National College of Ireland

Understanding the Subjective Aspect of Question Answer

MSc Research Project
Data Analytics

Ashish Patel
Student ID: x18182445

School of Computing
National College of Ireland

Supervisor: Christian Horn

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Ashish Patel

Student ID: X18182445

Programme: Data Analytics **Year:** 2019-20

Module: MSc Research Project

Supervisor: Christian Horn

Submission Due Date: 17/08/2020

Project Title: Understanding Subjective Aspect of Question Answer

Word Count: 6403 **Page Count** 16

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:

Date: 17/08/2019

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Understanding the Subjective Aspect of Question Answers

Ashish Patel
x18182445

Abstract

Question answer system these days are good when it comes to fact based or verifiable answers but when it comes to questions which seek recommendations, personal experiences or opinions, humans are much better at answering. One could say that humans are great at solving contextual problems that need a broader, multidimensional view of the context, something machines are not qualified to do well yet, as questions could be of several forms like multi sentence elaborations while others could be incomplete without proper context. Unfortunately, it is very hard to build a good subjective question answer system due of the lack of trained data. To rectify the problem, Google with the help of it's crowdsource team came out with a dataset which comprises of question answers pairs from various open source websites which are given scores between 0 to 1 on 30 different subjective aspect of the question answer pair like question is well written or not, answer provided is satisfactory or not etc., rated by the team itself. The aim of this research is to take the above dataset and create a model which could be able to score the question answer pair of their subjective aspect. To achieve the above results three different NLP techniques were used Word Embeddings, Universal Sentence Encoder and BERT transformer model and their results were compared. Throughout the result it was found that against the BERT model, which is considered gold standard in NLP, Universal sentence encoder gave equal if not better result for the data set

1 Introduction

1.1 An Overview of Question Answer System

Question answering system is one of the oldest NLP tasks which were first built on punch card system by (Simmons, Klein and McConlogue, 1964) in 1964, since then the question answering system have come a long way. As pointed out by (Ferrucci *et al.*, 2013), in 2011 IBM Watson won the famous Jeopardy contest where contestants compete, giving answers to various types of questions. Also the virtual assistant offered by various tech companies such Siri, Cortana and Ok google, Alexa all are advance question answering system which work really well with factual based questions such as "how many calories are there in the apple pie?" or "what is the average age of the onset of autism?".

According to (Ravichandran and Hovy, 2002) There are two main paradigms for any question answer system

- IR based approach which is followed by IBM Watson and Google commercial system. These systems whenever asked any question, convert it into a query and search in whole internet for the answer, it follows the search engine methodology for its answers.
- Then there is knowledge based and hybrid approaches which is followed by Apple Siri, Wolfram Alpha. These systems build a pure semantic representation of the query, they would come up with a semantic representation language for question that

they understand and then map these semantics with the structured database for the answers.

1.2 Background and Motivation

The question answer systems work well when they are given a factual or verifiable question like “what is the weather today?” but when the same system is given any opiniated or subjective question, humans outperform the question answer system. According to (Adiwardana *et al.*, 2020) question answer systems are designed to see number of characters, punctuation density, readability, entropy of POS tags (positional tags such as DT determiner, FW foreign word etc which describes the sentence structure) and question answer overlap in a question answer pair. Whereas when a human sees a question answer pair it looks, Is the question’s intent is understood well? Is the question interesting? Is the question looking for factual information? Does the answer satisfy the question intent? due the subjective approach of human towards a question makes it better at understanding the complex question and answers. Figure 1 from (Adiwardana *et al.*, 2020) describes sensibleness and specificity score for the human as well as all the modern question answer system such as Google Meena, Mitsuku etc. and it us quite evident from the figure that though the question answer system are excellent in giving factual answers but are unable to understand the questions objective.

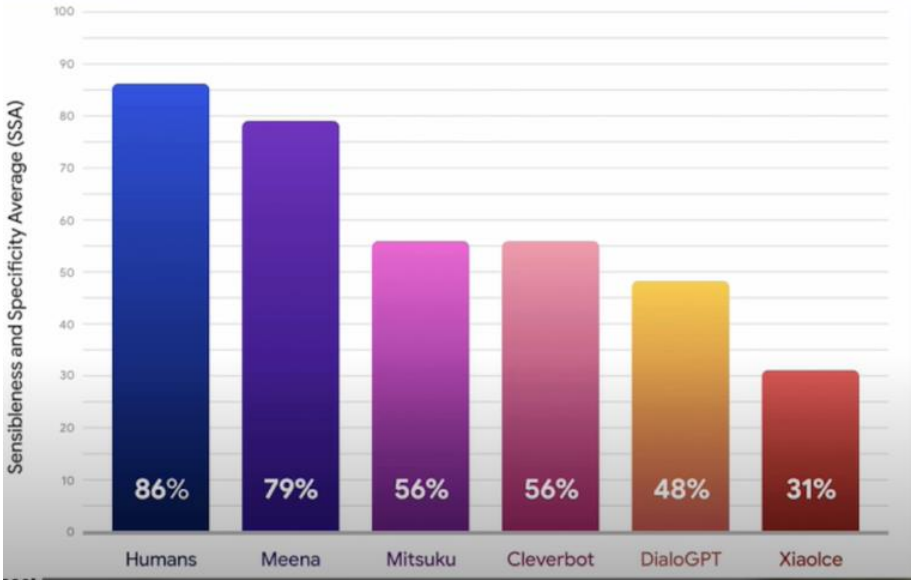


Figure 1 SAS average (Adiwardana et al., 2020)

Failure of a computers to understand the opinions, politics or emotions are mainly due to lack of data set, to build a good subjective question answers system. Google with the help of it crowdsource team came up with a dataset which contains question and answer pairs from 70 different websites and then each question answer pair is scored between 0 to 1 by the team itself on 30 different subjective attributes such as “question asker intend understood”, “question controversial”, “answer acceptable”, “answer relevance” etc. The main objective of the research is to build model which taken input any question answer pair and provide scores on all the 30 subjective attributes of question and answer

1.3 Research Question

“Are Deep Learning architectures capable of recognizing the subjective aspect of the question and answer pairs?”

