Top-\(k\) Entity Augmentation Using Consistent Set Covering

Julian Eberius*, Maik Thiele, Katrin Braunschweig and Wolfgang Lehner
Technische Universität Dresden, Germany
What is Entity Augmentation?

**ENTITY AUGMENTATION QUERY (EAQ)**
- Query of the form
  \[ Q_{EA}(a_+, E_{a_1,...,a_n}) \]
  
- Result: same set of entities with the augmented attribute added
- Example

\[
\begin{array}{ccc}
\text{n\_nationkey} & \text{n\_regionkey} & \text{n\_name} \\
0 & 0 & \text{Algeria} \\
1 & 1 & \text{Argentina} \\
2 & 1 & \text{Brazil} \\
3 & 1 & \text{Canada} \\
4 & 4 & \text{Egypt} \\
\end{array}
\]

\[
\begin{array}{cccc}
\text{n\_nationkey} & \text{n\_regionkey} & \text{n\_name} & \text{population} \\
0 & 0 & \text{Algeria} & \text{pop\_ALGERIA} \\
1 & 1 & \text{Argentina} & \text{pop\_ARGENTINA} \\
2 & 1 & \text{Brazil} & \text{pop\_BRAZIL} \\
3 & 1 & \text{Canada} & \text{pop\_CANADA} \\
4 & 4 & \text{Egypt} & \text{pop\_EGYPT} \\
\end{array}
\]

**ENTITY AUGMENTATION SYSTEM (EAS)**
- Manages a corpus of data sources
- Selects a subset D relevant to a query \( Q_{EA} \)

1. Dresden Web Table Corpus: 125M tables

Challenge 1: Information need expressed by a keyword → underspecified

\[ Q_{EA}(\text{\textquoteleft\textquoteleft revenue\textquote Right}}, E) \]

<table>
<thead>
<tr>
<th>Company</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Bank of China</td>
<td></td>
</tr>
<tr>
<td>Banco do Brasil</td>
<td></td>
</tr>
<tr>
<td>Rogers Communications</td>
<td></td>
</tr>
<tr>
<td>China Mobile</td>
<td></td>
</tr>
<tr>
<td>AT&amp;T</td>
<td></td>
</tr>
</tbody>
</table>

\[ S_1: \text{revenues 2013} \]

<table>
<thead>
<tr>
<th></th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agricultural Bank of China</td>
<td>( x_1 )</td>
</tr>
<tr>
<td>Banco do Brasil</td>
<td>( x_2 )</td>
</tr>
<tr>
<td>AT&amp;T</td>
<td>( x_3 )</td>
</tr>
<tr>
<td>China Mobile</td>
<td>( x_4 )</td>
</tr>
</tbody>
</table>

\[ S_2: \text{US market share} \]

<table>
<thead>
<tr>
<th></th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT&amp;T</td>
<td>0.92</td>
</tr>
</tbody>
</table>

\[ S_3: \text{telco sector} \]

<table>
<thead>
<tr>
<th></th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bank of China</td>
<td>( x_1 )</td>
</tr>
<tr>
<td>China Mobile</td>
<td>( x_2 )</td>
</tr>
<tr>
<td>AT&amp;T</td>
<td>( x_3 )</td>
</tr>
</tbody>
</table>

\[ S_4: \text{banking sector} \]

<table>
<thead>
<tr>
<th></th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bank of China</td>
<td>( x_1 )</td>
</tr>
<tr>
<td>Banco do Brasil</td>
<td>( x_2 )</td>
</tr>
<tr>
<td>Rogers</td>
<td>( x_1 )</td>
</tr>
<tr>
<td>China Mobile</td>
<td>( x_2 )</td>
</tr>
<tr>
<td>AT&amp;T</td>
<td>( x_3 )</td>
</tr>
</tbody>
</table>

\[ S_5: \text{Chinese market growth} \]

<table>
<thead>
<tr>
<th></th>
<th>Revenue</th>
<th>Growth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bank of China</td>
<td>( x_1 )</td>
<td>( y_1 )</td>
</tr>
<tr>
<td>China Mobile</td>
<td>( x_2 )</td>
<td>( y_2 )</td>
</tr>
</tbody>
</table>

\[ S_6: \text{revenue changes} \]

