Name: TAPAS: Weakly Supervised Table Parsing via Pre-training

Paper link: https://arxiv.org/abs/2004.02349

A. **Introduction**
   - Authors introduce an approach to answer NL questions from tables without generating logical forms (as is done in **semantic parsing models**)
   - **Weakly supervised (learning with incomplete, inexact and inaccurate supervision signal) pre training** on text segments and tables crawled from Wikipedia
   - Use **BERT architecture** to encode data in the table. Uses **transfer learning**
   - Selects relevant cells + applies relevant **AGGREGATION** operations (SUM, MAX, MIN etc)

B. **Description**:
   a. Issues with Semantic Parsing models: They rely on **generated logical form** of the input user query. This logical form comes with issues like **maintaining a logical formalism** and **label bias problem.**
   b. **Contribution**:
      i. **ARCHITECTURE** : BERT architecture + table specific additional embeddings + 2 classification layers (one for cell selection and one for applying aggregation)
      ii. **PRETRAINING** : BERT's MLM (Masked Language Modelling) on millions of Wikipedia text and tables. Textual and tabular information is masked and the model is made to predict it based on context.
      iii. **FINETUNING**: End-to-end fully differentiable training that allows TaPaS to be trained under **weak supervision** which allows:
         1. **Only cell selection**
         2. **Cell selection + Aggregation**
   c. Details:
      i. **ARCHITECTURE**
         1. **EMBEDDINGS**

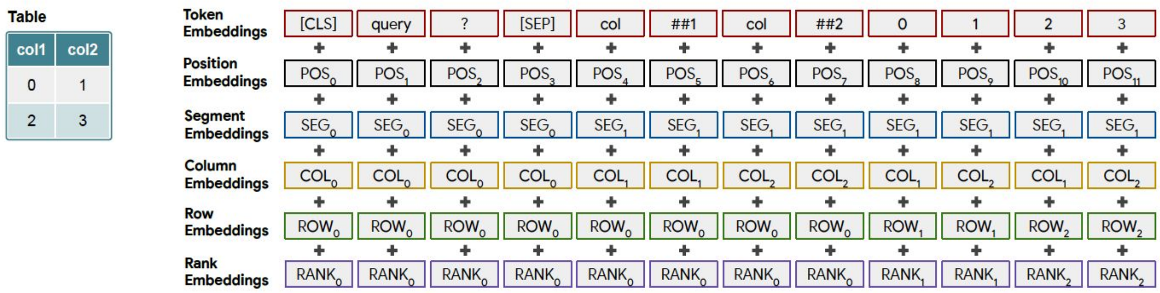
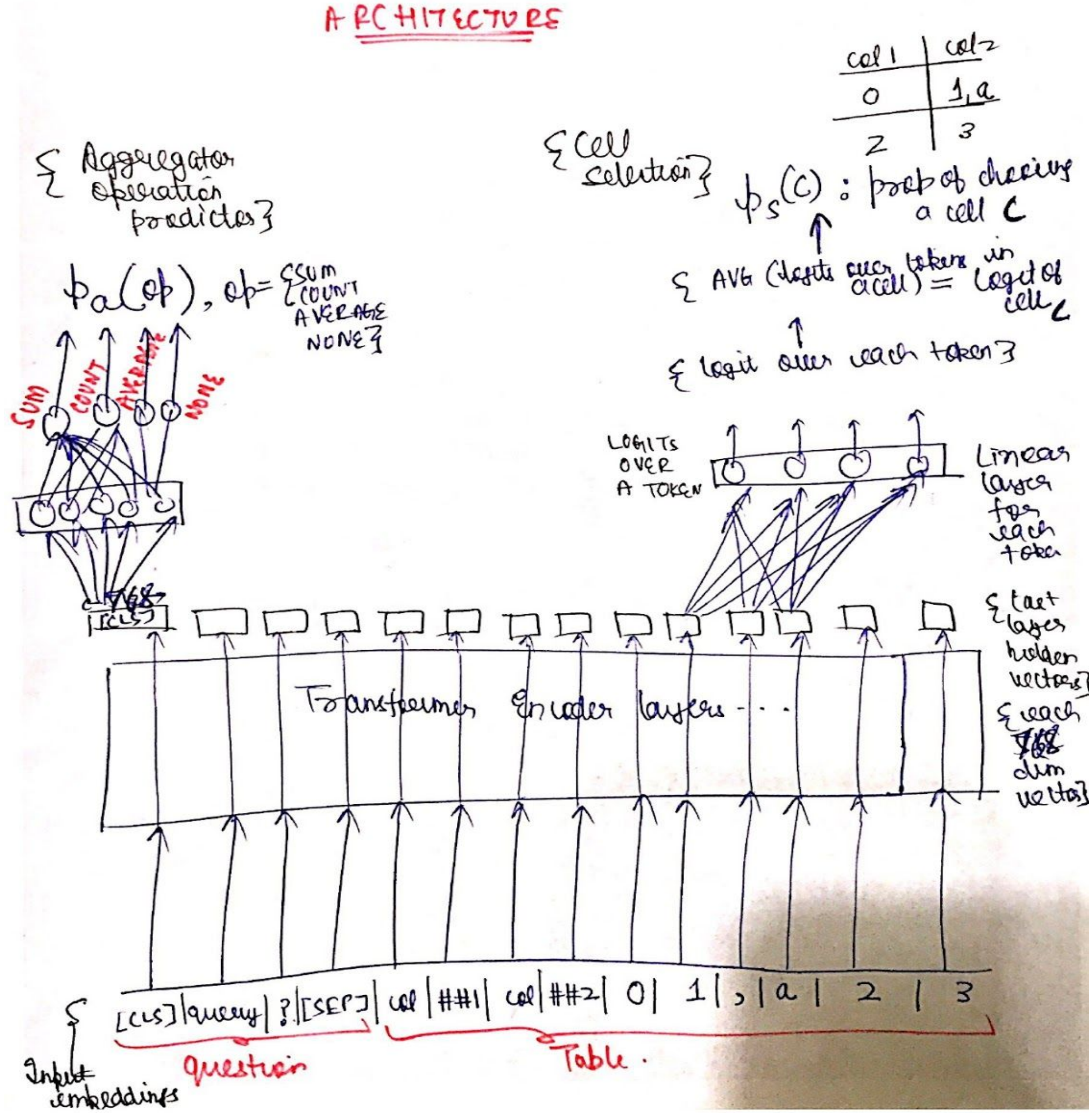


