology to generate corpus. Fig. 2 gives an overview of our implemented pipeline to generate the new corpus. The main idea was to collect data tables from different sports domains such as soccer, basketball, baseball, etc. since data tables coming from such kinds of sources are rich in numerical columns. For example, a soccer player statistic table of
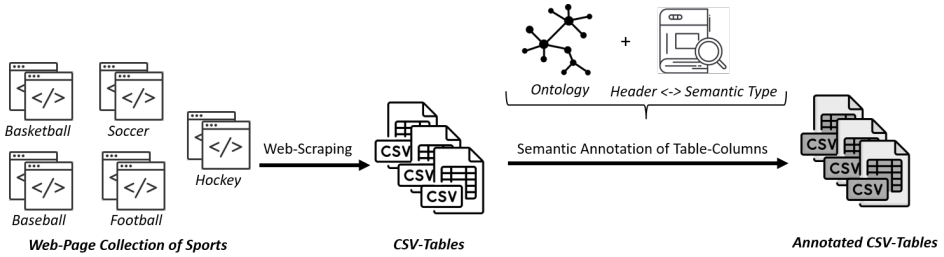
Fig. 2: Overview of the implemented pipeline to build SportsTables. We use web-scraping techniques to extract HTML tables from a manually defined web page collection for each selected sport and convert the tables to CSV files. With the help of a defined ontology and a manually created dictionary that maps column headers to semantic types, we annotate each table column with an appropriate semantic type.

a soccer season contains typically 3 textual columns (e. g., player name, team name, field position) and 18 numerical columns (e. g., goals, games played, assists). Hence, building a collection of such tables will lead to a corpus that contains many numerical columns which are in addition semantically interpretable. As a result, the corpus will enable to analyze the performance of semantic type prediction models in a much more rigorous manner regarding numerical data.

Scraping data from the web [Di19]. A vast amount of data covering information about player statistics, team statistics, coach statistics, or season rankings of different sports are available on various web pages. Therefore, for collecting the data, we built a data collection pipeline based on web scraping technology[Di19]. In the first step, we manually searched and defined a set of different web pages for each of the selected sports of which we want to scrape contained data tables (left side of Fig. 2). We first converted each HTML table on the web pages to Pandas-Dataframes using Python and then saved them as CSV files (center of Fig. 2), since this file format is most known and used to store raw structured data [Mi16]. During the scrape process, we kept the respective column headers from the original HTML table and used them as headers in the CSV file.

Annotating columns with semantic types. Due to the low granularity of existing ontologies (e. g. DBpedia) regarding semantics of a given sport, we manually created an ontology-like set of valid semantic types for all sports. For example, in DBpedia there is the type *Person.Athlete.BasketballPlayer*, but semantic labels in the particular that would match individual numerical columns such as *NumberOfGoals* are not defined. Next, we annotated all table columns with semantic types using a manually created dictionary that maps column headers to matching semantic types from our created set. Since the column headings were in many cases identical if the semantic content was the same, this procedure significantly reduces the manual labeling effort. In addition, to ensure that the labels are of very high quality in terms of correctness, we manually checked each assignment based on the content of the columns.

Sports	#Table	#Total Cols	Avg. #Text. Cols per Table	Avg. #Num. Cols per Table	Avg. #Rows per Table
Baseball	174	3,829	3.97	18.03	76.34
Basketball	180	3,801	1.78	19.34	152.5
Football	303	6,764	2.45	19.88	354.79
Hockey	257	5,347	2.1	18.7	247.15
Soccer	273	5,097	3.9	14.77	297.11
Total	1,187	24,838	2.83	18.1	246.72

Tab. 2: Corpus statistics about the number and size of included tables. Statistics are shown broken down by individual sports taken into account and in total. Across all sports, the average number of numeric columns is much higher than textual columns.

4 Analysis of the Corpus

This section describes the characteristics of SportsTables in detail and then demonstrates the significant impact of these characteristics on semantic type prediction frameworks in a small study where we apply the corpus to an existing type detection model.

4.1 Corpus Characteristics

In the following. we discuss the statistics of the SportsTables corpus and compare them to the existing corpora.

Data statistics (Tab. 1&Tab. 2). Using the described pipeline for creating SportsTables, a total of 1,187 tables which comprises 24,838 columns (approx. 86% numeric and 14% textual) are scraped from the web resulting in 20.93 columns (2.83 textual and 18.1 numerical) per table on average. This ratio of textual to numerical columns, as well as the total average number of columns in a table, differs significantly from existing corpora. To provide details about the contribution of different sports areas contained in SportsTables, Tab. 2 shows the main statistics by the individual areas of sports.

