Multilabel Classification

- *Sharvil Arjunwadkar*
- *Aadil Zikre*
- *Vipul Sarode*
- *Gowtham Behara*
- *Sai Naga Venkata Lakshmi Kalyan Medavarapu*

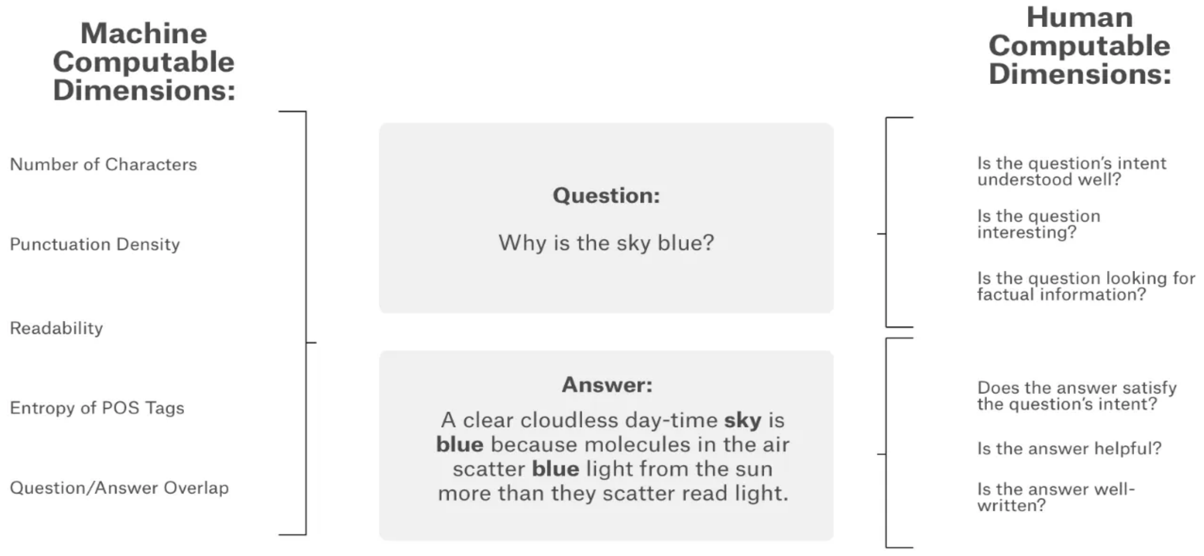
Introduction

The ever-evolving practice of Natural Language Processing (NLP) has seen some major progress and advancements in recent years, enabling computers to better understand, interpret, and generate human language. One crucial application of NLP is in the development of intelligent systems that can effectively respond to human queries.

Computers excel at providing answers to queries that have a definitive answer. However, when it comes to addressing questions that involve opinions, recommendations, or personal experiences, humans still have the upper hand. Human beings possess a superior capability for dealing with subjective questions that demand a comprehensive, nuanced grasp of the context. These questions can take many different forms, from being elaborately worded and multi-sentence to being straightforward inquiries or fully-fledged problems. They may have numerous purposes, such as soliciting guidance or viewpoints, and they can range from being helpful to merely intriguing. Certain questions may be unequivocally correct or incorrect, while others require interpretation and evaluation.

Google-QUEST Q&A Labeling is one such ambitious NLP project designed to improve the quality and relevance of question-answer interactions. Google QUEST (Quality Estimation from User-generated Text Sequences) Q&A Labeling is an innovative NLP model that leverages state-of-the-art techniques and a rich dataset to enhance the overall experience of users seeking information online. This project aims to address the challenges faced by traditional Q&A systems, such as providing incomplete or irrelevant answers, by employing cutting-edge algorithms that can better understand the nuances of user-generated content.

This paper delves into the creation and execution of the Google QUEST Q&A Labeling natural language processing (NLP) model. The article examines the model's fundamental framework, the distinctive characteristics that distinguish it from other NLP models, and how it enhances question-answer interactions across diverse online platforms. The aim of this paper is to furnish readers with an all-encompassing comprehension of the Google QUEST Q&A Labeling initiative and its possible effect on the future of information retrieval and user engagement.


Machine Computable Dimensions:
Number of Characters
Punctuation Density
Readability
Entropy of POS Tags
Question/Answer Overlap
Question:
Why is the sky blue?
Answer:
A clear cloudless day-time sky is blue because molecules in the air scatter blue light from the sun more than they scatter read light.
Human Computable Dimensions:
Is the question's intent understood well?
Is the question interesting?
Is the question looking for factual information?
Does the answer satisfy the question's intent?
Is the answer helpful?
Is the answer well-written?

<u>About Dataset</u>

The Google QUEST Q&A Labeling dataset is an extensive collection of question-answer pairs derived from various online platforms, designed to train, and evaluate models that can enhance the quality of user-generated content. This dataset was introduced as part of the Google-QUEST Q&A Labeling competition on Kaggle, with a purpose of encouraging the development of advanced NLP models for quality estimation tasks.

The dataset comprises over 6,000 question-answer pairs, sourced from Stack Exchange, a popular network of Q&A websites covering diverse topics such as programming, mathematics, and linguistics. These pairs are manually annotated by human raters to ensure the highest possible quality and relevance.

The annotation process involved the assignment of 30 different labels to each question-answer pair, spanning across nine categories. These categories include:

1. Question-related:

- Relevance
- Clarity
- Specificity
- Objectivity

2. Answer-related:

- Helpfulness
- Detail

- Relevance
- Clarity
- Completeness
- Objectivity

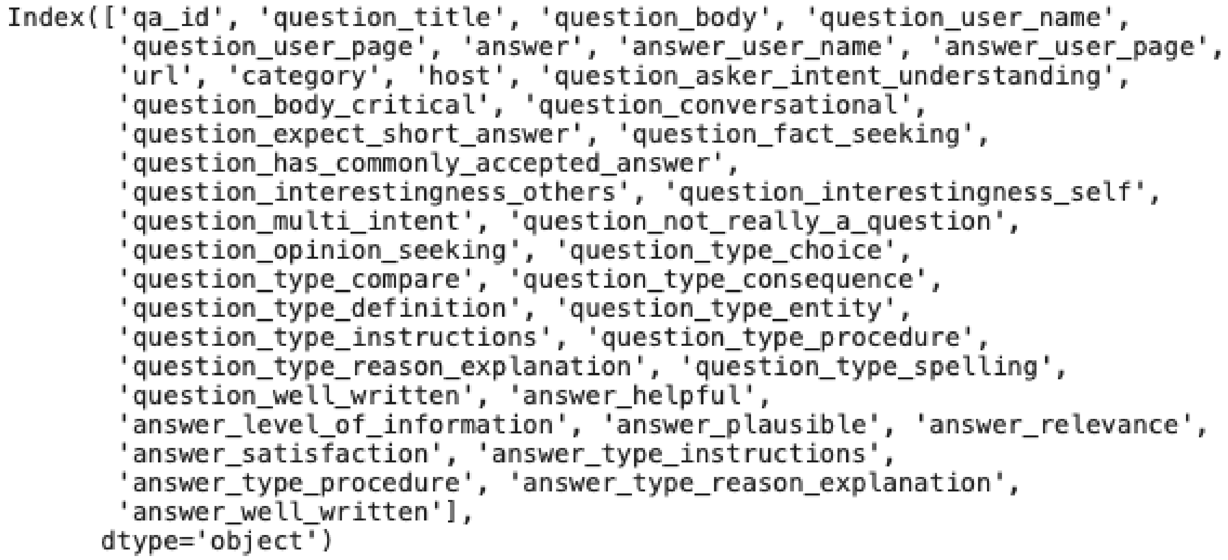
3. Overall:

- Quality
- Satisfaction

Different categories are used to fully assess the quality and helpfulness of each question-answer combination. The ratings range from 0 to 1, with higher values indicating better quality. The Google QUEST Q&A Labeling dataset is an extremely valuable tool for training and comparing NLP models for quality assessment tasks. Using this diverse and extensive dataset, models can be optimized to better understand user-generated content and provide more precise and useful responses to user questions.

<u>Exploratory Data Analysis (EDA)</u>

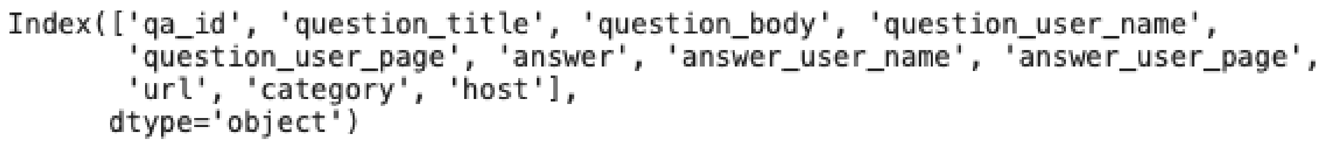
Train Data columns:



```
Index(['qa_id', 'question_title', 'question_body', 'question_user_name',
       'question_user_page', 'answer', 'answer_user_name', 'answer_user_page',
       'url', 'category', 'host', 'question_asker_intent_understanding',
       'question_body_critical', 'question_conversational',
       'question_expect_short_answer', 'question_fact_seeking',
       'question_has_commonly_accepted_answer',
       'question_interestingness_others', 'question_interestingness_self',
       'question_multi_intent', 'question_not_really_a_question',
       'question_opinion_seeking', 'question_type_choice',
       'question_type_compare', 'question_type_consequence',
       'question_type_definition', 'question_type_entity',
       'question_type_instructions', 'question_type_procedure',
       'question_type_reason_explanation', 'question_type_spelling',
       'question_well_written', 'answer_helpful',
       'answer_level_of_information', 'answer_plausible', 'answer_relevance',
       'answer_satisfaction', 'answer_type_instructions',
       'answer_type_procedure', 'answer_type_reason_explanation',
       'answer_well_written'],
      dtype='object')
```

The training data columns consist of all 30 q&a tags. We will use them to decide as the training features and training labels for out model training. By convention, the first 11 columns are used as training features and the remaining are used for categorizing the q&a tags.

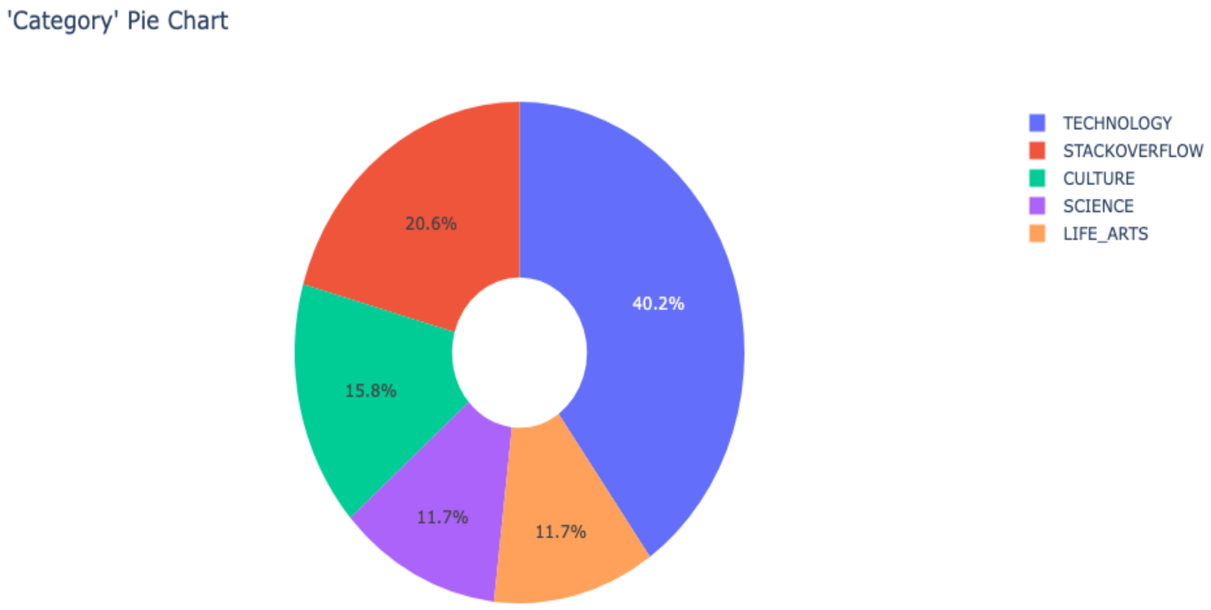
Test Data columns:



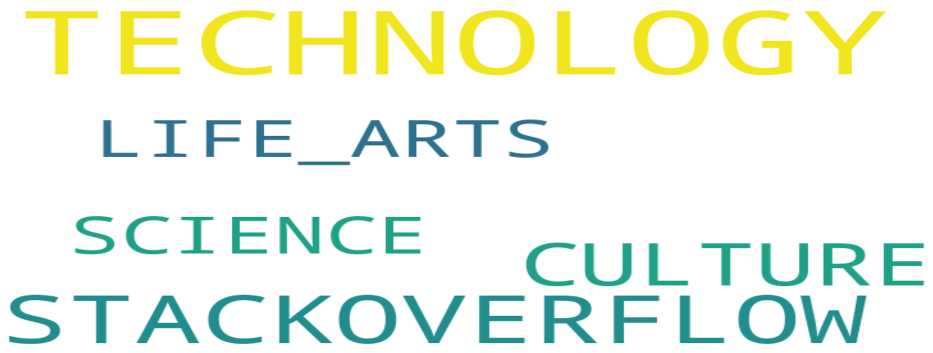
```
Index(['qa_id', 'question_title', 'question_body', 'question_user_name',
       'question_user_page', 'answer', 'answer_user_name', 'answer_user_page',
       'url', 'category', 'host'],
      dtype='object')
```

The test data columns consist of only the first 11 columns that are used for evaluating the performance of the model. We will use these features to compute the probabilities of each target label and decide the highest probability tag as the prediction to compute the accuracy of the model with actual results.

Category Pie Chart:



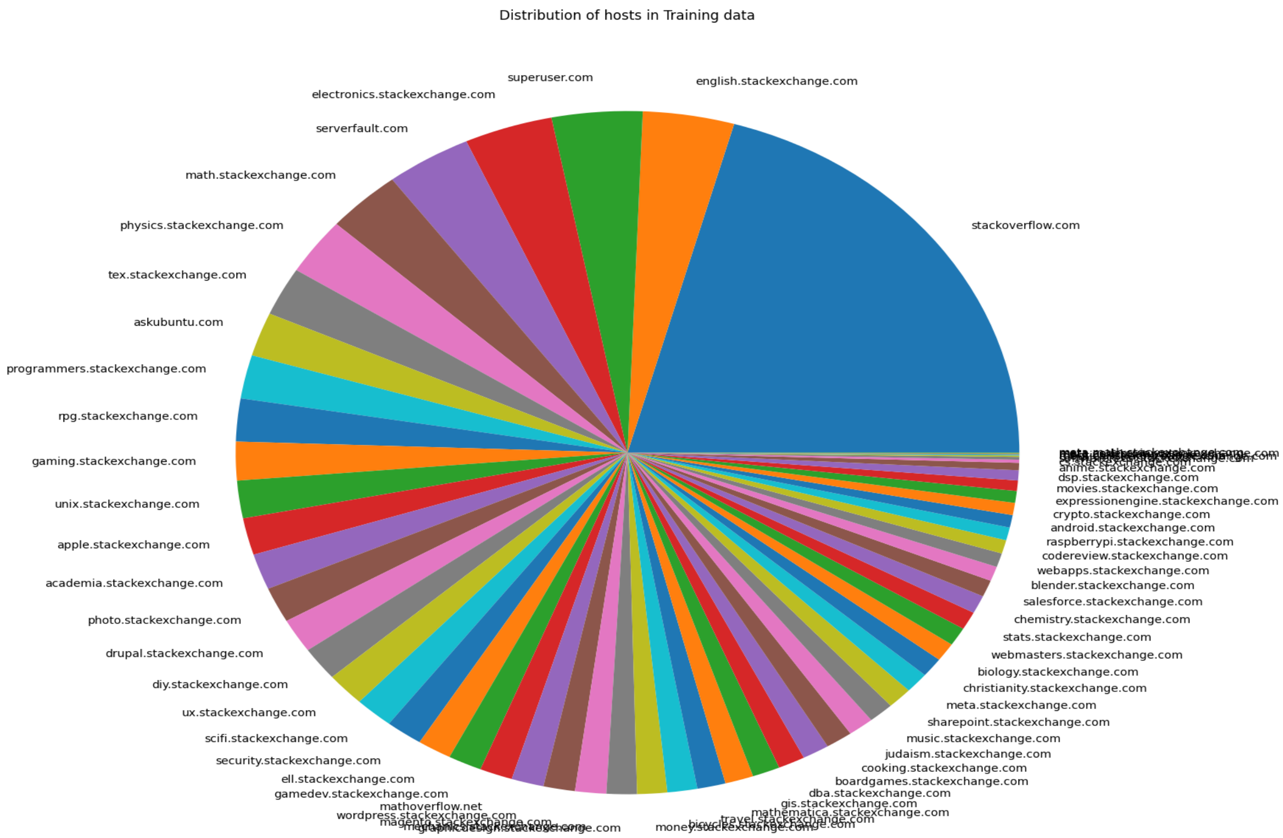
The category chart describes the percentage of each value present in the category column of the training feature. This explains how diversified the data is collected from different categories.