The motive of the research is to build a deep learning model which can understand the subjective aspect of the question answer pair which is lacking in most of the question answer system these days.

1.4 Research Objective

The main objective of the research is to take the dataset which is produced by Google and use state of the art NLP techniques such as text embedding, Universal Sentence Encoder and BERT to train a model which would be able to understand the subjective aspect of the question and answer pair such, as if question was controversial or not, or the answer provided is satisfactory or not etc. The research addresses the following tasks:

- Data pre-processing and doing some exploratory data analysis to understand the dataset.
- Implementation of text embedding, Universal Sentence Encoder and BERT model on the clean feature engineered data.
- Evaluate the performance of the all the 3 models and compare which models suits best for the dataset.

This research is organized as: Section 2 contains all the related work which has been done before. Section 3 describes the methodology followed while carrying out the research. Section 4 describes the machine learning model implementations. Section 5 describe the evaluation and results obtained after training the models.

2 Related Work

Coming to the 21st century according to the industry estimates only 20% of the total available data is in structured form. Data is being generated as we tweet, send message on WhatsApp or going through Facebook, and majority of this data is in textual form which is highly unstructured in nature. In order to produce significant and actionable insights from this data NLP (Natural language Processing) techniques are used. This research is using text embeddings, Universal Sentence Encoder and BERT transformer model technique to build the mode and compare their outcome, all the above-mentioned technique are mentioned below.

2.1 Text Embedding

Text embedding is an approach where a word is represented as vector of a real numbers and words with similar vectors are semantically similar, sometimes vectors are low dimension compared to the vocab size. There are various technique to convert words into vector form. In 2014 (Pennington, Socher and Manning, 2014) came up with global vectors for word representation which tend to improve upon the previous matrix factorization and shallow window approaches. Matrix factorization is one of the oldest word embedding technique still used today, and the problem as pointed out by (Pennington, Socher and Manning, 2014) is their inability to capture contextual data in the words neighborhood and the fact that they are almost simply taking the probability of word occurrences fail to distinguish between any sub or secondary meanings a word may tend to have. On the other hand there are shallow window approaches like word2vec (Mikolov *et al.*, 2013) where word representation is learned so that they can make predictions within a local contextual window. The problem with this sliding

window approach is that these schemes move the context window across the entire corpus which means the repetition of words and phrases are not utilized and hence do not account for co-occurrence statistics.

GloVe (Pennington, Socher and Manning, 2014) aims to rectify all the above problems with the previous embeddings by capturing the context of the word in the embedding through explicitly capturing the co-occurrences probabilities. This is empirically show in the paper as per the table 1, where the word ice and steam are compared to various probe words solid, gas, water, and fashion. We can see that the word ice is related more strongly towards solid than it is to gas and the converse is true for steam as seen by the ratios calculated in the bottom row, both terms have very similar large values with water and on the other hand very small values in the context of the word fashion. All this aims to argue the point that embedding should be built not on just the word probabilities but their co-occurrences probabilities within the context.

Table 1: probability and ratio for ice and steam against solid, gas, water, and fashion

Probability and Ratio	K=Solid	K=gas	k=water	k=fashion
P(k|ice)	1.09*10^-4	6.6*10^-5	3.0*10^-3	1.7*10^-5
P(k|steam)	2.2*10^-5	7.8*10^-4	2.2*10^-3	1.8*10^-5
P(k|ice)/P(k|steam)	8.9	0.085	1.36	0.96

The study by (Brochier et al., 2019) conducted on the web nodes state that previously Skip gram model and native sampling technique were used for link prediction and node classification. The researcher came up with a updated GloVe model which uses matrix factorization technique to provide better results than the previous used models. Similarly, in another research by (Lee and Dernoncourt, 2016), combination of RNN and CNN were used to build a classification model which uses transfer learning approach of using pre trained GloVe embeddings to build a classification model. In another research (El Mahdaouy et al., 2017) deep neural network model is built for the classification of Arabic words which also uses a transfer learning approach of using GloVe embedding to build a text classification model. looking at the previous researches this research also uses transfer learning to build a model also using the architecture similar to use in the research by (Lee and Dernoncourt, 2016), a combination of LSTM and CNN.

2.2 Universal Sentence Encoder

One of the biggest challenges faced in NLP is the lack of labelled or supervised data. This becomes a challenge for the deep learning models which are data hungry. In order to rectify the problem many models use transfer learning approach where they take the pre trained embedding like Word2Vec, GloVe and then train on the task specific data using that embedding. The researchers (Cer et al., 2018) from Google came up with a similar transfer learning approach where instead of word they would be encoding the whole sentences. They created two models with the help of transfer learning technique one big model with multiple layer and another small model with minimum layer required for the task. While comparing the result they found the sentence embedding model performed better that word embedding model.

Due to it's effectiveness many of the research start using sentence embedding in place of word embedding while building their models. Similarly in research by (Perone et al., 2018) uses the sentence embedding for the various downstream NLP task such as question answer system, language model etc. and compare it with the state of the art models which uses word embeddings. They found that in some cases the sentence embedding perform better while in other cases the old models perform better. In this research we will also be using both the transfer learning approaches where we will be building model taking a pre-trained GloVe

word embedding as well as pre-trained Sentence embedding and then compare which provides the best results when trained with this research dataset.

In the research by (Fu et al., 2020), the sentence embedding are used in conjunction with the CNN and LSTM architecture for the downstream NLP tasks. By using CNN and LSTM together the model trained on sentence embedding produce state of the art results in some of the downstream task. This research will also incorporate the same approach of transfer learning where the sentence embedding will be used to train a neural net model with the research dataset to create a model which categorizes the question answer pair into 30 different subjective attributes.