<table>
<thead>
<tr>
<th></th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rogers</td>
<td>( x_1 )</td>
</tr>
<tr>
<td>AT&amp;T</td>
<td>( x_2 )</td>
</tr>
<tr>
<td>Banco do Brasil</td>
<td>( x_3 )</td>
</tr>
</tbody>
</table>

\[ S_7: \text{revenues 2013} \]

<table>
<thead>
<tr>
<th></th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rogers</td>
<td>0.95</td>
</tr>
</tbody>
</table>

Challenge 2: Data sources unknown to the user → difficult to express precise queries
Why Consistency? (2)

**S₃: telco sector**

<table>
<thead>
<tr>
<th></th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rogers</td>
<td>x₁</td>
</tr>
<tr>
<td>China Mobile</td>
<td>x₂</td>
</tr>
<tr>
<td>AT&amp;T</td>
<td>x₃ 0.85</td>
</tr>
</tbody>
</table>

**S₄: banking sector**

<table>
<thead>
<tr>
<th></th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bank of China</td>
<td>x₁</td>
</tr>
<tr>
<td>Banco do Brasil</td>
<td>x₂ 0.85</td>
</tr>
</tbody>
</table>

**S₁: revenues**

<table>
<thead>
<tr>
<th></th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agricultural Bank of China</td>
<td>x₁</td>
</tr>
<tr>
<td>Banco do Brasil</td>
<td>x₂</td>
</tr>
<tr>
<td>AT&amp;T</td>
<td>x₃</td>
</tr>
<tr>
<td>China Mobile</td>
<td>x₄ 0.69</td>
</tr>
</tbody>
</table>

**S₅: Chinese market growth**

<table>
<thead>
<tr>
<th></th>
<th>Revenue</th>
<th>Growth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bank of China</td>
<td>x₁ y₁</td>
<td></td>
</tr>
<tr>
<td>China Mobile</td>
<td>x₂ y₂ 0.87</td>
<td></td>
</tr>
</tbody>
</table>

**S₆: revenue changes**

<table>
<thead>
<tr>
<th></th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rogers</td>
<td>x₁ 0.95</td>
</tr>
<tr>
<td>AT&amp;T</td>
<td>x₂</td>
</tr>
<tr>
<td>Banco do Brasil</td>
<td>x₃ 0.90</td>
</tr>
</tbody>
</table>

**S₇: revenues 2013**

<table>
<thead>
<tr>
<th></th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rogers</td>
<td>x₁</td>
</tr>
</tbody>
</table>

Challenge 3:
- EAQ explorative in nature
- Users may want to stumble on new aspects of their information need (similar to document search engines)

Cover 1

Cover 3
Our Top-k Approach

Uncertainty in Sources and Unclear User Intent

- Challenge 1: Information need expressed by a keyword → underspecified
- Challenge 2: Data sources unknown to the user → difficult to express precise queries
- Challenge 3: EAQ explorative in nature → users may want to stumble on new aspects of their information need (similar to document search engines)

We solve this problem...

- ..by presenting not one, but multiple, ranked, alternative covers → TOP-K
Top-1 Entity Augmentation

**INITIAL EXAMPLE**

Diagram showing entities ranked from top to bottom, labeled $c_1$ to $c_{11}$, against data sources by rank.
Top-1 Entity Augmentation (3)

**STRATEGY 1: TOP-RANKED**

- **Entities**
  - $c_{11}$
  - $c_{10}$
  - $c_9$
  - $c_8$
  - $c_7$
  - $c_6$
  - $c_5$
  - $c_4$
  - $c_3$
  - $c_2$
  - $c_1$

- **Problem**
  - Large number of data sources picked
  - Combines data sources that do not fit to each other → consistency!!