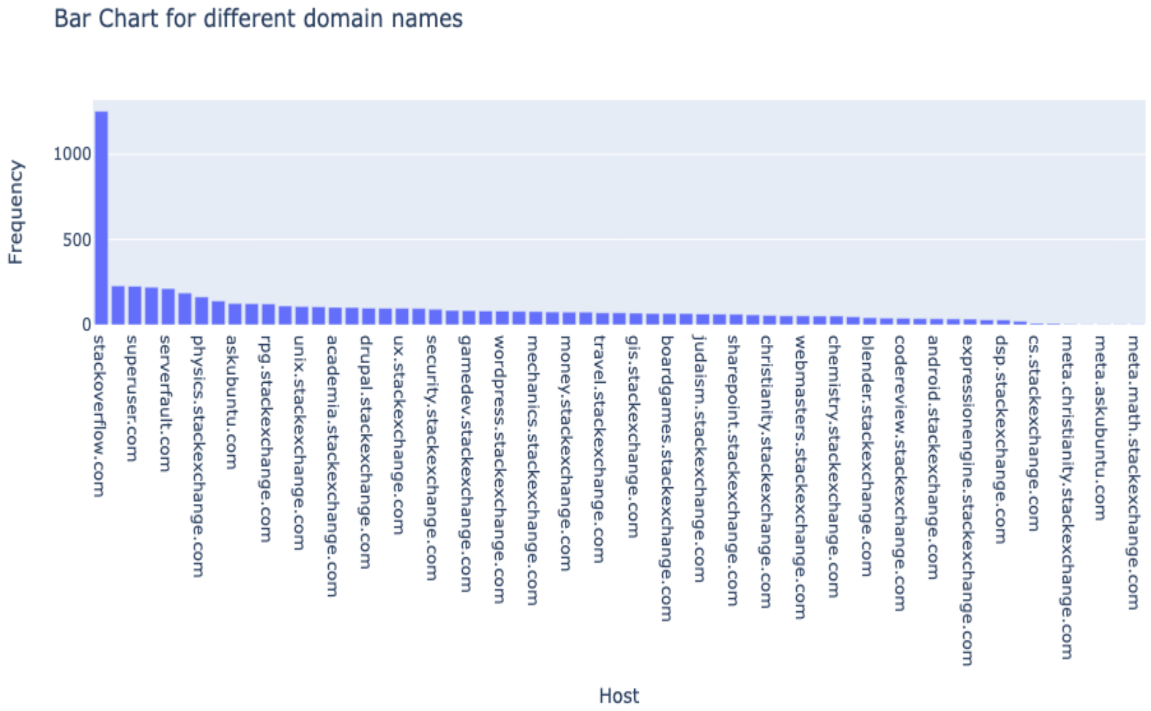
The above is a wordcloud representation of the category column values.

Distribution of hosts in Training Data:

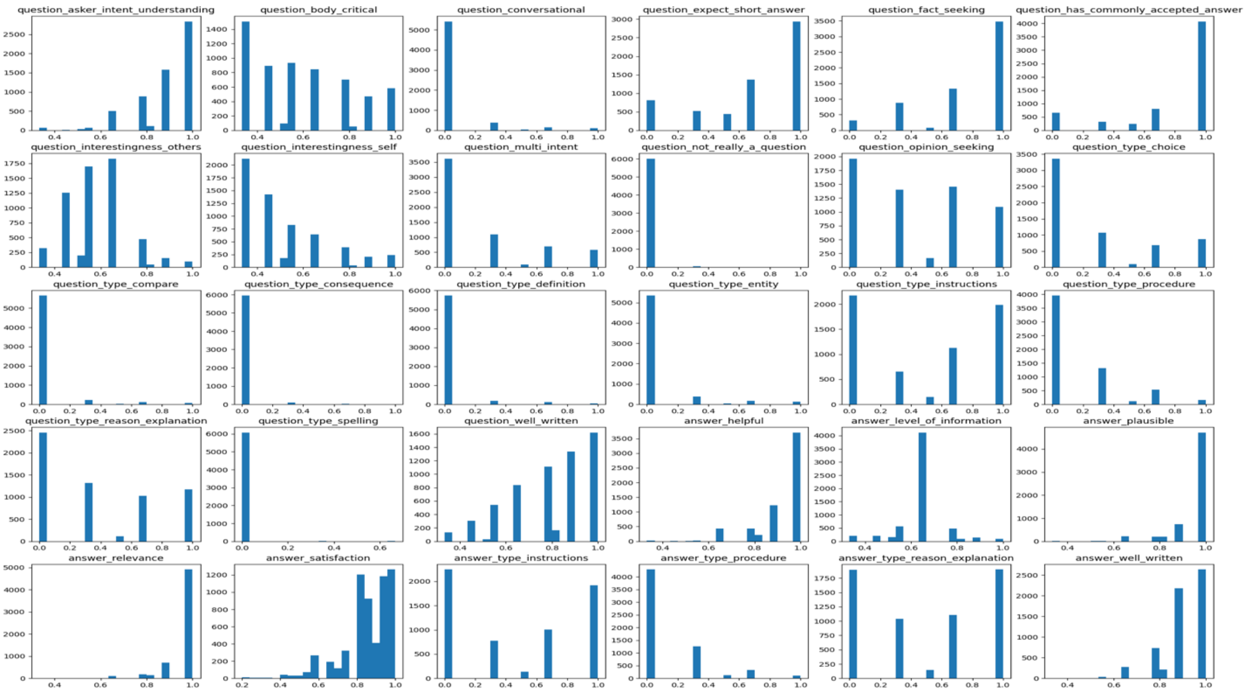
This is one more feature that we're visualizing to get a better understanding of the dataset. We can see that there are several q&a forums considered to extract the content from discussion forums. Stackoverflow occupies the highest share whereas all the subdomains of stackexchange like english.stackexchange.com and superuser.com and others are the next leading hosts.



