TAPAS: Weakly Supervised Table Parsing via Pre-training

Seminar “Domain-specific Question Answering”, Summer 2022

Angelina Basova

Heidelberg University
Institute of Computer Science
Student MSc. Data and Computer Science
angelina.basova@stud.uni-heidelberg.de

June 19, 2022

Outline

Outline

1. Introduction
2. TAPAS Model
3. Pre-Training
4. Fine-tuning
5. Results

Motivation

Table

Rank	Name	No. of reigns	Combined days
1	Lou Thesz	3	3,749
2	Ric Flair	8	3,103
3	Harley Race	7	1,799
4	Dory Funk Jr.	1	1,563
5	Dan Severn	2	1,559
6	Gene Kiniski	1	1,131

Example questions

#	Question	Answer	Example Type
1	*Which wrestler had the most number of reigns?*	Ric Flair	Cell selection
2	*Average time as champion for top 2 wrestlers?*	AVG(3749,3103)=3426	Scalar answer
3	*How many world champions are there with only one reign?*	COUNT(Dory Funk Jr., Gene Kiniski)=2	Ambiguous answer
4	*What is the number of reigns for* Harley Race?	7	Ambiguous answer
5	*Which of the following wrestlers were ranked in the bottom 3?*	{Dory Funk Jr., Dan Severn, Gene Kiniski}	Cell selection
	Out of these, who had more than one reign?	Dan Severn	Cell selection

Figure: A table (left) with corresponding example questions (right) [**Herzig2020**]

- answer questions with data from tables
- answer questions by aggregating data from tables
- answer questions based on a previous question

BERT Encoders

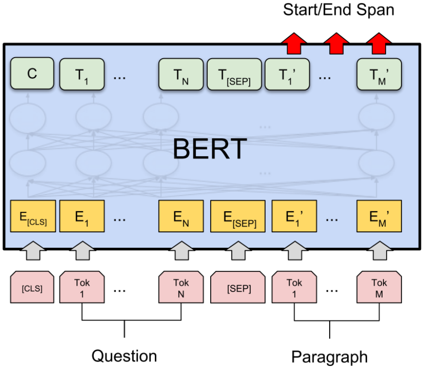


Figure: BERT architecture for question answering [**Devlin2019**]

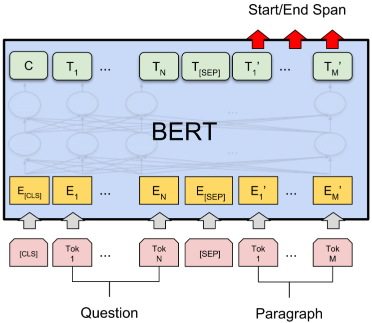
Weak supervision

- form of supervised learning
- heuristically generating training data with external knowledge bases, patterns/rules, or other classifiers
- programmatically generating training data—or, **programming training data** [**AlexRatner2019**]

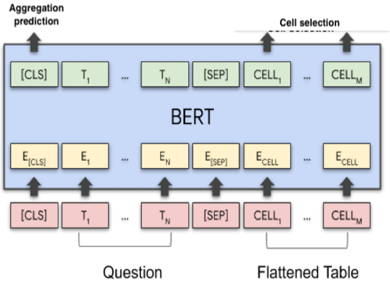
Outline

1. Introduction
2. TAPAS Model
3. Pre-Training
4. Fine-tuning
5. Results

Extension of BERT to TAPAS



(a) BERT architecture for question answering **[Devlin2019]**



(b) TAPAS architecture for table question answering **[Muller2020]**

Architecture

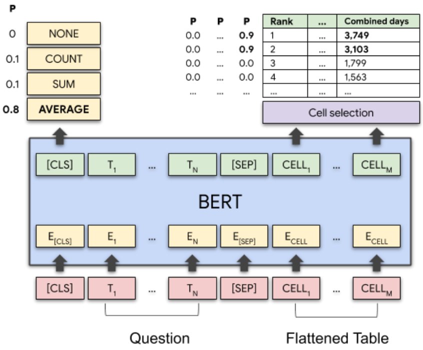


Figure: TAPAS architecture and tasks [**Muller2020**]