Fig. 3 shows a comparison of the average number of textual and numerical columns per table of SportsTables versus that of the existing corpora. Here we can see that numerical columns only exist in the corpora VizNet with 0.33, SemTab2021 with 1.09, and GitTables with 3.2 columns per Table. Compared to GitTables, in SportsTables there are thus on average over 6 times more numeric columns per table. Moreover, as we discuss below, our corpus uses a much richer set of numerical data types that better reflects the characteristics in real-world data lakes which is very different from GitTables. For example, when looking at the semantic types that are assigned to numerical columns in GitTables, more than half (393,925) of the columns are labeled with just a single type *Id*.

In terms of the total number of columns, the tables in SportsTables (20.93 columns per table) are on average about 3 times wider than in GitTables (6.82 columns per table), which contains the widest tables among the existing corpora. As such, the number of columns in tables
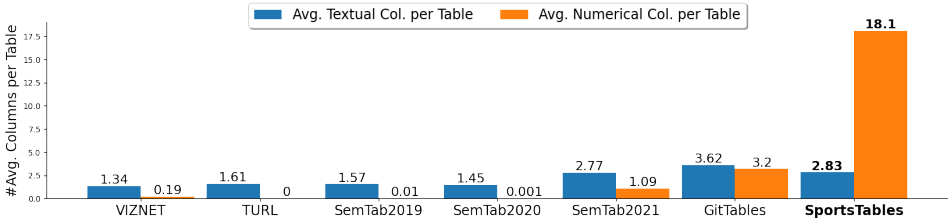
Fig. 3: Average number of textual and numerical columns per table for each existing annotated corpora and our new SportsTables corpus. This shows the absence of numerical data columns per table in most existing corpora and the dominance of textual data columns per table in all existing corpora. Instead, our new corpus SportsTables contains on average over 6 times more numerical columns than textual columns.
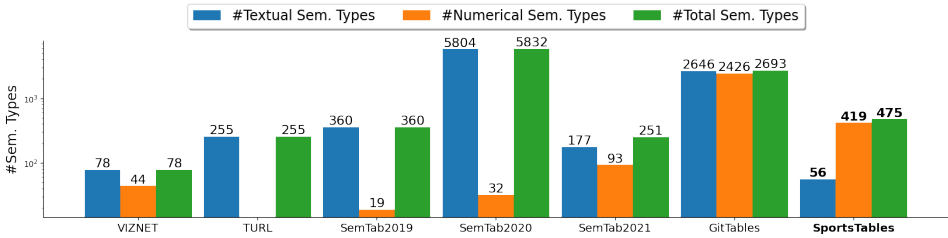

Fig. 4: Corpus statistics about the number of unique semantic types included. Showing that our new corpus has a higher proportion of numerical semantic types than textual semantic types in contrast to the existing corpora. In addition, there is a large overlap of semantic types used for textual and numeric columns in the existing corpora. In comparison, the semantic types in SportsTables are disjoint for the two column data types.

of SportsTables are reflecting better the width when comparing this to the characteristics of the tables in the real-world data lakes which we analyzed. Moreover, considering the average number of rows per table, it can be seen that the tables in SportsTables have on average 246.72 rows. In comparison, tables in SportsTables are larger on overage than in many other corpora where tables have typically fewer rows.

Annotation statistics. Semantic type annotation follows a two step process. First, we establish a directory with manually defined mappings from column header to semantic type for each existing header. Second, we label each column with the semantic type listed in the directory for its header. As a result, 56 textual and 419 numerical semantic types are present in the corpus. Thereby textual semantic types are those which specify textual columns and numerical types are those which specify columns containing numeric values. To compare the annotation statistics, we also counted the number of textual and numerical semantic types in an analysis of the existing corpora. The results of these analyses can be seen in Fig. 4. Different from our corpus, the sets of textual and numerical types are not disjoint in all other corpora (except TURL where no numeric values are present). This indicates that

