CLASSIFICATION OF ANSWERS AND QUESTIONS USING NATURAL LANGUAGE PROCESSING

BY:
OLIVIA RUTAYISIRE
213000127

UNIVERSITY OF RWANDA
COLLEGE OF BUSINESS AND ECONOMICS
AFRICAN CENTER OF EXCELLENCE FOR DATA SCIENCE

SUPERVISOR:
PROF. ANTHONY WAITITU

This thesis report is submitted to the African Centre of Excellence for Data Science of University of Rwanda in partial fulfillment of the requirements for the award of the degree of:

MSc. DATA SCIENCE (DATA MINING)

September, 2020

DECLARATION

I hereby declare that this dissertation entitled "Classification of answers and questions using Natural Language Processing" is the result of my own work to the best of my knowledge. It has not been previously submitted for any other degree at the University of Rwanda or any other institution.

Names: Olivia Rutayisire

Signature:  Date: 28th Sept 2020

APPROVAL SHEET

This dissertation entitled Classification of answers and questions using Natural Language Processing written and submitted by Olivia Rutayisire in partial fulfilment of the requirements for the degree of Master of Science in Data Science majoring in Data Mining is hereby accepted and approved. The rate of plagiarism tested using Turnitin is 16% which is less than 20% accepted by ACE-DS.

Prof. Anthony Waititu

Signature: 
Date: 28th Sept 2020

_____
Supervisor

_____
Head of Training

ACKNOWLEDGEMENTS

First and above all, I thank The Almighty God for all His Timely Provisions.

I would like to also thank the following people for helping with this research project:

My supervisor, Prof. Anthony Waititu, for providing guidance, feedback and encouragement throughout this project.

My fellow students for all your contributions throughout this project.

My parents, siblings and friends for your prayers, encouragements and other kinds of supports.

ABSTRACT

CLASSIFICATION OF ANSWERS AND QUESTIONS USING NATURAL LANGUAGE PROCESSING

BY:

OLIVIA RUTAYISIRE (MSc. DATA SCIENCE)

UNIVERSISITY OF RWANDA, 2020

The last decade has marked a rapid and significant growth of digital technology globally to drive our society. One of the key solutions that are being adopted by institutions to get aligned with this trend is the use of question answering systems and chatbots to automate some of the services that their users might need. Some of the automated services are questions asked by the users. The biggest challenge lies in the classification of questions and answers the way a human being would do. This research aims to identify advanced implementations that can be used to optimize the usage of question answering systems. Three different models have been built in python and trained on Kaggle labeled dataset of classified questions and answers from various perspectives, to better understand the questions and their respective answers. The labeled features are 21 questions related features and 9 answer features each ranked in a range of 0 and 1. The algorithms attempted to use in the models are Ridge Regression, Recurrent Neural Network using Long Short Term Memory, and a Neural Network using Keras library. Ridge regression obtained a maximum validation accuracy of 0.37, Recurrent Neural Network with Long Short Term Memory had an accurary 0.40, and Keras with Neural Network performed better on our training dataset with a validation accuracy of 0.58. Better models should be applied to text data for text classification such as BERT models and also consider using more features on the training model to classify better. Additionally, focusing on fewer perceptions but meaningful while choosing labeled features to boost the accuracy of the model.

Keywords: Text Classification, Question Answering systems, NLP, Ridge Regression, Deep QA, Long Short Term Memory, Keras, Neural Network.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

NLP: Natural Language Processing

QA: Question Answering

XML: Extensible Markup Language

CNN: Convolutional Neural Network

RNN: Recurrent Neural Network

LSTM: Long Short Term Memory

BLSTM: Bidirectional Long Short Term Memory

Q&A: Question and Answer

AI: Artificial Intelligence

NLU: Natural Language Understanding

SVM: Support Vector Machine

NN: Neural Network

TF- IDF: Term Frequency – Inverse document frequency

DAN: Deep Averaging Network

URL: Uniform Resource Locator

AUC: Area Under the ROC Curve

ROC: Receiver Operating Characteristic

BERT: Bidirectional Encoder Representations

1. INTRODUCTION

1.1 Background

Globally, the world is emerging in a society driven by digital technology. Rwanda experienced an increase from 7.9 percent in 2010 of internet penetration to 52.1 percent at the end of 2018 (Xuequan, 2019). This spread of internet facilitated business processes and digital marketing had a huge influence on people's interactions, work, purchases, and life habits. One of the ways for a company to engage with its target audience to pursue its business objectives is to improve the interactivity of the company and the customers. Companies started marketing their products and services over the internet because that is where their audience was, and also customers are adopting the culture of seeking support over the internet, either on social media platforms or business websites.

Traditionally, customers call or email the company in case of more information or when requiring support. These requests have to sometimes wait until working hours of a company which is not always the same time as when a customer will encounter a challenge.

Additionally, as the customer support team is serving many customers, the customer will have to queue until his/her turn.

Most of the time, they offered support or answered questions are sometimes on the website or don't really require any technical approach to be solved, instead only guidelines or additional information would set it up.

This process of acquiring this information and get support do sometimes take a long time and are locally solved by hiring more customer support workers which are not feasible in case the company does not have that budget. However, this is not even helpful to the customer service team as they keep on answering the same questions again and again. The traditional ways of capturing data either by messaging platforms (emails, website inquiries) or tokens, companies via the customer support teams are gathering a lot of data. This study attempted on mining for information retrieval gathered in the past and the automation of some of the questions that were being replied by the customer service team using natural language processing and data mining techniques.

Currently, one of the solutions adopted by big messenger platforms such as telegram, Facebook, and Slack, is chatbots. However, the field of question answering and chatbots in general still face the challenge of training the chatbots and question answering systems so that they understand and classify the questions and answers correctly. This challenge is related to not having a high-quality dataset that represents well the intents and entities supported by the chatbots or question answering systems causing them to have low accuracy compared to other simple text classification tasks (Ahmad Abdellatif, 2020) . Locally, training models on external datasets may not reflect the country's local culture and writing styles/language.

