Group11: Google QUEST Q&A Labeling

Yingzhuo Yu (yy56), Yuhong Shao (yuhongs2), Zixing Deng (zixingd2)

1 ABSTRACT

This project is adapted from Kaggle's *Google QUEST Q&A Labeling*[1] competition, which will focus on using different machine learning (ML) methods to improve the automated understanding of complex question-answer (Q&A) content. By using the dataset from Kaggle, which was collected by CrowdSource team at Google Research, we will predict target values of the ten most representative labels among thousands of questions and answers from various StackExchange properties. There are originally 41 variables in the dataset, and 30 of them are response variables. After exploratory data analysis, we find out the statistics and distribution of explanatory variables. Also, by using the Variance Inflation Factor (VIF), we narrowed down the response variables into 10 variables. We implemented three different models combining with BERT (Support Vector Machine (SVM), Multilayer Perceptron (MLP), and XGBoost Regression) and compare each model's performance to gain the most accurate results. Our final result achieved by BERT+MLP performs a satisfactory job with average Spearman's correlation coefficient score **0.3302** on 10 target variables.

2 INTRODUCTION

Starting from 1978[9], various online discussion forums enabled people to gather information and solve problems, either self-resolved or crowdsourced-solved. With the improvements in technology, scientists began to try to use machines and algorithms to answer questions in those Question-Answering (Q&A) communities. However, computers cannot handle subjective questions as perfectly as objective questions since subjective questions require multidimensional understanding. In this project, we will use Kaggle's [1] dataset to further develop algorithms for subjective aspects of Q&A and provide predictions for quantitative scores for the qualities of Q&As.

3 MOTIVATION

The ML techniques have become increasingly popular and relevant to solve Natural Language Processing (NLP) related problems in recent years, but machines are less helpful when encountering complicated situations that require a deeper and more complex understanding of the contexts. Professionals are still working on building better subjective Q&A algorithms to understand the multidimensional context. By completing this project, we aim to contribute to future intelligent Q&A systems becoming more human-like.

4 RELATED WORK

For exploratory analysis of text data, Etemadi et.al used the Apriori algorithm to conduct association rule mining on the topics of questions based on keywords in order to find out related topics that appear in a question [6]. Topics were extracted using Latent Dirichlet Allocation (LDA) proposed by Blei et. al [3], which is a Bayesian probabilistic model that extracts a number of topics represented by a list of keywords given a document term matrix. For predictive analysis, Wu et.al [11] stated that professional question contains various terminologies, making feature extracting harder, thus requiring the combination of text-matching BERT[5] model with a boosted tree to improve model accuracy. Annamoradnejad et.al [2] proposed a data preprocessing procedure that tokenizes original data utilizing the Python library *'huggingface transformers'* so that predictive models can be subsequently built using a pretrained BERT model.

In terms of evaluation metrics, Winter et al. conducted an empirical experiment comparing the performance of the more common Pearson Correlation Coefficient to that of the Spearman's Rank Correlation Coefficient and concluded that for heavily tailed distributions, Spearman's Rank Correlation Coefficient has more robust performance [4].

5 METHODOLOGY

Exploratory Data Analysis To examine the dataset and have an overview of the dataset, we performed exploratory data analysis (EDA) for both response variables (quantitative) and explanatory variables (qualitative) by practicing methods that were mentioned during the class. Since the dataset does not contain null values, we do not worry about dealing with them. We utilize a five-number summary as well as mean and standard deviations for our response variables inspection.

Due to limited computational power, we also implemented the variable selection metric Variance Inflation Factor (VIF) and feature selection method Principle Component Analysis(PCA) to select a subset of the response variables and build a predictive model based on these variables. Traditionally, VIF is used to measure multicollinearity issues in linear regression analysis. However, here we use VIF to measure the strength of the linear relationship between one response variable and all other response variables. In other words, we aim to measure the information gain introduced by every response variable, and then select a small subset of response variables that introduce as much new information as possible. To empirically verify the effectiveness of this approach, we also set up an experimental procedure as illustrated in (Fig.1). We compare the prediction performance between the following two methods: 1) training a linear regression model using all response variables; 2) setting up two regression models, one predicts the chosen response variables given training data, and the other predicts the remaining response variables given the chosen response variables. By applying the prediction of chosen variables between these two models, we would also be able to gain a complete prediction of all response variables.