- **Candidate Scoring**
  - Schema-/instance match using string distance functions
  - Metadata match using set distance
  - Popularity scores (Alexa Web information through AWS)

- **Data sources by rank**

---

Dresden Database Systems Group
Strategic 2: Minimal Number of Sources

- Consistency is maximized
- Quality according to the ranking function will be low

Cover \( c_{\text{min}} \)
Top-1 Entity Augmentation (5)

**STRATEGY 3: SIMILAR DATA SOURCES**

Coherence Measures
- Similarity of attribute, values, metadata, domains, tags
- Combined using a weighted sum

Cover $c_{\text{consistent}}$
Top-\( k \) Entity Augmentation

Orthogonal strategy: create multiple (Top-\( k \)) diverse results
Top-\(k\) Consistent Entity Augmentation

For a given...

- entity set, ...
- an attribute...
- and a large scale corpus of web tables...

...we propose new entity augmentation algorithms that...

- construct multiple (top-\(k\)), ...
- minimal and...
- consistent...

...covers, that are...

- complementary alternatives.

We solve the problem by...

- mapping the Top-\(k\) Consistent Entity Augmentation to the weighted set cover problem
Top-k Consistent Entity Augmentation

**Weighted Set Cover Problem**
- Universe of elements $U$
- Family of subsets of this universe $S$
- Associated with a weight $w_i$
- Find a subset $s$ of $S$ whose union equals $U$, such that $\sum_{i \in S} w_i$ is minimized

**Top-K Consistent Entity Augmentation**
- Set of entities $E$
- Set of data sources $D = \{d_1, ..., d_n\}$
- Data source quality $score(d)$ with respect to a query
- Find a cover $c = [d_i, ..., d_x]$ with $\bigcup_{d \in c} cov(d) = E$ such that $\sum_{d \in c} score(d)$ is maximized

So far we could trivially map our problem to the Weighted Set Cover problem but there are differences...
Top-k Consistent Entity Augmentation

ONE
- One minimal cover $c$

VERSUS
<>

MANY (TOP-K)
- List of covers $C = [c_1, \ldots, c_n]$

CONSISTENCY CONSTRAINT
- Covers consisting of similar datasets according to a function $sim()$

\[ \forall c \in C \quad \text{maximize} \quad \sum_{d_i, d_j \in c} \text{sim}(d_i, d_j) \]

DIVERSITY CONSTRAINT
- List of covers should not consist of the same or similar datasets, but be complementary alternatives

\[ \text{minimize} \quad \sum_{c_i, c_j \in C} \text{sim}(c_i, c_j) \]
Greedy Top-\(k\) Consistent EA

**Greedy Implementation**
- Initially empty cover \(c\)
- Free entity set \(F = E\)
- In each iteration pick the dataset that maximizes 

\[ \text{score}(d) \cdot |\text{cov}(d) \cap F| \]
- Until \(F = \emptyset\)

**Consistency**
- Considering documents already added to the partial cover result 

\[ \text{score}(d) \cdot |\text{cov}(d) \cap F| \cdot \text{sim}_F(d, c) \text{ with the special case of } \text{sim}_F(d, \emptyset) = 1 \]
- Encourages picks of data sources that are similar to data sources that were already picked for the current cover
Greedy Top-k Consistent EA (2)

**TOP-K**

- Basic idea: perform k consecutive runs of the greedy algorithm using the same input datasets to create k covers

**CREATE K COMPLEMENTARY COVERS (= MAXIMIZE RESULT DIVERSITY)**

- Extend the scoring function

\[
\frac{\text{score}(d) \cdot |\text{cov}(d) \cap F| \cdot \text{sim}_F(d, c)}{\text{redundancy}(d, F, C)}
\]

with \(\text{redundancy}(d, F, C) = \sum_{e \in F \cap \text{cov}(d)} \text{sim}_F(d, \text{coveredBy}(e, C))\)

- Function \(\text{coveredBy}(e, C)\): returns datasets that were used to augment an entity \(e\) in previously created covers \(C\)
Greedy Top-\(k\) Consistent EA (3)

