DAGOBAH
An End-to-End Context-Free Tabular Data Semantic Annotation System

Yoan Chabot
Orange Labs
@yoan_chabot

Thomas Labbé
Orange Labs
@tau_labbe

Jixiong Liu
Orange Labs

Raphaël Troncy
EURECOM
@rtroncy
Context & Goals

- Design a **semantic engine** able to query (semi-)structured data

  I want to have precise and relevant answers to my queries expressed in natural language, without having to know the target database(s) model(s)

- We focus on tabular data: annotate the content and structure of tabular data for searching and recommending datasets
Tabular Data to Knowledge Graph Matching

- **Goals**

- **1st step:** preprocessing to identify tables characteristics (orientation, key-column…)
- **2nd step:** annotations workflows
  - Method 1: Baseline lookups
  - Method 2: Embedding approach
- We focus on the CTA and CEA tasks
  CPA processing: list of properties associated to entities pairs, plus majority voting
Preprocessing (new homogeneity factor)

Datatable corpus (CSV, TSV, HTML, …)

Converter

Table in WTC format

Table orientation

Header detection

Table orientation

Key column detection

Homogeneity factor

\[ Hom(x) = \left[ \frac{1}{\text{len}(x)} \sum_{t \in x} \left(1 - 2 \cdot \frac{\text{count}(t_i)}{\text{len}(x)} \right)^2 \right]^2 \]

Lake | Area | Depth | Country | Hom. RH
---|---|---|---|---
Windermere | String_number | String_number | String unknown | 0.89
Kielder Reservoir | String_number | String_number | String unknown | 0.89
Ullswater | String_number | String_number | String unknown | 0.89
Bassenthwaite Lake | String_number | String_number | String unknown | 0.89
Derwent Water | String_number | String_number | String unknown | 0.89
Hom. CH | 0 | 0 | 0

Pre-processed tables

Mean(CH) < Mean(RH) $\rightarrow$ Horizontal

$\exists \text{col where } Hom(\text{col}[0:3]) \neq 0 \rightarrow \text{Header} = \text{true}$

[1] https://subversion.assembla.com/svn/commondata/WDCFramework/tags/1.0.3/
Baseline lookups

1. Lookups from all tables cells (4 external sources + 1 internal Wikidata ES)
2. DBpedia translation (uri & types)
3. Wikidata as pivot metadata
4. TF-IDF-like types scoring
5. Entities disambiguation with target type(s)

Entities Lookups

- Pre-processed tables
- Lake
  - Windermere 14.73 km²
  - Kielder Reservoir 10.86 km²

Entities Disambiguation

- CTA output
- CEA output
Embedding approach

1. Embedding enrichment through Wikidata ES server
2. Regex + Levenshtein lookup
3. K-means clustering over candidates space
4. Scoring algorithm to extract best cluster and deduce target type
5. Candidates disambiguation from clusters, types and entities scores

Q223687
Id: ["Q223687"],
label: ["Wes Anderson"],
aliases: ["Wesley Wales Anderson"],
types: ["Q5", "dbPedia.Person"],
subTypes: ["dbPedia.Director", "Q2526255", "Q36180"]

Title
Rushmore
Director
Anderson

Fight Club
Fincher

Clusters scoring

Candidates' types scoring

Candidates' entities scoring

CTA output

CEA output

Embedding approach example

<table>
<thead>
<tr>
<th>Title</th>
<th>Year</th>
<th>Director</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requiem For A Dream</td>
<td>2000</td>
<td>Aronofsky</td>
</tr>
<tr>
<td>Fight Club</td>
<td>1999</td>
<td>Fincher</td>
</tr>
<tr>
<td>Royal Tenenbaums</td>
<td>2001</td>
<td>Anderson</td>
</tr>
<tr>
<td>There's Something About Mary</td>
<td>1998</td>
<td>Farrelly</td>
</tr>
</tbody>
</table>

Entities scoring (CEA):
$$S_e(i) = 0.25 \times S_k(n) + 0.5 \times R_T + 0.2 \times S_c(i)$$

Entities disambiguation:
$$S_e(Wes Anderson) > \begin{cases} S_e(Paul Thomas Anderson), \\ S_e(Paul W.S. Anderson) \end{cases}$$

Candidates scoring (CTA): $$S_e(Q941209)$$

Clusters scoring: $$S_k(cluster#2)$$
Results

Table 1: Preprocessing results

<table>
<thead>
<tr>
<th>Task/Tool</th>
<th>DWTC</th>
<th>DAGOBAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orientation Detection</td>
<td>0.9</td>
<td>0.957</td>
</tr>
<tr>
<td>Header Extraction</td>
<td>Not evaluated</td>
<td>1.0</td>
</tr>
<tr>
<td>Key Column Detection</td>
<td>0.857</td>
<td>0.986</td>
</tr>
</tbody>
</table>

Table 2: Round 1 results (own evaluator < AI crowd evaluator)

<table>
<thead>
<tr>
<th>Task</th>
<th>CTA</th>
<th>CEA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Criteria</td>
<td>F1</td>
</tr>
<tr>
<td>Baseline</td>
<td>0.517</td>
<td>0.482</td>
</tr>
<tr>
<td>Baseline++</td>
<td>0.641</td>
<td>0.641</td>
</tr>
<tr>
<td>Embedding</td>
<td>0.683</td>
<td>0.683</td>
</tr>
</tbody>
</table>

Approach | Pros | Cons
---|------|------|
Baseline  | ▪ High coverage (multiple sources)  
▪ Computational efficiency | ▪ Lookup-services dependency (reliability)  
▪ Blackbox (indexing, scoring...)  
▪ Queries volume
Embedding | ▪ Lookup strategy independence  
▪ Relevant clustering even with few data  
▪ Generalization (no tailored cleaning + less heuristics in lookups and scoring) | ▪ Computational performances  
▪ K optimization  
▪ Embedding dependency
Discussion & Future Work

- Performance bottlenecks (due to the challenge context):
  - Light Data cleaning ... on purpose
  - Basic lookup strategies ... on purpose (e.g. no use of dictionary)
  - Missing Wikidata – DBpedia type mappings
  - Subset embedding (restricted to baseline candidates)

- Future work:
  - Test other Wikidata embeddings methods (on the whole space)
  - Compute joint embeddings with Wikipedia/DBpedia to enhance coverage
  - Experiment more clustering algorithms and parameters on different datasets
  - Learn data table embedding and find vectorial transformation(s) with KG embedding space
  - ...
DAGOBAH
Datatable-powered Accurate-knowledge Graph for Outstanding and Beautiful Answers to Humans

Thanks!