Building Predictive Algorithms for Improving Automated Understanding of Complex Question Answer Content

Kylie Vo
kylie.vo@ischool.berkeley.edu

Abstract

Question answering (QA) has been a benchmark task in natural language understanding. However, the development of a methodology for evaluation is still challenging. One of the key issues is to find the metrics that is used for evaluation to ensure quality. This document contains my work on the Google Quest QA Labeling competition on Kaggle. The challenge is to build predictive algorithms for different subjective aspects of question-answering. This competition requires participants to predict the scores given by human raters to questions and answers on various websites. The questions and answers are scored on 30 dimensions, including whether the question was well-written, relevant, helpful, satisfactory, contained clear instructions, etc. The purpose this work is to find a result that can hopefully foster the development of QA systems.

1 Introduction

Computers are really good at answering questions with single, verifiable answers. However, humans are often still better at answering questions about opinions, recommendations, or personal experiences.

Humans are better at addressing subjective questions that require a deeper, multidimensional understanding of context - something computers aren't trained to do well yet. Questions can take many forms - some have multi-sentence elaborations, others may be simple curiosity or a fully developed problem. They can have multiple intents, or seek advice and opinions. Some may be helpful and others interesting. Some are simple right or wrong.

Unfortunately, it's hard to build better subjective question-answering algorithms because of a lack of data and predictive models. The CrowdSource team at Google Research, a group dedicated to advancing NLP and other types of ML science via crowdsourcing, has collected data on a number of these quality scoring aspects.

The question-answer pairs were gathered from nearly 70 different websites, in a "common-sense" fashion. The raters received minimal guidance and training, and relied largely on their subjective interpretation of the prompts. As such, each prompt was crafted in the most intuitive fashion so that raters could simply use their common-sense to complete the task.

Demonstrating these subjective labels can be predicted reliably can shine a new light on this research area. Results from this will hopefully inform the way future intelligent QA systems will get built, and contributing to them becoming more human-like.

2 Backgrounds

Question answering is not a new research area. QA systems can be found in many areas of NLP research including natural language database systems and spoken dialog systems. Therein, community QA has already attract attention from researchers investigate information seeking behaviors and a wide range of other information-related behaviors. Product differentiation have been occurred in many QA sites. Many sites have restricted their scope in a variety of ways, such as Stack Overflow, which limits its scope to only questions about programming. This would benefit to the QA sites from both of the standpoint of user satisfaction and evaluation of the quality of the answers provided on the sites.

Some researchers suggest that the context that rises to an information need is unique for every individual and and answer that is useful to an in-

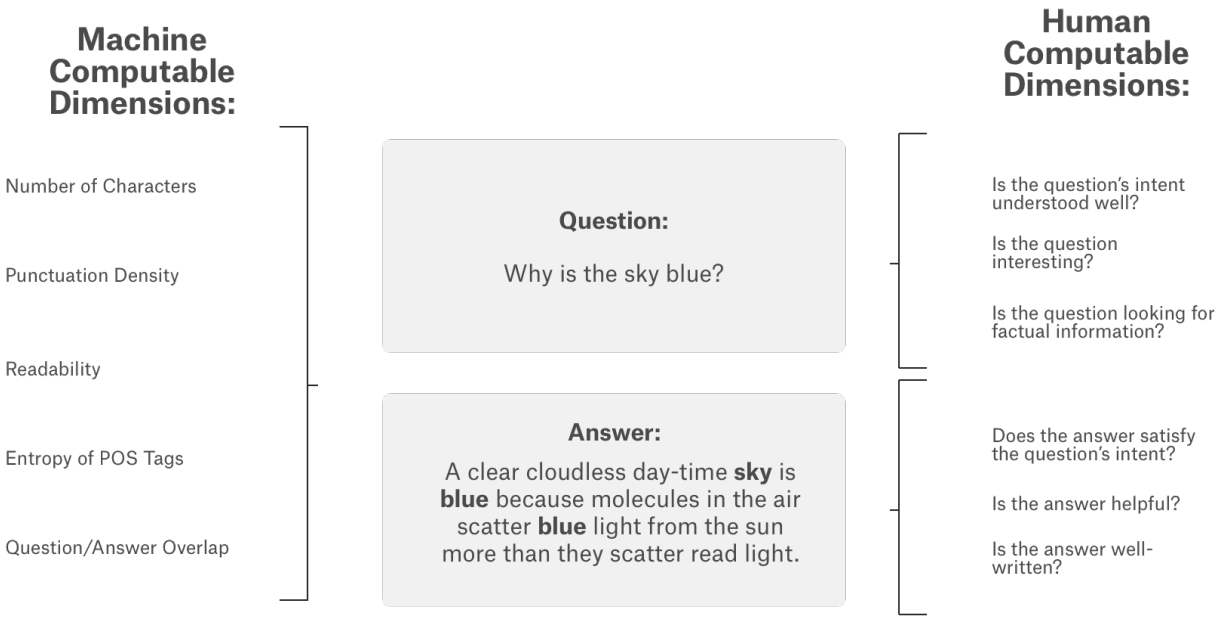

Figure 1: Machine-Human Computation Dimensions

dividual in a particular context may only partially useful to others in other contexts. Currently, the evaluation of the quality of questions and answers are largely classification problems. Most of these classification tasks are already sufficiently complex that human are able to performed more effectively than machine classification. However, the quality of an answer, or any information content, can be subjective. Giving a single predictive rating may not be sufficient. A quality assessment may depend on the relevant of the content, and among others, relevance itself is difficult to measure.