2.3 Transformer Model

NLP techniques pre-dominantly use LSTM networks for all the downstream tasks but these networks have issues as pointed out by (Goldberg, 2019). LSTM networks are slow to train because words are passed in sequentially so it can take significant number of time steps for the neural network to learn, it is also not the best at capturing the true meaning of the words even the bidirectional LSTM's as they learning left to right and right to left context separately and then concatenating them so the true context is slightly lost. To address the LSTM issues researchers (Vaswani *et al.*, 2017) in Google came up with transformer network which follow attention mechanism and were better than LSTM in speed as multiple words were trained in parallel also the context of the word was better understood by the transformers than LSTM.

The Transformer architecture comprises of two key component an encoder and decoder. Suppose for a NLP task we want to convert English to French. The encoder takes the input words simultaneously and generates embeddings for every word at the same time, these embedding are vectors that encapsulate the meaning of the word, similar words have closer number in their vectors. The decoder takes these embeddings from the encoder and the previously generated words of the translated French sentence and then it uses them to generate the next French word, the transformer architecture keep generating the French translation one word at a time until the end of sentence is reached.

Taking the advantage of the above transformer architecture researcher (Devlin *et al.*, 2019) in the Google came up with the BERT model which takes the encoder part of the transformer and stack them one by one to give us BERT (Bidirectional Encoder Representation from Transformers). The original transformer architecture is only used for the language translation but according to (Rogers, Kovaleva and Rumshisky, 2020) a BERT model can be used for all types of NLP downstream tasks such question answering system, sentiment analysis, text summarization etc. In order to train BERT for the above task, the training is done in two phases, the first phase is pre training where the model understands the language and the context and the second phase is fine tuning where the model learns how to solve the problem after it has learned the language in the first phase.

The research by (Munikar, Shakya and Shrestha, 2019) uses BERT model to do sentiment analysis. Sentiment analysis is one of the most important NLP tasks as it helps to understand the perception of people towards a topic, product or business. Most of the sentiment analysis problem earlier only focussed at the binary classification problem but the research uses a BERT model to resolve finer grained multi class classification problem. For the task research uses transfer learning approach where a pre trained BERT model is used, and an extra sigmoid layer was added at the end to get the classification. Similarly in another research by (Huang *et al.*, 2019), BERT model is used to build a classification model DCNN-BiGRU (Deep Convolutional Neural Network Bidirectional Gated Recurrent) which maps every word in the corpus with multi-dimensional matrix, which is better than the single dimension vectors mapping by previous models. Thus creating a classification model in which the word

embedding will have both local and contextual feature both. This research also takes the transfer learning approach of taking pre trained BERT model to build a model.

3 Research Methodology

The implementation of the research follows a CRISP DM methodology. The figure 2 represent the different stages on which the whole research is followed Business Understanding, Data Acquisition, Data Pre-Processing, Modelling and then Evaluation.

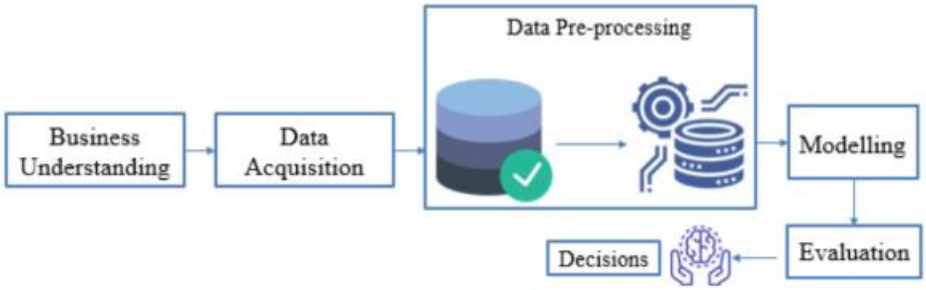


Figure 2 KDD methodology

3.1 Business Understanding

Due to rapid improving computer hardware and software, various NLP tasks such as sentiment analysis, and text classification give state of the art result. But question answering system which are really good at factual or verifiable questions still lack in understanding the subjective aspect of the question answers. Thus the objective of the research is to create a model which understands the subjective aspect of the question answer pair by giving a score in between 0 to 1 for all the 30 subjective attributes such as question conversational, question expect short answer, question opinion seeking, answer controversial, answer satisfactory etc.

3.2 Data Acquisition`

The crowdsource team of google came up with a dataset that addressed the industry wise issue of non-availability of supervised data in the NLP question answering tasks. The data comprises of question answer pairs from across 70 different websites covering almost all the different categories such as science, technology, philosophy, beauty and many more. Once the data is collected the inhouse team of google scored each question answer pair in the range of 0 to 1 on their 30 different subjective attribute such as question intent understood, answer satisfactory and many more. This is the first labelled dataset to address the subjective aspect and would be a key step required for the question answer system which still have not reached the human level when it comes to understanding the subjective aspect of the question and answer pair.

Table 2: Dataset description

Dataset	Record Count	Attribute Count
Train.csv	6079	41

3.3 Data pre-processing and EDA

Data pre-processing and EDA (Exploratory Data Analysis) is one the most crucial things in the research as all the data present in train.csv file is in the text format which is difficult for computers to understand. Thus, we have to analyse the text data and see for any irregularities in the data, if the data is clean then we will convert text data into numerical values which are taken as input to the deep learning models.

