KaushikAcharya at SemEval-2021 Task 9: Candidate Generation for Fact Verification over Tables

Kaushik Acharya
Philips India Ltd. / Bangalore, India
acharya.kaushik@gmail.com

Abstract

This paper describes the system submitted in the SemEval-2021 Statement Verification and Evidence Finding with Tables task. The system relies on candidate generation for logical forms on the table based on keyword matching and dependency parsing on the claim statements.

1 Introduction

Tables convey important information in a concise manner. This is true in many domains, scientific documents being one of them. Truth verification tasks in past(e.g. SemEval-2019 Fact Checking Task) have focused on written text without considering the tables. The current shared task (Wang et al., 2021) focuses on tables written in English language. It requires participants to develop systems to predict

- veracity of textual claims (statement verification)
- identify table cells forming relevant evidence for the claim (evidence finding)

The shared task [1] comprised of two sub-tasks:

1. Subtask A: **Table Statement Support**
2. Subtask B: **Relevant Cell Selection**

Subtask A is a classification problem in which the system needs to assign one of the following labels for the claim statement:

- *Entailed*: Table supports the statement.
- *Refuted*: Statement is contradicted by the table.
- *Unknown*: Not enough information available in the table to assess statement's veracity.

Country	N	Board network			Family network			State network			Political network		
		mean	SD	max	mean	SD	max	mean	SD	max	mean	SD	max
Hong Kong	133	5.12	6.1	33	2.62	4.51	26	1.00	1.41	6	0.67	1.37	6
Indonesia	169	1.64	3.31	23	0.95	2.64	17	0.14	0.38	2	0.22	1.09	9
Japan	126	1.84	2.33	15	0.07	0.42	3	0.09	0.31	2	0.00	0.00	0
South Korea	133	2.5	2.8	21	1.09	1.37	6	0.15	0.40	2	0.02	0.15	1
Malaysia	281	7.35	6.61	37	1.07	1.94	8	2.15	3.09	18	0.36	0.74	5
Philippines	98	8.52	8.91	38	5.33	6.16	21	0.71	1.59	10	0.20	0.81	6
Singapore	116	3.52	3.24	15	0.59	1.66	12	1.28	2.40	11	0.57	1.90	14
Taiwan	107	1.6	2.22	12	0.21	1.11	7	0.14	0.46	3	0.00	0.00	0
Thailand	127	5.11	5.04	23	1.58	3.15	19	0.73	1.99	11	0.29	1.16	8

Figure 1: Example table showing Networks across East Asia.

id	Claim statement
1	The n value is same for Hong Kong and Malaysia.
2	There are 9 different types country in the given table.

Table 1: Statement claims of table in Figure 1

Subtask B requires finding evidence(table cells) which are minimally required to either support or refute the claim statement. This is not applicable for the statements whose veracity is unknown.

These tables were sourced from scientific articles belonging to journals published by Elsevier and available on ScienceDirect. For details related to web scraping of the articles, selection criteria for choosing the tables, creating statement claims and assigning table cell evidence for the claim, please refer to the task description paper (Wang et al., 2021). An example table is shown in Fig. 1 with corresponding claims mentioned in Table 1.

System described in this paper generates logical form candidates on the table data frame based on the claim statement. It executes the most probable candidate and verifies the output to check whether it matches with the one mentioned in the statement. Averaged F1 score over the tables are shown in

[1] https://competitions.codalab.org/competitions/27748

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 1271–1275
Bangkok, Thailand (online), August 5–6, 2021.

Table 4. Source code has been released on github [2].

2 Related Work

Thorne et al. (2018) had conducted Fact Extraction and VERification (**FEVER**) Shared Task[3] to build systems to verify claims based on evidence from Wikipedia. Similar to the current shared task, systems had to label claim as **Supported**, **Refuted** or **NotEnoughInfo** (if there isn't sufficient evidence to either support or refute it). Along with that, system must extract textual evidence (sets of sentences) that support or refute the claim.

Pasupat and Liang (2015) had created question answering dataset (WikiTableQuestions) [4] from Wikipedia tables. Using question-answer pairs as supervision, they developed a logical-form driven parsing algorithm.

Herzig et al. (2020) build a question answering model over tables without generating logical forms by extending BERT's architecture (Devlin et al., 2019) with additional positional embeddings to encode tabular structure.

