Tables as Semi-structured Knowledge for Question Answering

ACL’16

Sujay Kumar Jauhar¹   Peter D. Turney²   Eduard Hovy¹

¹Carnegie Mellon University

²Allen Institute for Artificial Intelligence

Presented By : Dushyanta Dhyani
Outline

1. Problem Definition
2. Motivation
3. Tables - Structure & Formalism
4. Dataset Generation
5. Solving MCQ's
6. Evaluation & Results
7. Conclusion
Outline

1. Problem Definition
2. Motivation
3. Tables - Structure & Formalism
4. Dataset Generation
5. Solving MCQ’s
6. Evaluation & Results
7. Conclusion
Using tables for creating a Question Answering system.

This includes:

- Using tables for creating MCQ training Data.
- Building a QA model using tables
Outline

1. Problem Definition
2. Motivation
3. Tables - Structure & Formalism
4. Dataset Generation
5. Solving MCQ’s
6. Evaluation & Results
7. Conclusion
Motivation
Why Tables?

Primary Question: Why do we need tables?
Motivation
Why Tables?

Primary Question: Why do we need tables?

Subsequent Question: What are the alternatives to tables?
Motivation

Why Tables?

Primary Question: Why do we need tables?

Subsequent Question: What are the alternatives to tables?

WEB
Motivation

Why Tables?

**Primary Question**: Why do we need tables?

**Subsequent Question**: What are the alternatives to tables?

- WEB
- Knowledge Bases
Motivation

Why Tables?

Primary Question: Why do we need tables?

Subsequent Question: What are the alternatives to tables?

- WEB
- Knowledge Bases
- Databases
Motivation

Why Tables?

Primary Question: Why do we need tables?

Subsequent Question: What are the alternatives to tables?

- WEB
- Knowledge Bases
- Databases
Motivation

Why Tables?

WEB

- Available in large volumes.
- Contain a large number of entities (And also continuously increasing at a rapid scale)
- Thus difficult to reason with and interpret.
- Ensuring compositionality is difficult.
- Use Information Retrieval based methods.
Motivation

Why Tables?

Knowledge Bases

- Structured.
- Fixed Schema. Large number of entities and relations.
- Coverage is high.
- Compositionality is low.
Motivation

Why Tables?

Databases

- Few entities and relations.
- Ensure high compositionality.
- Extremely low coverage
Motivation

Why Tables?

TABLES!!!
Motivation

Why Tables?

TABLES!!!
Best of all the worlds
Motivation

Why Tables?

TABLES!!!
Best of all the worlds
Semi-structured
TABLES!!!
Best of all the worlds
Semi-structured
Can ensure compositionality
Motivation for carrying out the given work

- A few (77 out of 108) 4th grade science exam questions from the Regent’s dataset were manually annotated for alignment to tables.
- Built a QA System to solve the Aristo challenge.
- Rivaled the best solvers that AI2 had built till then.
Outline

1. Problem Definition
2. Motivation
3. Tables - Structure & Formalism
4. Dataset Generation
5. Solving MCQ’s
6. Evaluation & Results
7. Conclusion
Tables I
What are they?

- Semi-structured data

<table>
<thead>
<tr>
<th>Phase Change</th>
<th>Initial State</th>
<th>Final State</th>
<th>Form of Energy Transfer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Melting</td>
<td>causes a</td>
<td>solid</td>
<td>to change into a</td>
</tr>
<tr>
<td>Vaporization</td>
<td>causes a</td>
<td>liquid</td>
<td>to change into a</td>
</tr>
<tr>
<td>Condensation</td>
<td>causes a</td>
<td>gas</td>
<td>to change into a</td>
</tr>
<tr>
<td>Sublimation</td>
<td>causes a</td>
<td>solid</td>
<td>to change into a</td>
</tr>
</tbody>
</table>

- Contain general knowledge data.
- Cells contain free form text. Can be used independently as raw text.
- Thus also help in comparing the proposed model to an information retrieval based model.
What are they?