3.3.1 EDA

Host Website	Percentage
stackoverflow.com	20.61%
english.stackexchange.com	3.77%
superuser.com	3.73%
electronics.stackexchange.com	3.64%
serverfault.com	3.50%
math.stackexchange.com	3.08%
physics.stackexchange.com	2.70%
tex.stackexchange.com	2.30%
askubuntu.com	2.07%
programmers.stackexchange.com	2.06%
rpg.stackexchange.com	2.02%
gaming.stackexchange.com	1.83%
unix.stackexchange.com	1.78%
apple.stackexchange.com	1.74%

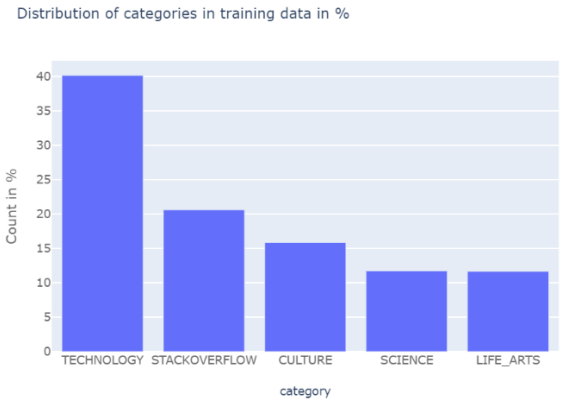


Figure 3 training data

Figure 3 above gives us the understanding of the data as the left side we could see the distribution of the dataset on the basis of websites from where the question answer pairs were taken and maximum of them were taken from stackoverflow.com, from which we could make out that most question answer pair in data set would be related to technology. The right image proves the above point when we plot a histogram for all the question answer category wise, we could make out that most of the questions answer pair are from technology field.

Table 3: 30 target variables in the dataset

1	question_asker_intent_understanding	11	question_opinion_seeking	21	question_well_written
2	question_body_critical	12	question_type_choice	22	answer_helpful
3	question_conversational	13	question_type_compare	23	answer_level_of_information
4	question_expect_short_answer	14	question_type_consequence	24	answer_plausible
5	question_fact_seeking	15	question_type_definition	25	answer_relevance
6	question_has_commonly_accepted_answer	16	question_type_entity	26	answer_satisfaction
7	question_interestingness_others	17	question_type_instructions	27	answer_type_instructions
8	question_interestingness_self	18	question_type_procedure	28	answer_type_procedure
9	question_multi_intent	19	question_type_reason_explanation	29	answer_type_reason_explanation
10	question_not_really_a_question	20	question_type_spelling	30	answer_well_written

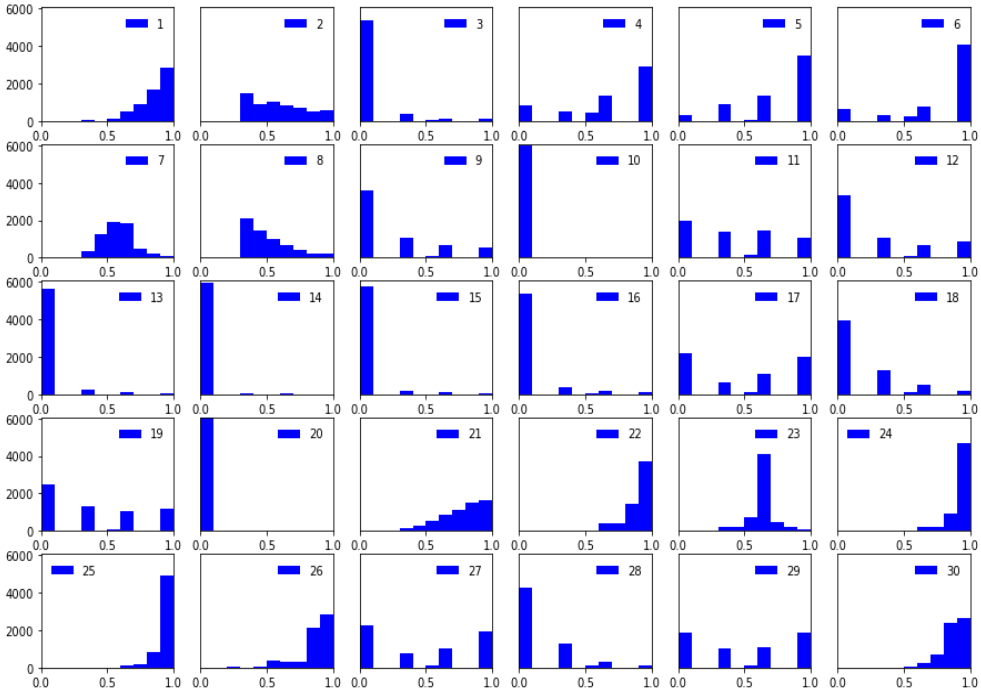


Figure 4 Target variable Distribution

Figure 4 shows the distribution of each of 30 target variables (which are given in a table above the figure) which account for the subjective aspect of a question answer, from the distribution graph we could make out that there is no pattern to be found in any of the attribute, all the attributes are properly scored by the crowdsource team of the google also there were no missing values in the dataset.

3.3.2 Data Cleaning

When doing any NLP task one of the most important aspect is data cleaning, for this research we have striped out the words which will not be necessary for building the model. Normalized or shortened misspell words such as “can’t, couldn’t” which should be “cannot, could not” thus a complete dictionary of misspelled or shortened words was created and those words were replaced with the actual word as shown above. Also, the stop words such as “how”, “a”, “the” and many more were removed to decrease the size of whole corpus which could lead to the better performance from the model. All the letters in the text were made to lower case so that the same letter one in upper case and other in lower should not be taken as two separate characters by the tokenization algorithm. Once the text data is clean then we can proceed with the data pre-processing steps, for each three different model build.

3.3.3 Data Pre-processing for Word Embedding

The word embedding model in this research uses transfer learning approach where pre-trained word vectors are used to train the model, In this research GloVe word embeddings are used because of their properties such as nearest neighbors where words which are semantically similar are given vectors similar to each other. For example, all varieties of frog i.e. toad, rana, lizard, litoria all have vector space near to frog as they all lie in the toad family. Linear substructure is also important property shown by GloVe as in other word embeddings fail to distinguish between men and women as they use similarity matrices and as the men and women occur mostly in same scenarios the vector most of the time is similar, but GloVe embedding add difference vector, as show in the figure 5 taken from (Pennington,

Socher and Manning, 2014) man woman, queen king have similar difference, same way company and CEO have similar difference vector

.