3 Model Description

3.1 XML Data

Input xml documents were parsed using the *ElementTree XML API* [5]. Each document is composed of table element(s). Table element is composed of row elements and statements. The row elements describe the rows and columns of the table and contains the text of each table cell. Optionally, legend and caption texts were also provided for a portion of tables.

3.2 Loading table

Table element was parsed and loaded into *pandas dataframe* [6]. The challenging part was in identifying column labels having hierarchical indexing [7]. An example of hierarchical columns is shown in Fig. 1, which has four columns: Broad network, Family network and so on. All these columns have three sub-columns: mean, SD, max. A simple approach to identify whether multiple table rows represent column labels was applied: Table rows from top were considered as part of column labels until all the columns of the table are filled. In the example table, *Broad network* is mentioned in table cell col:2 and *Family network* belongs to col:5 The in-between columns (i.e. 3,4) were filled in next row. Hence the first two rows were considered as column labels.

#	Candidate
1	Column superlative value
2	Column values identical or unique
3	Comparison of column values for a row or vice-versa
4	Column value range

Table 2: Partial list of Candidates

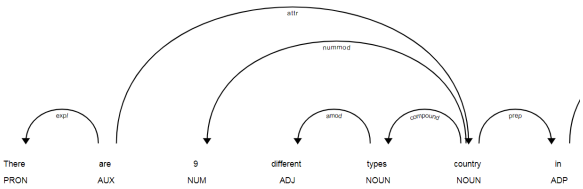

Figure 2: Dependency tree for statement id=2 in Table 1

3.3 Candidate Selection

A list of candidates were defined along with pattern rules to identify them and corresponding operations to execute. For instance, for the candidate superlative(highest/lowest) of column, corresponding operation on pandas dataframe gets executed. Statements are parsed to identify the candidate and fact verification is done in the following steps:

1. Match column and row labels (if available) based on approximate keyword matching.
2. Candidate operation(s) are identified based on keyword and dependency tag matching.
3. Candidate logical form is executed.
4. Output of candidate operation is matched with the one mentioned in the statement.

Table 2 enumerates subset of candidates.

For matching columns and row labels, sliding window over the tokens of statments are matched using Jaro metric (Cohen et al., 2003). Number

[2] https://github.com/kaushikacharya/statement_verification_evidence_finding

[3] https://fever.ai/2018/task.html

[4] https://ppasupat.github.io/WikiTableQuestions/

[5] https://docs.python.org/3/library/xml.etree.elementtree.html

[6] https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html

[7] https://pandas.pydata.org/pandas-docs/stable/user_guide/advanced.html

of tokens in column/row label is considered as the window size. As an example, for matching the row label *Hong Kong* of Figure 1 in Table 1, sliding window of two tokens is considered. If overlapping window match the same column/row label, the one with maximum Jaro metric is chosen. A span of statement tokens is considered matched if the Jaro metric is above a threshold (value considered 0.85).

Examples: Statement id=1 in Table 1 matches the candidate: *comparison of two rows for a column value*. The two rows referred by *Hong Kong* and *Malaysia* are compared for the value corresponding to the column *N*. Logical form: Comparing whether the cell value for the matched column corresponding the matched rows is same/different.

Statement id=2 refers to the candidate: *unique count of the values under the column* **Country**. Candidate is chosen based on the keyword *different* and matching of a single column **Country**. Candidate value (that needs to be verified) is identified based on the dependency tree shown in Figure 2. The token *9* with part of speech tag *NUM* has the head token of column *country* through dependency tag *nummod*. The logical form that has been assigned for the candidate is unique count over the matched column of the *pandas dataframe*.

4 Experimental Setup

4.1 Data

The data splits used were the same as provided by the task organizers. Split statistics is shown in Table 3.

split	docs	tables	claims
train	424	981	4506
dev	21	52	556
test	36	52	653

Table 3: Data split count statistics

4.2 External libraries

- spaCy (version: 2.3.2) [8]
- word2number [9]

4.3 Evaluation Metrics

Task A - Fact Verification The goal of this task is to determine whether a statement can be entailed/refuted by the given table, or cannot be determined from the table. The classification algorithm is evaluated using the standard F1-score. Two different evaluation results were generated:

1. Two way
2. Three way

Two way is an easier evaluation in which **unknown** ground truth labels were ignored. Whereas in *three way* all the three labels were considered. This tests whether the classification algorithm understands cases where there are insufficient information to make prediction.

