

THE IBM HACK CHALLENGE

User Query on Stack Overflow

PROBLEM STATEMENT

The problem statement aims at building a solution that helps to find the right answers that are relevant to the developer issues on Stack Overflow.

A lot of content is present in form of stack overflow questions and answers, various studies point that developers face problems while development life cycles and they ask questions on stack overflow which gets answered by fellow developers across the globe. For a new developer to understand a concept or solve an issue, it could be very difficult to identify the problems. The proposed solution should help to identify most relevant questions to a query using text similarity including identify the matching tags and pick top relevant questions from stack overflow and identify top (k) solutions of the problem based on sentiment analysis of reviews of the given solutions on the Stack Overflow.

TEAM SIZE - 3

YUDHIK AGRAWAL



SAYAK KUNDU

SAMYAK JAIN



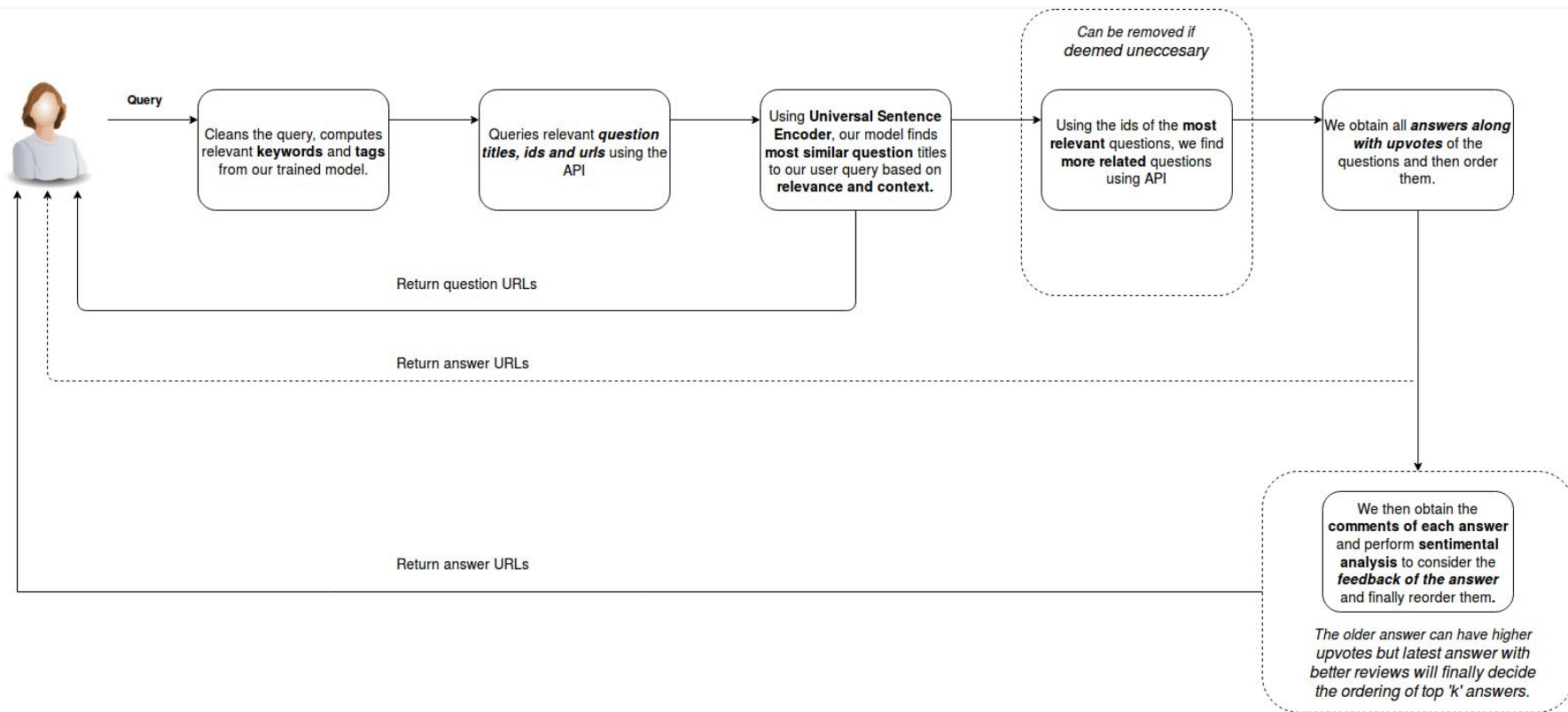
ABOUT/CONTRIBUTION TEAM MEMBERS

- **Samyak Jain:** [*Deep Learning, NLP, CV*]
 - Optimizing deployment of server by making it faster.
 - **Universal Sentence Encoder** to further trim the set of questions achieved by ***Tag filtration***.
 - UI/UX design.
- **Sayak Kundu:** [*Pipeline design, API handling, NLP*]
 - Smartly handling the API's for ***identifying Tag*** and generating the questions based on tag similarity and upvotes.
 - Limiting the set of top solutions using ***statistical analysis***.
 - ML model that extracts relevant tags from the question titles.
- **Yudhik Agrawal:** [*Deep Learning, NLP, CV*]
 - **Top(k)** answers of a query by formulating a metric which involves sentiment-analysis on the feedback of each answer.
 - Data Collection from StackExchange Data Explorer and Google BigQuery.
 - **Universal Sentence Encoder** to further trim the set of questions achieved by ***Tag filtration***.