**function** \texttt{GREEDY-\(TOP\)-\(k\)-COVERS}(
\[
C \leftarrow \emptyset \\
U \leftarrow \begin{pmatrix}
0 & \dots & 0 \\
\vdots & \ddots & \vdots \\
0 & \dots & 0
\end{pmatrix}_{|E| \times |D|}
\]
while \(|C| < k\) do 
\[
c \leftarrow \text{COVER}(E, D, U) \\
\text{for all } (e \rightarrow d) \in c \text{ do} \\
\quad U[e, d] \leftarrow U[e, d] + 1 \\
\text{if } c \notin C \text{ then} \\
\quad C \leftarrow c
\]
return \(C\)

**function** \texttt{COVER}(\(E, D, U\))
\[
c \leftarrow \emptyset \\
F \leftarrow E \\
\text{while True do} \\
\quad \text{if } |F| = 0 \text{ then} \\
\quad \quad \text{return } c \\
\quad d \leftarrow \arg \max_{d \in D} \frac{\text{score}(d) \cdot |\text{cov}(d) \cap F| \cdot \text{sim}_{Agg}(d, c)}{\text{REDUNDANCY}(d, D, F, U)} \\
\quad \text{for all } e \in F \cap \text{cov}(d) \text{ do} \\
\quad \quad F \leftarrow F \setminus e \\
\quad \quad c \leftarrow c \cup (e \rightarrow d) \\
\text{return } c
\]
Greedy* Top-k Consistent EA

**Drawback of Greedy Top-K Consistent EA**

- After the first cover is created the search is mainly guided by using different datasets for each cover
- New combinations of used data sets are often not considered in the basic greedy algorithm

→ **Greedy*, consisting of two phase**

- Phase 1: uses the Greedy Top-K Consistent EA algorithm to create $s$ time more covers than requested
- Phase 2: select the $k$ best covers from the a pool of $s \times k$ solutions, using a greedy approach
  - Pick the best solution by score and consistency
  - Pick further solutions that maximize these criteria while using dissimilar datasets (scoring function defined on the level of covers instead)

```plaintext
function GREEDY*-TOPK-COVERS(k, s, E, D)
    C ← GR-TOPKCovers(k * s, E, D)
    C ← SELECT(k, C)
    return C
```
Genetic Top-\(k\) Consistent EA

**POPULATION**

- Initialized by creating covers with the Greedy algorithm
- Until all candidate sources have been used in at least one cover \(\rightarrow\) slightly modification of the Greedy algorithm to favor unused data sources

**GENES AND GENOME**

- Gene = one data source
- Genome = set of data sources

**FITNESS FUNCTION**

- Same as the scoring function in the Greedy algorithm

\[
\frac{\text{score}(d) \cdot |\text{cov}(d) \cap F| \cdot \text{sim}_F(d,c)}{\text{redundancy}(d,F,C)}
\]
Genetic Top-k Consistent EA

CROSSOVER FUNCTION

- Combining two sets of data sources does not necessarily yield a set that is again a set cover
  - Uncovered entities
  - Cover may not minimal (if too many genes are passed)
- Use the basic Greedy approach applied to the union of genes instead of the whole set D

MUTATION STRATEGY

- Randomly flipping bits in the genome
- Fill and prune
  - Remove datasets that may have become redundant
  - Add random datasets to fill holes created by mutation
Genetic Top-K Consistent EA

function GENETIC-TOPK-COVERS(k, s, E, D)

\[
U \leftarrow \begin{pmatrix}
0 & \ldots & 0 \\
\vdots & \ddots & \vdots \\
0 & \ldots & 0
\end{pmatrix}
\]  \[|E| \times |D|\]  ▷ Usage matrix

\[Pop \leftarrow \text{INIT-GREEDYTOPKCOVERS}(k \times s, E, D, U)\]

for \(i \in 0..k \times s\) do

\[c_1, c_2 \leftarrow \text{pickRandom}(Pop)\]  ▷ Select Parents

\[w_1, l_1 \leftarrow \text{tournament}(c_1, \arg \max_{x \in Pop \setminus c_2} \text{sim}(c_1, x))\]