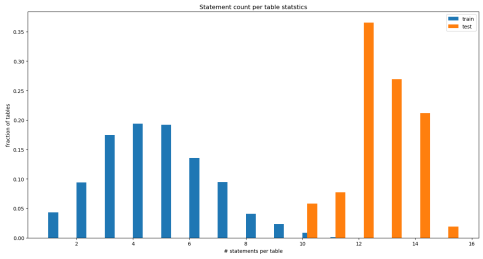

Figure 3: Distribution of number of statements per table.

Task B - Evidence Finding The goal of this task is to determine whether a table cell is needed to entail/refute the given statement. In other words, whether a statement can be entailed/refuted given only the table cells marked as *relevant*. F1 score is computed for each table with *relevant* cells as the positive class and *irrelevant* cells as the negative class.

Fig. 3 shows that unlike test data, train data has many tables with very few statements. This indicates that comparing averaged F1 score(as shown in Table 4) between train and test data is not a good indicator how well the model works on unseen test data compared to its performance on train/dev data. Instead comparing confusion matrix(as shown in Table 5 and Table 6) is a better indicator.

5 Results

Averaged F1 score over the tables are shown in Table 4. These are the scores at the time of writing this paper. Train data didn't had the ground truth for the relevant cell selection. Hence value is non-available for Task B on train data. Due to absence of *unknown* class in train data, 2-way and 3-way averaged F1 scores are same. Confusion matrix

[8] https://github.com/explosion/spaCy
[9] https://github.com/akshaynagpal/w2n

is displayed in Table 5 for train data and Table 6 for test data. Predicted claim class *unknown* also contains the claim statements for which the system failed to identify candidate. Hence this class shows a high value.

split	Task A (2 way)	Task A (3 way)	Task B
train	0.3669	0.3669	NA
dev	0.3804	0.4314	0.3687
test	0.4037	0.4909	0.3849

Table 4: F1 score averaged over tables

truth	predicted		
	entailed	refuted	unknown
entailed	863	472	1483
refuted	49	861	778

Table 5: Confusion Matrix for Task A on train data. Row represents truth classes and column represents the predicted classes.

truth	predicted		
	entailed	refuted	unknown
entailed	83	39	152
refuted	8	127	113
unknown	3	15	113

Table 6: Confusion Matrix for Task A on test data. Rows and columns represents truth and predicted classes respectively.

type	# statements
Total wrongly predicted statements	2782
Neither column nor row matched	786
No column matched but row(s) matched	386

Table 7: Matching columns/rows in wrongly predicted statements in train data.

Error Analysis: The errors can be categorized broadly into the following categories:

1. Absence of semantic matching
2. Lack of enough candidate generation rules
3. Classifying candidate in a deterministic way

Due to keyword matching the system fails to identify the columns which are mentioned with different words in the statement even though semantically they are same. Table 7 gives a glimpse of *probable* failures under this category.

The set of candidate generation rules needs to be extended. The current system misses out candidate generation in several statements because of the absence of these not yet defined candidates. The high number of *unknown* predictions in the confusion matrices shown in Table 5 and 6 is a proof of this issue. There's a need for a scoring system which considers multiple probable candidates for logical forms. The current system selects a single candidate based on the one which matches first in the order listed for candidate match.

6 Conclusion

I have described the system used for submission to the Statement Verification and Evidence Finding with Tables task. The problem has been framed as a candidate generation for logical forms over dataframe using keyword matching and dependency parsing. Future work would include extending the defined list of candidates and usage of scoring based system to identify the most probable candidate. This improvement would take ideas from (Pasupat and Liang, 2015) for feature extraction and build a log-linear model to compute score for the candidates.

References

William W. Cohen, Pradeep Ravikumar, and Stephen E. Fienberg. 2003. A comparison of string distance metrics for name-matching tasks. In *Proceedings of the 2003 International Conference on Information Integration on the Web*, IIWEB'03, page 73–78. AAAI Press.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333, Online. Association for Computational Linguistics.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480, Beijing, China. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018. The fact extraction and VERification (FEVER) shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.

Nancy Xin Ru Wang, Diwakar Mahajan, Marina Danilevsky, and Sara Rosenthal. 2021. SemEval-2021 Task 9: Fact Verification and Evidence Finding for Tabular Data in Scientific Documents (SEM-TAB-FACTS). In *Proceedings of the 15th international workshop on semantic evaluation (SemEval-2021)*.