- However, each row exhibits a well-defined recurring filler pattern.
- Help in providing rich alignment annotations.
- **Content Columns**: Columns with explicitly specified headers. These columns contain concepts, entities, processes, etc.
- **Filler columns**: Columns with no headers. Contain a recurring pattern.
The topics for creating these tables were determined from the Training set of 2 evaluation datasets: **Regents & Monarch**

**Regents** Dataset:
- Public Dataset
- Single table containing 108 questions.
- Questions revolve around 4th grade science.

**Monarch** Dataset:
Unreleased Dataset also related to 4th grade science.
Tables for this task?

- AI2’s Aristo Tablestore
- 65 hand crafted tables organized by topics.
- Table Topics - Bounded & Unbounded
- Total: 3851 facts (Row=fact)

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Facts</th>
<th>Table Name</th>
<th>Facts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orbital Event Daylight Hours</td>
<td>4</td>
<td>Country Hemispheres</td>
<td>267</td>
</tr>
<tr>
<td>Phase Transitions</td>
<td>6</td>
<td>Device Energy Conversion</td>
<td>65</td>
</tr>
<tr>
<td>Average Weights of Animals</td>
<td>1225</td>
<td>Wordnet Definitions</td>
<td>2467</td>
</tr>
</tbody>
</table>
• Amazon Mechanical Turk!!
• Structural constraints are imposed while creating questions.
• In case of insufficient choices for an answer, Turkers provide answers on their own.
<table>
<thead>
<tr>
<th>Phase Change</th>
<th>Initial State</th>
<th>Final State</th>
<th>Form of Energy Transfer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Melting</td>
<td>solid</td>
<td>liquid</td>
<td>adding heat</td>
</tr>
<tr>
<td>Vaporization</td>
<td>liquid</td>
<td>gas</td>
<td>adding heat</td>
</tr>
<tr>
<td>Sublimation</td>
<td>solid</td>
<td>gas</td>
<td>adding heat</td>
</tr>
<tr>
<td>Freezing</td>
<td>liquid</td>
<td>solid</td>
<td>removing heat</td>
</tr>
<tr>
<td>Deposition</td>
<td>gas</td>
<td>solid</td>
<td>removing heat</td>
</tr>
<tr>
<td>Condensation</td>
<td>gas</td>
<td>liquid</td>
<td>removing heat</td>
</tr>
</tbody>
</table>

**Answer**

Cells that could be a part of the question

Distractors / Alternate Answers
TabMCQ

How efficient is it?

<table>
<thead>
<tr>
<th>Task</th>
<th>Avg. Time (s)</th>
<th>$/hour</th>
<th>% Reject</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rewrite</td>
<td>345</td>
<td>2.61</td>
<td>48</td>
</tr>
<tr>
<td>Paraphrase</td>
<td>662</td>
<td>1.36</td>
<td>49</td>
</tr>
<tr>
<td>Add choice</td>
<td>291</td>
<td>2.47</td>
<td>24</td>
</tr>
<tr>
<td>Write new</td>
<td>187</td>
<td>5.78</td>
<td>38</td>
</tr>
<tr>
<td>TabMCQ</td>
<td>72</td>
<td>5.00</td>
<td>2</td>
</tr>
</tbody>
</table>
For generating MCQ data, given a cell in a table, we generated question from cell’s row and candidate answers from the cell’s column.

Now the task at hand is reversed.

Given question-answer pairs, find a cell that best confirms the assertion.
Log-linear Model that assigns a score to every cell in every table according to their relevance to each question-answer pair

Formally, Given:

- \( Q = \{ q_1, q_2, \ldots, q_N \} \) - set of Questions
- \( A_n = \{ a^1_n, a^2_n, \ldots, a^k_n \} \) for a given question \( q_n \)
- \( T = \{ T_1, T_2, \ldots, T_M \} \) - Set of Tables

\[
\log p(t_{ij}^m | q_n, a^k_n; A_n, T) = \sum_d \lambda_d f_d(q_n, a^k_n, t_{ij}^m; A_n, T) - \log Z
\]

where,

- \( \lambda_d \) is a set of parameters to be learned.
- \( f_d(\ldots) \) is a set of features
- \( Z = \sum_{m,i,j} \exp \left( \sum_d \lambda_d f_d(q_n, a^k_n, t_{ij}^m; A_n, T) \right) \) is the partition function.
- \( t_{ij}^m = Cell(i, j) \) in the \( m^{th} \) table.
We try to assess the significance/salience of each cell for a given Q-A pair.