Figure 2: Encoding of the question "query?" and a simple table using the special embeddings of TAPAS. The previous answer embeddings are omitted for brevity.

2. **CELL SELECTION + AGGREGATOR SELECTOR:** We choose the relevant cells + appropriate aggregator using the architecture shown below to get the inference/ output

CELL SELECTION and AGGREGATION SELECTOR

Each cell modelled as independent Bernoulli RV. $p_s(c) > 0.5$ i.e. selected.



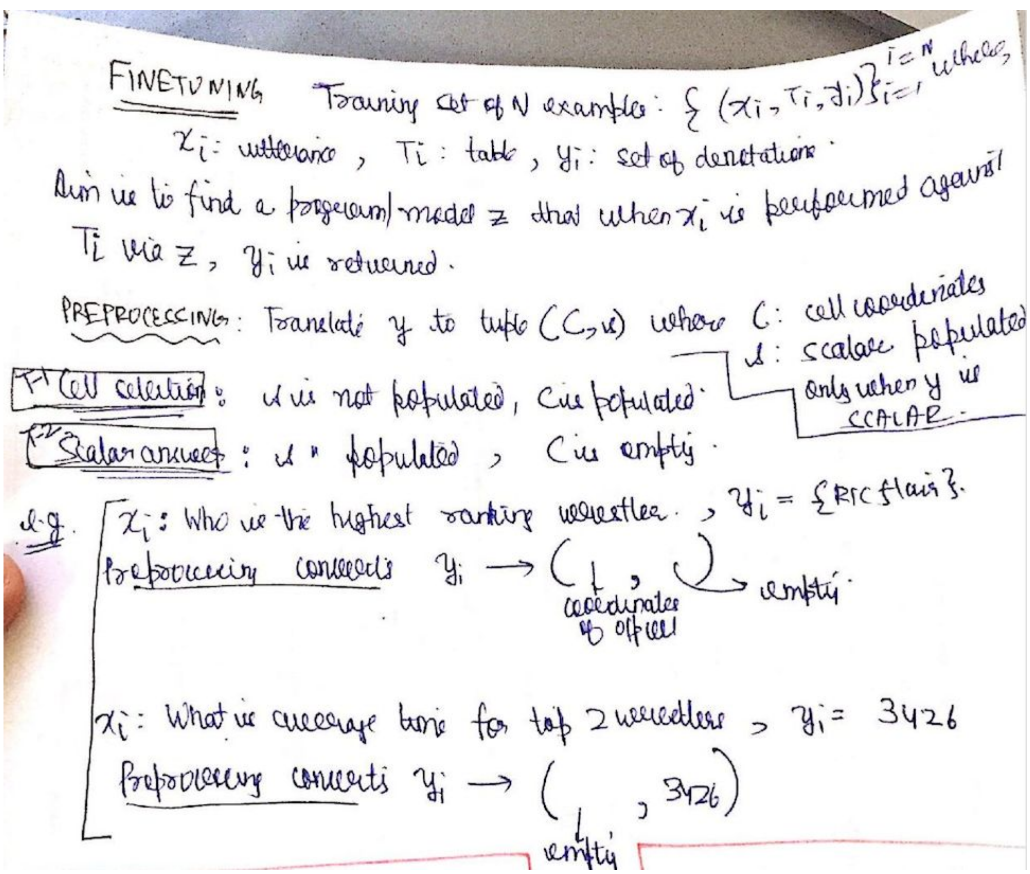
ii. **PRE TRAINING:**

1. **EXTRACTED DATA:** 6.2M tables (max 500 cells) and 21.3M relevant text snippets (table caption, article title, article description,

segment title, text of segment) from Wiki to form extracted text-table pairs.

2. **CONVERT DATA TO PRETRAINING EXAMPLES**:
   len(tokenized text + table cells) <= 128
   Randomly choose a **word piece snippet of length 8 to 16** from the associated text. To fit the table, we start by only adding the **first word of each column name and cell**. We then keep adding words turn-wise until we reach the word piece budget. **For every table we generate 10 different snippets in this way**
3. **OBJECTIVE:** MLM pre training objective with whole word + whole cell masking
4. Allows the model to learn correlations between text and table , and between cells of column and the header.
5. Used as initialization for the table parsing task.

iii. **FINETUNING**:



FINETUNING Training set of N examples: $\{(x_i, T_i, y_i)\}_{i=1}^{i=N}$, where,

x_i: utterance, T_i: table, y_i: set of denotations.

Aim is to find a program/model z that when x_i is performed against T_i via z, y_i is returned.

PREPROCESSING: Translate y to tuple (C, s) where C: cell coordinates, s: scalar populated only when y is SCALAR.

T-1 Cell selection: s is not populated, C is populated.

T-2 Scalar answer: s is populated, C is empty.

e.g. x_i: Who is the highest ranking wrestler., y_i = {Ric Flair}.