Input Embeddings

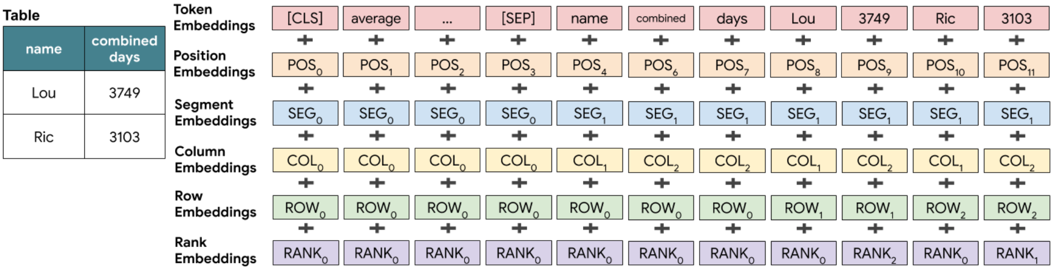


Figure: Encoding of the questions average _ and a simple table using the special embeddings of TAPAS [**Herzig2020**]

Tasks

1. aggregation operator prediction
   - supported aggregation operations (op): SUM, COUNT, AVERAGE, NONE
   - operator selected by linear layer followed by softmax on top of the final hidden vector of the first token
   - linear layer denoted as $p_a(op)$
2. cell selection
   - cells modelled as independent Bernoulli variables
3. inference
   - predict most likely aggregation operator together with subset of cells
   - select table cells with probability > 0.5

Outline

Pre-Training Data

pretrain on 6.2M tables from Wikipedia.

- 3.3M of class Infobox
- 2.9M of class WikiTable
- consider tables with max. 500 cells
- horizontal tables with a header row with column names
- as proxy for questions extract data about the table (table caption, article title, article description, segment title, text of segment the table occurs)
- pre-trainig examples format: (extracted text, table)

Pre-Training

objective: (Whole Word) Masked language model

- restrict word piece sequence length to 128
- word piece sequence length = length of tokenized text and table cells
- use whole word masking for the text
- use whole cell masking to the tables

Outline

1. Introduction
2. TAPAS Model
3. Pre-Training
4. Fine-tuning
5. Results

Overview

Training set

A training set for table parsing in a weakly supervised setup is a set of N examples:

$$\left\{(x_i, T_i, y_i)\right\}_{i=1}^{N}$$

where x_i is utterance, T_i table and y_i set of denotations

Goal: learn a model that maps a new utterance x to a program z, such that when z is executed against the corresponding table T , it yields the correct denotation y.
learn model m: $x \rightarrow z$
$z(T) = y$

- program z comprises a subset of the table cells and an optional aggregation operator.
- table T maps a table cell to its value

Overview

- for each example: translate set of denotations y to a tuple (C, s),
  $y \to (C, s)$
  where C are cell coordinates and s is a scalar
- scalar s is only populated when y is a single scalar
- guide training according to (C, s)

Cell selection

- $(C, s) = (C, \emptyset)$
- train the model to select the cells in C

Scalar answer

- $(C, s) = (\emptyset, s)$
- train model to predict an aggregation over the table cells that amounts to s

Overview

- for each example: translate set of denotations y to a tuple (C, s),
  $y \rightarrow (C, s)$
  where C are cell coordinates and s is a scalar
- scalar s is only populated when y is a single scalar
- guide training according to (C, s)

Cell selection

- $(C, s) = (C, \emptyset)$
- train the model to select the cells in C

Scalar answer

- $(C, s) = (\emptyset, s)$
- train model to predict an aggregation over the table cells that amounts to s

Cell selection

goal: train model to select relevant cells

- $(C, s) = (C, \emptyset)$
- search done in set of cell coordinates (C)
- scalar s not populated
- y is mapped to a subset of the table cell coordinates C

Procedure

use hierarchical model

(1) select a single column

(2) select cells from within that column

Cell selection

Procedure

use hierarchical model

(1) select a single column

  select column with highest number of cells in C

  if C is empty **then**

    select the additional empty column corresponding to empty cell selection

(2) select cells from within that column

- select cells $C \cap col$

Cell selection

loss components:

1. **average binary cross-entropy loss over column selections**

$$\mathcal{J}_{\text{columns}} = \frac{1}{|\text{Columns}|} \sum_{\text{co}\in\text{Columns}} \text{CE}(p_{\text{col}}^{(\text{co})}, \mathbb{1}_{\text{co}=\text{col}})$$

   where the set of columns Columns includes the additional empty column, CE($\cdot$) is the cross entropy loss