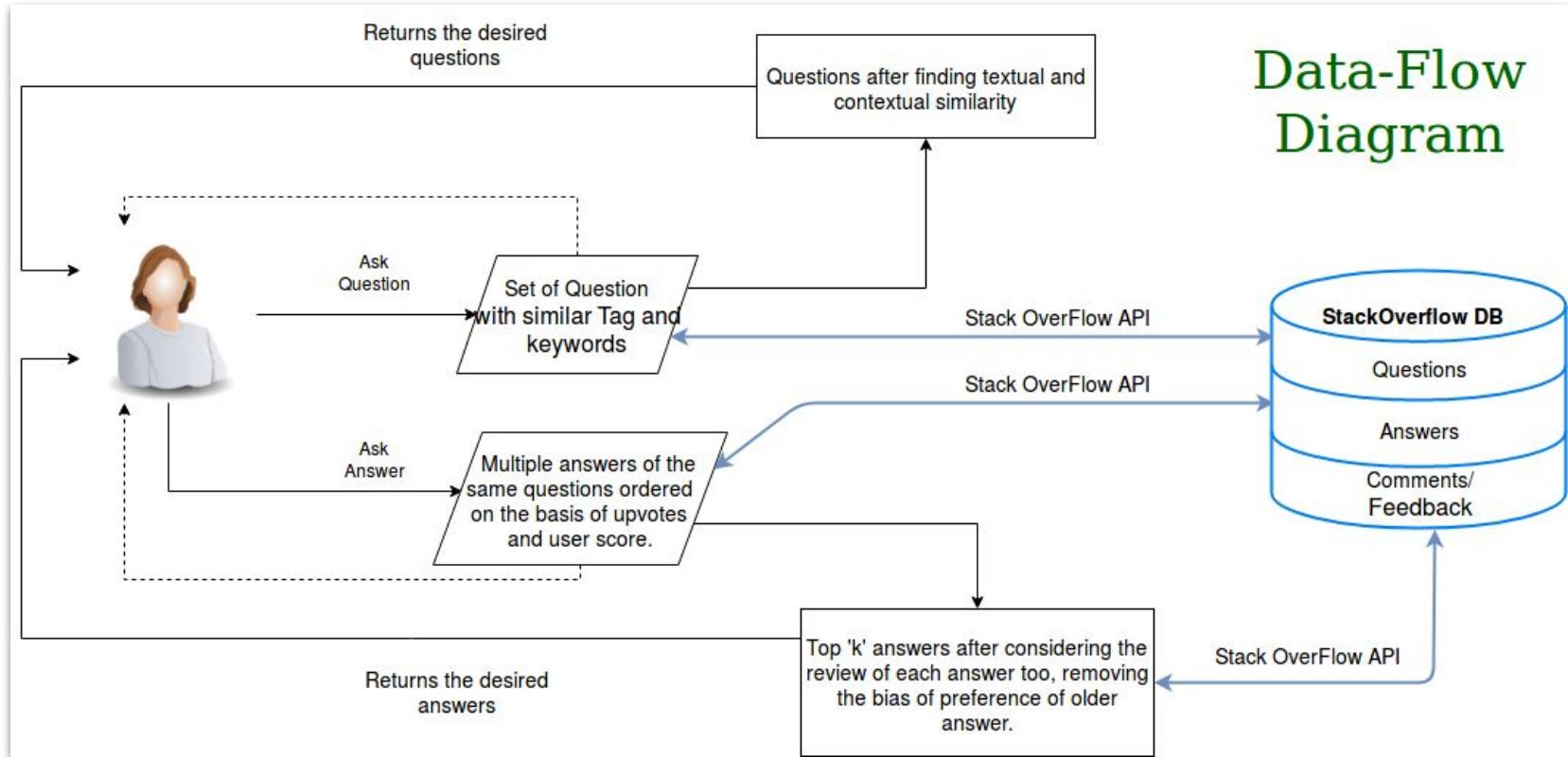
TECHNOLOGY/PLATFORM/API'S USED

- Platform:
 - Django
- Technology
 - Linux/Unix
 - StackExchange Data Explorer
 - Python
 - Deep Learning Libraries [Tensorflow]
- Api's
 - Tensorflow-Hub [Universal Sentence Encoder]
 - IBM Watson [Sentiment Analysis]
 - Stack Overflow API's

