# TAPAS at SemEval-2021 Task 9: Reasoning over tables with intermediate pre-training

**Thomas Müller, Julian Martin Eisenschlos, Syrine Krichene**

Google Research, Zürich
{thomasmueller,eisenjulian,syrinekrichene}@google.com

## Abstract

We present the TAPAS contribution to the Shared Task on Statement Verification and Evidence Finding with Tables (SemEval 2021 Task 9, Wang et al. (2021)). SEMTABFACT Task A is a classification task of recognising if a statement is entailed, neutral or refuted by the content of a given table. We adopt the binary TAPAS model of Eisenschlos et al. (2020) to this task. We learn two binary classification models: A first model to predict if a statement is neutral or non-neutral and a second one to predict if it is entailed or refuted. As the shared task training set contains only entailed or refuted examples, we generate artificial neutral examples to train the first model. Both models are pre-trained using a MASKLM objective, intermediate counter-factual and synthetic data (Eisenschlos et al., 2020) and TABFACT (Chen et al., 2020), a large table entailment dataset. We find that the artificial neutral examples are somewhat effective at training the first model, achieving 68.03 test F1 versus the 60.47 of a majority baseline. For the second stage, we find that the pre-training on the intermediate data and TABFACT improves the results over MASKLM pre-training (68.03 vs 57.01).

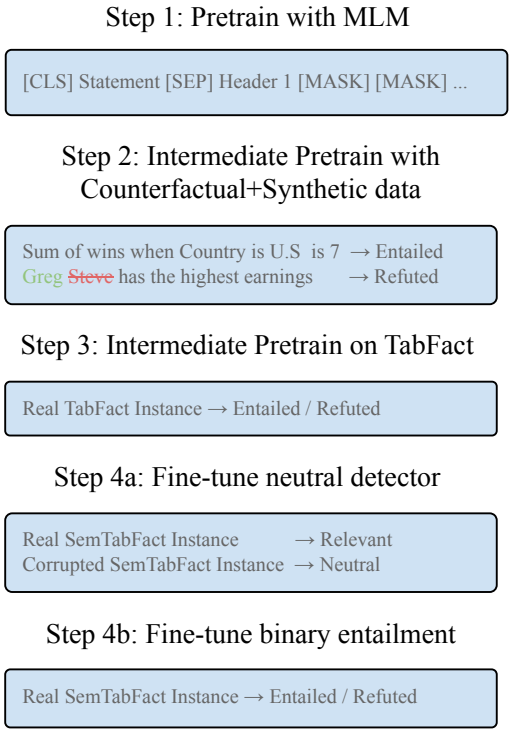


Figure 1: Overview of the training pipeline use in our system. We use intermediate pre-training on *Counterfactual+Synthetic* data (Eisenschlos et al., 2020) and then fine-tune on TABFACT (Chen et al., 2020).

## 1 Introduction

Recently, the task of Textual Entailment (TE) (Dagan et al., 2005) or Natural Language Inference (NLI) (Bowman et al., 2015) has been adapted to a setup where the premise is a table (Chen et al., 2020; Gupta et al., 2020). The Shared Task on Statement Verification and Evidence Finding with Tables (SemEval 2021 Task 9, Wang et al. (2021)) follows this line of work and provides a new dataset consisting of tables extracted from scientific articles and natural language statements written by crowd workers. In this paper, we discuss a system for tackling task A, which is a multi-class classification task that requires finding if a statement is entailed, neutral or refuted by the contents of a table. The training set contains only entailed and refuted examples and requires data augmentation to learn the neutral class. Additionally, this data set is composed of English language data and requires sophisticated contextual and numerical reasoning such as handling comparisons and aggregations.

A successful line of research on table entailment (Chen et al., 2020; Eisenschlos et al., 2020; Gupta et al., 2020) has been driven by BERT-based models (Devlin et al., 2019). These approaches reason over tables without generating logical forms to directly predict the entailment decision. Such models



*Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 423–430
Bangkok, Thailand (online), August 5–6, 2021. 

are known to be efficient on representing textual data as well reasoning over semi-structured data such as tables. In particular, TAPAS-based models (Herzig et al., 2020) that encode the table structure using additional embeddings, have been successfully used to solve binary entailment tasks with tables (Eisenschlos et al., 2020).