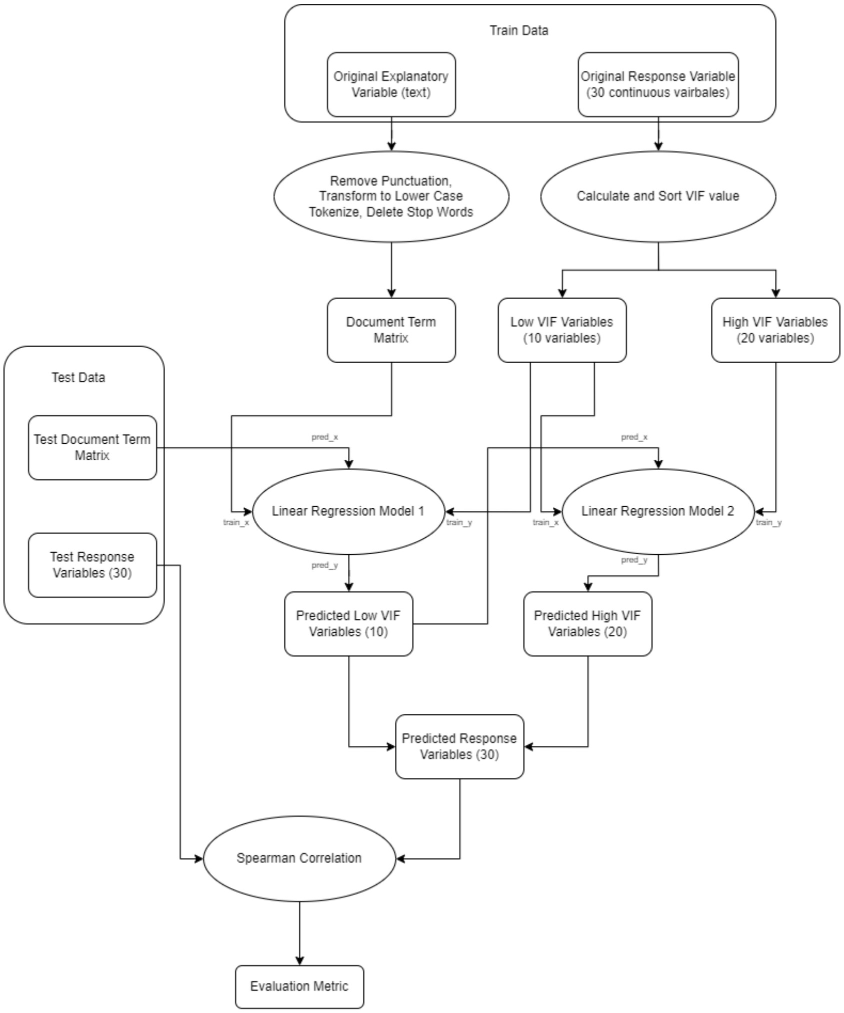

Train Data
Original Explanatory Variable (text)
Original Response Variable (30 continuous vairbales)
Remove Punctuation, Transform to Lower Case Tokenize, Delete Stop Words
Calculate and Sort VIF value
Document Term Matrix
Low VIF Variables (10 variables)
High VIF Variables (20 variables)
Test Data
Test Document Term Matrix
Test Response Variables (30)
pred_x
pred_x
train_x
Linear Regression Model 1
train_y
train_x
Linear Regression Model 2
train_y
pred_y
pred_y
Predicted Low VIF Variables (10)
Predicted High VIF Variables (20)
Predicted Response Variables (30)
Spearman Correlation
Evaluation Metric

Fig. 1. Empirical verification procedure

As our explanatory variables are qualitative, we use bar charts and distribution plots to discover the overall patterns of the data as well as the patterns in different categories. Moreover, we also perform the statistics overview (i.e. five-number summary, mean, standard deviation) for explanatory variables by counting the occurrences of the data in each category.

Model Architecture To capture the contextual meaning of the input questions and answers, we applied BERT[5] as the backbone encoder. Motivated by BERT fine-tuned on Question Answering task, we used the same input format as [CLS], question tokens, [SEP], and answer tokens. In this way, the BERT language model is able to differentiate two parts of the input and analyze the contextual connection bidirectionally. To fine-tune the pretrained model on our dataset, regression techniques are applied to the average of the last three hidden states embedding from BERT. In this project, we decided to conduct experiments on three different models: BERT+Mutlilayer Perceptron, BERT+SVM, and BERT+XGBoost Regression. 1) For Multilayer Percetion, it's a class of feedforward artificial neural networks that stack multiple layers of perceptron. To make the mapping function non-linear, we used the ReLU function as the activation function. And since the target value ranges from 0 to 1, we add a sigmoid function to the output layer. 2) For SVM, we mentioned its usage as a classification in the lecture. Since it tries to maximize the margin to find the optimal boundary between different classes, it performs better than simple linear classification. Using the same principle as SVM, Support Vector Regression tries to find the best fit hyperplane to the data points within a threshold value. Since it also applies kernel techniques that map the data into higher dimensions, we expect this model can yield good regression performance. 3) XGBoost, Extreme Gradient Boosting is a gradient boosting technique that gives the prediction model by the ensemble of weak prediction models. It can be directly used for regression modeling and we expect this gradient-boosting algorithm can help improve our model performance.