A range of approaches has also emerged for developing evaluation criteria used in studied of QA. Zhu et al.[2] identified a set of 13 criteria when askeing human assessors to judge the quality of the information from both questions and answers. In this report, 30 criteria are used as different aspects of answer quality. Extracting necessary features is need to construct fairly reliable models.

3 Methods

3.1 Proposed Works

The purpose of the project is to build predictive algorithms for different subjective aspects of question-answering and give the answer a score base on its adequacy. I see this problem under two sub problems, first to embed/encode the text data, and second to use a model to find the correlations between predictor variables. For my initial approach, I consider using ridge regression as my baseline model and BERT as my advanced model.

To enhance the model accuracy and efficiency, I consider using k-fold validation to tune the models. The Kaggle challenge is helpful to evaluate my models performance relative to other participants' approaches.

3.2 The Data

The data for this competition includes questions and answers from various StackExchange properties. The task is to predict target values of 30 labels for each question-answer pair.

The list of 30 target labels are the same as the column names in the "sample_submission.csv" file. Target labels with the prefix question_ relate to the question_title and/or question_body features in the data. Target labels with the prefix answer_ relate to the answer feature.

Each row contains a single question and a single answer to that question, along with additional features. The training data contains rows with some duplicated questions (but with different answers). The test data does not contain any duplicated questions.

This is not a binary prediction challenge. Target labels are aggregated from multiple raters, and can have continuous values in the range [0,1]. Therefore, predictions must also be in that range.

Since this is a synchronous re-run competition, the test set is no larger than 10,000 rows, and less than 8 Mb uncompressed.

Data file descriptions:

- train.csv - the training data (target labels are the last 30 columns)
- test.csv - the test set (predict 30 labels for each test set row)
- sample_submission.csv - a sample submission file in the correct format; column names are the 30 target labels

3.3 Exploratory Data Analysis

The Exploratory Data Analysis is conducted on both of the train and test sets. The training set has a total of 6079 samples. The first 11 columns are the feature columns, including the question tile, question body, answer, category, question user page, etc. The last 30 columns are the target variables, including the scores on the questions/answers. The test set has 476 samples and only includes the feature columns.

The following aspects of the data were examined:

- Question category distribution
- Repeated questions titles/bodies and most popular questions
- Word count distribution of the question titles/bodies and the answers
- Common words used in the question titles/bodies and the answers
- Target variables distribution
- Correlation between target variables
- Distribution of Scores by Categories

Both of the train and test sets have no missing data. There are five categories in the samples: culture, technology, life arts, stackoverflow, and science. The train data set appears to have more samples on the science category, whereas the test data set appears to have more samples on the technology category.

Figure 2 shows the distribution plots of five target variables. This tell that most of the answers are rated very high on "answer_well_written", but rated low on "answer_type_procedure". The scores for "answer_type_instructions" and "answer_type_reason_explanation" seem to be equally

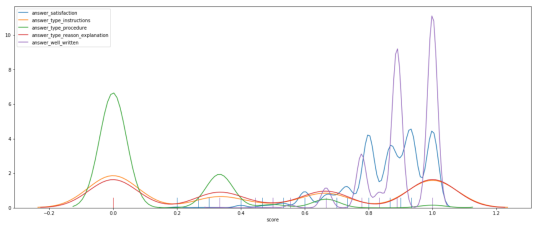

Figure 2: Examples of Answer Score Distribution

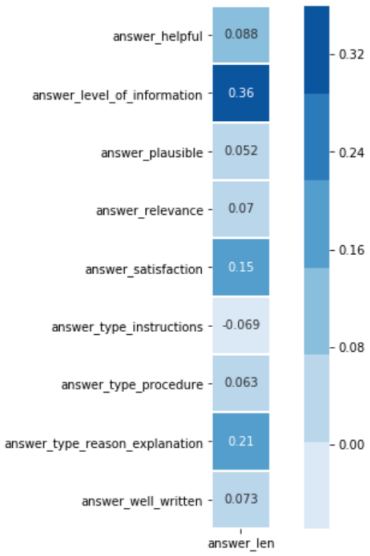

Figure 3: Answer Length Vs. Answer Related Scores

distributed between samples. Other score distributions can be found in the EDA Jupyter Notebook.