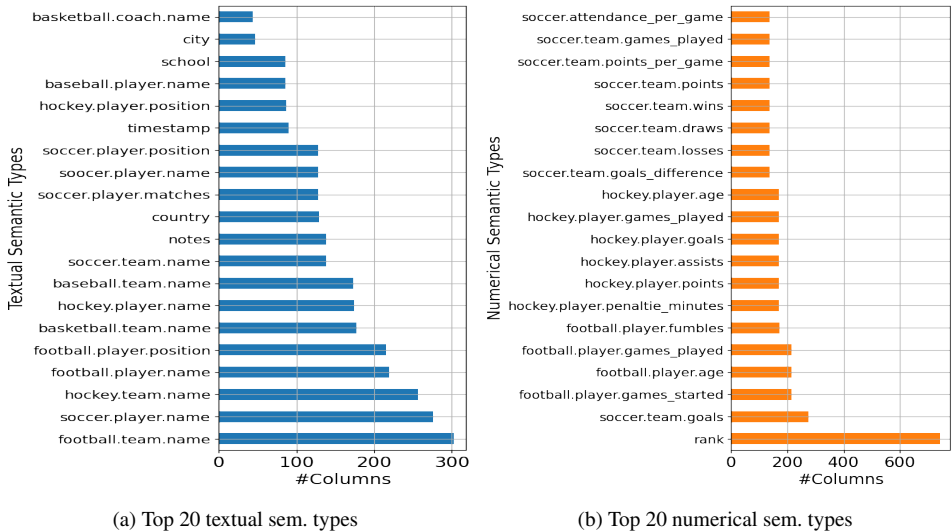(a) Top 20 textual sem. types   (b) Top 20 numerical sem. types

Fig. 5: Semantic type annotation statistics of SportsTables. (a) Shows column annotation counts of the top 20 textual semantic types. Across all kinds of sports, *player.name* and *team.name* are the most common. (b) Shows column annotation counts of the top 20 numerical semantic types. A dominant type here is *rank*, which describes a column containing the placements of e. g. a team in a season standings table.

individual semantic types were assigned to both textual and numerical columns which is problematic if semantic type detection models should be trained and tested on these corpora. In particular, GitTables has a very large overlap and almost all semantic types are used in both column data types. To give an example, in GitTables the semantic types *comment, name* and *description* are assigned to both column data types. Next, we take a closer look into the semantic types of our corpus.

Fig. 5a and Fig. 5b shows the top 20 semantic types (textual and numerical) in regards to how often they were assigned to a table column. It can be seen that the most common textual types across all sports are *player.name* and *team.name*. These are types that occur in almost every table. Other types such as *country* or *city* are also common, describing, for example, the player's origin or the team's hometown. Among numeric semantic types, *rank* is by far the most common and is present in almost all tables. The type describes a column containing the placement of e. g., a team in a "seasons standing" table or a player in a "top scorer" table. All other numeric semantic types show mainly an equal distribution of the frequency, which is a good precondition for training machine learning models. In order to show not only the frequency of the top 20 semantic types, Fig. 6 plots all semantic types (separated in textual and numerical) by the frequency of occurrences. Here we see that 19 textual and 66 numerical semantic types occur only once in the entire corpus. For

the training and testing of prediction models, we would suggest not considering these types due to the low occurrences.

SportsTables vs. GitTables. Since GitTables is the largest corpus with the most tables, one could argue that a subset of GitTables would result in a new corpus with similar characteristics as SportsTables. To analyze this, we executed a small experiment in which we filtered out only tables from GitTables where the number of textual and numerical columns (min. 3 textual and 18 numerical columns) is at least the same as it is in SportsTables. The result was a corpus containing a total of 16,909 tables and 743,432 columns. On average a table has 12.53 textual columns, 31.43 numerical columns, and 17.35 rows. However, looking at the semantic types that are assigned to numerical columns, more than half (393,925) of the columns are labeled with the type *Id*. In terms of training and validating semantic type detection models, this is rather an unfavorable type representing no semantically meaning. Moreover, the next 5 most common numerically based semantic types are *parent, max, comment, created* and *story editor*, constituting a large proportion of the columns. The assignment of these types to numerical data is slightly less understandable and indicates a lack of quality in the automatically generated labels for table columns.

4.2   An Initial Study of Using SportsTables

In the following, we report on the initial results of using Sato, a recent semantic type detection model, on our new corpus. With this, we want to measure how well the semantic types in our corpus can be inferred by the model with a special focus on how it performs on textual and numerical columns.