Preprocessing converts $y_i \rightarrow (\underset{\text{coordinates of cell}}{\downarrow}, \ ) \rightarrow$ empty.

x_i: What is average time for top 2 wrestlers, y_i = 3426

Preprocessing converts $y_i \rightarrow (\underset{\text{empty}}{\downarrow}, 3426)$

Task-1 Cell selection: $\{C \neq \phi, s = \phi\}$

* Train model to choose column 'col' that has highest # of cells in C.

$$\boxed{\begin{array}{lllllll} \text{Loss} & = & \text{Loss in choosing column} & + & \text{Loss in choosing cells within column} & + & \text{loss applying NONE operator} \\ & & L_{cols} & + & L_{cells} & + & L_{\cancel{cells}\ aggr} \end{array}}$$

$$= \frac{1}{|\text{columns}|} \sum_{co \in \text{column list}} CE\left(p_{col}^{(co)}, \mathbb{1}_{co=col}\right) + \frac{1}{|\text{Cells}(col)|} \sum_{c \in \text{Cells}(col)} CE\left(p_s^{(c)}, \mathbb{1}_{c \in C}\right) - \alpha \log p_a(op_0 = \text{NONE})$$

Task-2 Scalar answers (s) $\{C = \phi, s \neq \phi\}$

* y is a scalar which does not appear in table. Hence it is assumed to be an o/p of aggregation.
* OP: COUNT, AVERAGE, SUM.
* $$s_{pred} = \sum_{i=1} \hat{p}_a(op_i) * \text{compute}(op_i, p_s, T)$$ calculated using SOFT IMPLEM.

$$\hat{p}_a(op_i) = \frac{p_a(op_i)}{\sum_{i=1} p_a(op_i)}$$ except $op = \text{NONE}$.

* Calculate HUBER Loss B/W s and s_{pred} (L_{scalar})
  Also calculate $L_{aggr} = -\log\left(\sum_{i=1} p_a(op_i)\right)$ except $op_0 = \text{NONE}$

$$\boxed{\text{Loss} = L_{scalar} + \beta L_{\cancel{scalar}}}\ {}_{aggr}$$

op	compute(op, p_s, T)
COUNT	$\sum_{c \in T} p_s^{(c)}$
SUM	$\sum_{c \in T} p_s^{(c)} \cdot T[c]$
AVERAGE	$\frac{\text{compute}(\text{SUM}, p_s, T)}{\text{compute}(\text{COUNT}, p_s, T)}$

Task-3 Ambiguous answer $\{C \neq \phi, s \neq \phi\}$

↳ Value s is available in table.

↳ Let model decide whether it wants cell selection OR scalar answer based on threshold S.

$0 < S < 1$

$$\boxed{\begin{array}{ll} p_a(op_0) = p_a(\text{NONE}) \geq S, & \text{CELL SELECTION} \\ \text{ELSE}, & \text{SCALAR ANSWER} \end{array}}$$

iv. **EXPERIMENTATION**:

1. Experiment on semantic parsing datasets : **WIKITQ, SQA, WIKISQL**
2. Metric: Denotational accuracy and report median for 5 independent runs.
3. From ablation studies, it is concluded that **model pretraining** and **column/row input embeddings** are very important

v. **LIMITATION:**

1. Fails to capture very large tables which cannot fit in memory
2. Fails to capture questions which have multiple aggregations.
3. Fails to handle cases when the denotation is text that does not appear in the table

C. My Assessment/ Analysis

a. Authors present a new idea of directly extracting answers from tables without intermediate logical form.
b. Uses Transfer Learning (BERT-large architecture), pre-training on WIKI and fine tuning on specific dataset for TABULAR DATA
c. Combination of 2 tasks: Cell selection and aggregation.
d. Authors very cleverly calculate SOFT DIFFERENTIABLE ESTIMATION for each operator instead of pinpointing one single operator. I.e. s_pred is calculated using all the operators.
e. Can be fine tuned /trained via **weak supervision** where supervision signal is either **Cell selection** OR **Scalar answer.** i.e. **y => (C,s)** via code during preprocessing. The training for 2 tasks as mentioned in b. is done using **(C,s) tuple**

D. LIMITATIONS/ CONFUSIONS for me

a. Very simplistic assumption of considering cells as independent Bernoulli Random Variables.
b. Adding INDUCTIVE BIAS of choosing a column and then cells within it might not be applicable for all scenarios.
c. Does not seem to work well on complex queries across multiple columns and aggregations.
d. Aggregations are limited to AVERAGE, SUM, MAXIMUM, NONE. Not sure how to extend this.