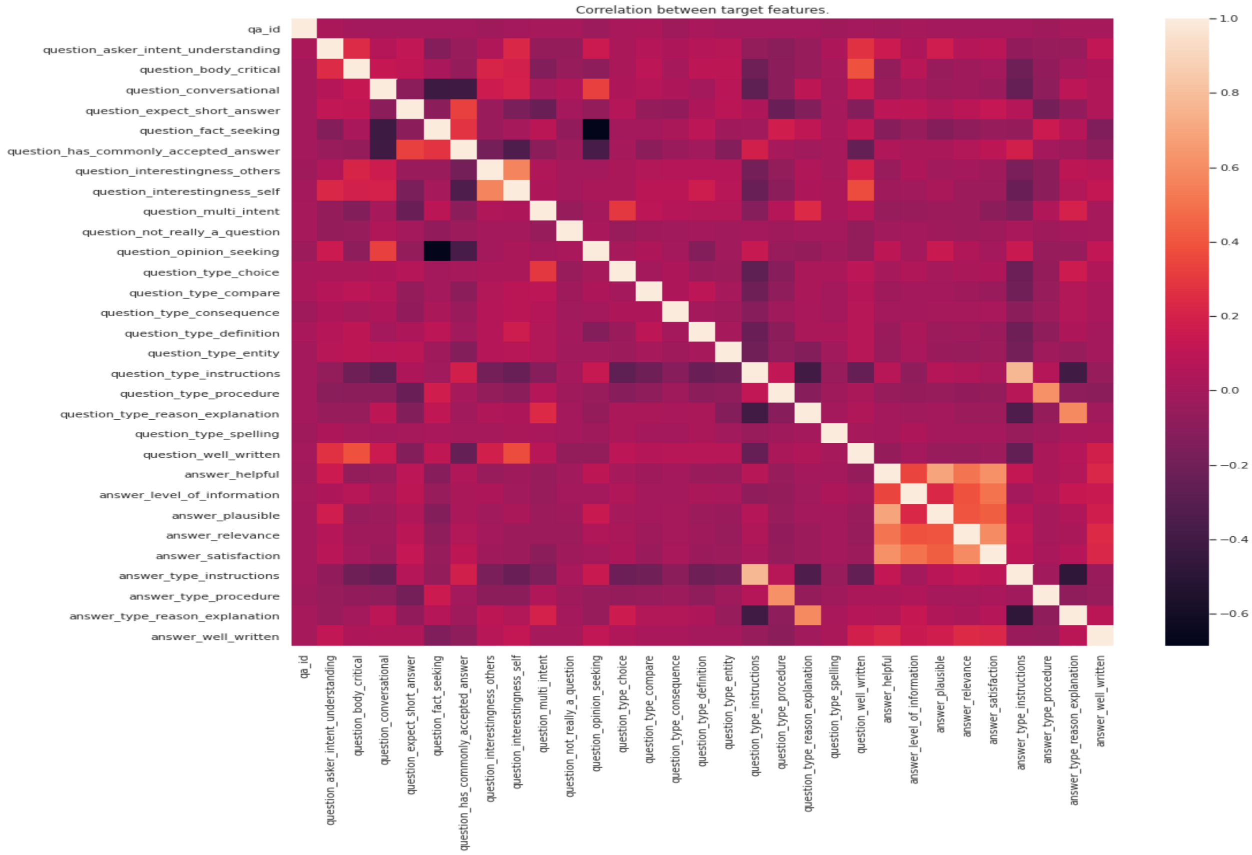
Distribution of hosts in Training data



Distribution of Target variables:



Heatmap:



From the heatmap, we can see that tags that closely related to each other have higher correlation value. For example, ‘question_type_instructions’ and ‘answer_type_instructions’ are highly correlated values and hence, have highest degree of correlation. We can notice a similar kind of pattern being present between ‘question_type_procedure’ - ‘answer_type_procedure’ and ‘question_type_reason_explanation’ – ‘answer_type_reason_explanation’.

Our assigned undertaking was to predict 30 specific target labels based on the given question_title, question_body, and answer. However, it's worth noting that the initial 21 target labels pertain exclusively to the question_title and question_body, and are not influenced by the answer provided. Conversely, the remaining 9 target labels relate solely to the answer, but a few of them still consider the information provided in the question_title and question_body.

Architecture and Model Training Results

1. Baseline Model : 2 Dense Layer on Stacked Input

For the baseline, we made the dense representation of 4 texts - question_title, question_body, category and answer- depending on the label we were trying to make the model for. For these dense representations we used bert uncased pretrained model. For the representation, we used the pooler output from the bert output. Pooler Output is the processed representation of CLS Token that is appended to each sentence and research has showed that it effectively captures the essence of the sentence well. For making the input, we stacked the embeddings on top of each other to get 1 big embedding of size 3072 when the label to be predicted was answer related and 2304 when the label to be predicted was question related. We passed this input to first hidden layer which was a dense layer with 128 output nodes. Activation we used in this layer was tanh. For the second layer, we used another dense layer with 64 output nodes. Same as the first layer, activation function we used was tanh. For the output, we had 2 output logit on which we applied sigmoid activation to get the probabilities for each class (2 in each case as the labels are binary). Loss we used was binary-crossentropy and we used the default learning rate. For training, we used a batch size of 64. Dropout Layers were also added between Dense Layers to avoid overfitting.

For each label, we made a separate model. While preparing the data for the model, we also used random oversampling for the minority class to handle label imbalance problem. Each Model for trained for 10 epochs. Loss Metric we used was Accuracy and last epoch metrics are summarized in the table below.