Experiment setup. For the first experiments, we only considered the soccer data from SportsTables. Thereby, we split the corpus into different sizes of train and test sets (5/95, 10/90, 15/85, 20/80), to show the results of scenarios where the model has less and more training data available. We use the pre-trained Sato model, which was trained on the VizNet corpus, and re-trained it with the different training set sizes. During re-training, we replaced the last layer of Sato to support the number of semantic types that occur in SportsTables and then re-trained the entire neural network. To measure the performance, we applied the re-trained model to the corresponding test data set.

Results of study. Fig. 7 shows the results of the experiments reporting F1-Scores using the defined different sizes of train and test splits as described before. We plot macro and weighted average F1-Score across all semantic types to show the total performance, but also separate average F1-Score for only textually and numerically based semantic types, respectively. As we can see in the figure, while Sato can detect numerical types, there is a significant performance difference between predicting textual and numerical semantic types for all setups. At the data split 20/80, all textual columns can be predicted correctly by the model, whereas for numerical columns only an F1-Score of 0.56 is achieved. On average, the difference in F1-Score between textual and numeric types is 0.41 across all setups.
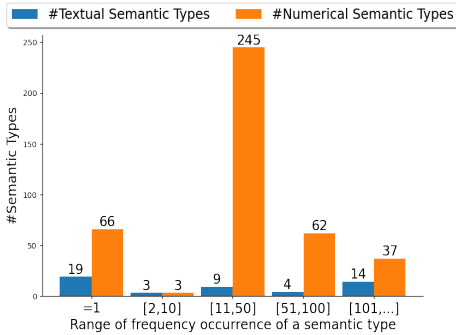
Fig. 6: Shows how often semantic types occur in SportsTables using buckets of varying widths, which represents the frequency of occurrences. For example, 19 textual and 66 numerical types occur only once in the entire corpus.
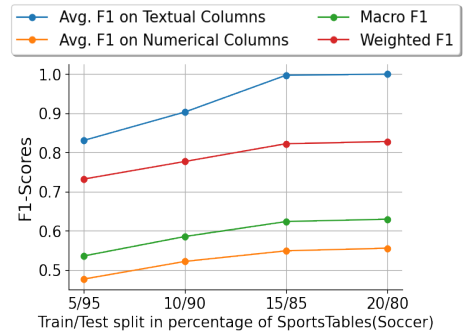

Fig. 7: Initial results using the Sato model on our new SportsTables corpus with different train/test split sizes. The differences in F1-Scores for predicting textual and numeric columns indicate that the model can handle textual data more effectively than numeric data.

These results indicate that the model is more able to handle textual data and determine the associated semantic type more accurately than for numeric data. Across all semantic types, the weighted F1-Score increase from 0.73 to 0.82 while the macro F1-Score range from 0.53 to 0.63, which are rather moderate score values for semantic type prediction models.

5 Further Research Challenges

Detecting semantic types in real-world data lakes comes with many more challenges that need to be addressed. In particular, based on our findings of the analysis using Sato in Section 4, we think that new model architectures are needed for detecting numerical data types which have very different characteristics from non-numerical data. In the following, we list some of the challenges we think are important to be addressed. We hope that our corpus enables research on those challenges.

Embedding numerical data: Most state-of-the-art models apply language models like BERT [De19] to encode literals to infer the semantic type of a table column. Since such approaches are optimized for textual data, the performance on numerical data of such models is not entirely analyzable with the existing corpora.

Leveraging numerical context: To improve the semantic type prediction of a table column, recent approaches like Sato [Zh20], TURL [De21] and Doduo [Su22] incorporate also context information like the table-topic or values from neighboring columns of the same table. Given that tables in existing corpora contain almost entirely textual columns, the contexts (e. g. values from neighboring columns) used are rich in information and therefore also lead to performance improvements. However, it is unclear how effective this approach

is in case the tables contain many numerical columns and only a few textual columns since the context information provided is reduced due to the lower entropy of numeric values as described before.

Supporting wide tables: Existing datasets for semantic type detection consist of tables with small numbers of columns and rows. In nearly all corpora, the existing tables contain on average less than two columns and less than 40 rows (see Tab. 1). Therefore, at the current state, it has not been analyzed how state-of-the-art models can handle such large tables. To give an example of why large tables could be a problem for recent models, we will briefly discuss Doduo[Su22]. Doduo uses pre-trained language models (e. g., BERT) and hence they have to convert the entire table into token sequences with a fixed tensor length of 512 elements so that the table and its entries can be meaningfully processed by the language model. To accomplish this, Doduo serializes the complete table and its entries as follows: for each table that has n columns $T = (c_i)_{i=1}^n$, where each column has N_m column values $c_i = (v_i^j)_{j=1}^m$, they let $serialize(T) ::= [CLS]v_1^1...[CLS]v_1^n...v_m^n[SEP]$, where the special token [CLS] marks the beginning of a new table column and [SEP] the end of a token sequence. With this methodology of serialization and the fixed given tensor length, increasing the number of table columns means that decreasing number of values of each column can be included for serialization. For example, a table with 512 columns would allow only one value per column to be considered and this would most likely result in an insufficient semantic representation of the column based on that one value.