6 EMPIRICAL RESULTS

6.1 EDA Results

As mentioned in previous parts, among 41 variables in the dataset[1], we have 30 response variables. For response variables selection, we first plotted the correlation heatmap and noticed that there exist signs of high correlation (greater than 0.6) between variables. Then we calculated the VIF for each variable and selected the top ten response variables (i.e., *question_type_spelling*, *question_not_really_a_question*, *question_type_consequence*, *question_type_compare*, *question_type_definition*, *question_type_entity*, *question_conversational*, *question_multi_intent*, *question_type_choice*, *answer_type_procedure*) as our research focus. Then, we compared the prediction performance measured by Spearman's Correlation Coefficient. The score for full model regression is **0.1393**, and the score for the reduced model regression is **0.1076**, which is not significantly lower, indicating that when computational power is limited, it is acceptable to trade some accuracy for simplicity. After variable selection, we performed the five-number summary for response variables. Observed from the summary, the minimum and median for all variables are 0, and the maximum for all variables is 1 except for *question_type_spelling*, which is approximately 0.67. Moreover, most variables' third quartile is 0 except for the last three variables (*question_multi_intent* (0.33), *question_type_choice* (0.67), *answer_type_procedure* (0.33)), indicating that most data points lay in the last 25-percentile. After applying PCA analysis to response variables, we don't see any variables having a significant contribution to a principle component. In other words, the transformed response variables would no longer be interpretable. Therefore, we decided to discard this feature extraction method.[2]

For the rest 11 explanatory variables, we put more focus on the fraction of categories in the dataset. There are five categories in the dataset, and we can see that *Technology* is about 40% of the total samples, while the other four categories are less than 20% (Fig.2). We also found some duplicate questions in the dataset, indicating that several answers are corresponding to one question. Additionally, we count the words in the question and the answer body and make them into distribution plots (Fig.3). The results show that most question body contains about 81 words and most answer body contains either 30 or 69 words. Nevertheless, the maximum words and minimum words have significant differences.[2]

[1] Please see the full dataset at https://www.kaggle.com/competitions/google-quest-challenge/data

[2] For more detailed information, please refer to the appendix page *Response Variables and Explanatory Variables Overview* section.

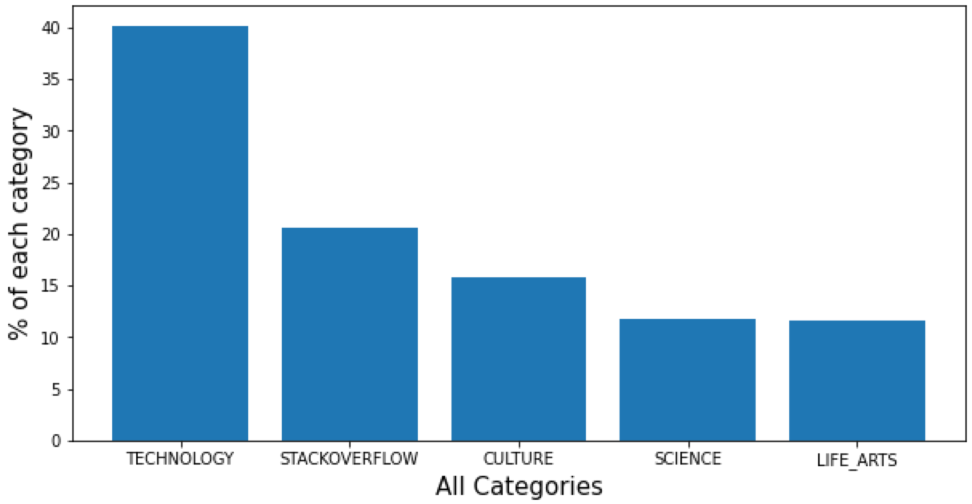


Fig. 2. Bar chart for percentages of each category

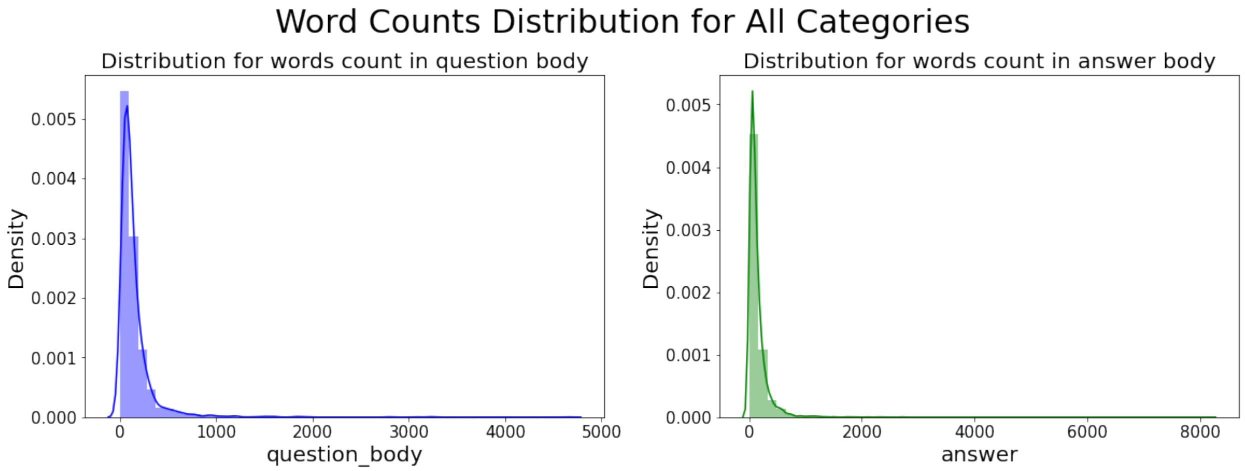


Fig. 3. Question body[Left] and answer body[Right] words count distribution for all categories.