Relevant cells, if present, assert to the hypothetical claim/fact made by a given QA pair.

During Inference, we decide upon the answer choice that gets the maximum score out of all the rows, i.e.

\[ a^*_n = \arg \max_{a^k_n} \max_{m,i} \sum_{j} \sum_{d} \lambda_d f_d(q_n, a^k_n, t^{ij}_m; A_n, T) \]
For training, apart from the features, we also need predictor values. Since we are dealing with probability of cell relevance, the predictor value should numerically quantify the alignment of rows to questions and columns to answer choices.

Authors selected the following methodology from a few (tested) other scoring heuristics.

- For a **correct** QA pair,
  - Assign a score of **1.0** to cell that exactly answers the question.
  - If a cell does not answer the question, but is used in the construction of the question, assign a score of **0.5**
  - Otherwise assign a score of **0.0**

- For an **incorrect** QA pair:
  - With a probability of 1%, assign a score of **0.1** to random cells from all the (other) tables that have no alignment to the given QA pair.
Cross-Entropy Loss Function

\[ L(\lambda) = \sum_{q_n} \sum_{m,i,j} p(t^*_{ij}|q_n, a^k_n; T) \cdot \log p(t^*_{ij}|q_n, a^k_n; A_n, T) \]

where \( p(t^*_{ij}|q_n, a^k_n; T) \) is the normalized probability of the true alignment scores

Adaptive Gradient Descent (AdaGrad) is used to minimize the above loss.
Several features are used at various granularity levels: Table, Row, Column, Cell.

Certain features are supplemented with their soft matching variants.

Features that are assigned high weights ($\lambda_d$) during training are then used to form a compact FRETS model.
<table>
<thead>
<tr>
<th>Level</th>
<th>Feature</th>
<th>Description</th>
<th>Intuition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>Table score</td>
<td>Ratio of words in ( t ) to ( q + a )</td>
<td>Topical consistency</td>
</tr>
<tr>
<td></td>
<td>†TF-IDF table score</td>
<td>Same but TF-IDF weights</td>
<td>Topical consistency</td>
</tr>
<tr>
<td>Row</td>
<td>Row-question score</td>
<td>Ratio of words in ( r ) to ( q )</td>
<td>Question align</td>
</tr>
<tr>
<td></td>
<td>Row-question w/o focus score</td>
<td>Ratio of words in ( r ) to ( q - (a_r + q_f) )</td>
<td>Question align</td>
</tr>
<tr>
<td></td>
<td>Header-question score</td>
<td>Ratio of words in ( h ) to ( q )</td>
<td>Prototype align</td>
</tr>
<tr>
<td>Column</td>
<td>Column overlap</td>
<td>Ratio of elements in ( c ) and ( A )</td>
<td>Choices align</td>
</tr>
<tr>
<td></td>
<td>Header answer-type match</td>
<td>Ratio of words in ( c_h ) to ( a_f )</td>
<td>Choices hypernym align</td>
</tr>
<tr>
<td></td>
<td>Header question-type match</td>
<td>Ratio of words in ( c_h ) to ( q_f )</td>
<td>Question hypernym align</td>
</tr>
<tr>
<td>Cell</td>
<td>†Cell salience</td>
<td>Salience of ( s ) to ( q + a )</td>
<td>QA hypothesis assert</td>
</tr>
<tr>
<td></td>
<td>†Cell answer-type entailment</td>
<td>Entailment score between ( s ) and ( a_f )</td>
<td>Hypernym-hyponym align</td>
</tr>
<tr>
<td></td>
<td>Cell answer-type similarity</td>
<td>Avg. vector sim between ( s ) and ( a_f )</td>
<td>Hypernym-hyponym sim.</td>
</tr>
</tbody>
</table>
TF-IDF scores are computed for all words in all the tables.