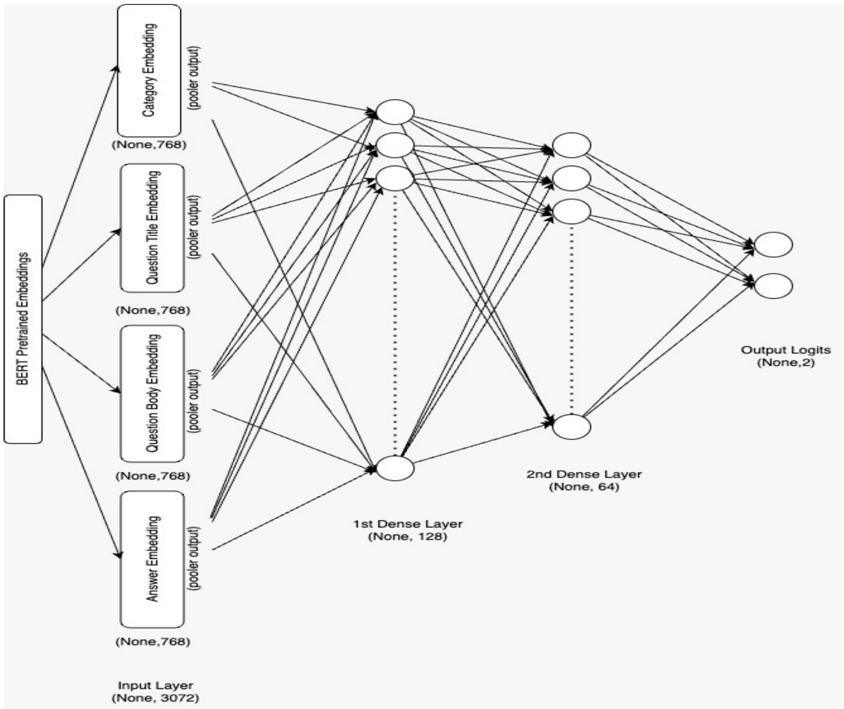
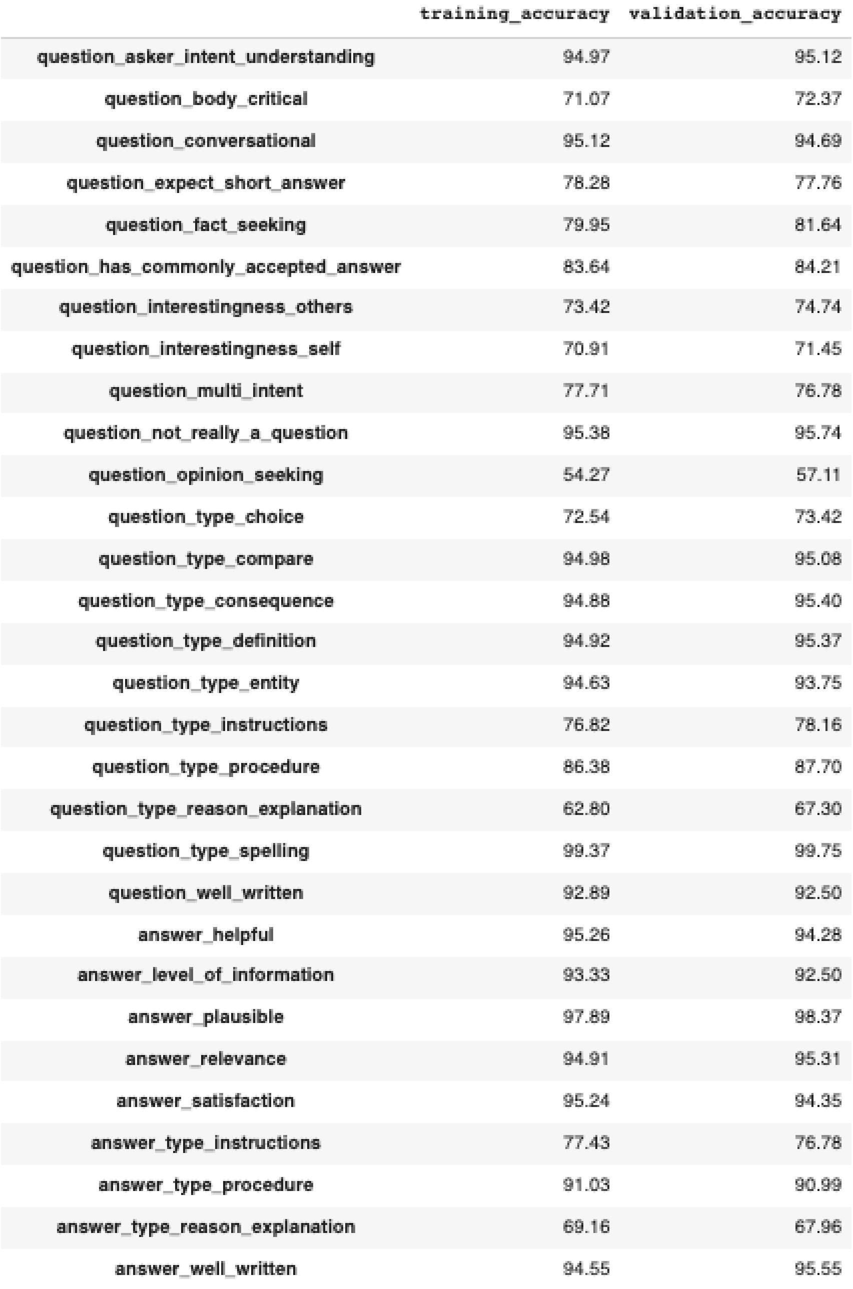


Fig: Architecture for the baseline model



	training_accuracy	validation_accuracy
question_asker_intent_understanding	94.97	95.12
question_body_critical	71.07	72.37
question_conversational	95.12	94.69
question_expect_short_answer	78.28	77.76
question_fact_seeking	79.95	81.64
question_has_commonly_accepted_answer	83.64	84.21
question_interestingness_others	73.42	74.74
question_interestingness_self	70.91	71.45
question_multi_intent	77.71	76.78
question_not_really_a_question	95.38	95.74
question_opinion_seeking	54.27	57.11
question_type_choice	72.54	73.42
question_type_compare	94.98	95.08
question_type_consequence	94.88	95.40
question_type_definition	94.92	95.37
question_type_entity	94.63	93.75
question_type_instructions	76.82	78.16
question_type_procedure	86.38	87.70
question_type_reason_explanation	62.80	67.30
question_type_spelling	99.37	99.75
question_well_written	92.89	92.50
answer_helpful	95.26	94.28
answer_level_of_information	93.33	92.50
answer_plausible	97.89	98.37
answer_relevance	94.91	95.31
answer_satisfaction	95.24	94.35
answer_type_instructions	77.43	76.78
answer_type_procedure	91.03	90.99
answer_type_reason_explanation	69.16	67.96
answer_well_written	94.55	95.55

Table: Baseline Model Accuracies for Each Label at the 10th Epoch for Training

2. **Final Model : BiLSTM Classifier with 2 Hidden Dense Layers**
   For the improved model, we chose to take a different approach to classification. To further efficiently condense the information in the different embedded vectors, we decided to utilize BiLSTM Architecture. The construct was such that each text was put as a time step and each time step had 768 features (which are nothing but the pooler output features for each text). We chose Bi-LSTM to capture context from both ends appropriately. Each LSTM had output nodes of 64. Attached to the output of BiLSTM

was 1st Dense Layer which had 128 output nodes. Activation function we used for this layer was ReLU. For the 2nd Dense Layer, output nodes were 64 and activation function was ReLU again. For the last layer, the output layer, the nodes were 2 as each label had 2 classes. Activation function in the final layer was sigmoid to preserve the probabilities. Dropout Layers were also added between the dense layers to avoid overfitting.
For each label, we made a separate model. While preparing the data for the model, we also used random oversampling for the minority class to handle label imbalance problem. In this case, we tried two different rates for oversampling. In one case oversampling was done in such a way that the minority label was atleast 5% of the total records for any particular label in consideration. In the other one, the ratio was upped to 15%. Each Model for trained for 10 epochs. Loss Metric we used was Accuracy and last epoch metrics are summarized in the table below.