6.2 Implementation Details

To obtain the best model among the three for predicting the results, we split the training dataset into the train(80%), validation(10%), and test(10%) subsets. After using the training dataset to train the initial model, we used the validation dataset to tune the hyper-parameters for each model by performing grid searches. We finally used Spearman's correlation coefficient as the metrics to compare the performance of BERT+MLP, BERT+SVM, and BERT+XGBoost models. In addition, we conducted a data cleaning step and removed some special characters that may not be recognized by BERT.

BERT + MLP For the first model combination, we added a multilayer perceptron to the embedding from BERT. We first added the dropout layer that randomly dropped the parameters from the BERT embedding with the probability of 0.5 to avoid overfitting and make the model more robust. We then input the embedding into one hidden layer with shape 768×4000 and ReLU activation function. Finally, the output layer with shape 4000×10 outputs ten prediction values at the same time, and each output is applied sigmoid function to make the value range from 0 to 1. Besides, we initialized the weight of the hidden layer with the Kaiming initialization method[7], which takes the non-linearity of ReLU into account. Finally, we used backpropagation to train our model with the criterion Mean Squared Error Loss to minimize the error between the prediction values and target values. The hyperparameters are listed in Table 1.

BERT + SVM and **BERT + XGBoost** For the next two model combinations, we built separate regression models for each target value on BERT output embeddings using SVM and XGBoost. Since we have ten response variables in total, for both SVM and XGBoost, we train ten sub-models, each corresponding to one response variable. Note that for the first response variable *question_type_spelling*, since approximately 98% of the response variable is 0, the validation data and test data may be completely consist of 0, thus making it impossible to calculate a Spearman's correlation coefficient, for it depends on the variation within data. Therefore, we drop the first response variable when building a predictive model.
In terms of tuning hyper-parameters, we use Spearman's correlation coefficient as the evaluation metric. For each sub-model, we first conduct grid search with 2-fold cross-validation on training data to obtain three candidate sub-models with highest cross-validation scores. Then, we re-train the three candidate sub-models using all train data(80%), evaluate the sub-model based on validation data(10%), and obtain the final sub-model based on Spearman's correlation coefficient between true response variable and model prediction. Finally, we combine all ten sub-models together, and compute the overall Spearman's correlation coefficient for test data(10%). The relevant hyperparameters are shown in Table 2 and Table 3.

Hyperparameter	Value
Batch size	12
Learning Rate	4e-5
Optimizer	AdamW
Bert Model	bert-base-cased
Criterion	Mean Squared Error Loss
Maximum Token Number (question)	300
Maximum Token Number (answer)	209

Table 1. BERT+MLP Hyperparameters

Hyperparameter	2	3	4	5	6	7	8	9	10
Kernel	sigmoid	poly	rbf	rbf	sigmoid	linear	rbf	sigmoid	poly
Regularization Parameter	4	9	2	2	7	9	1	7	2

Table 2. SVM Hyperparameters for each sub-model

Hyperparameter	2	3	4	5	6	7	8	9	10
Maximum Depth	10	20	5	10	10	10	40	5	15
Number of Estimators	20	110	6	300	30	20	30	15	4
Learning Rate	1e-3	1e-2	1e-3	1e-3	1e-2	1e-2	1e-3	1e-3	1e-2

Table 3. XGBoost Hyperparameters for each sub-model

6.3 Final Model Results

We have the results of three models in table 4. With the BERT+MLP model obtaining the highest Spearman's correlation coefficient score **0.3302**, we take the BERT+MLP model as our final output.

Domain	BERT+MLP	BERT+SVM	BERT+XGBoost
Training set	0.3686	0.1331	0.3904
Validation set	0.3439	0.0592	0.0002
Test set	**0.3302**	0.0335	0.0309

Table 4. Three Models Performance on training, validation and test set

To further show the performance of the BERT+MLP model on questions and answers for different categories, we separately calculated Spearman's correlation coefficient score for each category shown in Fig. 4. The plot shows that the prediction on *Life Art* has the highest Spearman's correlation while *StackOverFlow* has the lowest performance. Generally, the two categories *Stack Overflow* and *Technology* tends to have relatively lower Spearman's correlation coefficient score than that of the rest three categories. Combine this information with the fact that there exists some special characters that are not English words as we have mentioned in the Empirical Results section, we speculate that the reason for model not being able to make accurate predictions for categories *Stack Overflow* and *Technology* is because questions and answers in these two categories typically contain more special characters. Therefore, to improve model performance in the future, one possible direction would be conducing more thorough data cleaning.