6 Conclusion

Existing corpora for training and validating semantic type detection models mainly contain tables with only or a very high proportion of textual data columns and no or just a limited number of numerical data columns. Therefore, it has not been studied precisely how well state-of-the-art models perform on a dataset with a very high percentage of numerical columns as it occurs in real-world data lakes. Moreover, tables in existing corpora are very small regarding the total number of columns and rows. To tackle these shortcomings, we built a new corpus called SportsTables which contains tables that have on average approx. 3 textual columns, 18 numerical columns, and 250 rows. With our new corpus, semantic type detection models for table columns can now be holistically validated against numerical data. We show initial results by using Sato – a state-of-the-art model – on our new corpus and report significant differences in the performance of predicting semantic types of textual data and numerical data. The corpus is available on `https://github.com/DHBWMosbachWI/SportsTables.git`. Finally, we think that the corpus is just a first step to stimulate more research on new model architectures that can better deal with numerical and non-numerical data types.

Bibliography

[ASKR21] Abdelmageed, Nora; Schindler, Sirko; König-Ries, Birgitta: , fusion-jena/BiodivTab, October 2021. https://doi.org/10.5281/zenodo.5584180.

[Au07] Auer, Sören; Bizer, Christian; Kobilarov, Georgi; Lehmann, Jens; Cyganiak, Richard; Ives, Zachary: DBpedia: A Nucleus for a Web of Open Data. In (Aberer, Karl; Choi, Key-Sun; Noy, Natasha; Allemang, Dean; Lee, Kyung-Il; Nixon, Lyndon; Golbeck, Jennifer; Mika, Peter; Maynard, Diana; Mizoguchi, Riichiro; Schreiber, Guus; Cudré-Mauroux, Philippe, eds): The Semantic Web. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 722–735, 2007.

[BND15] Bhagavatula, Chandra Sekhar; Noraset, Thanapon; Downey, Doug: TabEL: Entity Linking in Web Tables. In: The Semantic Web - ISWC 2015. Springer International Publishing, Cham, pp. 425–441, 2015.

[Ca08] Cafarella, Michael J.; Halevy, Alon; Wang, Daisy Zhe; Wu, Eugene; Zhang, Yang: WebTables: Exploring the Power of Tables on the Web. In: VLDB. volume 1. VLDB Endowment, p. 538–549, 2008.

[Cu20] Cutrona, Vincenzo; Bianchi, Federico; Jiménez-Ruiz, Ernesto; Palmonari, Matteo: , Tough Tables: Carefully Evaluating Entity Linking for Tabular Data, November 2020. GT and Target files have different formats in 2T and 2T_WD (CEA). 2T complies with the SemTab2019 CEA format (tab_id, col_id, row_id, entity), while 2T_WD follows the SemTab2020 CEA format (tab_id, row_id, col_id, entity). Visit the SemTab challenge website (https://www.cs.ox.ac.uk/isg/challenges/sem-tab/) for more details.

[De19] Devlin, Jacob; Chang, Ming-Wei; Lee, Kenton; Toutanova, Kristina: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: NAACL-HLT 2019. Association for Computational Linguistics, Minneapolis, Minnesota, pp. 4171–4186, June 2019.

[De21] Deng, Xiang; Sun, Huan; Lees, Alyssa; Wu, You; Yu, Cong: TURL: Table Understanding through Representation Learning. In: VLDB. volume 14. VLDB Endowment, pp. 307 – 319, 2021.

[Di19] Diouf, Rabiyatou; Sarr, Edouard Ngor; Sall, Ousmane; Birregah, Babiga; Bousso, Mamadou; Mbaye, Sény Ndiaye: Web Scraping: State-of-the-Art and Areas of Application. In: International Conference on Big Data. pp. 6040–6042, 2019.