ARCHITECTURE DIAGRAM

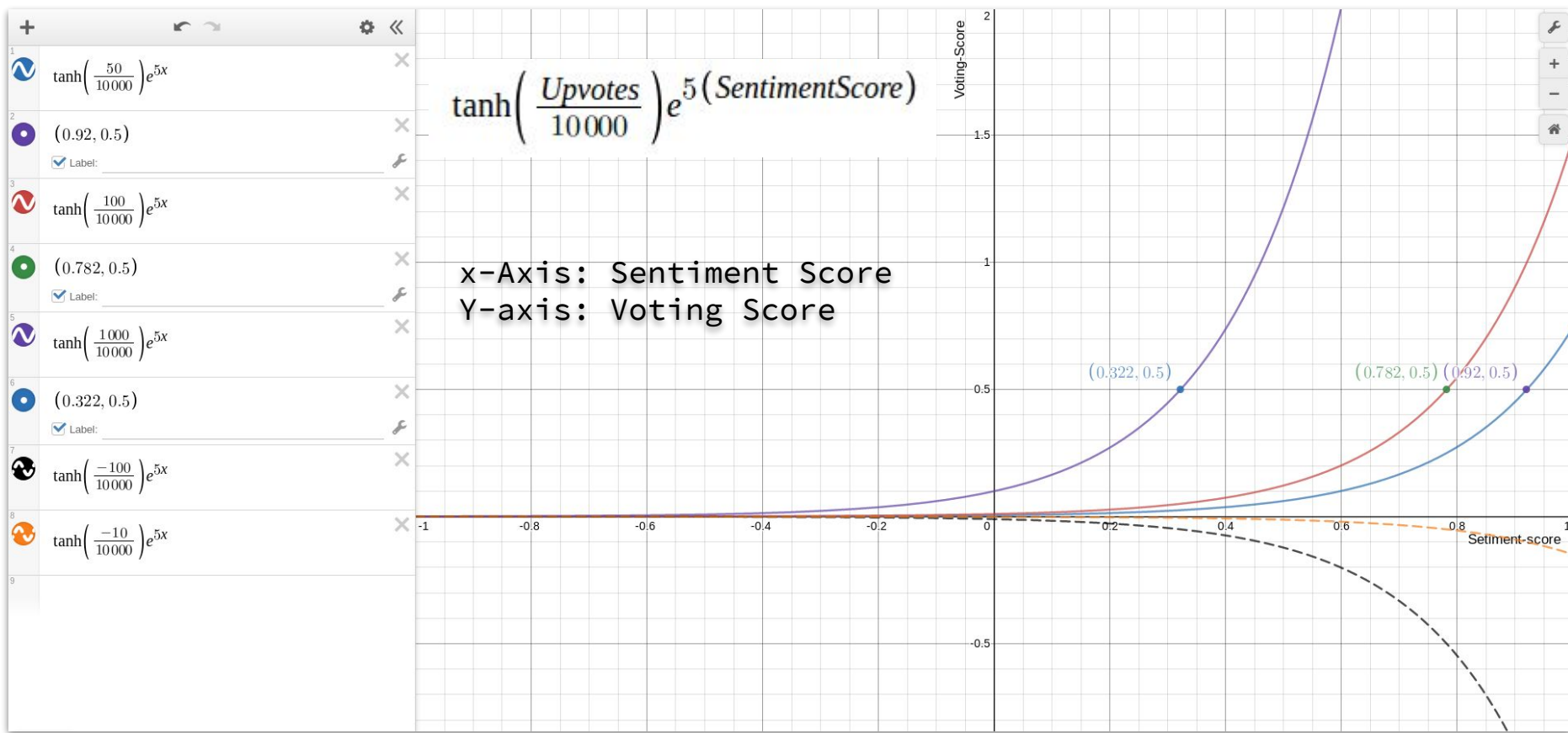


DATA-FLOW DIAGRAM



NOVELTY'S

- **Identifying Most relevant questions**
 - We do not just rely on *text similarity*
 - We use *Universal Sentence Encoder* (USE) which is a better metric than *Bleu Score* based on 'context and relevance' along with text similarity.
- **Identifying Most Relevant Tags from questions**
 - Tag extraction happens using a neural network, which removes the dependency on sequence matching. It reduces the search space which increases efficiency.
- **Identifying top 'k' solutions**
 - We consider the feedback of the answer by doing sentiment analysis on the comments along with the upvotes of the answer and give score [[here](#)].
- **Model Deployment**
 - Improved UX by making it *faster to resolve a query* by loading all the pre-trained models and creating sessions while starting the server.



Formula-Range

Upvotes: $[-10000, 10000]$ (General Assumption for scaling)

SentimentScore: $[-1, 1]$ (Output of the IBM-Watson)

THE GRAPH

Three Answers with upvotes having 50, 100, 1000 votes have the top priority with Sentiment Score (0.92, 0.782, 0.322) and all have Voting Score of 0.5.

This removes the bias of older answer with high upvotes from the more accurate with lower upvotes.

Setting Environment Variables for Node to retrieve

- Answer Id and Link - [34154491](#)

Sentiment: 0.1557 Upvotes: 173 Score: 0.0376

- Answer Id and Link - [22312793](#)

Sentiment: -0.251 Upvotes: 358 Score: 0.0101

- Answer Id and Link - [22312868](#)

Sentiment: -0.102 Upvotes: 90 Score: 0.0053


Real-Like Example:

Answer with 173 upvotes was given higher priority over answer with 358 upvotes due to sentiment score and the positive comments in the above example is good evidence.

Ref: <https://stackoverflow.com/questions/22312671/setting-environment-variables-for-node-to-retrieve>

▲
358
▼

24 If you're using `fish` instead of `bash`, you need to use: `env USER_ID=239482 my_command`. For example, for setting environment variables for node.js' `debug` library: `env DEBUG='*' node some_file.js` [fishshell.com/docs/current/faq.html#faq-single-env](#) – [SilentSteel](#) Oct 22 '14 at 15:21

1 I found I had to remove the quotes around `""` for it to work: `env DEBUG=* node some_file.js` – [divillysausages](#) Jul 8 '15 at 21:20 

▲
173
▼

Awesome man, You save my day. Thank you – [Nguyễn Anh Tuấn](#) Dec 28 '16 at 10:17

I once was blind, now I can see – [webdevinci](#) May 9 '17 at 12:19

Best answer I've seen on the topic yet. Thank you. – [Dave Voyles](#) - [MSFT](#) Aug 8 '17 at 21:25

FRAMEWORKS/TOOLS USED FOR UI/UX DESIGN

We used **Django** as it is easily scalable and compatible while deploying various deep learning models.

As the Django uses **DRY** principle it makes the process of ***creating sessions one-time while deployment*** making it faster for the rest of the world.

Jinja2 is a *templating library* which simplifies the process of *generating HTML* for ***Python web apps***.

REFERENCES

- Text similarity
 - [Universal Sentence Encoder](#)
 - [InferSent](#)
- Sentimental analysis
 - [IBM Watson Sentimental analysis](#)
 - [Reading Wikipedia to Answer Open-Domain Questions](#)
- [Scalable API](#)
- [Django Documentation](#)
- [TensorFlow Hub | TensorFlow](#)