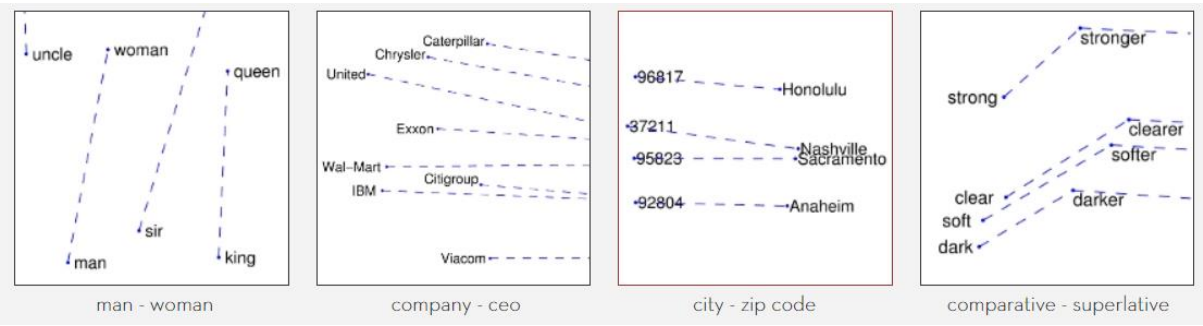


Figure 5 linear relation between words (Pennington et al., 2014)

Once the text data is cleaned all the words in the dataset are mapped to word vectors from pre trained GloVe file which is trained on the huge Wikipedia corpus comprising of more than 400000 words, then these word vectors are taken as input to the deep learning model.

3.3.4 Data Pre-processing for Universal Sentence Encoder

Like above this method also uses transfer learning approach where a pre trained sentence encoder model from TensorFlow hub is imported and then that model is used to convert all the sentences in the dataset to vector space. Once the sentence is converted into vector then this vector is fed as an input to the neural net model.

3.3.5 Data Pre-Processing For Transformer Model

BERT is a transformer model which uses the encoder part of a transformer put together sequentially, BERT often requires the input data in specific format where every sentence should start with a [CLS] token and in between two sentences there must me a [SEP] token. Also, the way BERT works is from the main corpus some words are masked and BERT tries to predict those masked words thus training itself, so we need to mask 15 % words with [MASK] token. Figure 6 taken from (Devlin *et al.*, 2019) gives a clear picture of the inputs required.

- Token Embedding refers to the embedding of each input word, mapped from the vocb.txt present in the BERT model when downloaded from TensorFlow hub.
- Sentence Embedding refers to the numerical value which distinguishes between two sentences.
- Positional embedding refers to the position of each token in the input corpus.

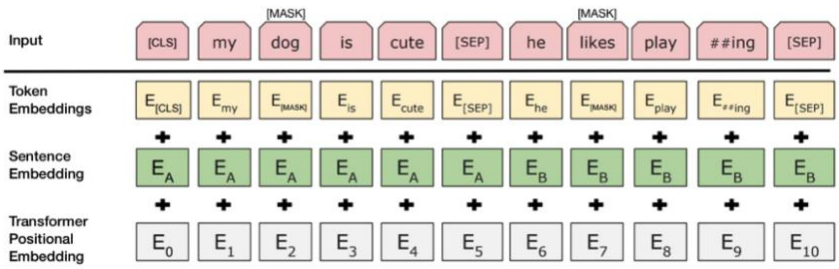


Figure 6 BERT Input (Devlin et al., 2019)

4 Design Specification

The research is divided into three stages data preparation, modelling and evaluation as shown in figure 7.

- In data preparation part as explained above the data is cleaned and the necessary exploratory data analysis is done, then the text data is converted to numerical data with the help of various techniques and given as input to the downstream models.
- In modelling stage three models were trained and various combinations of architecture and hyperparameter tuning were tried with the only motive to increase the performance of the model.
- At the evaluation stage all the output and the performance parameters of all the three models are compared against each other to find which model suit better for the give dataset.