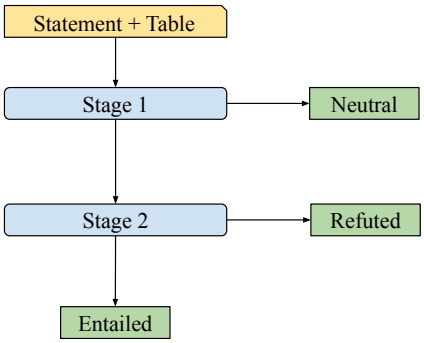


Figure 2: Overview of the complete system. Stage 1 classifies into neutral and non-neutral statements. Stage 2 into entailed and refuted. Both stages are based on binary TAPAS classifier models.

To address multi-class classification entailment we decompose the main task into two sub-tasks and use two TAPAS models as described in Figure 2. A first model classifies the statement into neutral or non-neutral, and a second into entailed or refuted. The two models are learned separately: we created artificial neutral statements to fine-tune the first model. Examples are extracted by randomly pairing statements and tables from the SEMTABFACT training set. We also generate harder examples by creating new tables from the original tables by removing columns that contain evidence to refute or entail the statement. This procedure is discussed in Section 3.

We follow Eisenschlos et al. (2020) and pre-train the two TAPAS models with a MASKLM objective (Devlin et al., 2019) and then with counterfactual and synthetic data as shown in Figure 1. We additionally fine-tune both models on the TABFACT dataset. Details are given in Section 4.

We find that our artificial neutral statement creations out-performs a majority baseline and that pre-training help for both the first and the second stage. Our best models achieve 68.03 average micro f1-score on the test set.

## 2 Related Work

**Entailment on Tables** Recognizing textual entailment (Dagan et al., 2010) has expanded from a text only task to incorporate more structured data, such knowledge graphs (Vlachos and Riedel, 2015), tables (Jo et al., 2019; Gupta et al., 2020) and images (Suhr et al., 2017, 2019).

The TABFACT dataset (Chen et al., 2020) for example, uses tables as the *premise*, or source of information to resolve whether a statement is entailed or refuted. The TAPAS architecture introduced by Herzig et al. (2020) can be used to obtain transformer-based baselines, as shown in Eisenschlos et al. (2020), by using special embeddings to encode the table structure. Zhang et al. (2020); Chen et al. (2020) also use BERT like models but obtain less accurate results due possibly to not using table-specific pre-training.

**Intermediate Pre-training** Our system relies on intermediate pre-training, a technique that appears in different forms in the literature. Language model fine-tuning (Howard and Ruder, 2018), or domain adaptive pre-training (Gururangan et al., 2020) are useful applications for domain adaptation. In a similar manner than Pruksachatkun et al. (2020), we use the *Counterfactual+Synthetic* tasks from Eisenschlos et al. (2020) to improve the discrete and numeric reasoning capabilities of the model for Table entailment.

**Synthetic data** The use of synthetic data to improve learning in NLP is ubiquitous (Alberti et al., 2019; Lewis et al., 2019; Wu et al., 2016; Leonandya et al., 2019). Salvatore et al. (2019) focus on textual entailment and probes models with synthetic examples. In semantic parsing Wang et al. (2015); Iyer et al. (2017); Weir et al. (2020) use templates to augment the training data for text-to-SQL tasks and Geva et al. (2020) do so to improve numerical reasoning, as do Eisenschlos et al. (2020) on tabular data. They also create minimal contrastive examples (Kaushik et al., 2020; Gardner et al., 2020) by automatically swapping entities in the statements by plausible alternatives that exists elsewhere in the table.

## 3 System

Our system is a two stage process that first decides whether a statement is neutral, and then decides if non-neutral statements are entailed or refuted. Both stages are implement using a binary TAPAS

| Dataset | Statements | Tables | Entailed | Refuted | Neutral |
|---|---|---|---|---|---|
| Crowdsourced Train | 4,506 | 981 | 2,818 (62.54%) | 1,688 (37.46%) | |
| Auto-generated Train | 179,345 | 1,980 | 92,136 (51.37%) | 87,209 (48.63%) | |
| Stage 1 train | 9,012 | 1,915 | 4506 (50%) | | 4506 (50%) |
| Dev | 556 | 52 | 250 (44.96%) | 213 (38.31%) | 93 (16.73%) |
| Test | 653 | 52 | 274 (41.96%) | 248 (37.98%) | 131 (20.06%) |

Table 1: SEMTABFACT (Wang et al., 2021) statistics. The training data for the first stage was created from the crowdsourced training data using artificial neutral statements created by deleting columns with evidence or swapping statements randomly. For the second stage we use the crowdsourced training data.

classifier. TAPAS (Herzig et al., 2020; Eisenschlos et al., 2020) is a variation of BERT (Devlin et al., 2019), extended with special token embeddings that give the model a notion of the row and column a token is located in and what is its numeric rank with respect to the other cells in the same column.