1.2 Research Objectives

This study aims to identify advanced implementations that can be used to optimize the usage of question answering systems using Natural Language Processing (NLP) and other data mining techniques.

Objective 1: To create a database of labeled questions and answers for improved question answering systems.

Objective 2: To apply the Ridge Logistic Regression algorithm to classify the questions and answers.

Objective 3: To apply Neural Network algorithms to classify the questions and answers

Objective 4: To compare the Ridge Logistic regression algorithm with the Neural Network algorithms.

1.3 Scope of the research

This study used an open-source dataset. The data has been explored using data mining techniques to classify the questions and answers through the NLP algorithm. The NLP algorithm has been trained on text data only. The study only focused on understanding and classifying questions and answers used in real life so that so it can be used to train question answering systems to perform better.

1.4 Structure and timeframe of the thesis

This introduction gives an overview of the study and introduced the research Objectives. The next course of my research is a thorough Literature Review to provide a state-of-the-art review of existing literature on the research subject. The third chapter the Research Methodology, explaining the selection process of the research method and design of the chosen approach. The Results chapter focused on analyzing and presenting the findings generated from the implementation of the research methodologies and experiments phase. The final chapter, Conclusion, summarizes the results, proposes recommendations and areas for further research.

2. LITERATURE REVIEW

As mentioned earlier, one of the technologies adopted by researchers to improve the digital communication of business entities as well as other institutions is Chatbots. A chatbot is a computer program that processes natural-language input from a user and generates relative responses that are then sent back to the user (Das, 2018). Currently, one of the solutions adopted by big messenger platforms such as telegram, Facebook, and Slack. Such challenges can be resolved by Questions Answering systems as well. A Question Answering(QA) system is an information retrieval system in which a direct answer is expected in response to a submitted query, rather than a set of references that may contain the answers (R.Mervin, 2013). Looking at customer experience, often customers be required to spend more time by going through some full documents or guides from a website looking for the answer, instead, customers are looking for short and precise answers, just like the way a customer support agent would have replied to that question. The difference between a Chabot and a QA system is that a chatbot are able to engage in deep dialogs to help perform specific tasks and sometimes mimic human interactions using intents and also as the chatbots engage deep conversations with users, and develop a way of learning from them for future conversations with the users with a friendlier manner, though some chatbots designers might decide to not include interactivity functions. Another difference is that chatbot have been pattern matching systems but question answering systems generally use machine learning technology (Endicott, 2016). However, a QA system is supposed to answer questions related to a specific source of information. The benefit of using a QA system is that instead of pooling the whole document where one will have to read it to fill the answer, the system will extract only the piece of information in that document needed. This may be in a form of a short answer or a paragraph containing the answer, depending on the nature of the question. On the other hand, a chatbot can also do information retrieval to only pull the targeted answer within a document using the QA system technologies as their backbone to perform such tasks and can also help one go through the required steps to accomplish a task further to providing information (Taylor, 2019). As we want to focus on the automation of customer service answers for improved digital communication, we do not need to hold a long conversation with the user, instead of providing a quick answer to the user's question would be enough as the system will reduce the search time to get the exact time as well even getting the answer that would have missed to the user. Therefore, the emphasis on the QA system over a chatbot will be explored in this study.

1.1 History of Question Answering Systems

The systems that were able to process queries using Natural Languages started in the 1950s, the "Imitation Game" widely known as "Turing Test" that allows human and machine converse through an interface such as the “Natural Language Interface to Databases (NLIDB)” which was the first QA system. It created the avenue for users to present queries in natural language and retrieve responses from databases (Androutsopoulos et al., 1993).

The other two first invented question answering systems were BASEBALL and LUNAR. BASEBALL was designed to answer questions on the US baseball league over a period of one year. LUNAR, was targeting the geological analysis of rocks returned by the Apollo moon missions kind of questions. These mentioned question answering systems were very successful in their domains of interest. LUNAR was showcased in 1971 at a lunar science convention and was able to answer 90% of the questions in its domain asked by persons who were not trained on it. New closed-domain question answering systems were designed in the following years. The common feature of all the mentioned systems is that they all had a database or knowledge system that was written by experts of the chosen domain. The language abilities of BASEBALL and LUNAR used techniques much the same to ELIZA and DOCTOR (Wikipedia, 2020). The evolution of Question answering systems continued and as a very highly effective question-answering program in the late 1960s and early 1970s has been developed by Terry Winograd named SHRDLU. This program was a closed domain with physics rules of physics designed to be easily encoded by a computer program and this was one of its key strengths.

Knowledge bases were introduced in the 1970s, and question answering systems experts arranged them and produced more recurring and existing responses within the field of that knowledge base.

Other later developed question answering systems are EAGLi (for health and life scientist’s domains), and Wolfram Alpha. As the evolution of question answering is still progressing, currently the most known successful system is an IBM computer system named Watson, famous to date after winning the Jeopardy challenge in February 2011 over World's best

champions (Tampa Bay Times, 2014). IBM Watson was developed and trained to answer questions on the quiz show Jeopardy.

1.2 The Framework and architecture of Question Answering systems

1.2.1. Framework of Question Answering Systems

The evolution of question answering programs introduced various types of implementing such systems. Question answering systems are also categorized based on their available resource for answers. The answers also vary depending on the users' needs of general information on a general topic while other users need specific information from a particular application domain (Anjali Saini, 2017).

The types of answers are classified into Open-domain question system and Closed domain question system.