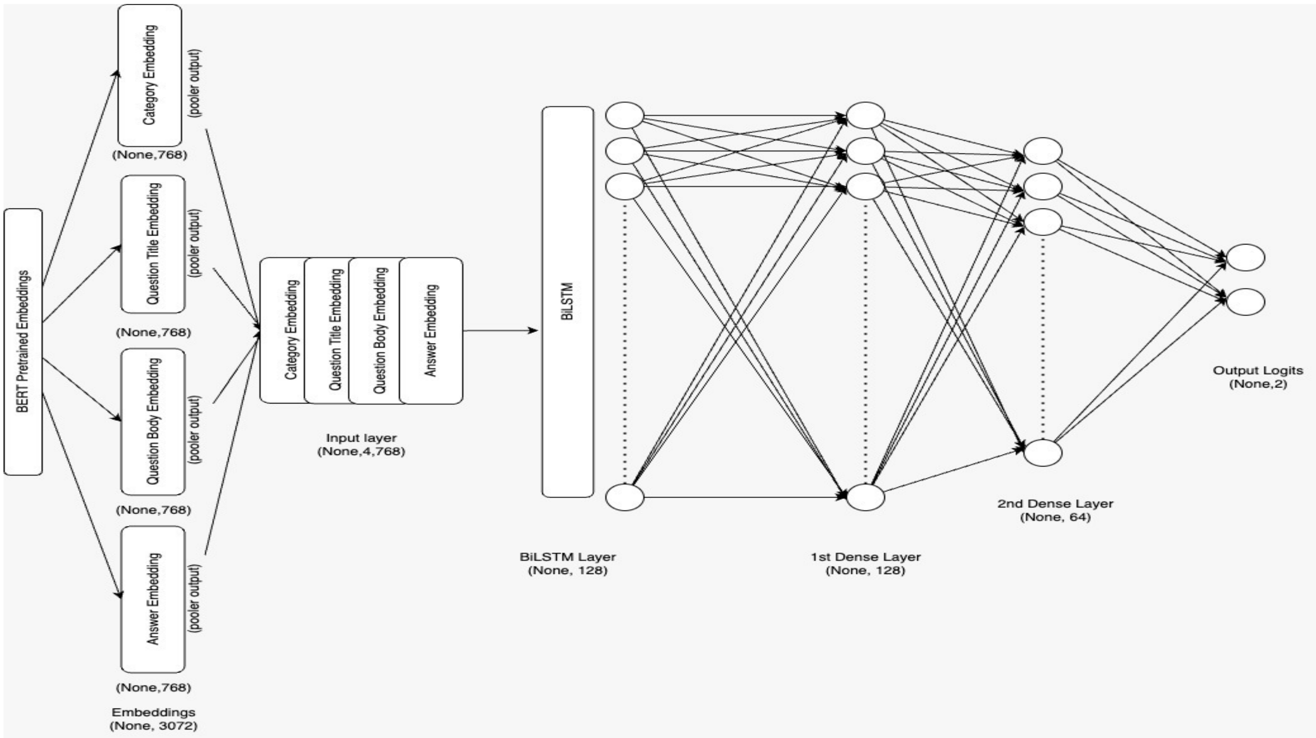
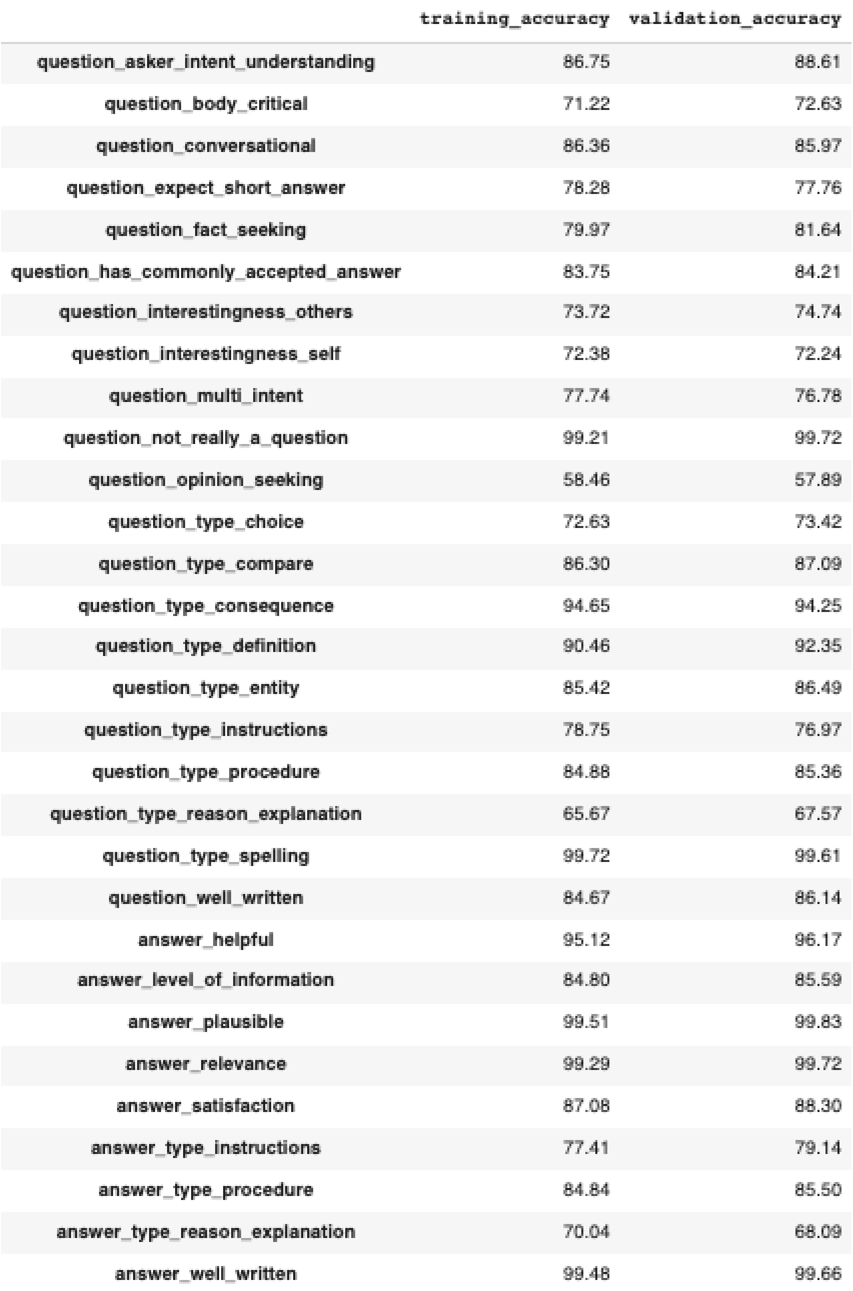