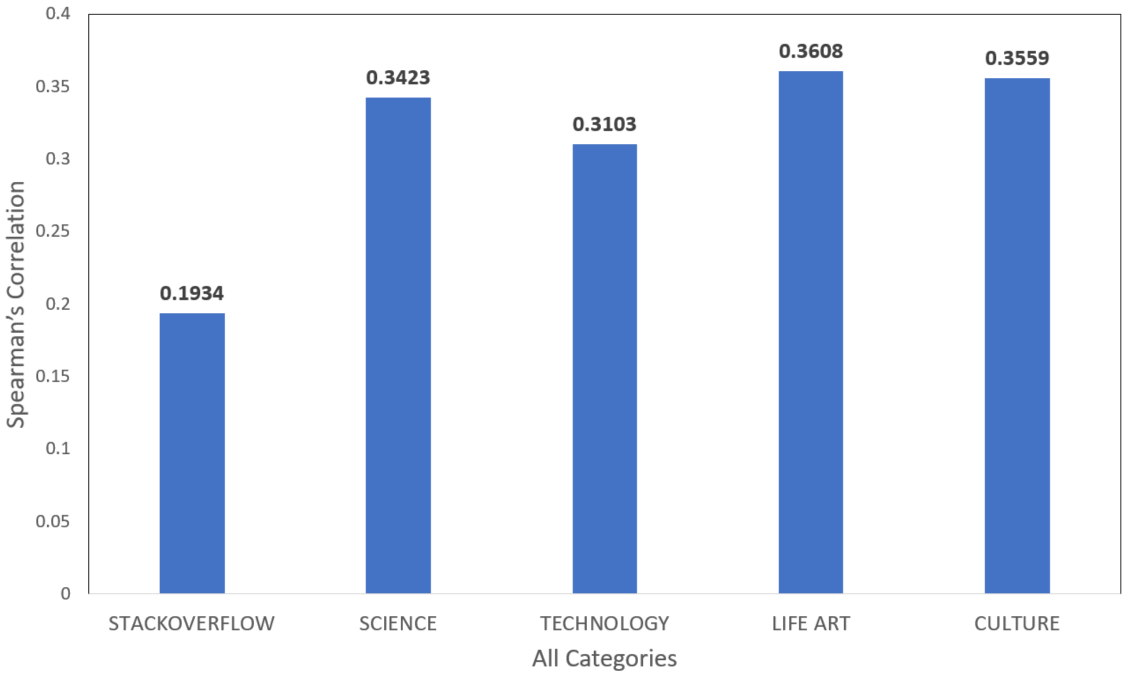


Fig. 4. BERT+MLP model spearman's correlation score in each category

7 CONCLUSIONS/DISCUSSIONS

Overall, our final model performance reached the Spearman's correlation of **0.3302** for the whole dataset. For the relatively low Spearman's correlation 0.1934 for *StackOverFlow* category in the final model, one of the possible causes for the result might be the limitation of BERT when dealing with special character tokens of complex math formula in Latex. Due to the time limit and the computation limit, we were not able to try more methods to deal with the situation as well as carry out different models for respective categories. Besides, we also noticed that the token number of questions and answers is much longer than the maximum input token numbers of BERT, 512 and thus, the truncation of the input text may make us lose much information. For future investigation, we may want to introduce MathBERT[10] to deal with the special character problem and ColBERT[8] to efficiently deal with long paragraph input to make the results more precise as well as dig into training different models for various categories. In a nutshell, although our final results are not as good as Kaggle Competition Winner 0.4628, we had a chance to practice what we learned from the class and dive deep into different regression models. As NLP techniques are developing rapidly, as we can see from the latest chatbot *ChatGPT* by OpenAI. Therefore, we will keep an eye on the changes in different ML methods and apply them to solve more real-world problems.

REFERENCES

[1] 2019. Google QUEST Q&A Labeling. https://www.kaggle.com/competitions/google-quest-challenge/overview

[2] Issa Annamoradnejad, Mohammadamin Fazli, and Jafar Habibi. 2020. Predicting subjective features from questions on Q&A websites using bert. *2020 6th International Conference on Web Research (ICWR)* (2020). https://doi.org/10.1109/icwr49608.2020.9122318

[3] A. Y. Ng D. M. Blei and M. I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.* 3 (2003). https://doi.org/10.5555/944919.944937

[4] Joost C. de Winter, Samuel D. Gosling, and Jeff Potter. 2016. Comparing the Pearson and spearman correlation coefficients across distributions and sample sizes: A tutorial using simulations and empirical data. *Psychological Methods* 21, 3 (2016), 273–290. https://doi.org/10.1037/met0000079

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional Transformers for language understanding. https://arxiv.org/abs/1810.04805