### 3.1 Pre-training

The original TAPAS model (Herzig et al., 2020) was pre-trained with a Mask-LM objective (Devlin et al., 2019) on tables extracted from Wikipedia. It was later found (Eisenschlos et al., 2020) that its reasoning capabilities can be improved by further training on artificial counter-factual entailment data. This led to substantial improvements on the TABFACT dataset (Chen et al., 2020), a binary table entailment task similar to SEMTABFACT. On that dataset the test set accuracy for a BERT-base-sized model improved from 69.6 to 78.6. In this work, we use models fine-tuned on TABFACT as the foundation for both stages. We also experimented with using models fine-tuned on INFOTABS (Gupta et al., 2020) and SQA (Iyyer et al., 2017) as the initial models but did not find that to achieve better accuracy. The overall pre-training strategy is described in Figure 1, where we also show how we use these checkpoints to use the two classification models described below.

### 3.2 Neutral Identification Stage

As discussed, the first stage of the system identifies if a statement is neutral. Training a system for this task is challenging as the SEMTABFACT training data does not contain neutral statements. We therefore created artificial neutral statements from two sources. Following the recommendation of the shared-task organizers, we created neutral statement by randomly pairing statements from the training set with new tables. Additionally, we created neutral statements by identifying columns that contained evidence for deciding whether a statement is entailed and then randomly removing one of these columns. Our assumption is that it should not be possible to decide whether the statement is entailed when an evidence column has been removed. We do not remove the first column of a table since that often contains the name of the row entries. In order to detect the columns containing the evidence, we trained an ensemble of 5 TAPAS QA models on the automatically generated SEMTABFACT training set. Note that the auto-generated data is generated from templates and in contrast to the crowdsourced training data does have evidence cell annotations. The models are trained to predict the binary entailment decision as well as the evidence cells at the same time, and are initialized using a TAPAS model fine-tuned on SQA. The model is trained to predict the binary entailment decision as well as the evidence cells at the same time. Our models take as input $[CLS]s_1...s_n[SEP]t_1...t_m$ where $s_1, ..., s_n$ represents the tokenized statement and $t_1, ..., t_m$ the tokenized table. For each token $t$ of the table the model outputs a score for the token to be an evidence for the statement $S$, $s(t \in S) \in \mathbb{R}$. Additionally, it outputs the scores of the entailment decision using the $[CLS]$ tokens $s([CLS]) \in \mathbb{R}$.

We use the same hyper-parameters as SQA (as discussed in Herzig et al. (2020)). We then run these models over the crowdsourced training data and for all examples where the majority of the models correctly predicts the entailment label, we extract all columns for which a majority of the ensemble predicted at least one evidence cell. Evaluation on the SEMTABFACT development set showed that the precision of this column selection process is 0.87 (87% of the extracted columns contain a reference cell). For each column, we then create a new artificial neutral example by removing the