\[w_2, l_2 \leftarrow \text{tournament}(c_2, \arg \max_{x \in Pop \setminus c_1} \text{sim}(c_2, x))\]

\[c_+ \leftarrow \text{COVER}(E, w_1 \cup w_2, U)\]

\[\text{MUTATE}(c_+)\]

\[\text{FILLANDPRUNE}(c_+)\]

\[\text{removed} \leftarrow \text{minScore}(l_1, l_2)\]

\[Pop \leftarrow (Pop \setminus \text{removed}) \cup c_+\]

\[U \leftarrow \text{UPDATEUSAGE}(U, c_+, \text{removed})\]

\[C \leftarrow \text{SELECT}(k, Pop)\]

return \(C\)
Evaluation

 SETUP

- All approaches\(^1\) and baseline algorithms implemented in Scala version 2.11, executed on a version 1.8.0 Oracle JVM
- Dresden Web Table Corpus (DWTC)\(^2\): 125 million web tables
- All queries are run against the full corpus
  - Companies (attributes: revenue, employees, founded)
  - Countries (attributes: population, population growth, area)
  - Large cities (attribute: population)
- Gold standard: human judges evaluated for each entity in each result set the relevance of the data source and the correctness of the entity match

 INDICATORS

- Coverage, minimality, consistency, score, diversity and precision

\(^1\) http://github.com/JulianEberius/REA
\(^2\) http://wwwdb.inf.tu-dresden.de/misc/dwtc
Baseline Algorithms

**VALUEGROUPING**
- Generates one cover (Infogather approach)
- Collects all values into per-entity group
- Clusters these groups according to their values
- Selects the highest scoring value from the cluster with the highest sum of scores
  → Favoring high ranking data sources that are supported by many other sources with similar values

**TOPRANKED**
- Generates k covers
- Selects the highest ranked data sources that can produce a value for each entity
- Next cover is constructed from each second best ranked value for each entity, until k results have been constructed
- Consistency and diversity agnostic
Single Entity Precision

![Graph showing precision at rank 1 for different datasets and methods: Greedy, Greedy*, Genetic, TopRanked, ValGroup.](image)
Inherent Score
Consistency and Minimality

- Greedy
- Greedy*
- Genetic
- TopRanked
- ValGroup

Graphs showing the consistency and minimality across different ranks for various algorithms.
Diversity

- Blue line: Greedy
- Red line: Greedy*
- Brown line: Genetic
- Gray line: TopRanked
- Star: ValGroup
Relaxing Coverage

coverage = 1.0

coverage = 0.75

coverage = 0.0
Relaxing Coverage (2)

coverage = 1.0
coverage = 0.75
coverage = 0.0
Relaxing Coverage (3)

coverage = 1.0

coverage = 0.75

coverage = 0.0
Relaxing Coverage (4)

coverage = 1.0

coverage = 0.75

coverage = 0.0
Runtime Performance

- **Greedy**
- **Greedy***
- **Genetic**
- **Top Ranked**
- **ValGroup**
- **$|D|$**

![Graph 1](image1.png)

- **Runtime in ms**
- **Number of entities $|E|$**

![Graph 2](image2.png)

- **Runtime in ms**
- **Number of covers requested $k$**
Summary

EAQ BASED ON LARGE CORPORA OF DATA SOURCES SHOULD BE PROCESSED...

- as Top-k queries while ensuring consistency and minimize of the individual augmentations and the diversity of the result as a whole

THREE NEW ALGORITHMS FOR CONSISTENT MULTI-SOLUTION SET COVERING

- Greedy...
- Greedy* and...
- Genetic Top-k Consistent EA

EXPERIMENT SHOWED THAT

- Genetic set covering based approach improves both consistency and minimality of the results significantly...
- without loss of coverage...
- while producing a diverse set of results
Backup
Why Web Tables