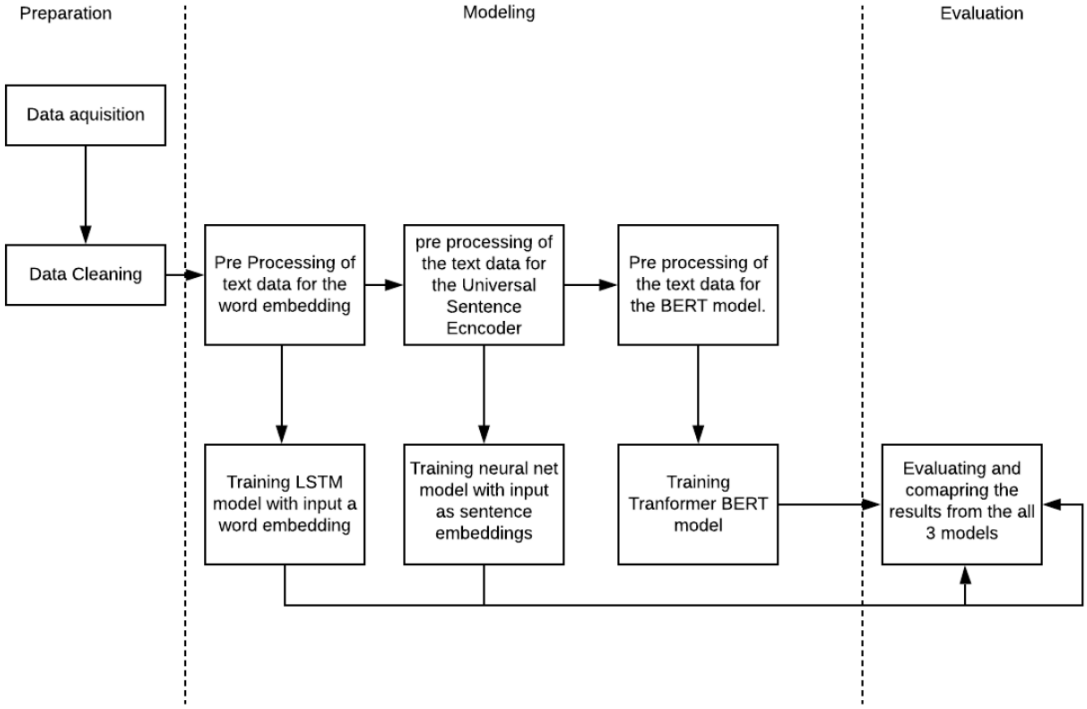


Figure 7 Design Flowchart

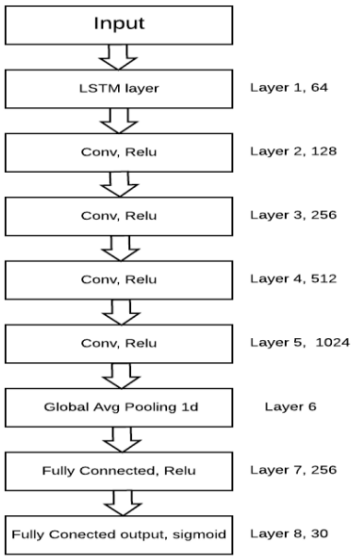
5 Implementation

The implementation in this research is done on three separate deep learning models which have been trained on the similar data to find out which gives the best performance, the hyper parameters of the models were tweaked in order to reduce the training time as much as possible, also increasing the performance at the same time.

5.1 GloVe Word Embedding Model

Once the text data is pre-processes to word vectors with the help of transfer learning the embedding is given as input to the deep learning model with the architecture as shown in figure 9. The deep learning architecture was inspired from the research by (Lee and

Dernoncourt, 2016) where the combination recurrent neural net and the convolutional neural net was used to get the best results for the model. As shown in the below figure 8 the model comprises of a LSTM layer which takes embedding inputs, then it is connected to the 4 convolutional layer with each layer having a drop out of 0.2 and having nodes 126, 256, 512 and 1024. Then there is global avg pooling layer to convert it into flat 1d array which is taken as input by a fully connected layer of 256 neurons, which yields to the final output layer with sigmoid activation function to get the final score / probability of each attribute between 0 to 1. For the research a custom call back function "SpearmanRhoCallback" was created, which was used to calculate the performance of the model after every epoch

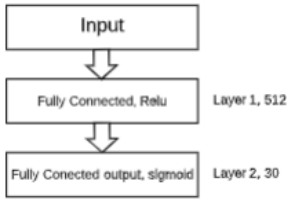



```
model = bilstm_conv_model()
model.compile(loss='binary_crossentropy', optimizer=Adam(lr=0.0005))
history=model.fit(X_tr, y_tr,
                  epochs=15,
                  batch_size=32,
                  validation_data=(X_val, y_val),
                  callbacks=[SpearmanRhoCallback(training_data=(X_tr, y_tr),
                                                 validation_data=(X_val, y_val), patience=2)])
```

Figure 8 GloVe Embedding Model and hyperparameters

5.2 Universal Sentence Encoder Model

One of the major difference from the word embedding approach we had earlier is that, in the case of universal sentence encoder whole sentences in corpus id converted into a vector which is created with the help of universal sentence encoder model taken from the TensorFlow hub. Once we had the input vectors, then various combinations of deep leaning architecture were tried but the best performance came from simple neural net architecture with single fully connected dense layer having a relu activation function. The input from the top layer is then used by the output layer with 30 neurons and sigmoid function as activation function to provide a score/probability value between 0 to 1 for 30 subjective aspects or attributes of a question answer pair taken from various open source websites. Like in the previous case this this models also uses a custom call back function "SpearmanRhoCallback" which is used as metric for the performance of the model after every epoch.

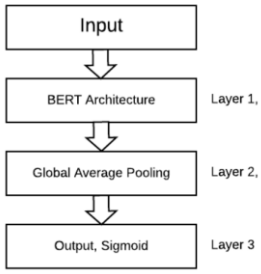



```
uni_model.compile(loss='binary_crossentropy', optimizer=tf.keras.optimizers.Adam(lr=0.0001))
history=uni_model.fit(xtrain, ytrain,
                      epochs=200,
                      batch_size=16,
                      validation_data=(xval, yval),
                      callbacks=[SpearmanRhoCallback(training_data=(xtrain, ytrain),
                                                     validation_data=(xval, yval), patience=5)])
```

Figure 9 Universal Sentence Encoder

5.3 Transformer Model

The transformer model approach used in the research is transfer learning approach where open source pre-trained BERT model which is trained on the huge Wikipedia corpus is taken from the TensorFlow hub, adding to that we added Global Average Pooling layer for the output from the BERT to make it 1 dimensional. Then the final layer of 30 neurons with a sigmoid activation function is used to get the probability value between 0 and 1 for all the 30 subjective attributes of the question answer pair. Same as the previous two models a custom call back function "SpearmanRhoCallback" was used to get the performance of the model after every epoch.




```
model = bert_model()
model.compile(loss='binary_crossentropy', optimizer=tf.keras.optimizers.Adam(lr=0.00005))
history=model.fit(X_tr, y_tr,
                  epochs=4,
                  batch_size=8,
                  validation_data=(X_val, y_val),
                  callbacks=[SpearmanRhoCallback(training_data=(X_tr, y_tr),
                                                 validation_data=(X_val, y_val), patience=5)])
```

Figure 10 BERT Model

6 Evaluation

As the dataset comprises of question answer pairs with score of in the range between 0 to 1 for all the 30 attributes the best way to evaluate the model would be to take the spearman correlation between the predicted column and the actual column then averaging the correlation value for all the 30 columns which gives us a correlation score that would be used to measure the performance of the various models applied in the research, also as mentioned above custom call back function have been used in the research which calculates the average spearman correlation for the validation set after each epochs to measure the performance of the model. Below are the experiment results with all the three models used in the research.