| Stage 1 | Stage 2 | Dev | | | | Test | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | f1 2-way | | f1 3-way | | f1 2-way | | f1 3-way | |
| | | Median | Ensemble | Median | Ensemble | Median | Ensemble | Median | Ensemble |
| Majority | Majority | 51.44 | | 42.80 | | 52.41 | | 42.15 | |
| Majority | TABFACT | **78.33** $\pm$**0.45** | **80.25** | 66.40 $\pm$0.66 | 68.29 | **75.33** $\pm$**0.79** | **75.21** | 60.64 $\pm$0.65 | 60.47 |
| MASKLM | TABFACT | 74.98 $\pm$0.39 | 78.38 | 70.81 $\pm$0.66 | 72.80 | 74.32 $\pm$0.84 | 74.84 | 67.76 $\pm$0.50 | 67.67 |
| BERT | TABFACT | 75.54 $\pm$0.75 | 77.01 | 70.33 $\pm$0.59 | 72.04 | 72.73 $\pm$0.86 | 73.18 | 66.15 $\pm$0.38 | 67.70 |
| Inter | TABFACT | 75.77 $\pm$0.50 | 78.28 | **71.21** $\pm$**0.34** | 72.79 | 72.94 $\pm$0.88 | 74.01 | **67.99** $\pm$**0.78** | 67.98 |
| TABFACT (drop) | TABFACT | 78.02 $\pm$0.45 | 80.06 | 67.88 $\pm$0.87 | 69.47 | 74.92 $\pm$0.76 | 75.02 | 62.41 $\pm$0.51 | 61.67 |
| TABFACT (random) | TABFACT | 75.97 $\pm$0.73 | 78.50 | 69.62 $\pm$0.93 | 71.81 | 74.77 $\pm$0.97 | 74.67 | 66.64 $\pm$0.16 | 67.11 |
| TABFACT | BERT | 54.41 $\pm$0.51 | 55.09 | 52.00 $\pm$0.96 | 52.87 | 56.14 $\pm$0.45 | 56.49 | 53.29 $\pm$1.17 | 54.15 |
| TABFACT | MASKLM | 61.76 $\pm$1.06 | 65.09 | 58.95 $\pm$0.64 | 61.62 | 58.49 $\pm$0.15 | 60.04 | 55.89 $\pm$0.40 | 57.01 |
| TABFACT | Inter | 74.00 $\pm$0.32 | 76.68 | 68.86 $\pm$0.48 | 71.33 | 71.08 $\pm$0.78 | 72.14 | 64.94 $\pm$0.34 | 66.43 |
| TABFACT | TABFACT | 75.74 $\pm$0.18 | 78.33 | 70.76 $\pm$0.55 | **72.95** | 73.74 $\pm$0.95 | 74.01 | 67.67 $\pm$0.96 | **68.03** |

Table 2: Stage 1 and 2 ablation at 20,000 steps. majority, TABFACT (drop) and TABFACT (random) use majority voting (always predicting non-neutral), only the artificial data created by removing columns and only the random neutral statements respectively. All other models use both kinds of artificial statements.

respective column from the table. This procedure yields 651 unique new instances from the 4506 training examples. However, similarly to the first approach of pairing random statements and tables, the process is not perfect. It may happen that refuted statements continue to be refuted after removing some of the evidence, but in practice we find it beneficial to generate examples in this fashion.

The final training data is then created by taking the original crowdsourced training examples as positive examples and randomly sampling an equally-sized set of negative examples, where half of the negatives are random combinations of a statement with a table and the other half are drawn with replacement from the 651 artificial examples.

### 3.3 Entailment Stage

Training the entailment stage is rather straightforward, we train the model on the crowdsourced training data using the same hyper-parameters as Eisenschlos et al. (2020).

### 3.4 Calibration and Ensemble

As our training data for stage 1 is balanced but the development data is skewed we find it to improve accuracy if we trigger for examples with a logit larger than 4.0 (rather than 0.0). Empirically we also find the threshold of 4.0 to work better for the second stage. This could be explained by the fact that the development set has a different label distribution than the training set.

We train 5 models per stage and use them as an ensemble. The ensemble score is defined as the median of all the model scores. Using the median worked better than the mean and voting in preliminary experiments.

## 4 Experimental Setup

In this section we explain the SEMTABFACT task and dataset and give additional details about the experimental setup we used.

The SEMTABFACT dataset consists of statements and tables from the scientific literature. It is much smaller than similar datasets such as TABFACT (Chen et al., 2020) and INFOTABS (Gupta et al., 2020). It is note-worthy that the training set only contains entailed and refuted statements while the dev and test set also contain neutral (unknown) statements. The statements were written by crowd workers, which presented with 7 different types of statements were instructed to write one statement of each type. The types of statements were using aggregation, superlatives, counting, comparatives, unique counting and the usage of the caption or common-sense knowledge.

The main metric of the task is the micro f1-score computed over the statements belonging to a table. The 3-way score takes all statements into account while the 2-way score is restricted to refuted and entailed statements.