Answer word length appears to have positive correlation with "answer_level_of_information", but does not seems to have any strong correlation on the plausibility or relevance of the the answers. Answer length can be a good indicator on the level of information of the answer, but does not provide much context on the correctness or helpfulness of the answers.

Figure 4 shows that the target scores can be different from one category to another. Most of the answer from culture sites are rated low on the the "answer_type_instruction", but answers from Stackoverflow and technologies sites are rated higher on "answer_type_instruction". This makes sense because culture sites usually provide information instead of providing instructions.

3.4 Completed Works:

Before approaching to fit the models, I had review the fundamental of the two BERT and ridge re-

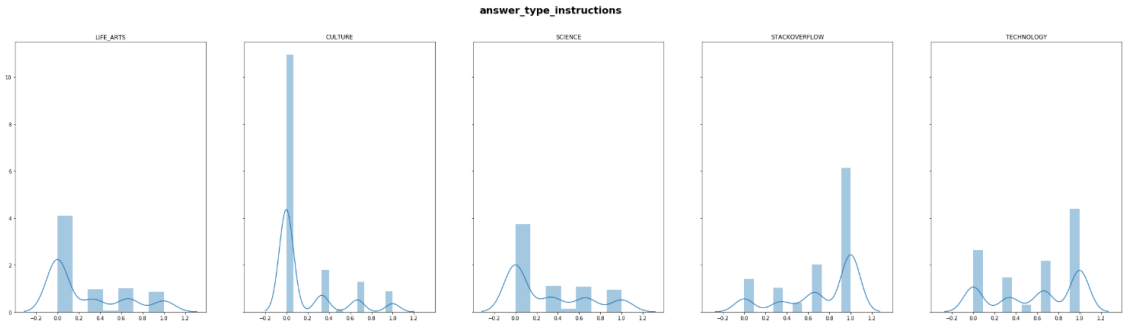

Figure 4: Answer_type_instructions Scores by Category

gression models.

The Ridge Regression is the Variation of Linear Regression. People also called it as Tikhonov regularization. This is widely used as a method of regularization of ill-posed problems. This model is preferred to use in models with large numbers of parameters (multicollinearity). This means the model can pick up correlation between word and word, character and character really quick and fast.

On the other hand, BERT model is a deeply bidirectional, unsupervised language representation, pre-trained using only a plain text corpus. In this case, Bert model convert all the word into vectors or matrix of vectors. Words with strong correlation with each other are usually represent as mostly the same vector. Therefore, in this case, where we need to pick up the phrase of word and determine if it is meaningful and makes some senses to the questions, we would prefer BERT model over Ridge Regression model. Even though Ridge Regression model can do really well in picking up correlation between word and word, character and character, it can not pick up the correlation among phrases.

Both models are trained in the order of:

1. Import Clean Data
2. Pre-process Data
3. Transfer text data to ”number” by encoding/embedding
4. Applied model
5. Test the result
6. Tune the parameter
7. Finalize results

3.4.1 Baseline Model:

In Ridge Regression model portion, I used all of the samples in the training data to train the model. I used TfidfVectorizer to convert text to vector with ngram = 2 for both answer and question portions. I also splitted train data set into two equally portions, then randomly split each portion into train and validation set in order to train and test the model. For this initial work, I got a result of 0.75 accuracy rate averagely for both the portions of the data with ridge regression model.

3.4.2 Bert Model:

For BERT model, I first import data then preprocess them with 'bert-base-uncased'. After that, I encoded all the text data by using function 'encode_plus'. This will give me a new matrix that represents the same type of data, but in number. These numbers are the vector, the correlation, and represent how important, strong a word or phrase is. It also tell us if a word can connect with another word to make a perfect phrase.I trained the model with BERT model and used GroupKFold of 5 to validate the result. Using only 5 percent of training data, I got a min loss of 0.4161. This model will need further tuning.

4 Results and Discussion

4.0.1 Baseline Model:

1. Accuracy and Loss: NA.
2. Error Analysis: NA.

***I have not tuned the model yet

4.0.2 Bert Model

1. Accuracy and Loss: NA.
2. Error Analysis: NA.

***I have not tuned the model yet

The current model accuracy/loss of the models cannot be compared at this stage. Only 5 percent of data has been used for BERT model. The accuracy of the Ridge Regression model could be lower the the current value.

5 Next Steps

For the rest of the semester, my main focus will be on tuning both models. More analysis on EDA and the models will be added. Model approach and performance will be added in greater details. The introduction and background will be also edited.

References

[1] Zhao, Yiming and Zhang, Jin and Xia, Xue and Le, Taowen 2018. *Evaluation of Google question-answering quality*. Library Hi Tech.

[2] Zhu, Z., Bernhard, D., Gurevych, I. 2009. *Multi-dimensional Model for Assessing the Quality of Answers in Social QA Sites.* Technical Report TUD-CS-2009-0158. Technische Universität Darmstad.