6.1 Experimenting with the Word Embedding Model:

On going through the architecture of the model taken from the research by (Lee and Dernoncourt, 2016) where LSTM and CNN layers were combined could lead to the better performing model but the max correlation score which we could reach in the research for this approach was 0.25 validation score and 0.23 test score which is average spearman correlation for all the 30 predicted vs the actual columns. On looking at the below graph in figure 11 which leads to prove that the model was trained well without much of over fitting or under fitting. Various hyperparameters such epochs and batch size were tested to get the best performance from the LSTM model.

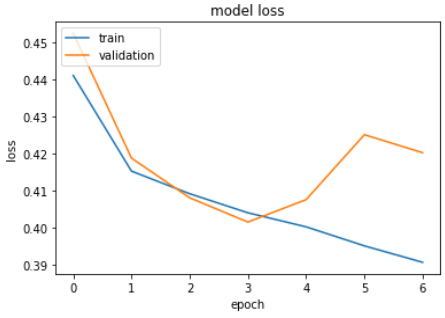


Figure 11 Train vs Validation Loss

6.2 Experimenting with the Transformer Model

For the transformer model various architecture were used but adding a sigmoid output layer gave us a validation score of 0.35 which is the average spearman correlation of all the 30 attributes for the question answer pairs in the validation set. While training the model directly with the training data it seems the models is overfitting whereas when the k-fold validation taking k=5, the model seems less overfit.

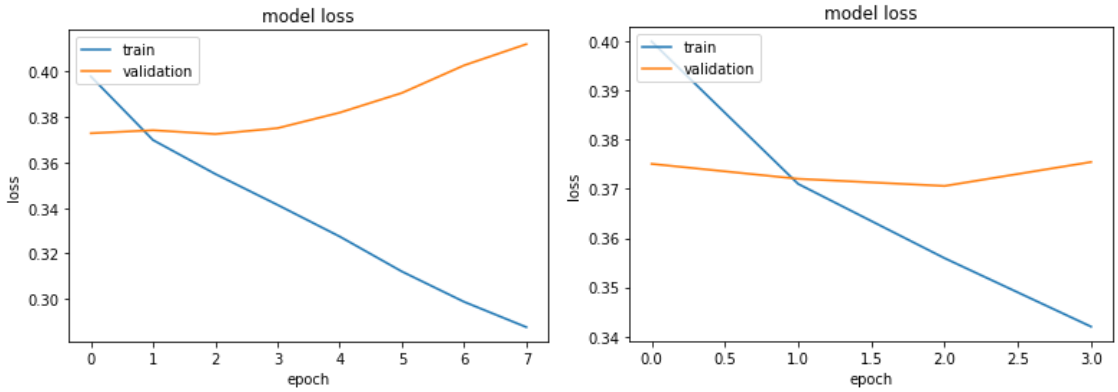


Figure 12 Training Data directly vs Training Data with K-fold validation

While looking at the dataset for the research it comprises of question and answer pair from various categories such as 'LIFE_ARTS', 'CULTURE', 'SCIENCE', 'STACKOVERFLOW', 'TECHNOLOGY', in order to increase the performance we came up with the hypothesis that if a model is created taking similar categories in one group could lead to a better performing model. Thus, from the above categories only 'SCIENCE', 'SATCKOVERFLOW', 'TECHNOLOGY' was chosen to train the BERT model with same architecture. The results were not as expected as the spearman correlation score went down to 0.35, hence rejecting the hypothesis that taking a subset of the dataset could lead to better performance.

6.3 Experiments with Universal Sentence Encoder Model

For the sentence encoder model transfer learning approach was taken where all the sentence were converted into vector form and then given as an input to neural net model with just one fully connected hidden layer, while training the data the validation score is 0.38 and the test score 0.37 which is correlation between the predicted and the actual columns in the dataset. On looking at the figure 13 one can make out that model has trained quite well, it is neither showing signs of under fitting or over fitting.

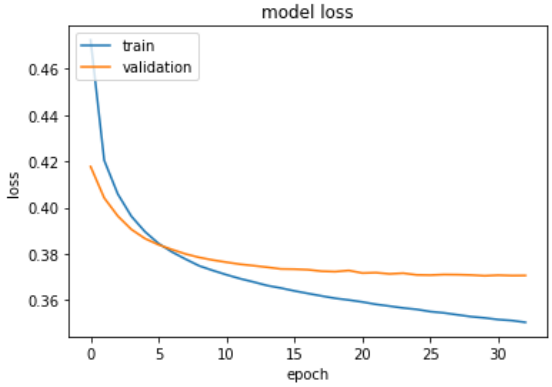


Figure 13 Train vs Validation Loss

As mentioned earlier the final score is the average of the correlation score off all the 30 predicted columns with the actual columns, on looking at the separate correlation score off all the 30 attributes, to increase the performance a hypothesis was taken where if we reduce the attributes or column which have least correlation score and creating a model after removing those columns may lead to a better performing model. While going through the correlation score most of the columns were correlation score were near or more than 0.37 which Is a average correlation thus this hypothesis too was rejected.

6.4 Discussion

This research is based on the fact that in the question answer system today date gives good performance when subjected to factual or verifiable answers but when the question are subjective or opiniated, the question answer system fail to understand it. The major cause for the lack of understanding is due to lack of labelled data present for the domain. To rectify that problem Google came up with a dataset that is hand labelled by crowdsource team on the 30 different subjective aspect of question answer pair. Various models were applied to figure out which models suits best for the dataset. At the beginning of the research it was perceived that BERT model will suit best for the data but from the table 4 we can make out that for this particular dataset Universal Sentence Encoder model gives performance equal if not better than the BERT model which makes it interesting find as for other state of the art NLP downstream task BERT models outperforms the Sentence Encoder model.