[GBM16] Guha, R. V.; Brickley, Dan; Macbeth, Steve: Schema.Org: Evolution of Structured Data on the Web. Commun. ACM, 59(2):44–51, jan 2016.

[Go22] Google: , Freebase Data Dumps. https://developers.google.com/freebase, 2022.

[Ha19] Hassanzadeh, Oktie; Efthymiou, Vasilis; Chen, Jiaoyan; Jiménez-Ruiz, Ernesto; Srinivas, Kavitha: , SemTab 2019: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching Data Sets, October 2019. https://doi.org/10.5281/zenodo.3518539.

[Ha20] Hassanzadeh, Oktie; Efthymiou, Vasilis; Chen, Jiaoyan; Jiménez-Ruiz, Ernesto; Srinivas, Kavitha: , SemTab 2020: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching Data Sets, November 2020. https://doi.org/10.5281/zenodo.4282879.

[Ha21]   Hassanzadeh, Oktie; Efthymiou, Vasilis; Chen, Jiaoyan; Jiménez-Ruiz, Ernesto; Srinivas, Kavitha: , SemTab 2021: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching Data Sets, November 2021. https://doi.org/10.5281/zenodo.6154708.

[HDD21]   Hulsebos, Madelon; Demiralp, Cagatay; Demiralp, Paul: , GitTables benchmark - column type detection, November 2021. https://doi.org/10.5281/zenodo.5706316.

[HDG21]   Hulsebos, Madelon; Demiralp, Çagatay; Groth, Paul: GitTables: A Large-Scale Corpus of Relational Tables. CoRR, abs/2106.07258, 2021.

[Hu19a]   Hu, Kevin; Gaikwad, Snehalkumar 'Neil' S.; Hulsebos, Madelon; Bakker, Michiel A.; Zgraggen, Emanuel; Hidalgo, César; Kraska, Tim; Li, Guoliang; Satyanarayan, Arvind; Demiralp, Çağatay: VizNet: Towards A Large-Scale Visualization Learning and Bench-marking Repository. In: CHI '19. CHI '19, ACM, New York, NY, USA, p. 1–12, 2019.

[Hu19b]   Hulsebos, Madelon; Hu, Kevin; Bakker, Michiel; Zgraggen, Emanuel; Satyanarayan, Arvind; Kraska, Tim; Demiralp, Çagatay; Hidalgo, César: Sherlock: A Deep Learning Approach to Semantic Data Type Detection. In: SIGKDD. KDD '19, ACM, New York, NY, USA, p. 1500–1508, 2019.

[Mi16]   Mitlöhner, Johann; Neumaier, Sebastian; Umbrich, Jürgen; Polleres, Axel: Characteristics of Open Data CSV Files. In (IEEE, ed.): International Conference on Open and Big Data (OBD). IEEE, pp. 72 – 79, 2016.

[NUP16]   Neumaier, Sebastian; Umbrich, Jürgen; Polleres, Axel: Automated Quality Assessment of Metadata across Open Data Portals. J. Data and Information Quality, 8(1), oct 2016.

[OP21]   Oliveira, Daniela; Pesquita, Catia: , SemTab 2021 BioTable Dataset, October 2021. https://doi.org/10.5281/zenodo.5606585.

[Pl18]   Plotly: , Plotly. https://chart-studio.plotly.com/feed/, 2018.

[Sh48]   Shannon, Claude Elwood: A Mathematical Theory of Communication. The Bell System Technical Journal, 27:379–423, 1948.

[Su22]   Suhara, Yoshihiko; Li, Jinfeng; Li, Yuliang; Zhang, Dan; Demiralp, Çağatay; Chen, Chen; Tan, Wang-Chiew: Annotating Columns with Pre-Trained Language Models. In: SIGMOD. ACM, New York, NY, USA, pp. 1493—-1503, 2022.

[Vi07]   Viegas, Fernanda B.; Wattenberg, Martin; van Ham, Frank; Kriss, Jesse; McKeon, Matt: ManyEyes: A Site for Visualization at Internet Scale. IEEE Transactions on Visualization and Computer Graphics, 13(6):1121–1128, nov 2007.

[Zh20]   Zhang, Dan; Hulsebos, Madelon; Suhara, Yoshihiko; Demiralp, Çağatay; Li, Jinfeng; Tan, Wang-Chiew: Sato: Contextual Semantic Type Detection in Tables. In: VLDB. volume 13. VLDB Endowment, p. 1835–1848, jul 2020.