[6] Vahid Etemadi, Omid Bushehrian, and Reza Akbari. 2017. Association rule mining for finding usability problem patterns: A case study on stackoverflow. *2017 International Symposium on Computer Science and Software Engineering Conference (CSSE)* (2017). https://doi.org/10.1109/csicsse.2017.8320144

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV).*

[8] Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. *CoRR* abs/2004.12832 (2020). arXiv:2004.12832 https://arxiv.org/abs/2004.12832

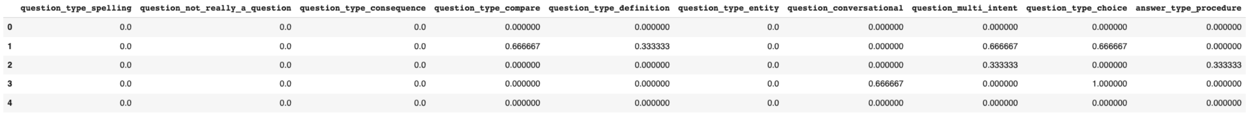
[9] Matt Lake. 2009. Timeline: The evolution of online communities. https://www.computerworld.com/article/2526581/timeline--the-evolution-of-online-communities.html

[10] Shuai Peng, Ke Yuan, Liangcai Gao, and Zhi Tang. 2021. MathBERT: A Pre-Trained Model for Mathematical Formula Understanding. https://doi.org/10.48550/ARXIV.2105.00377

[11] Ziming Wu, Jun Liang, Zhongan Zhang, and Jianbo Lei. 2021. Exploration of text matching methods in Chinese disease Q&A systems: A method using ensemble based on Bert and Boosted Tree Models. *Journal of Biomedical Informatics* 115 (2021), 103683. https://doi.org/10.1016/j.jbi.2021.103683

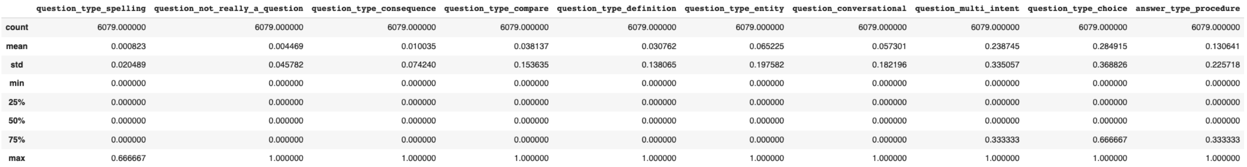
A APPENDIX

A.1 Response Variables and Explanatory Variables Overview



	question_type_spelling	question_not_really_a_question	question_type_consequence	question_type_compare	question_type_definition	question_type_entity	question_conversational	question_multi_intent	question_type_choice	answer_type_procedure
0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.000000
1	0.0	0.0	0.0	0.666667	0.333333	0.0	0.000000	0.666667	0.666667	0.000000
2	0.0	0.0	0.0	0.000000	0.000000	0.0	0.000000	0.333333	0.000000	0.333333
3	0.0	0.0	0.0	0.000000	0.000000	0.0	0.666667	0.000000	1.000000	0.000000
4	0.0	0.0	0.0	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.000000

Fig. 5. Head rows for response variables after variable selection



	question_type_spelling	question_not_really_a_question	question_type_consequence	question_type_compare	question_type_definition	question_type_entity	question_conversational	question_multi_intent	question_type_choice	answer_type_procedure
count	6079.000000	6079.000000	6079.000000	6079.000000	6079.000000	6079.000000	6079.000000	6079.000000	6079.000000	6079.000000
mean	0.000823	0.004469	0.010035	0.038137	0.030762	0.065225	0.057301	0.238745	0.284915	0.130641
std	0.020489	0.045782	0.074240	0.153635	0.138065	0.197582	0.182196	0.335057	0.368826	0.225718
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.333333	0.666667	0.333333
max	0.666667	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Fig. 6. Statistics for selected response variables