**Very Important Source of Information**

- Used as a compact and efficient way to present relational information
- Have many applications
  - Knowledge management, e.g. ontology enrichment (Mulwad et al.)
  - Information retrieval, e.g. entity augmentation (Eberius et al.), factual search (Yin et al.), table search
- Dresden Web Table Corpus (DTWC)
  - 125M tables based on Common Crawl (3.6 billion web pages, 266TB)
  - [http://wwwdb.inf.tu-dresden.de/misc/dwtc](http://wwwdb.inf.tu-dresden.de/misc/dwtc)
Evaluation

**Setup**

- All approaches and baseline algorithms implemented in Scala version 2.11, executed on a version 1.8.0 Oracle JVM
- Dresden Web Table Corpus (DWTC): 125M web tables
- All queries are run against the full corpus
  - Companies from the Forbes 2000 list (attributes: revenue, employees, founded)
  - Countries (attributes: population, population growth, area)
  - Large cities from the Mondial database (attribute: population)
- Default query set
  - 4 queries for each domain-attribute pair, with 20 entities \(|E| = 20\) each, and \(k = 10\)
  - Company and city domain: top and tail queries, i.e. queries for the top 100 entities, queries with randomly picked entities
- Gold standard: human judges evaluated for each entity in each result set the relevance of the data source and the correctness of the entity match

1. [http://wwwdb.inf.tu-dresden.de/misc/dwtc](http://wwwdb.inf.tu-dresden.de/misc/dwtc)
Evaluation – Indicators

**Coverage**
- Percentage of the queried entities that are augmented with a value

**Precision**
- Percentage of entities for which a relevant value was retrieved
- E.g. precision = 1.0 → each entity was augmented with a value that was judged relevant with respect to the query keyword, but the augmentation is not necessary consistent

**Consistency**
- Average similarity between the datasets that were used to create the respective cover
- Higher value → cover was created from more similar datasets
Evaluation – Indicators (2)

MINIMALITY

- Number of datasets used in a cover in relation to the number of entities it covers
  \[ 1 - \frac{|c|}{\sum_{d \in c} |cov(d)|} \] or \[ 1.0 \frac{|c|}{s} = 1 \]

- High value \( \rightarrow \) cover was created from a small number of data sources
- Value of 0.0 \( \rightarrow \) each entity was covered using a different data source

SCORE

- Average ranking score each individual dataset received from the scoring function

DIVERSITY (FOR A SET OF COVERS)

- Average distance between all covers in the set (average distance between the datasets of two covers)
Baseline Algorithms

**VALUEGROUPING**
- Generates one cover
- Collects all values into per-entity group
- Clusters these groups according to their values
- Selects the highest scoring value from the cluster with the highest sum of scores
→ Favoring high ranking data sources that are supported by many other sources with similar values

**TOPRANKED**
- Generates k covers
- Selects the highest ranked data sources that can produce a value for each entity
- Next cover is constructed from each second best ranked value for each entity, until K results have been constructed
- Consistency and diversity agnostic
Evaluating Entity Coverage

Results are comparable to Infogather+ (SIGMOD’13)
Evaluating Single Entity Precision (2)
Evaluating Single Entity Precision (3)

**Percentage of queries with best solutions not on rank 1, and improvements over rank 1**

<table>
<thead>
<tr>
<th>Method</th>
<th>Best Result not on Rank 1</th>
<th>Avg. Improvement over Rank 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Genetic</td>
<td>14%</td>
<td>12%</td>
</tr>
<tr>
<td>Greedy</td>
<td>14%</td>
<td>10%</td>
</tr>
<tr>
<td>Greedy*</td>
<td>23%</td>
<td>9%</td>
</tr>
<tr>
<td>TopRanked</td>
<td>32%</td>
<td>7%</td>
</tr>
</tbody>
</table>
Evaluating Cover Quality (3)

SETUP

- $\text{thCov}=1.0$ (demanding full covers) and $\text{thCons}=0.0$

Graph shows:
- Greedy
- Greedy*
- Genetic
- TopRanked
- ValGroup

Coverage vs. Rank graph.
Top-1 Entity Augmentation (2)

**RELATED WORK: AGGREGATION**

- Aggregating all candidate values into one result (Infogather: Yakout et al. SIGMOD’12, Zhang et al. SIGMOD’13)
- Generating **one** augmentation