Open Domain Question System can answer questions from all domains. The questions are not restricted to any specific domain and provide a short answer to a question, addressed in natural language. Open Domain systems help search engines find the required answer. The web is the best source to get information with huge potential to extend the usage of the internet and most of the Web-based Question Answering systems work for open domains (Deepa Yogish, 2016). This is appropriate for a huge number of casual users(users searching for various questions in different domains) (A. Chandra Obula Reddy, 2017),

Closed Domain Question System refers to question answering systems that can answer only questions in a specific domain. The answered questions will therefore be restricted to that specific domain. As the system learns from a specific domain from a limited repository, the quality of answers is high. Closed domain QA systems are designed to get answers from structure data (such as databases), unstructured data (free-texts), and semi-structured data (such as XML). As the closed domain system results in more accurate answers compared to the open domain, multiple closed domain systems can be combined to get an Open-domain QA system with higher accuracy (A. Chandra Obula Reddy, 2017).

The types of questions that the system will answer also has to be defined beforehand to use efficient techniques.

The various types of questions are:

Factoid type questions

The factoid type questions are questions that require a single fact (Mark A. Greenwood). These questions normally start with “what”, “which”, “when”, “who” or “how”. Questions like “What is the largest country in Africa” having short answers like “Algeria”. These questions are simple to answer and need answers in a single sentence or short phrase. The performance of QA systems on factoid questions actually gives a satisfactory (Anjali Saini, 2017). To reach to satisfactory, factoid systems use named entities answer types. Named Entities are sets of elements that the system picks as important while trying to understand text data. Some common used entities are parts of speech (like nouns, verbs, adjectives, etc). Named Entity Recognition is used while understanding the details within a sentence by particularly using nouns (DeepAI, n.d.).

List type questions

These are the kind of questions that require multiple facts to be replied as an answer to a question. Questions like “Restaurants open in Kigali Town before 8 am”. Such questions will give a list of all possible answers. Such answers types are named entities to enhance a good accuracy on answers provided by the QA systems.

Confirmation Questions

These are the kind of questions that requires answers in the form of YES or NO answers. Questions like "Is the capital city of Rwanda Kigali?", look for answers yes or no. To answer the confirmation questions world knowledge, inference mechanism, and common sense reasoning are required (A. Chandra Obula Reddy, 2017).

Causal Questions [why or how]

Causal questions are the kind of questions that needs the reason for a certain thing as an answer. They are not named entities as factoid or list type questions but they are instead looking for the description about an entity. This deals with questions such as "Why can we choose to use causal questions over other types of questions?"

Hypothetical Questions

These are the questions to request for information about a hypothetical event. These questions usually start with "What would happen if?". The accuracy of these answers is low and depends upon users and context (A. Chandra Obula Reddy, 2017).

Complex questions

Complex questions are more difficult to answer and whose answers generally consist of a list of "nuggets". Complex questions such as "What are the reasons for Covid-19 spread?" often require inferring and synthesizing information from multiple documents to get multiple nuggets as answers. Complex procedures are needed to answer complex questions (A. Chandra Obula Reddy, 2017).

1.2.2. The general architecture of a QA system

The architecture of Question answering systems generally follows the pipeline of three major modules namely the Question Analysis, Document Retrieval, and the Answer Extraction (Bolanle Ojokoh, 2019). The whole pipeline is shown in the figure below.

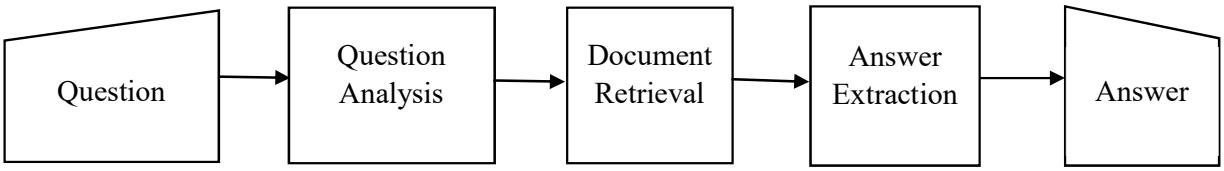


Figure 1: Question Answering System architecture

Question Analysis

As discussed above, there are many types of questions that can be asked to a QA system, the question analysis in the QA architecture serves as a stage to classify the asked question in the relevant type of question so that the system understands what the user is trying to ask and thus extract the relevant answers. In the question Analysis part, first, when a question is posted in the QA system, the question is then broken into tokens of words. Tokens are the building block of a sentence, which can either be words, character or sub words (Pai, 2020) . The answer type detector then recognizes the type of answer using the created tokens (Heba Kurdi, 2014). In the question classification, the system also determines the question focus of further steps by extracting the relevant keywords from the tokens regenerated. Sometimes, depending on the system, reformulating the question in a form that the system will be able to better understand (Manvi Breja, 2018). Below are some examples of words found in a question and their expected answers (Rosy Madaa, 2012).

Question Classification	Expected Answer Type
When	Date
Which	Person/Date/Location
Why	Reason
What	Person/Date/Location
Who	Person

Table 1: Question classification and expected answer type

Various NLP techniques such as tokenization (word tokens), stemming, lemmatization, Part-Of-Speech(POS), Parsing, and NER are used to have good question classification as cited above on question types.

Document Retrieval

This aims to look in the database for the relevant document or information that might answer the asked question. This module searches a list of related documents and extracts a set of paragraphs based on the focus of the question. This stage receives the queries formulated from the question analysis module and searches information sources for suitable answers to the asked questions. Various answers from various sources such as the Web or online databases can also be stored for reference (Bolanle Ojokoh, 2019). This process is actually broken down into three main actions: processing, retrieval, and ranking. The processing stage involves the use of database query for the relevant answers in the database, the retrieval stage then involves the matching of documents to the asked questions according to their resemblance (Bolanle Ojokoh, 2019). Most appropriate passages are selected according to passage score for an answer. (A. Chandra Obula Reddy, 2017). It returns a ranked list of relevant documents in response to a reformulated question (Manvi Breja, 2018).

