LeQua@CLEF2022: Learning to Quantify
Andrea Esuli, Alejandro Moreo, Fabrizio Sebastiani
Istituto di Scienza e Tecnologie dell’Informazione - Consiglio Nazionale delle Ricerche
56124, Pisa, Italy

LeQua@CLEF2022
Quantification is the task of predicting the prevalence (i.e., relative frequency) of a property in a sample of elements from a domain. LeQua 2022 is the first edition of the “Learning to Quantify” lab, hosted within the CLEF 2022 Conference.

Tasks
Two tasks are offered:
- T1 The Vector Task:
  - Participant teams are provided with vectorial representations of the documents.
  - Mostly for teams not into text mining.
- T2 The Raw-Documents Task:
  - Participant teams are provided with the raw documents.
  - Mostly for teams wanting to test end-to-end systems.

For each task, two subtasks are offered:
- A The Binary Subtask:
  - 2 classes
  - Classes are sentiment-related (Positive and Negative)
- B The Multiclass Subtask:
  - 28 classes
  - Classes are topic-related (e.g., Automotive, Baby, Beauty, ...)

Dataset
The data are obtained from a crawl of ≥100M Amazon reviews; from these we remove:
- all reviews shorter than 200 characters,
- all reviews that have not been recognized as “useful” by any users,
- (for the binary “sentiment-based” task) all reviews with 3 stars.

The 2 training sets L_B (binary) and L_M (multiclass):
- L_B consists of 5,000 documents and L_M consists of 20,000 documents
- L_B and L_M are sampled from the ≥100M-strong dataset Ω via stratified sampling on the dimension of interest (resp. sentiment, topic), so as to have “natural” prevalence values for all the class labels.

The 2 test sets:
- We use 1,000 development samples of 250 documents each for the binary task and 1,000 development samples of 1,000 documents each for the multiclass task.
- The sets of development samples D_B and D_M are generated from Ω \ L_B and Ω \ L_M via the Craemer algorithm for sampling uniformly from the unit simplex.
- The goal of this sampling algorithm is generating samples characterised by a variety of (equi-probable) class distributions, with class prevalence values not from a predefined grid of values.

Timeline
<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dec 1, 2021</td>
<td>Release of train and dev set</td>
</tr>
<tr>
<td>Apr 22, 2022</td>
<td>Release of test set</td>
</tr>
<tr>
<td>May 5, 2022</td>
<td>Run submission deadline</td>
</tr>
<tr>
<td>May 13, 2022</td>
<td>Release of results</td>
</tr>
<tr>
<td>May 27, 2022</td>
<td>Paper submission deadline (optional)</td>
</tr>
<tr>
<td>Sep 5, 2022</td>
<td>LeQua @ CLEF2022</td>
</tr>
</tbody>
</table>

Sampling
Generating test samples that cover all the possible spectrum of prevalence values is key to test the robustness of quantifiers to the variation of prevalence from training data to test data. For example, drawing n random values (for a problem with n classes) uniformly at random from the interval [0,1] and then normalizing them so that they sum up to 1 (IID method), produces samples biased towards the centre of the unit (n−1)-simplex. The Kraemer algorithm generates samples that uniformly cover the entire spectrum of prevalence values for all classes:
- Given a set of classes Y, generate a vector A = (a_1, ..., a_(|Y|−1)) of points sampled uniformly at random from [0,1].
- Sort the a_i’s to obtain B = (b_1 ≤ ... ≤ b_(|Y|−1)), and define b_0 = 0 and b_Ω = 1.
- Obtain a vector P = (p_1, ..., p_Ω) by defining p_i = b_i − b_(i−1) for all i ∈ {1, ..., |Ω|}.
- Use P as the distribution of class prevalence values for generating sample σ.

Visualization of distribution of 2-dimensional samples generated using different methods:

Evaluation
Relative frequencies of classes are represented by probability distributions. The true probability distribution p for each set is compared to predicted one ˆp, by means of Relative Absolute Error:

RAE(p, ˆp) = \frac{1}{n} \sum_{y \in Y} |\hat{p}(y) − p(y)| p(y)  \tag{1}

The final score is the mean RAE across all the samples in the test set.

Links
Web: https://lequa2022.github.io/
Data: https://zenodo.org/record/5734465
Twitter: @LeQua2022

Acknowledgements
This work has been supported by the SoBigdata++ project, funded by the European Commission (Grant 871042) under the H2020 Programme INFRAIA-2019-1, and by the AI4Media project, funded by the European Commission (Grant 951911) under the H2020 Programme ICT-48-2020.