Fall 2022 CS 6220 Big Data Systems Live FAQ

Project Proposal

Andrea Covre Anshul Gupta Prahlad Jasti Akash Nainani
Rishabh Thukral

Motivation and Objectives

Over the last couple of years, people are spending more time consuming digital content or participating in online meetings whether for official work purposes, school classes, or meeting for social activities. Consequently, there have been many services that are developed to make the online experience feel more like an in-person environment. But we have identified one problem that still persists in the digital world but is an essential part of any in-person activity. The element of interaction, specifically interactive questions.

In any meeting of several people, the audience often asks questions with follow-ups to participate in discussions. In a conference happening virtually, the participants often put up their questions in a Q&A/Chat box which are usually suppressed by other questions, most of which are quite similar, falling in the category of what can be considered as an FAQ! Moreover, a rare but important question might get suppressed among these frequent & similar questions. For an in-person event, quite often a similar question is asked that can be clarified for everyone.

Through this project, we seek to understand all the burning questions that such an audience might have by consuming all the incoming questions, summarizing them, and generating a list of FAQs that is continuously evolving with time as the event progresses. This would be a streaming application generating results in real-time supported by an active online machine learning model. Such a system would bridge the gap of interactivity that seems to be lacking in the virtual world.

Related Work

There are live Q&A solutions like slido [2] that aim to provide a platform to host live Q&A for events as well as a large number of streaming platforms that allow live streams like Youtube, Zoom have the option for Q&A. These platforms have basic ways for the person answering to sort the questions asked: based on likes, latest, and oldest. Beyond this, slido provides an option to view sentiments of the questions as one of its advanced features [2]. There has been some work done that aims in clustering together similar questions and also ranking them [3], Yunbo et al proposes a way to rank question search results by clustering. Alian [4] suggests using canopy-K-mean and hierarchical-K-means clustering to cluster together questions. Aggarwal[9] has done work in clustering streaming data source that combines and creates clusters where data arrives.

Proposed Work

Novelty

This project is novel in its high-level goal to provide ease-of-use to live Q&A sessions via natural language processing. While a message queue / distributed streaming pipeline to consume textual data has always existed, a system to consume multiple sources of data and perform NLP-based unsupervised learning has not been proposed in any recent literature. The clustering portion of the analytics pipeline is the main driver towards generating a ranking of questions that contributes to the high-level goal.

Current solutions used for live Q&A either list down questions as they come or have a feature to vote questions to give priority to popular questions. These approaches at times may eclipse some relevant questions which might not be as frequent.

Methodology

The project is divided into two major components:

1. Big Data System: This part deals with developing an infrastructure that is distributed in nature and supports streaming data. This project will utilize Apache Kafka [1], a distributed message queue that can store messages coming from various sources and forward them to a required destination. We chose Kafka since it is distributed in nature, and we can upscale or downscale the services based on incoming traffic. It provides a publisher-subscriber interface for transferring messages.
2. Analytics Machine Learning: This part involves developing a sophisticated NLP model which can interpret messages to detect and classify questions. Given a set of questions, the model could find semantically similar questions by using sentence-level embeddings. Using some similarity metrics (eg: Dot-product of embedding vectors), an unsupervised algorithm generates clusters of questions. Each cluster would be associated with a representative question. Finally, clusters are ranked to find the most active questions to generate a list of FAQs based on the user-provided thresholds.

Technologies

There are several technologies that enable this project making this a novel and useful application. Some of the technologies that we anticipate utilizing in the development of implementation, along with their intended purpose, include the following:

- Python - Apache Kafka (Distributed Streaming)
- Python - Numpy & Pandas (Data Analysis)
- Python - Matplotlib & Seaborn (Data Visualization)
- Python - Sklearn, Tensorflow, SciPy (Machine Learning Frameworks)
- Google Universal Sentence Encoder, SentenceTransformers (Sentence Embedding)
- Data Source Open APIs (Datasets)
- Javascript/Typescript - React (Frontend Dashboard)

Deliverables

- Finalized architecture
- Datasets

- Output samples
- Source code
- System analysis report

Architecture

The following figure presents a high-level architecture of the Live FAQ project:

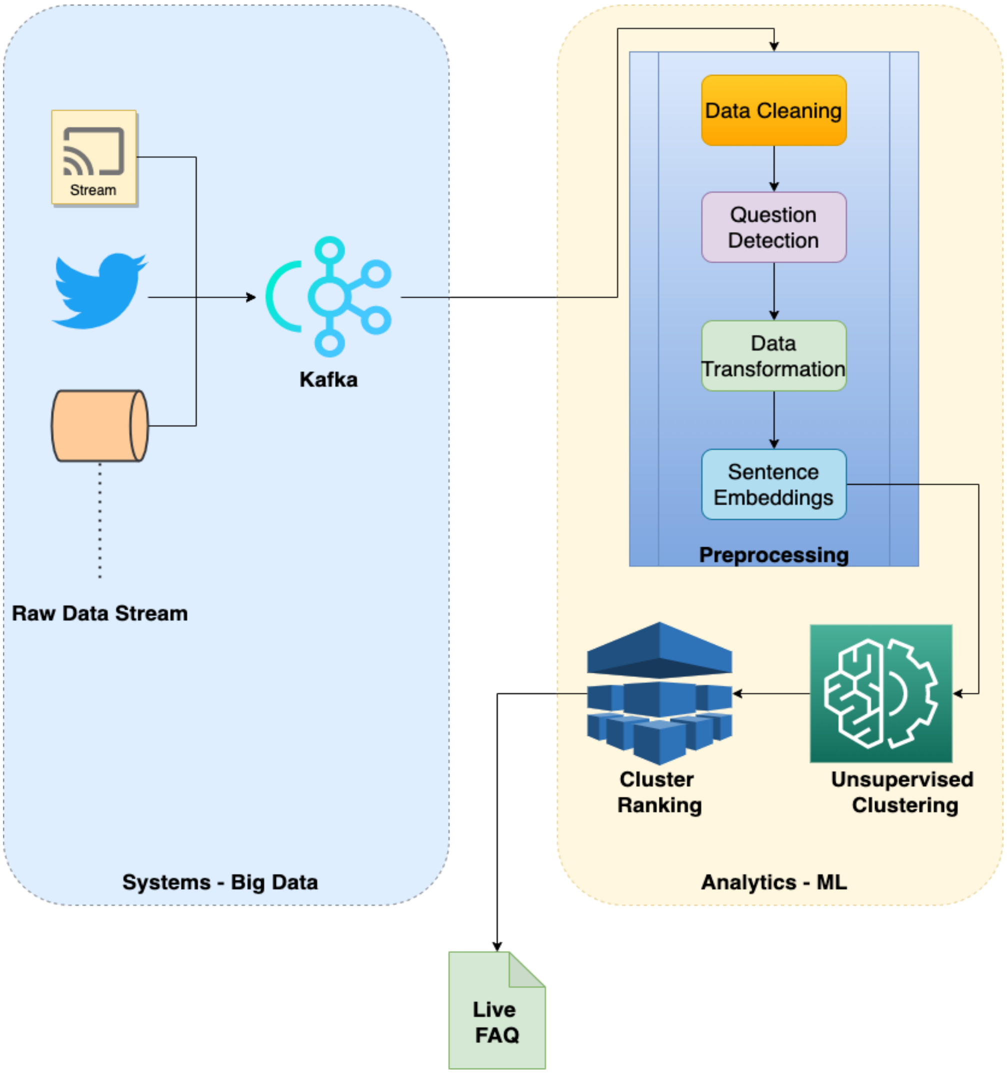

Figure: High-level architecture for LiveFAQ

Plan of Action

We plan to use **GitHub** for collaboration among the team members and use **Trello** as our kanban board for tracking the progress of features, tasks, and issues among the peers.

Resources

- AWS/Azure/GCloud: We plan to utilize student plans/ trial credits offered by cloud vendors for any infrastructure requirements.
- GPU: Some resources are available locally and can be obtained from cloud vendors if needed or through requesting PACE clusters for the purpose of training if needed.
- A combination of real and artificial data sources for system & analytical evaluation.

Schedule

Goal	Due Date	Task size
Finalize architecture	Oct 5	S
Collect data and create datasets	Oct 10	M
Big Data component implementation	Oct 23	L
Big Data component Testing + Debug	Oct 27	S
Analytics component implementation	Nov 6	L
Analytics component Testing + Debug	Nov 9	S
Integrate Big Data + Analytics	Nov 13	M
System Testing + Debug	Nov 21	M
System Profiling	Nov 30	S
Create sample outputs	Nov 30	S
Make System analysis	Nov 30	S
Write the final report and deliverable	Dec 4	M

Evaluation and Testing Method

The system will be evaluated based on:

- FAQs/Clustering Quality
  - Comparative analysis of different clustering algorithms such as:
    - K-Means Clustering (batch based)
    - Density-Based Clustering (batch based)
    - Hierarchical Clustering (batch based)
    - BIRCH [8] (iterative/streaming based)
  - Are questions within a cluster semantically equivalent to each other?
- Resource usage & Performance
  - Latency & Throughput
  - GPU Hours & cloud resource cost
- Robustness & Transferability
  - Can the system perform well in scenarios with large clusters of any given question and large numbers of clusters?
  - Can the system detect malicious and/or random questions?
  - Can the system perform well in other contexts and in other applications? E.g.:
    - Rank opinions in live streams expressed by users
    - Cluster tweets based on breaking news

Bibliography

[1] Apache Kafka. https://kafka.apache.org/. [Online]. Available: https://kafka.apache.org/

[2] www.slido.com

[3] Yunbo Cao, Huizhong Duan, Chin-Yew Lin, and Yong Yu. (2011). Re-ranking question search results by clustering questions. J. Am. Soc. Inf. Sci. Technol. 62, 6 (June 2011), 1177–1187. https://doi.org/10.1002/asi.21529

[4] Alian, M., Al-Naymat, G. (2022). Questions clustering using canopy-K-means and hierarchical-K-means clustering. *Int. j. inf. tecnol.* https://doi.org/10.1007/s41870-022-01012-w

[5] Haponchyk, I., Uva, A., Yu, S., Uryupina, O., & Moschitti, A. (2018). Supervised Clustering of Questions into Intents for Dialog System Applications. *EMNLP*.

[6] Zhang W-N, Liu T, Yang Y, Cao L, Zhang Y, Ji R (2014). A Topic Clustering Approach to Finding Similar Questions from Large Question and Answer Archives. PLoS ONE 9(3): e71511. https://doi.org/10.1371/journal.pone.0071511

[7] Chinea-Rios, M., Sanchis-Trilles, G., Casacuberta, F. (2015). Sentence Clustering Using Continuous Vector Space Representation. In: Paredes, R., Cardoso, J., Pardo, X. (eds) Pattern Recognition and Image Analysis. IbPRIA 2015. Lecture Notes in Computer Science(), vol 9117. Springer, Cham. https://doi.org/10.1007/978-3-319-19390-8_49

[8] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. 1996. BIRCH: an efficient data clustering method for very large databases. SIGMOD Rec. 25, 2 (June 1996), 103–114. https://doi.org/10.1145/235968.233324

[9] Aggarwal, C.C., Yu, P.S. On clustering massive text and categorical data streams. Knowl Inf Syst 24, 171–196 (2010). https://doi.org/10.1007/s10115-009-0241-z