Table 4: Results for all the experiments conducted

Experiment	Validation Score	Test Score
BERT	0.37	0.35
BERT with Specific categories	0.30	0.28
Sentence Encoder	0.38	0.37
Word Embedding	0.21	0.19

7 Conclusion and Future Work

Table 5: Individual Correlation Score from Universal Sentence Encoder Model

Attribute	Correlation Score	Attribute	Correlation Score
question_asker_intent_understanding	0.378096383	question_type_entity	0.460414953
question_body_critical	0.635775816	question_type_instructions	0.725225189
question_conversational	0.327981595	question_type_procedure	0.320265667
question_expect_short_answer	0.262187033	question_type_reason_explanation	0.6343764
question_fact_seeking	0.300049061	question_type_spelling	0.543322211
question_has_commonly_accepted_answer	0.437480984	question_well_written	0.522398824

question_interestingness_others	0.35310593	answer_helpful	0.16854419
question_interestingness_self	0.48207745	answer_level_of_information	0.47672418
question_multi_intent	0.44263126	answer_plausible	0.12449847
question_not_really_a_question	0.05935281	answer_relevance	0.14108906
question_opinion_seeking	0.38307042	answer_satisfaction	0.36345406
question_type_choice	0.60298167	answer_type_instructions	0.70431503
question_type_compare	0.38779169	answer_type_procedure	0.22026872
question_type_consequence	0.19181927	answer_type_reason_explanation	0.63881359
question_type_definition	0.38277411	answer_well_written	0.07518665

This research primarily focuses on creating a model which given any question answer pair would be able to rate the subjective attribute of each question answer between 0 to 1. On looking at the results of correlation score 0.38 which is average correlation score of all the 30 columns, may not seem very good. But when we go deeper and look at the correlation score of each column from table 5. We could find that some subjective attributes such as "question_type_instructions", "question_type_choice", "answer_type_reason_explanation" which scored 0.70, 0.60, 0.64 respectively and some attributes like question_type_consequence have score of 0.19. Ongoing through the individual results of the attributes which does not have good correlation score, one could say that there is a possibility that few of the attributes labelled by the team may not be correct. The complete dataset is also not released by the Google as this dataset is a part of Google ongoing competition.

There is lot of possibility for the future work as the max final score which this research could reach is .38 thus there could be one area which could be taken by other research to increase the correlation score. This research is totally based on transfer learning approach where the pre-trained embedding/models trained on the Wikipedia corpus were used to get the results, but there could be scenario where the pre-trained model does not contain all the words present in the dataset such as various technology questions. In that case it would be a wise option to pre-train BERT / GloVe / Sentence encoder with the complete data of stack exchange website as the question and answer pairs in the dataset were taken from various StackExchange websites, once trained we should take the embedding and then use it to train the model with given dataset. This approach could result in better correlation score for the trained models.

References

Adiwardana, D. *et al.* (2020) 'Towards a Human-like Open-Domain Chatbot', *arXiv:2001.09977 [cs, stat]*. Available at: http://arxiv.org/abs/2001.09977 (Accessed: 29 July 2020).

Devlin, J. *et al.* (2019) 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding', in *Proceedings of the 2019 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). *NAACL-HLT 2019*, Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. doi: 10.18653/v1/N19-1423.

Ferrucci, D. *et al.* (2013) 'Watson: Beyond Jeopardy!', *Artificial Intelligence*, 199–200, pp. 93–105. doi: 10.1016/j.artint.2012.06.009.

Goldberg, Y. (2019) 'Assessing BERT's Syntactic Abilities', *arXiv:1901.05287 [cs]*. Available at: http://arxiv.org/abs/1901.05287 (Accessed: 30 July 2020).

Huang, H. *et al.* (2019) 'DCNN-BiGRU Text Classification Model Based on BERT Embedding', in *2019 IEEE International Conferences on Ubiquitous Computing Communications (IUCC) and Data Science and Computational Intelligence (DSCI) and Smart Computing, Networking and Services (SmartCNS)*. *2019 IEEE International Conferences on Ubiquitous Computing Communications (IUCC) and Data Science and Computational Intelligence (DSCI) and Smart Computing, Networking and Services (SmartCNS)*, pp. 632–637. doi: 10.1109/IUCC/DSCI/SmartCNS.2019.00132.

Lee, J. Y. and Dernoncourt, F. (2016) 'Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks', *arXiv:1603.03827 [cs, stat]*. Available at: http://arxiv.org/abs/1603.03827 (Accessed: 31 July 2020).

Mikolov, T. *et al.* (2013) 'Efficient Estimation of Word Representations in Vector Space', *ICLR*.

Munikar, M., Shakya, S. and Shrestha, A. (2019) 'Fine-grained Sentiment Classification using BERT', in *2019 Artificial Intelligence for Transforming Business and Society (AITB)*. *2019 Artificial Intelligence for Transforming Business and Society (AITB)*, pp. 1–5. doi: 10.1109/AITB48515.2019.8947435.

Pennington, J., Socher, R. and Manning, C. (2014) 'Glove: Global Vectors for Word Representation', in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. *EMNLP 2014*, Doha, Qatar: Association for Computational Linguistics, pp. 1532–1543. doi: 10.3115/v1/D14-1162.

Ravichandran, D. and Hovy, E. (2002) 'Learning surface text patterns for a Question Answering System', in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. *ACL 2002*, Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, pp. 41–47. doi: 10.3115/1073083.1073092.

Rogers, A., Kovaleva, O. and Rumshisky, A. (2020) 'A Primer in BERTology: What we know about how BERT works', *arXiv:2002.12327 [cs]*. Available at: http://arxiv.org/abs/2002.12327 (Accessed: 31 July 2020).

Simmons, R. F., Klein, S. and McConlogue, K. (1964) 'Indexing and dependency logic for answering english questions', *American Documentation*, 15(3), pp. 196–204. doi: 10.1002/asi.5090150306.

Vaswani, A. *et al.* (2017) 'Attention is All You Need', in. Available at: https://arxiv.org/pdf/1706.03762.pdf (Accessed: 1 April 2020).