Ranking Multi-Word Expressions by Difficulty

Set-up and Results of the Crowdsourcing Experiment

Jaka Čibej

Centre for Language Resources and Technologies, University of Ljubljana

Göteborg, 5 December 2018
Idea

- Rank a set of expressions by difficulty.
- Can this be done through crowdsourcing?
- How?
- We need a simple task.
- Manageable workload.
- (Relatively) reliable results.
Ranking Multi-Word Expressions by Difficulty

Ranking – How?

• Ranking the entire list?
  • Task can’t be divided between multiple participants.

• Ranking a subset of tasks?
  • Combinations might affect results.
  • Still not very user-friendly.
  • Difficult to merge?
  • Which combinations?

• Ranking a (small) subset of tasks, e.g. 4?
  • Again, which combinations?
Best-Worst Scaling

- Ranking method
- Choosing the best and worst unit in a combination of (ideally) 3–4 candidates
- Example:

  - 6 possible binary relations between the 4 elements
    - J ~ K, J ~ L, J ~ M, K ~ L, K ~ M, L ~ M
  - BWS with 4 elements
    - K = 3, M = 2, J = 2, L = 1
    - J < K, J > L, J ~ M, K > L, K > M, L < M
    - 5 out of 6 relations (83 %)
    - (at least) 2 clicks
  - Ranking all 4 elements:
    - 6 out of 6 relations (100 %)
    - (at least) 4 clicks
    - twice the workload!
Best-Worst Scaling

- 60 expressions
- How many tasks to rank them all?
- Which combinations?
- We based our selection process on the premise that we need the information on all binary relations to make sure the final ranking is as accurate as possible.

- 60 expressions
- 1,770 binary relations
- 487,635 possible combinations of 4 elements
  - too much work
  - too much repetition
  - no point in collecting information on binary relations multiple times from the same crowdsourcers
Selecting the Optimal Number of Combinations

Ranking Multi-Word Expressions by Difficulty

60 expressions

1,770 binary relations

1,764 binary relations remaining

…

487,635 combinations (4 units)

Combination 1
(6 new binary relations)

JKLM

Combination 2
(with repetitions from previous combinations)

JKLO

…

Combination N
(6 new binary relations)

PQRS

…

1,758 binary relations remaining

…
Selecting the Optimal Number of Combinations

- We go through all combinations and choose only the ones where no relation is repeated (in order to avoid tasks where we get too many repeated relations, which are practically useless).

- We continue by selecting tasks with only 1 repeated relation, then 2, then 3, then 4, then 5 (until we cover all possible binary relations).

- Why?
  - To minimize the number of (completely) redundant tasks.

- 60 expressions
  - 1,770 binary relations
  - 1,362 (77%) relations covered with non-repetitive combinations.
  - 33 combinations where 1 relation is already known.
  - 50 combinations where 2 relations are already known.
  - 12 combinations where 3 relations are already known.
  - 3 combinations where 4 relations are already known.
  - 1 combination where 5 relations are already known.
Final Set of Tasks

• 326 tasks
• 77% are non-repetitive.
• 23% are partially repetitive (as little as possible).

PREDICTIONS:

IF:
• Number of crowdsourcers: 20
• Average response time: 30 seconds
• Responses per task: 5

THEN:
• Time per crowdsourcer: **0.68 hours**, which equals **40.75 minutes**
# PyBossa Interface

<table>
<thead>
<tr>
<th>Easiest</th>
<th>Expression</th>
<th>Hardest</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>a lot</td>
<td></td>
</tr>
<tr>
<td></td>
<td>once upon a time</td>
<td></td>
</tr>
<tr>
<td></td>
<td>as it happens</td>
<td></td>
</tr>
<tr>
<td></td>
<td>deadly dull/serious, etc.</td>
<td></td>
</tr>
</tbody>
</table>

**Save**

as it happens

**Meaning:** something that you say in order to introduce a surprising fact  
**Example:** As it happens, her birthday is the day after mine.

- **phone compatibility** (not too wide or too long, etc.)
- user-friendly (or is it?)
- foreseen error scenarios - warnings helped limit any technical mistakes during annotation
  - e.g. only one ticked expression,
  - same expression in both columns
  - no ticked expression

---

Current task ID number: 689222

You have solved 1 task(s) out of a total of 326. You are expected to solve 82.

You can fill in the feedback questionnaire to describe how you made your decisions.
## Results – Metadata

- 2 projects with 326 tasks
- Up to 7 responses per task (at least 5).
- A total of 26 annotators.

<table>
<thead>
<tr>
<th>Metadata</th>
<th>Adverbs</th>
<th>Verbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean response time</td>
<td>47.4 seconds</td>
<td>50.38 seconds</td>
</tr>
<tr>
<td>Median response time</td>
<td>22.9 seconds</td>
<td>26.67 seconds</td>
</tr>
<tr>
<td>Total time spent on tasks</td>
<td>27.88 hours</td>
<td>31.25 hours</td>
</tr>
<tr>
<td>Mean response time (no outliers over 30 seconds)</td>
<td>18.54 seconds</td>
<td>20.12 seconds</td>
</tr>
<tr>
<td>Median response time (no outliers over 30 seconds)</td>
<td>18.3 seconds</td>
<td>20.02 seconds</td>
</tr>
<tr>
<td>Total time spent on tasks (no outliers over 30 seconds)</td>
<td>7.26 hours</td>
<td>7.24 hours</td>
</tr>
<tr>
<td>Time per crowdsourcer (no outliers over 30 seconds)</td>
<td>0.28 hours</td>
<td>0.29 hours</td>
</tr>
</tbody>
</table>
Results – Agreement (Verbs)

• Inter-annotator agreement (Krippendorff’s Alpha)
Results – Agreement (Adverbs)

- Inter-annotator agreement (Krippendorff’s Alpha)
Merging the Results

- Collection of ranked expressions obtained in two ways:
  - Linear scale using average ranks (presented by Jaka)
  - Clustering and multi-dimensional visualization using vector embeddings (presented by David)

- The Linear Scale approach
  - Download and view eNetCollect_Linear_Scales-verbs_and_adverbs.xlsx
  - a more brute-force approach
  - take all annotations for a specific expression (regardless of the expressions it appears with) and average the sum to get the expression's average rank
  - the premise: harder/easier expressions should consistently/more frequently be annotated as more difficult (rank 3) or easier (rank 1)
Thank you.

Jaka Čibej
jaka.cibej@cjvt.si