## 5 Results

Table 2 compares our system to multiple baselines. Unless stated otherwise all baselines have been trained with the same neutral data generation as discussed above and for 20,000 steps. All numbers are based on 5 independent model runs. For all setups we report the median of the individual runs as

well as the results for a system based on the median logit of the 5 models. We report error margins for the medians as half the inter-quartile range.

Looking at the first stage of the system in Table 2, we see that the system based on TABFACT is the best choice for the initialization, out-performing a simple BERT model as well as models trained with only the mask-lm and intermediate pre-training on both dev and test ensemble accuracy. However, the model trained on the intermediate data gives higher median dev and test accuracy (e.g. 72.12 vs 70.76).

With respect to the data generation we observe that any kind of neutral data generation outperforms the majority baseline. Combining the column removal and random statements yields the best results. The drop in the 2-way metrics going from the majority Stage 1 model to a learned model is expected as that metric ignores all neutral statements in the eval set.

On the second stage of our system (Table 2), we see that a TAPAS model based on TABFACT outperforms the other baselines by a bigger margin than for Stage 1. For example, a model based on only MASKLM pre-training achieves 57.01 test f1 score while the TABFACT-based modle achieves 68.03. We also found that for this stage there is a more pronounced difference between BERT and MASKLM (54.15 vs 57.01) and MASKLM and intermediate pre-training (57.01 vs 66.43).

Table 5 in the appendix shows the results for different number of steps and thresholds showing that results can be slightly tweaked by tuning them.

## 6 Analysis

Table 3 shows that the recall and precision on the *neutral* class are 37.6 and 71.4, respectively. Inspecting some instances of false positives, we find that the system is quite easily fooled; for example classifying the statement "*The lowest Factor 8 is 0.027*" as non-neutral for a table that has 5 columns labeled as Factor 1 to 5. False negatives are sometimes caused by failing to map words with typos ("paramters" vs "parameters") or abbreviations ("measurement errors" vs "ME"). Adding harder examples of neutral statements to the training set could potentially further improve the identification. We also see that the recall on the refuted class (74.3) is lower than the recall of the entailed class (85.2) while there precision values are similar.

In Table 4 we construct mutually excluded groups of the validation set. Each set is identified by specific keywords appearing in the statement, for example *Comparatives* must contain "higher", "better", "than", etc. The full list is defined in the appendix of Eisenschlos et al. (2020). We observe that comparatives and aggregations have the largest total error rates, meaning that the biggest gains in overall accuracy can be made by improving those reasoning skills. Between these two, Comparatives have the lowest in-group accuracy. Table 6 and Table 7 in the appendix show the some anaylsis for Stage 1 and Stage 2, respectively. The trend for Stage 2 is similar to the overall trend whereas Stage 1 accuracy is relatively stable across the different groups except for comparatives where the accuracy drops from 87% overall to 81%.

Another class of examples with relatively low accuracy are statements around unique counting. We find that statements containing the word *different* have an accuracy of 51.3 (vs. 71% overall) and account for 3.4 percentage points of the total error rate. Examples include "*There are six different classes*" and "*They have ten different parameters*".

| Reference \ Prediction | Non-neutral | Neutral | Recall |
|---|---|---|---|
| Non-neutral | 449 | 14 | 97.0 |
| Neutral | 58 | 35 | 37.6 |
| Precision | 88.6 | 71.4 | |
| **Reference \ Prediction** | **Refuted** | **Entailed** | **Recall** |
| Refuted | 153 | 53 | 74.3 |
| Entailed | 36 | 207 | 85.2 |
| Precision | 81.0 | 79.6 | |

Table 3: Confusion matrix for Stage 1 and Stage 2 on the development set.

| | Size | Acc | Baseline | ER |
|---|---|---|---|---|
| Overall | 100.0 | 71.0 | 45.0 | 29.0 |
| Superlatives | 15.8 | 73.9 | 50.0 | 4.1 |
| Aggregations | 13.8 | 61.0 | 46.8 | 5.4 |
| Comparatives | 12.2 | 58.8 | 47.1 | 5.0 |
| Negations | 3.1 | 82.4 | 41.2 | 0.5 |
| Multiple of the above | 5.9 | 72.7 | 63.6 | 1.6 |
| Other | 49.1 | 75.1 | 43.6 | 12.2 |

Table 4: Accuracy and total error rate (ER) for different question groups derived from the same word heuristics defined in Eisenschlos et al. (2020). The baseline is simple class majority and the error rate in each group is taken with respect to the full set. Comparatives show the biggest margin for future improvements comparing with the overall system accuracy.