2. **average binary cross-entropy loss over column cell selections**

$$\mathcal{J}_{\text{cells}} = \frac{1}{|\text{Cells}(\text{col})|} \sum_{c\in\text{Cells}(\text{col})} \text{CE}(p_{\text{s}}^{(c)}, \mathbb{1}_{c\in C})$$

   where Cells(col) is the set of cells in the chosen column

Cell selection

3. **aggregation loss**

$$\mathcal{J}_{\text{aggr}} = -\log p_{\text{a}}(op_0)$$

total loss:

$$\mathcal{J}_{CS} = \mathcal{J}_{columns} + \mathcal{J}_{cells} + \alpha \mathcal{J}_{aggr}$$

where α is scaling hyperparameter

Overview

- for each example: translate set of denotations y to a tuple (C, s),
  $y \rightarrow (C, s)$
  where C are cell coordinates and s is a scalar
- scalar s is only populated when y is a single scalar
- guide training according to (C, s)

Cell selection

- $(C, s) = (C, \emptyset)$
- train the model to select the cells in C

Scalar answer

- $(C, s) = (\emptyset, s)$
- train model to predict an aggregation over the table cells that amounts to s

scalar answer

goal: train the model to select aggregation function over cells

- set of cell coordinates is empty ($C \in \emptyset$)
- scalar s is populated

Training set

A training set for table parsing in a weakly supervised setup is a set of N examples:

$$\left\{(x_i, T_i, y_i)\right\}_{i=1}^{N}$$

y is scalar s that does not appear in the Table

Scalar answer

$\mathbf{compute}(\mathbf{op}, \mathbf{p_s}, \mathbf{T})$: estimation for an operator, given the token selection probabilities ($\mathbf{p_s}$) and the table values ($\mathbf{T}$)

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

Table: Aggregation operators soft implementation. Note that probabilities p_s outside of the column selected by the model are set to 0.

Scalar answer

compute expected results

$$s_{pred} = \sum_{i=1} p_a(op_i) * compute(op_i, p_s, T)$$

where $p_a(op_i) = \frac{p_a(op_i)}{\sum_{i=1} p_a(op_i)}$ is a probability distribution normalized over aggregation operators excluding NONE.

Scalar answer

1. **scalar answer loss**

$$\mathcal{J}_{\text{scalar}} = \begin{cases} 0.5 \cdot a^2 & a \leq \delta \\ \delta \cdot a - 0.5 \cdot \delta^2 & \text{otherwise} \end{cases}$$

   where $a = |s_{pred} - s|$, and δ is a hyperparameter

2. **answer loss**

$$\mathcal{J}_{\text{aggr}} = -\log(\sum_{i=1} p_{\text{a}}(op_i))$$

Scalar answer

total loss

$$\mathcal{J}_{SA} = \mathcal{J}_{aggr} + \beta \mathcal{J}_{scalar}$$

where β is scaling hyperparameter,

Ambiguous answer

An ambiguous answer is a scalar answer s that also appears in the Table and in some cases the question implies aggregation and in other cases a table cell should be predicted

Let model dynamically choose supervision(cell selection or scalar answer):

Select cell selection:

$$p_a(op_0) \geq S$$

where $0 < S < 1$ is a threshold hyperparameter

Select scalar answer:

$$p_a(op_0) < S$$

Outline

Results

Evaluation datasets: WIKITQ, SQA, WIKISQL

	SQA (SEQ)		WIKISQL		WIKITQ	
all	39.0		84.7		29.0	
-pos	36.7	-2.3	82.9	-1.8	25.3	-3.7
-ranks	34.4	-4.6	84.1	-0.6	30.7	+1.8
-{cols,rows}	19.6	-19.4	74.1	-10.6	17.3	-11.6
-table pre-training	26.5	-12.5	80.8	-3.9	17.9	-11.1
-aggregation	-		82.6	-2.1	23.1	-5.9

Figure: Table 6: Dev accuracy with different embeddings removed from the full model: positional (pos), numeric ranks (ranks), column (cols) and row (rows) [**Herzig2020**]

1. pre-training on masked language model task and using column/row embeddings significantly improve accuracy
2. removing the scalar answer and aggregation losses (i.e., setting JSA=0) results into accuracy drops for both datasets

Limitations

Error phenomena:

- 16% of the cases the gold denotation has a textual value that does not appear in the table
- 13% of the cases TAPAS selected no cells, which suggests introducing penalties for this behaviour
- 10% of the cases require complex temporal comparisons which could also not be parsed with a rich formalism such as SQL
  "what country had the most cities founded in the 1830's?"

Major limitations:

- Fail to capture large tables or databases
- multiple aggregations not possible
  "number of actors with an average rating higher than 4"

Questions

Questions

References I