Each table is treated as a unique document.

"...At run-time we discount scores by table length as well as length of the QA pair under consideration to avoid disproportionately assigning high scores to large tables or long MCQs."
**Question & Answer Focus:** Parse questions to find question type & desired answer type.

e.g. Given "What form of energy is required to convert water from a liquid to a gas?"

**Answer Type:** Form of Energy

Based on question patterns in the data, a rule-based parser (using a set of hand-coded regular expressions) is used to find answer-types from queries.

This parser is designed such that it produces answer types only in high confidence situations.

Similar operation is performed for questions.
- **Salience** for a pair of strings is evaluated by computing Point-wise Mutual Information (PMI) statistics between this pair from a large corpus.
- The higher the salience score, the higher the relevance of the cell for the QA hypothesis.
- **Entailment** refers to the confidence in the truthfulness of one string, given that another string is **true**.
- Features used for evaluating entailment - Overlap, Paraphrase probability, lexical entailment likelihood and ontological relatedness.
## Features - Soft Matching & Compactness

<table>
<thead>
<tr>
<th>Level</th>
<th>Feature</th>
<th>S-Var</th>
<th>Cmpct</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>Table score</td>
<td>◊</td>
<td>●</td>
</tr>
<tr>
<td></td>
<td>†TF-IDF table score</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Row</td>
<td>Row-question score</td>
<td>◊</td>
<td>●</td>
</tr>
<tr>
<td></td>
<td>Row-question w/o focus score</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Header-question score</td>
<td>◊</td>
<td></td>
</tr>
<tr>
<td>Column</td>
<td>Column overlap</td>
<td>◊</td>
<td>●</td>
</tr>
<tr>
<td></td>
<td>Header answer-type match</td>
<td>◊</td>
<td>●</td>
</tr>
<tr>
<td></td>
<td>Header question-type match</td>
<td>◊</td>
<td></td>
</tr>
<tr>
<td>Cell</td>
<td>†Cell salience</td>
<td>◊</td>
<td>●</td>
</tr>
<tr>
<td></td>
<td>†Cell answer-type entailment</td>
<td></td>
<td>●</td>
</tr>
<tr>
<td></td>
<td>Cell answer-type similarity</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Most of the features used, primarily follow a bag of words based model.

Thus a hard overlap feature would define a score between two bag of words $S_1$ & $S_2$ as $|S_1 \cap S_2|/|S_1|$

However, it’s quite possible that a new QA pair might not contain the terms in the vocabulary of the model.

Thus a soft-matching counterpart of the above features is proposed.
Similarity between words in an embedding space is used as an alternate to $|S_1 \cap S_2|$

Thus the corresponding soft overlap score would be:

$$\frac{1}{|S_1|} \sum_{w_i \in S_1} \max_{w_j \in S_2} \text{sim}(\vec{w}_i, \vec{w}_j)$$

Word embeddings are obtained by training on 300 million words of the **WMT-2011** shared task and were improved by retrofitting [1] them to PPDB - paraphrase database [2]
Outline

1. Problem Definition
2. Motivation
3. Tables - Structure & Formalism
4. Dataset Generation
5. Solving MCQ's
6. Evaluation & Results
7. Conclusion
Evaluation Dataset:

- **Regents** - publicly available; 129 MCQ’s
- **Monarch** - Unreleased; 250 MCQ’s
- **Elementary School Science Questions (ESSQ)** - Public dataset; 855 MCQ’s

Since the training tables of **Regents & Monarch** were used in the construction of the Aristo Tablestore, only the testing set was used for evaluation. All tables from ESSQ were used.
**Information Retrieval Method**
- Uses Lucene search engine.
- Table structure is ignored. The rows are used as simple text.
- Top results from Lucene are then used to rank the different answer choices.