The Taxonomy of QA systems usually differs on the ways used to retrieve the documents, as the document/information retrieval stage of a QA is both the core objective of the whole system and its implementation is complex. It requires both the use of Natural Language techniques and conventional Information Retrieval systems (Bolanle Ojokoh, 2019). There have been various ways of doing so but the most used and frequently categories are only two among others. The Knowledge-based system is one of the most used QA system taxonomy, which draws upon manual rules, ontologies, and large-scale knowledge graphs to deduce answers (Feuerriegel, 2018). Another most used taxonomy is the Information Retrieval module selects candidate

documents based on a chosen similarity metric, while a subsequent module then processes these to extract the answer (Feuerriegel, 2018).

Answer Extraction

The answer extraction module process the retrieved documents with their corresponding scores and the answers can be extracted from the high-ranked sentences (Pooja, 2012). The extracted answer with the highest rank is the one taken as the correct answer and the rest answers are taken as candidate answers. But also this extraction is done by using the similarity checker by considering the counts of matching keywords between the question and each retrieved answer; and answer ranker which sorts the answers in descending order of their relevance (A. Chandra Obula Reddy, 2017).

Deep QA

To improve the performance of the QA systems under the information retrieval stage, the extension and attempts of novel neural network models have been proven to be successful. The attempted neural network models are Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) such as LSTM and BLSTM, among others (Li Deng, 2018). The use of these Neural Network models has been incorporated both under the Knowledge Base and other QA taxonomies for better performances. These attempts can be categorized into two main models, either the information extraction style or the semantic parsing style (Li Deng, 2018). Semantic parser involves that the system first understands the real meaning of a question before providing its answer. Meaning that it transforms natural language sentences into a meaningful representation that the computer can better understand. It involves finding relationships in the text. Used by IBM Watson, Text summarization tasks. But on the other hand, Information Extraction extracts meaningful data within a text and uses it to extract the corresponding topic entities, ranks the candidates' answers to conclude with the answer with the highest rank. The below graph shows their graphic processing.

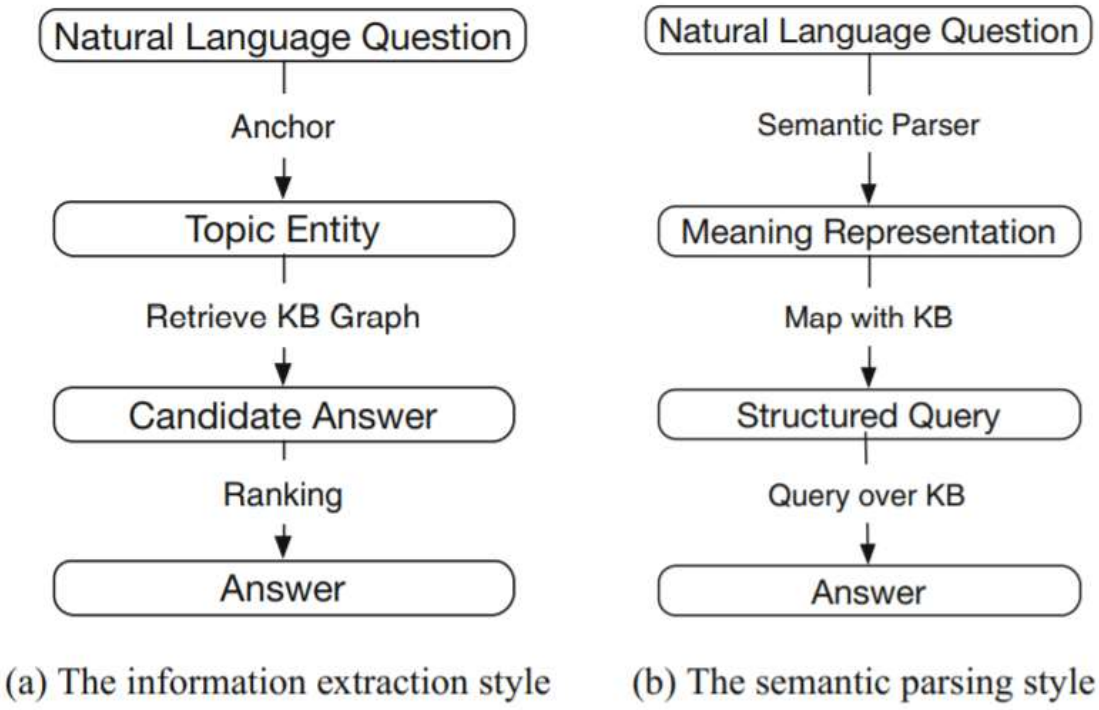


Figure 2: Information extraction style vs Semantic parsing style (Li Deng, 2018)

DeepQA is an effectual and expansible architecture that can be used as a pillar for combining, deploying, evaluating, and advancing a wide range of algorithmic techniques to rapidly advance the field of question answering (David Ferrucci, 2010).

3. RESEARCH METHODOLOGY

The thesis began by providing a comprehensive literature review to build up a theoretical framework around different technologies that would resolve our problem. Both Chatbots and Question answering systems have been evaluated and concluded that a question answering system would be more ideal for the study than a chatbot, based on the literature review findings. The history and existing frameworks for a question answering system implementation have been reviewed. This research subject is still in evolution and more new technologies are being tested over the question answering systems. Sometimes the later technologies were not much researched on, such as various deep QA methods. However, more insights taken to answer this research question were still missing. The purpose of the research was to understand the contribution of questions and answers understanding to successful question answering system implementations, how success can be defined, and what best practices could be identified.

This chapter introduces the selected research methods used in the study and then continued to explain how data gathering, processing and analysis were executed.