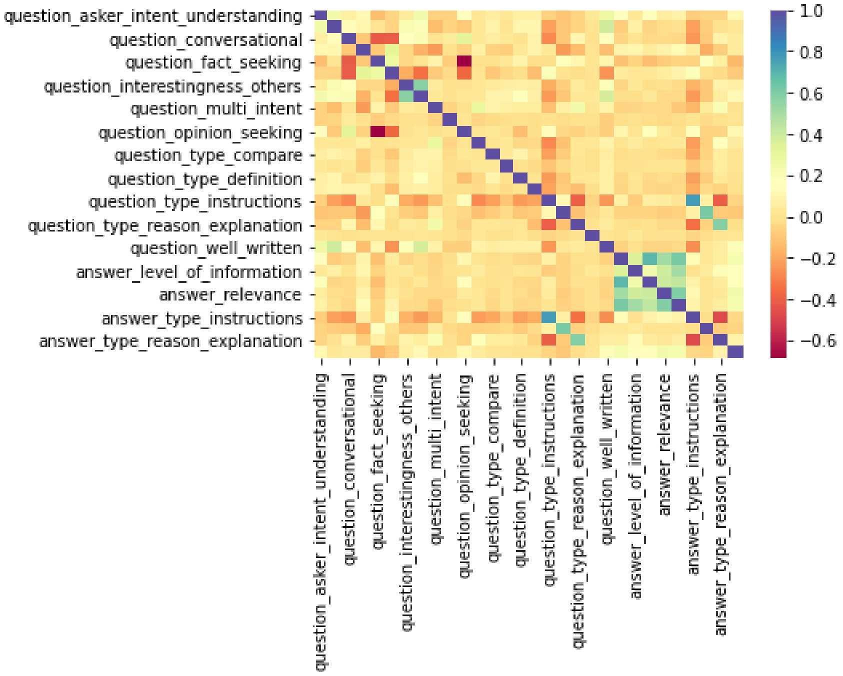
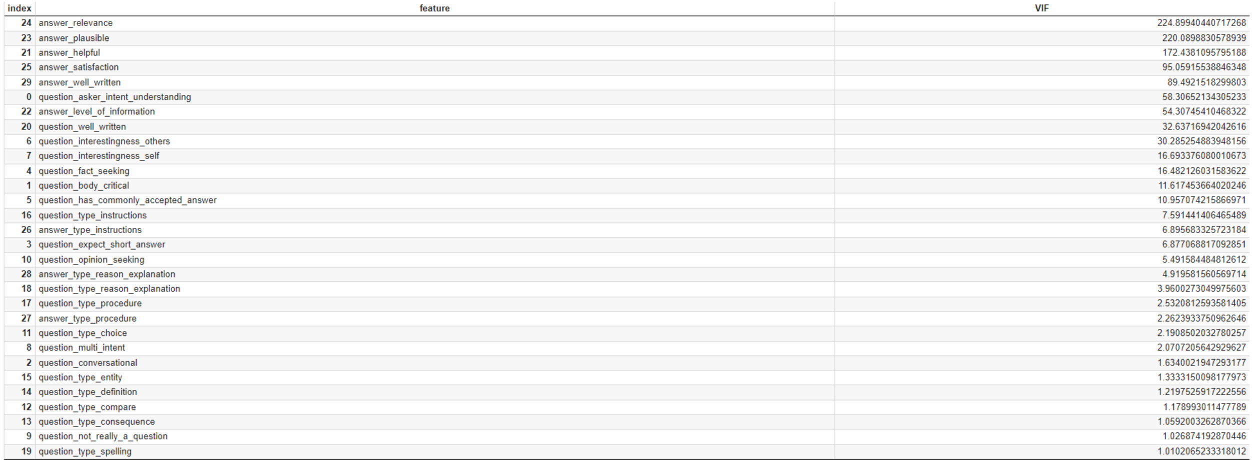


Fig. 7. Correlation heatmap for response variables



index	feature	VIF
24	answer_relevance	224.89940440717268
23	answer_plausible	220.0898830578939
21	answer_helpful	172.4381095795188
25	answer_satisfaction	95.05915538846348
29	answer_well_written	89.4921518299803
0	question_asker_intent_understanding	58.30652134305233
22	answer_level_of_information	54.30745410468322
20	question_well_written	32.63716942042616
6	question_interestingness_others	30.285254883948156
7	question_interestingness_self	16.693376080010673
4	question_fact_seeking	16.482126031583622
1	question_body_critical	11.617453664020246
5	question_has_commonly_accepted_answer	10.957074215866971
16	question_type_instructions	7.591441406465489
26	answer_type_instructions	6.895683325723184
3	question_expect_short_answer	6.877068817092851
10	question_opinion_seeking	5.491584484812612
28	answer_type_reason_explanation	4.919581560569714
18	question_type_reason_explanation	3.9600273049975603
17	question_type_procedure	2.5320812593581405
27	answer_type_procedure	2.2623933750962646
11	question_type_choice	2.1908502032780257
8	question_multi_intent	2.0707205642929627
2	question_conversational	1.6340021947293177
15	question_type_entity	1.3333150098177973
14	question_type_definition	1.2197525917222556
12	question_type_compare	1.178993011477789
13	question_type_consequence	1.0592003262870366
9	question_not_really_a_question	1.026874192870446
19	question_type_spelling	1.0102065233318012