Fig: Architecture for the BiLSTM Model



	training_accuracy	validation_accuracy
question_asker_intent_understanding	86.75	88.61
question_body_critical	71.22	72.63
question_conversational	86.36	85.97
question_expect_short_answer	78.28	77.76
question_fact_seeking	79.97	81.64
question_has_commonly_accepted_answer	83.75	84.21
question_interestingness_others	73.72	74.74
question_interestingness_self	72.38	72.24
question_multi_intent	77.74	76.78
question_not_really_a_question	99.21	99.72
question_opinion_seeking	58.46	57.89
question_type_choice	72.63	73.42
question_type_compare	86.30	87.09
question_type_consequence	94.65	94.25
question_type_definition	90.46	92.35
question_type_entity	85.42	86.49
question_type_instructions	78.75	76.97
question_type_procedure	84.88	85.36
question_type_reason_explanation	65.67	67.57
question_type_spelling	99.72	99.61
question_well_written	84.67	86.14
answer_helpful	95.12	96.17
answer_level_of_information	84.80	85.59
answer_plausible	99.51	99.83
answer_relevance	99.29	99.72
answer_satisfaction	87.08	88.30
answer_type_instructions	77.41	79.14
answer_type_procedure	84.84	85.50
answer_type_reason_explanation	70.04	68.09
answer_well_written	99.48	99.66

Table: BiLSTM Model Accuracies for Each Label at the 10th Epoch for Training

While we noticed a few labels where the accuracy reduced in the final model, that is due to the fact that the new model was much more robust to label imbalances and hence produced slightly less accuracy. This is especially true for the labels where the label imbalance was huge like question_answer_intent_understanding and answer satisfaction.

Prediction

Prediction was done on the test set following the same regime. In the output, we get uncaliberated probabilities which can be used with calibration. Or they can also be converted into Binary Classes depending on the use case.

Conclusion

Q&A tags labeling provides several benefits, including:

1. Improved accuracy: By categorizing questions and answers with specific tags, machine learning models can better understand the content, context, and relevance of user-generated content, leading to more accurate and relevant responses.

2. Quality assessment: Q&A tags labeling allows for the assessment of the quality and usefulness of each question-answer pair based on predefined criteria, such as accuracy, completeness, specificity, and relevance.

3. Topic modeling: By tagging questions and answers with specific topics or categories, Q&A tags labeling enables the identification of common themes and patterns within user-generated content, facilitating topic modeling and content analysis.

4. Personalization: Q&A tags labeling can be used to personalize responses to user queries based on their preferences, interests, and past behavior, leading to a better user experience and increased engagement.

Overall, Q&A tags labeling is a crucial component of question answering systems that aims to enhance the quality, accuracy, and relevance of user-generated content.

Future Use Cases

The development of the Google-QUEST Q&A Labeling NLP model holds significant potential for various future use cases that can enhance user experiences and streamline information retrieval processes across numerous domains. Some of these use cases include:

1. **Improved search engines**: Incorporating the Google QUEST Q&A model into search engine algorithms can lead to more accurate and relevant search results, by better understanding the quality of user-generated content and serving higher-quality answers to user queries.

2. **Enhanced customer support**: The model can be integrated into chatbots and virtual assistants to improve their ability to provide precise, helpful, and comprehensive responses to user inquiries, reducing response times and improving customer satisfaction.

3. **Online educational platforms**: The NLP model can be employed to identify high-quality explanations, solutions, or study materials, enabling students and educators to access reliable and accurate information more efficiently.

4. **Content moderation**: The model can be utilized to automatically evaluate and moderate user-generated content on online forums, social media platforms, and Q&A websites, ensuring that the information provided is relevant, clear, and helpful.

5. **Personalized recommendations**: By understanding user-generated content's quality, the model can be used to develop personalized recommendation systems that suggest high-quality resources, articles, or discussions tailored to users' interests and preferences.

6. **Corporate knowledge bases**: The Google QUEST Q&A model can be applied to optimize internal knowledge management systems within organizations, facilitating the quick retrieval of accurate and relevant information by employees.

7. **Medical consultations**: The NLP model can be integrated into medical chatbots or online health platforms to provide users with reliable, high-quality health advice and recommendations.

8. **Sentiment analysis**: By evaluating the quality of user-generated content, the model can be further adapted to understand user sentiments and emotions, enabling businesses to gain insights into customer opinions and make more informed decisions.

9. These use cases highlight the potential impact of the Google QUEST Q&A Labeling NLP model on various industries and applications, ultimately contributing to more efficient and effective information retrieval and enhancing user experiences across the board.

References:

- Explanation of BERT Model - NLP - GeeksforGeeks
- Word Embeddings in NLP - GeeksforGeeks
- Google QUEST Q&A Labeling | Kaggle
- Transformers (State-of-the-art Natural Language Processing) | by Sarthak Vajpayee | Towards Data Science
- Understanding BERT — (Bidirectional Encoder Representations from Transformers) | by Sarthak Vajpayee | Towards Data Science
- Long Short Term Memory | Architecture Of LSTM (analyticsvidhya.com)