3.1 Selecting the research method

By definition, primary data is data that is collected by a researcher in person, using methods like surveys, interviews, or experiments. It is collected with the research project in mind, directly from primary sources. Whereas, secondary data is the kind of data gathered from studies, surveys, or experiments that have been run by other people or for other research (Glen, 2018). The training of a question answering system requires a huge dataset as the algorithm's accuracy increases with the increase of the training dataset. Is our case, using primary data would require us a lot of time to reach to a considerable dataset. However, managed to pick the best question answering dataset amongst many other open-source question answering datasets that match with the research topic objectives, the "google quest Q&A labeling dataset". Google quest Q&A labeling dataset is an open dataset among google open-source datasets specifically designed to contribute to building better subjective question-answering algorithms as this field was still challenged by the lack of data and thus effective predictive models as well (Kaggle, 2020). The Google quest Q&A labeling is a Kaggle challenge launched on November 22nd,2019 whose dataset was used to predict algorithms for different subjective aspects of question-answering. The question-answer pairs were collected from nearly 70 different websites, in a "common-sense" like fashion with quality scoring aspects for both questions and

answers. The raters (for scoring the questions and answers) recorded in the dataset received minimal guidance and training and completed the work by relying almost on their subjective interpretation of the instant. As such, each instant was formulated most intuitively so that raters could just use their common-sense to do the task.

The dataset contains questions and answers from various StackExchange properties. The training dataset contains 30 target labels consisting of 21 question-related labels and 9 answer related labels in addition to the provided question and answer text and other question text details. Both the questions and answers labels are given in a continuous range between 0 and 1. Below are the features columns and both the questions and answers target labels in our used dataset.

Table 2: Google Quest features

Features columns	**Questions Related labels**	**Answers related labels**
1.Qa_id	1.question_asker_intent_understanding	1.answer_helpful
2.Question_title	2.question_body_critical	2.answer_level_of_information
3.Question_body	3.question_conversational	3.answer_plausible
4.Question_user_name	4. question_expect_short_answer	4.answer_relevance
5.Question_user_page	5. question_fact_seeking	5. answer_satisfaction
6.Answer	6.question_has_commonly_accepted_answer	6. answer_type_instructions
7.Answer_user_name	7. question_interestingness_others	7. answer_type_procedure
8.Answer_user_page	8. question_interestingness_self	8.answer_type_reason_explanation
9.url	9. question_multi_intent	9. answer_well_written
10.Category	10. question_not_really_a_question	
11.Host	11. question_opinion_seeking	
	12.question_type_choice	
	13. question_type_compare	
	14. question_type_consequence	
	15. question_type_definition	
	16. question_type_entity	
	17. question_type_instructions	
	18. question_type_procedure	
	19.question_type_reason_explanation	
	20. question_type_spelling	

	21. question_well_written	

In addition to the above features, this training dataset is made of 6079 rows of examples and the test dataset contains 477 questions and answers to classify.

The benefits of this dataset over other available QA systems training sets is that training our algorithm with the use of the questions and answers labels will improve the performance of question answering algorithm as the training dataset contains labeled questions and questions, which was not the case on the previously released training datasets.

3.2 Selecting analysis tools

Python

The data analysis and building the QA algorithm requires a programming language. In our endeavor to identify the best programming language for our research objectives, Python has taken a big lead.

Python is one of the most popular programming languages used by developers today. Guido Van Rossum created it in 1991 and ever since its inception has been one of the most widely used languages along with C++, Java, etc (Cuelogic Insights, 2016).

Python is available to all Operating systems and is also an open-source software offering titled CPython which is garnering widespread popularity as well.

Python also offers the flexibility to use a variety of libraries and frameworks be it its large range of pre-built libraries like Numpy and Scipy and also support other Machine Learning libraries like Scikit Learn, Keras, Pytorch, NLP libraries like NLTK, and also frameworks to deploy the algorithms such as Flask and which are very suitable to QA algorithms.

Kaggle

Kaggle is the largest community of data scientists and machine learners in the world. Kaggle started by only offering machine learning competitions but extended towards a public cloud-based data science platform. Kaggle is not only a platform that helps to solve difficult problems, hire strong teams, and showcase the power of data science but it does also help access various libraries and frameworks that would have been incompatible to the author's local device.Kaggle

3.3 Research Technique: Machine Learning

Machine learning is an artificial intelligence (AI) application providing systems the ability to automatically be trained and ameliorate from experience without being definetly programmed. Machine learning emphasizes on the computer programs development having accessibility to those data and make use of them to learn for themselves (What is Machine Learning? A Definition, 2020). Machine learning fit existing data to some model, creating a representation of the general data representation trend that can make decisions or generate predictions on new data based on mined patterns. In practice, the above is implemented by choosing a model that best unfold relationships between the target data and the input, specifying a form that includes parameters and features, then using some optimization procedure to minimize the error of the model on the training data. The model would then be used to new data on which it will produce prediction returning labels, probabilities, membership, or values based on the model format. As our training dataset was labeled and was enabling the computer system to learn from it and be able to accurately predict future observations, the author used machine learning techniques to achieve the goals of this study. Machine learning bein computer systems, the data should be in the form that will be understood by the computer, which is generally in a numeric form. But in our case, we are dealing with text data, which requires other machine learning techniques such as Natural Language Processing to process and manipulate such data.

Natural Language Processing (NLP) is a text data mining deals with the relationship between human natural languages, that are not understood by computers and artificial intelligence. NLP has therefore been used in our study to process our text features. This mainly involved the use of Natural Language Understanding (NLU) which is a sub-topic of NLP that breaks down the human language into a machine-readable format. NLU uses grammatical rules and common syntax to understand the overall context and meaning of “natural language”. NLU goal is to understand written or spoken language the way a human would.

3.3.1. Machine Learning Models

Our target labels on both questions and answers are the ranks between 0 and 1 of each label among all the 30 labels to predict. Our study is a text classification problem but should return the classification values in terms of the ranks of each class. Such classification required advanced techniques such as machine learning models. Below are different machine learning models used for this classification.