## 7 Conclusion

We presented our contribution to the SEMTABFACT task (Wang et al., 2021) on table entailment. Our system consists of two stages that classify statements into non-neutral or neutral and refuted or entailed. Our model achieves 68.03 average micro f1-score on the test set. We showed that our procedure for creating artificial neutral statements improves the system over a majority baseline but results in a relatively low recall of 37.6. Other methods for creating harder neutral statements might further improve this value. In line with Eisenschlos et al. (2020), we find that pre-training on intermediate data improves the system accuracy over a system purely pre-trained with a MASKLM objective. While these initial results look promising, we find that the model struggles with statements that involve complex operations such as comparisons and unique counting.

## References


Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic QA corpora generation with roundtrip consistency. In *Proceedings of the Association for Computational Linguistics*, pages 6168–6173, Florence, Italy. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020. Tabfact: A large-scale dataset for table-based fact verification. In *Proceedings of the International Conference on Learning Representations*.

Ido Dagan, Bill Dolan, Bernardo Magnini, and Dan Roth. 2010. Recognizing textual entailment: Rationale, evaluation and approaches. *Journal of Natural Language Engineering*, 4.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Julian Eisenschlos, Syrine Krichene, and Thomas Müller. 2020. Understanding tables with intermediate pre-training. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 281–296, Online. Association for Computational Linguistics.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 1307–1323, Online. Association for Computational Linguistics.

Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting numerical reasoning skills into language models. In *Proceedings of the Association for Computational Linguistics*, pages 946–958, Online. Association for Computational Linguistics.

Vivek Gupta, Maitrey Mehta, Pegah Nokhiz, and Vivek Srikumar. 2020. Infotabs: Inference on tables as semi-structured data. In *Proceedings of the Association for Computational Linguistics*, Seattle, Washington. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the Association for Computational Linguistics*, Seattle, Washington. 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 Association for Computational Linguistics*, pages 4320–4333, Online. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the Association for Computational Linguistics*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Shankar Iyer, Nikhil Dandekar, , and Kornél Csernai. 2017. Quora question pairs.

Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. 2017. Search-based neural structured learning for sequential question answering. In *Proceedings of the Association for Computational Linguistics*, pages 1821–1831, Vancouver, Canada. Association for Computational Linguistics.

Saehan Jo, Immanuel Trummer, Weicheng Yu, Xuezhi Wang, Cong Yu, Daniel Liu, and Niyati Mehta. 2019. Aggchecker: A fact-checking system for text summaries of relational data sets. *International Conference on Very Large Databases*, 12(12):1938–1941.

Divyansh Kaushik, Eduard H. Hovy, and Zachary Chase Lipton. 2020. Learning the difference that makes A difference with counterfactually-augmented data. In *Proceedings of the International Conference on Learning Representations*, Addis Ababa, Ethiopia.

Rezka Leonandya, Dieuwke Hupkes, Elia Bruni, and Germán Kruszewski. 2019. The fast and the flexible: Training neural networks to learn to follow instructions from small data. In *Proceedings of the International Conference on Computational Semantics*, pages 223–234, Gothenburg, Sweden. Association for Computational Linguistics.

Patrick Lewis, Ludovic Denoyer, and Sebastian Riedel. 2019. Unsupervised question answering by cloze translation. In *Proceedings of the Association for Computational Linguistics*, pages 4896–4910, Florence, Italy. Association for Computational Linguistics.

Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R. Bowman. 2020. Intermediate-task transfer learning with pretrained models for natural language understanding: When and why does it work? In *Proceedings of the Association for Computational Linguistics*, Seattle, Washington. Association for Computational Linguistics.

Felipe Salvatore, Marcelo Finger, and Roberto Hirata Jr. 2019. A logical-based corpus for cross-lingual evaluation. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 22–30, Hong Kong, China. Association for Computational Linguistics.

Alane Suhr, Mike Lewis, James Yeh, and Yoav Artzi. 2017. A corpus of natural language for visual reasoning. In *Proceedings of the Association for Computational Linguistics*, pages 217–223, Vancouver, Canada. Association for Computational Linguistics.

Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. 2019. A corpus for reasoning about natural language grounded in photographs. In *Proceedings of the Association for Computational Linguistics*, pages 6418–6428, Florence, Italy. Association for Computational Linguistics.

Andreas Vlachos and Sebastian Riedel. 2015. Identification and verification of simple claims about statistical properties. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 2596–2601, Lisbon, Portugal. 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)*.

Yushi Wang, Jonathan Berant, and Percy Liang. 2015. Building a semantic parser overnight. In *Proceedings of the Association for Computational Linguistics*, pages 1332–1342, Beijing, China. Association for Computational Linguistics.

Nathaniel Weir, Prasetya Utama, Alex Galakatos, Andrew Crotty, Amir Ilkhechi, Shekar Ramaswamy, Rohin Bhushan, Nadja Geisler, Benjamin Hättasch, Steffen Eger, Ugur Cetintemel, and Carsten Binnig. 2020. Dbpal: A fully pluggable nl2sql training pipeline. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, SIGMOD '20, page 2347–2361, New York, NY, USA. Association for Computing Machinery.

Changxing Wu, Xiaodong Shi, Yidong Chen, Yanzhou Huang, and Jinsong Su. 2016. Bilingually-constrained synthetic data for implicit discourse relation recognition. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 2306–2312, Austin, Texas. Association for Computational Linguistics.

Hongzhi Zhang, Yingyao Wang, Sirui Wang, Xuezhi Cao, Fuzheng Zhang, and Zhongyuan Wang. 2020. Table fact verification with structure-aware transformer. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 1624–1629, Online. Association for Computational Linguistics.

## Appendix

The appendix contains additional results and analysis tables.

## A Results

Table 5 shows the results for different number of steps and thresholds showing that results can be slightly tweaked by tuning them.

| Steps | Thresh | f1 2-way | | f1 3-way | |
|---|---|---|---|---|---|
| | | Median | Ensemble | Median | Ensemble |
| 20K | 0.0 | 74.97 $\pm$0.35 | 76.41 | 69.96 $\pm$1.05 | 71.03 |
| 10K | 4.0 | **75.97** $\pm$**1.48** | 76.47 | 70.68 $\pm$1.36 | 72.19 |
| 10K | 0.0 | 75.84 $\pm$1.33 | 76.55 | 70.63 $\pm$1.21 | 72.27 |
| 20K | 4.0 | 75.74 $\pm$0.18 | **78.33** | **70.76** $\pm$**0.55** | **72.95** |

Table 5: Ablation of steps and threshold on the dev set.

## B Analysis

Table 6 and Table 7 show the error rate contributions of different types of statements for Stage 1 and Stage 2, respectively. The trend for Stage 2 is similar to the overall trend (Table 4) whereas Stage 1 accuracy is relatively stable across the different groups except for comparatives where the accuracy drops from 87% overall to 81%.

| | Size | Acc | Baseline | ER |
|---|---|---|---|---|
| **Overall** | 100.0 | 87.1 | 83.3 | 12.9 |
| **Superlatives** | 15.8 | 90.9 | 89.8 | 1.4 |
| **Aggregations** | 13.8 | 88.3 | 87.0 | 1.6 |
| **Comparatives** | 12.2 | 80.9 | 79.4 | 2.3 |
| **Negations** | 3.1 | 88.2 | 64.7 | 0.4 |
| **Multiple of the above** | 5.9 | 93.9 | 87.9 | 0.4 |
| **Other** | 49.1 | 86.1 | 81.7 | 6.8 |

Table 6: Accuracy and total error rate (ER) for different question groups for Stage 1.

| | Size | Acc | Baseline | ER |
|---|---|---|---|---|
| **Overall** | 100.0 | 80.2 | 54.1 | 19.8 |
| **Superlatives** | 16.9 | 80.3 | 53.9 | 3.3 |
| **Aggregations** | 14.0 | 66.7 | 54.0 | 4.7 |
| **Comparatives** | 11.6 | 71.2 | 59.6 | 3.3 |
| **Negations** | 2.4 | 90.9 | 63.6 | 0.2 |
| **Multiple of the above** | 6.2 | 75.0 | 71.4 | 1.6 |
| **Other** | 48.8 | 86.3 | 53.0 | 6.7 |

Table 7: Accuracy and total error rate (ER) for different question groups for Stage 2.