Fig. 8. VIF for response variables

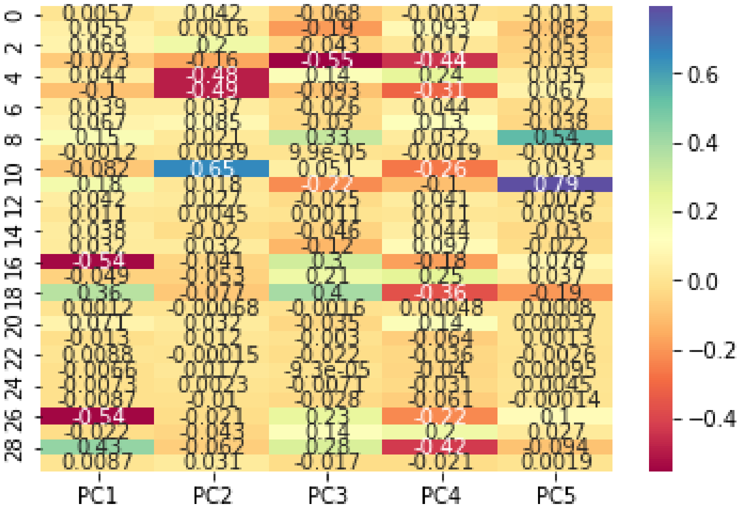
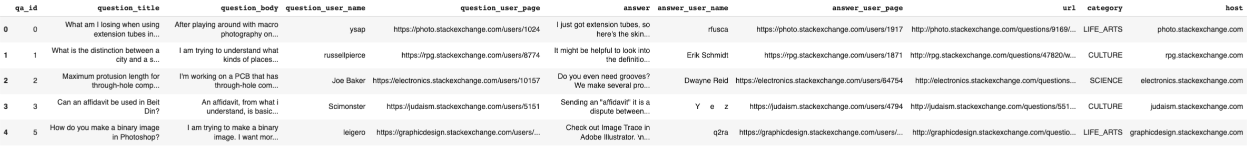
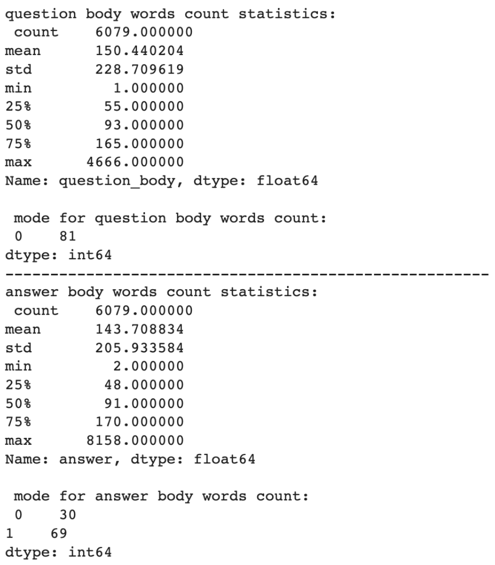


Fig. 9. Principle Component Analysis



	qa_id	question_title	question_body	question_user_name	question_user_page	answer	answer_user_name	answer_user_page	url	category	host
0	0	What am I losing when using extension tubes in...	After playing around with macro photography on...	ysap	https://photo.stackexchange.com/users/1024	I just got extension tubes, so here's the skin...	rfusca	https://photo.stackexchange.com/users/1917	http://photo.stackexchange.com/questions/9169/...	LIFE_ARTS	photo.stackexchange.com
1	1	What is the distinction between a city and a s...	I am trying to understand what kinds of places...	russellpierce	https://rpg.stackexchange.com/users/8774	It might be helpful to look into the definitio...	Erik Schmidt	https://rpg.stackexchange.com/users/1871	http://rpg.stackexchange.com/questions/47820/w...	CULTURE	rpg.stackexchange.com
2	2	Maximum protusion length for through-hole comp...	I'm working on a PCB that has through-hole com...	Joe Baker	https://electronics.stackexchange.com/users/10157	Do you even need grooves? We make several pro...	Dwayne Reid	https://electronics.stackexchange.com/users/64754	http://electronics.stackexchange.com/questions...	SCIENCE	electronics.stackexchange.com
3	3	Can an affidavit be used in Beit Din?	An affidavit, from what i understand, is basic...	Scimonster	https://judaism.stackexchange.com/users/5151	Sending an "affidavit" it is a dispute between...	Y e z	https://judaism.stackexchange.com/users/4794	http://judaism.stackexchange.com/questions/551...	CULTURE	judaism.stackexchange.com
4	5	How do you make a binary image in Photoshop?	I am trying to make a binary image. I want mor...	leigero	https://graphicdesign.stackexchange.com/users/...	Check out Image Trace in Adobe Illustrator. \n...	q2ra	https://graphicdesign.stackexchange.com/users/...	http://graphicdesign.stackexchange.com/questio...	LIFE_ARTS	graphicdesign.stackexchange.com

Fig. 10. Head rows for explanatory variables



```
question body words count statistics:
 count    6079.000000
mean      150.440204
std       228.709619
min         1.000000
25%        55.000000
50%        93.000000
75%       165.000000
max      4666.000000
Name: question_body, dtype: float64

 mode for question body words count:
 0    81
dtype: int64
------------------------------------------------------
answer body words count statistics:
 count    6079.000000
mean      143.708834
std       205.933584
min         2.000000
25%        48.000000
50%        91.000000
75%       170.000000
max      8158.000000
Name: answer, dtype: float64

 mode for answer body words count:
 0    30
1    69
dtype: int64
```

Fig. 11. Statistical summaries for question body and answer body words count