Ridge Logistic Regression

The author selected ridge regression among other many parametric machine learning models because previous researches have shown that ridge logistic regression has successfully been used in text categorization problems and reached to the same performance as other powerful models such as the SVM but with the main benefit of returning a probability value rather than a score (Sujeevan Aseervatham,Anestis Antoniadis, Eric Gaussier,Michel Burlet,Yves Denneulin, 2012). This corresponds to our research objectives of ranking the questions and answers to a feature label with values between 0 and 1. Ridge regression also works well when the number of features is larger than the number of observations and/or when the features are highly correlated. In our case, the dataset has many labeled features and has to be optimized with the use of parameters and weights(alphas) on each feature to control the complexity of the classifier. Logistic regression almost works like a simple linear regression but differs slightly from it by its way of adding a penalty to regularize the model to avoid overfitting, which will used in our study for better results. Regularization means to shrink the weights of the features to zero to avoid overfitting on the training model. The normal Linear regression equation used to predict the value of the dependent variable is $h(x) = \hat{Y} = \omega_0 + \omega_1 x_1 + w_2 x_2 + \cdots \omega_n x_n$, whereby the machine learning model tries to assign the best weights(ω) to each variable (x). To set the optimized weights to each variable, the regression considers particular criteria while choosing a certain weight over other suggested weight values. The linear regression applies a cost function which adds an error term to the weights so that the weights are overfitted. The cost function formulae(also called the squared error function) to minimize the weight for $h(x) = \hat{Y} = \omega_0 + \omega_1 x$ is

$$J[\omega_0, \omega_1] = \frac{1}{2m}\sum_{i=1}^{m}(h_0(x^{(i)}) - (y^{(i)})^2$$

Where the actual value and of y and the predicted value is called the error term.

The above equation provides constants weights that satisfy the equation, but we still want to choose the weight with the least cost function (least error term) to be attributed to our linear regression equation. Regression uses gradient descent to reach the minimal possible point of the cost. The gradient descent doesn't only reduce the cost function but also other functions for the linear regression. Gradient Descent accomplishes the task of moving towards the sharpest descent (global minima) by using updated equation for weight with gradient descent is:

$\omega_j = \omega_{j-\alpha}(d\omega_{j-1})$ *where* ω_j is the cost function, α is a learning rate

Ridge regression also uses all the above functions used by linear regression but also uses the *l2* norm penalty to its cost function. The equation of ridge regression is, therefore:

$$cost(\omega) = \frac{1}{(2 * n)} \sum_{i=1}^{i=n} (y_i - \overline{y_i})^2 + \alpha \sum_{j=1}^{j=D} \omega_j^2$$

The α fixed values in the second term of our above equation are hyper-parameters used on ridge regression is tuned and set it to a particular value based on one's choice. This hyper-parameter and squaring the cost function makes Ridge regression suitable for training datasets with many variables. Each variable's cost function(ω) has been tuned to its own α to minimize the overfitting. A manual grid search for suitable alpha values for each value has been computed and applied to our ridge regression model.

Deep QA

As stated above, the improvement of question answering systems is now being tested and upgraded using deep QA. Our target labels on both questions and answers are the ranks between 0 and 1 of each label among all the 30 labels to predict. Our study is a text classification problem.

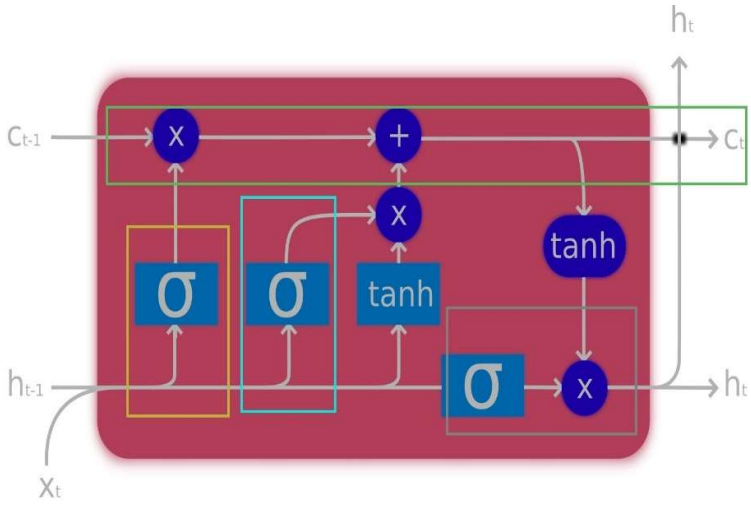
To ensure the improvement of question answering systems 2 different neural network algorithms have been used throughout the rest of the work, namely Long Term Short Memory (LSTM) and Keras with Neural Network.

Neural Network

A neural network is a collection of neurons with respective weights connecting them. Neurons process records one at a time or in batches and learn by comparing their classification with the actual classification for classification problems. The duty of the input layer is to receive the input signals from the outer system. Neuron Network also contains hidden layer between the input and output layer that contains neurons as well. Although the learning of the neural network can either be supervised or unsupervised, the fully supervised for the input provides to the neural network has an answer or output.

Long Short Term Memory (LSTM)

LSTM is a Recurrent Neural Network that makes use of sequential information by using their memory of what has been calculated so far and their sequence. A single LSTM uses both the feedforward and backpropagation to adjust the weights and is composed of 4 main components namely a forget gate, input gate, output gate, and a cell state. Below is their architecture:



Highlights:

- Yellow: Forgot gate
- Blue: Input gate
- Green: Cell State
- Grey: Output gate

Figure 3: LSTM architecture

- The forget layer is done by the first sigmoid function and is responsible for deciding which information is to be rejected or kept from the last step.
- The input layer decides which values will be updated in the cell state and whereby the tanh layer creates a vector of new candidate values.
- The cell state is the memory of the LSTM. This layer makes LSTM better than the vanilla RNN.
- The output layer gives the output coming from the cell state after being passed through a hyperbolic function tanh for filtering (Rahuljha, 2020).