**Markov Logic Networks (MLN) [3]:**
- Highly structured models.
- Results directly cited from [4].
- Thus results only available for Regent’s dataset.
3 different combinations of training data were used:

- Aristo tables only constructed for Regent’s dataset. Total = 40 tables.
- Aristo tables only constructed for Monarch dataset. Total = 25 tables.
- All the above. Total = 65 tables.

For the Lucene baseline, they also experimented with the Waterloo corpus that contain $5 \times 10^{10}$ words
## Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Data</th>
<th>Regents Test</th>
<th>Monarch Test</th>
<th>ESSQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lucene</td>
<td>Regents Tables</td>
<td>37.5</td>
<td>32.6</td>
<td>36.9</td>
</tr>
<tr>
<td></td>
<td>Monarch Tables</td>
<td>28.4</td>
<td>27.3</td>
<td>27.7</td>
</tr>
<tr>
<td></td>
<td>Regents+Monarch Tables</td>
<td>34.8</td>
<td>35.3</td>
<td>37.3</td>
</tr>
<tr>
<td></td>
<td>Waterloo Corpus</td>
<td>55.4</td>
<td>51.8</td>
<td>54.4</td>
</tr>
<tr>
<td>MLN (Khot et al., 2015)</td>
<td>-</td>
<td>47.5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>FRETS (Compact)</td>
<td>Regents Tables</td>
<td>60.7</td>
<td>47.2</td>
<td>51.0</td>
</tr>
<tr>
<td></td>
<td>Monarch Tables</td>
<td>56.0</td>
<td>45.6</td>
<td>48.4</td>
</tr>
<tr>
<td></td>
<td>Regents+Monarch Tables</td>
<td>59.9</td>
<td>47.6</td>
<td>50.7</td>
</tr>
<tr>
<td>FRETS</td>
<td>Regents Tables</td>
<td>59.1</td>
<td><strong>52.8</strong></td>
<td>54.4</td>
</tr>
<tr>
<td></td>
<td>Monarch Tables</td>
<td>52.9</td>
<td>49.8</td>
<td>49.5</td>
</tr>
<tr>
<td></td>
<td>Regents+Monarch Tables</td>
<td>59.1</td>
<td>52.4</td>
<td><strong>54.9</strong></td>
</tr>
</tbody>
</table>
Without the **Waterloo Corpus**, the Lucene baseline has scores far less than FRETS.

Thus FRETS is able to outperform an unstructured model given small amount of data.

In certain cases, FRETS performs significantly better than MLN, a model that is highly structured, with a much complex data formalism.

In the **Lucene** baseline, the performance decreases in case of **Monarch** or **Regents+Monarch** tables as training data. However, that is not the case with FRETS or not very significant in case of FRETS(Compact). This shows the addition of tables (presumably not useful) does not affect the performance of FRETS.
A set of features were removed and the model’s performance was analyzed.

<table>
<thead>
<tr>
<th>Model</th>
<th>REG</th>
<th>MON</th>
<th>ESSQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRETS</td>
<td>59.1</td>
<td>52.4</td>
<td>54.9</td>
</tr>
<tr>
<td>w/o tab features</td>
<td>59.1</td>
<td>47.6</td>
<td>52.8</td>
</tr>
<tr>
<td>w/o row features</td>
<td>49.0</td>
<td>40.4</td>
<td>44.3</td>
</tr>
<tr>
<td>w/o col features</td>
<td>59.9</td>
<td>47.2</td>
<td>53.1</td>
</tr>
<tr>
<td>w/o cell features</td>
<td>25.7</td>
<td>25.0</td>
<td>24.9</td>
</tr>
<tr>
<td>w/o ◊ features</td>
<td>62.2</td>
<td>47.5</td>
<td>53.3</td>
</tr>
</tbody>
</table>
Conclusion

- Tables can be used as knowledge bases for QA.
- A connected framework is proposed for both dataset generation and MCQ solving.
- Trade off between structure and reasoning ability of tables is efficiently handled.
- A large dataset of more than 9000 MCQ questions is publicly released.


Peter Clark, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Turney, and Daniel Khashabi. Combining retrieval, statistics, and inference to answer elementary science questions. 

*30th AAAI, 2016.*