Problem statement:

Build a web application for Q&A on beauty products, which are sold on Amazon. This application would be used by any user for finding out the best answer's and supporting *critical* reviews related to merchandising product functionalities. The problem statement would be: given a query on product functionalities find out the most relevant (top-3) reviews that are most relevant to the query and answers yet give a different perspective to the answer. That is, if the existing answer is saying positive about the product, the reviews must bring the complementary view points on the product and its functionalities. Currently the Q&A database contains few queries and their answers, the intention for this task is to

a) Enriching/augmenting the existing answers with other supporting information that can be extracted from the product reviews (reviewed previously by experts, which can be determined by `helpful` index)

For Example:
'question': 'can I use 'B00028OSI0'on my face?',
'answer': 'All over! Buy it its worth every penny.' – *positive sentiment*

Reviews:

[1] "Someone please tell me this product is alcohol free? It smells like alcohol and I'm not sure I should put it on my face. I do not see positive results yet either." – *negative sentiment*
[2]"Initially I began but putting this on my face after washing it. I must say that it can't be used alone. ……." – *negative sentiment*

[3] "The ingredients are non-toxic, non-drying and very helpful. It didn't work on my sensitive, picky, facial skin, but I do use it as a hair gel. …". – *negative sentiment*

Dataset:

1. Amazon question/answer data on beauty products (**http://jmcauley.ucsd.edu/data/amazon/qa/qa_Beauty.json.gz**) , containing the below mentioned fields

- `asin` - ID of the product, e.g. B000050B6 Z
- `questionType` - type of question. Could be 'yes/no' or 'open-ended'
- `answerType` - type of answer. Could be 'Y', 'N', or '?' (if the polarity of the answer could not be predicted). Only present for yes/no questions.

- `answerTime` – raw answer timestamp
- `unixTime` – answer timestamp converted to unix time
- `question` – question text
- `answer` – answer text

Total 42,422 questions.

2. Amazon product reviews on beauty products (http://snap.stanford.edu/data/amazon/productGraph/categoryFiles/reviews_Beauty_5 .json.gz), containing the below mentioned fields

- `reviewerID` - ID of the reviewer, e.g. A2SUAM1J3GNN3 B
- `asin` - ID of the product, e.g. 000001371 4
- `reviewerName` - name of the reviewer
- `helpful` - helpfulness rating of the review, e.g. 2/3
- `reviewText` - text of the review ● `overall` - rating of the product
- `summary` - summary of the review
- `unixReviewTime` - time of the review (unix time)
- `reviewTime` - time of the review (raw)

Total 198,502 reviews.

Validation techniques:

Okapi BM25+ [1] metrics should be used for determining the relevance of answers to queries. Both relevance and sentiments would be used for determining the final ranked-list of reviews against a query and its answer. In the UI user can type a question while answer to his/her query would be shown along with supporting reviews that would provide complementary views wrt the answers. Reviews would be ranked order wrt their relevance score against the input query. Both *relevance score* and *sentiment score* of the answer and the reviews should be shown on the UI.

Team Structure and Review Criteria:

Reference:

1. https://pdfs.semanticscholar.org/51bb/d9ac2850f28b50dc47c881c9eb580b18d80f.pdf
2. http://cseweb.ucsd.edu/~jmcauley/pdfs/www16b.pdf