As explained above, LSTM uses both feed-forward propagation and backpropagation applying the below equations to each layer:

Feed-Forward Propagation:

$$a^t = W^t.[h^{t-1}, x^t] + B$$

where

W is the weight of the layer, $[h^{t-1}, x^t]$ is the output from the previous layer and the layer's function and B is the bias

Then,

$f = sigmoid\ (a)$ or $f = tanh\ (a)$

Back Propagation:

$z^t = W^t . z^{t-1} + b^l$

where

W is the weight of the layer, and B is the bias and t-1 is the result from the previous layer

Then,

$f = sigmoid\ (z)$ or $f = tanh\ (z)$

Text classification proved to classify better if the sequence of words is taken into consideration, hence the use of the LSTM model on our training dataset.

Keras with Neural Network

Keras is a deep learning Python library containing efficient various libraries Theano and TensorFlow. Keras offers an embedding layer that can be used for neural networks on text data, the author opted to use it on the neural network for better outputs.

3.3.1. Feature engineering

A profound feature engineering has been conducted to ensure the best results. Some of the techniques used on our dataset are mainly NLP-related techniques namely the text normalization and word vectorization. The text normalization consisted of putting all the answers and questions in our dataset on a level playing field to not cause the algorithm to learn unnecessary patterns. The text normalization included lowering all the letters of our text, removing both punctuations, accent marks, and other signs; removing numbers; removing empty spaces, expanding abbreviations, removing stop words, space terms, and particular words. Additionally, other normalization techniques such as Tokenisation and Stemming has been also added depending on the word vectorization used. In machine learning, to train text format to a machine, the text must be transformed into its understandable version. Therefore, the answers and questions have been transformed into vectors (array of numbers), also known as word embedding. The text vectorization refers to various techniques used to extract

information from a text corpus and associating each word to a vector. In our study, the word vectorization method used is N-grams using TF-IDF Vectorizer and Universal Sentence Encoder. An N-gram is a sequence of N words considered while performing the TF-IDF vectorization. An N-gram model predicts the occurrence of a word based on the occurrence of its N – 1 previous word. TF-IDF in NLP stands for Term Frequency – Inverse document frequency. TF-IDF Vectorizer is among the very popular topics in NLP for generally dealing with human languages. Tfidf features are the method to convert the textual information into the vector space, they are a measure of how important a word in a text is within a document. The term frequency (TF) calculates the ratio of the frequency of a term in a document over the total number of terms in all the documents (Moradi, 2020).

$$Term\ Frequency(TF) = \frac{Frequency\ of\ the\ term\ in\ the\ document}{total\ number\ of\ terms\ in\ documents}$$

$$Inverse\ Document\ Frequency(IDF) = \log(\frac{total\ number\ of\ documents}{number\ of\ documents\ with\ term\ t})$$

$$TF.IDF = TF * IDF$$

The above mentioned TF-IDF vectorizer has been applied to one of the used machine learning techniques, Ridge regression that has been elaborated later.

Universal Sentence Embedding is a word vectorization that encode text into high dimensional vectors that can be used for text classification, semantic similarity, clustering, and other natural language tasks. It is trained and optimized for greater-than-word length text, such as sentences, phrases or short paragraphs. It is trained on a variety of data sources and a variety of tasks to dynamically accommodate a wide variety of natural language understanding tasks. Universal Sentence Embedding comes with two variations, one trained with a Transformer encoder and other trained with Deep Averaging Network (DAN) to enable building functional transfer learning models. The pre-trained model gives the model the transfer learning advantage and thus enables more powerful learning over new data. Universal Sentence Embedding has been used on both of our neural network (LSTM and Neural Network) models explained above (Mahapatra, 2019). Below is the architecture of DAN

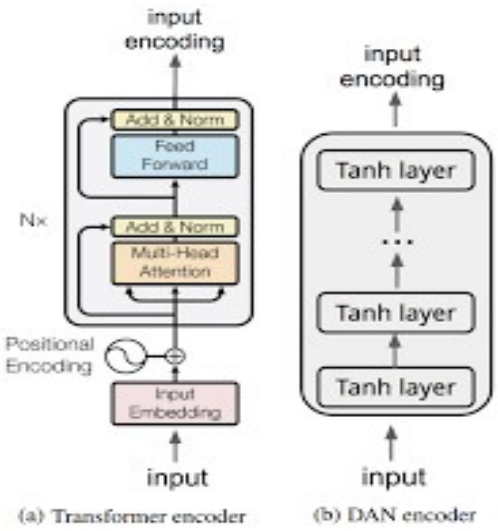


Figure 5:Universal Sentence Embedding1 (Keras + Universal Sentence Encoder = Transfer Learning for text data, 2018)

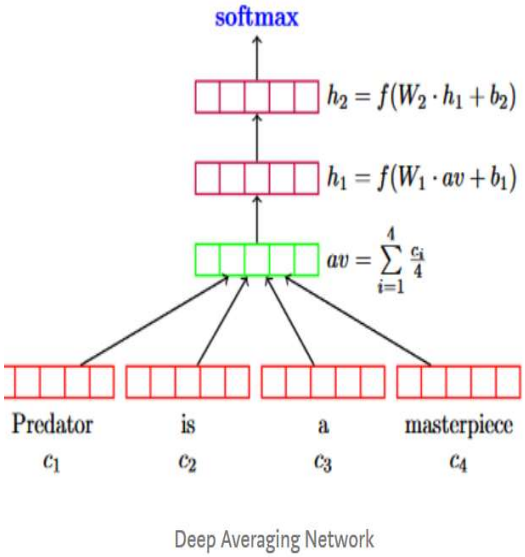


Figure 4: Universal Sentence Embedding2 (Shervin Minaee,Nal Kalchbrenner,Erik Cambria,Narjes Nikzad Khasmakhi, 2020)

Evaluation metrics

One of the metrics used while calculating the accuracy between predicted values and the real values on our models is spearman rank correlation. This is because it works well on textual data when looking for similarities. Spearman correlation has been used on the ridge regression and LSTM model. Below is the Spearman rank correlation equation:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

where d=difference between the ranks squared and *n*=number of observations

AUC roc score has been used on ridge regression too, to compute the accuracy of the classification only (1 to value above 0.5 and 0 to values less than 0.5)

The accuracy metric provided by Keras on deep learning models has been used on the Keras with Neural Network model. Below is how the accuracy in Keras is computed:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

4. RESULTS

4.1. Pre-analysis

The dataset is composed of 30 questions whereby 21 features are question-related labels and 9 answer related labels. Below is the distribution of each label:

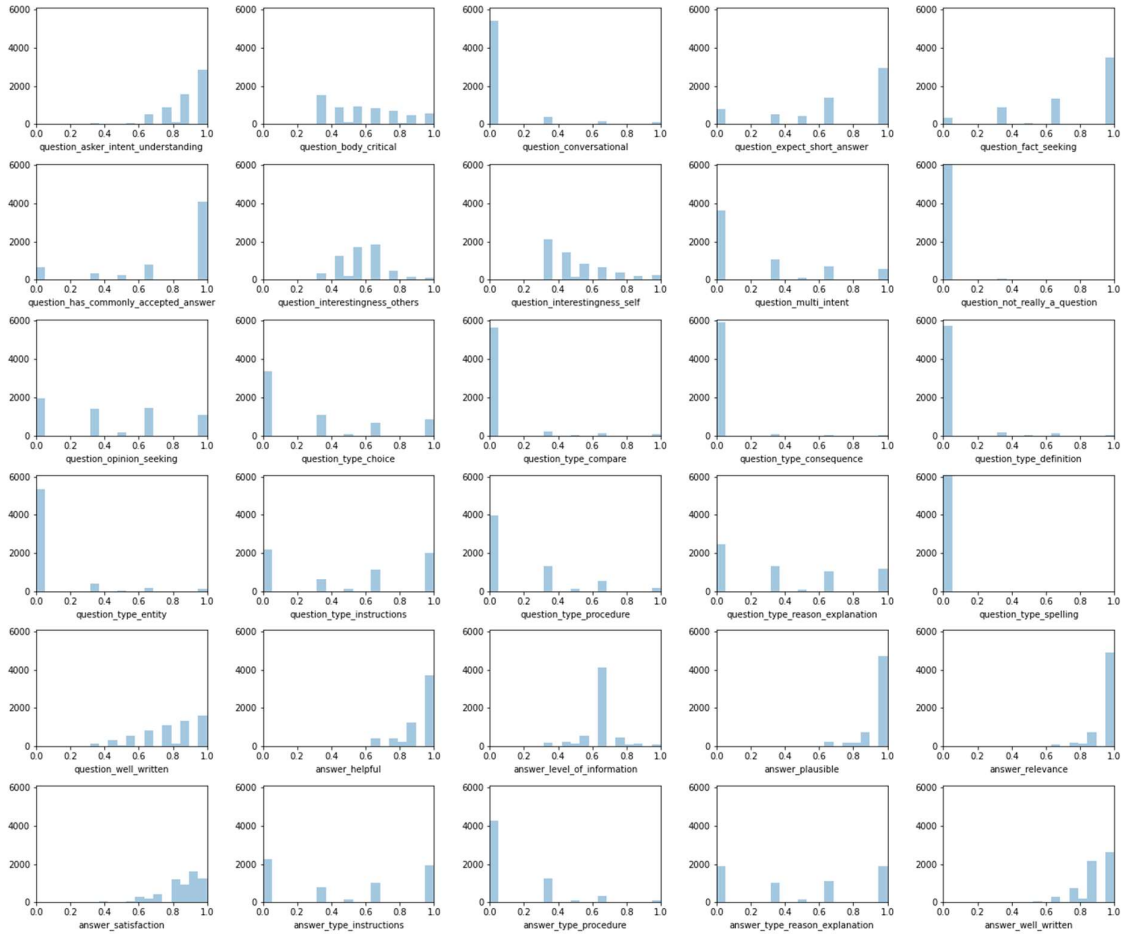


Figure 6:Labelled features distribution

The above figure shows that the data is quite imbalanced and some of them are almost zero.

However, the dataset has no null values for the features which provide more training data to the algorithm.

Additionally, our dataset had various feature columns to be used for training our model. Each feature column has been checked to look for the ones worth using while training the models.

The feature columns analysed are 'question_title', 'question_body', 'question_user_name', 'question_user_page', 'answer', 'answer_user_name', 'answer_user_page', 'url', 'category', 'host'.

Both the train and test datasets have almost the same distributions of categories which are either “Technology”, “Stackoverflow”, “culture, science” and “life_arts". This shows that our dataset is not trained for a closed domain, but an open domain question answering system although the available training dataset is very small for an open question answering system. The categories over the questions and answers were not bringing much to our learning process, thus removed from the features to consider.

The user checks showed that the same answer and question users tend to respond to the same kind of questions. Also, only unique question and answer users in both the training and the testing set were very few. This means that the users' questions and answers selected in both the training and testing have almost the same style, which predicted the testing dataset more accurately. However, the user information features have not been used while training our model as it was considered as not instructing much on our text as well as on the future inputs of question answering systems.

Word cloud has been used on both the training and testing dataset to assess the words than were mostly occurring. The word clouds showed that the train and test datasets have almost similar distribution on the question title, question body and answer body.

The features selected for training the models have been 'question_title', 'question_body' and ‘answer' only, as they are the minimal and more instructing features that a question answering system can learn from.

Besides, the feature engineering enabled the author to look at the features in a more detailed way and helped in the conclusion of the first objective. The data exploration showed that the "question_type_instructions" & "answer_type_instructions’, "question_type_procedure" & "answer_type_procedure" and "question_type_reason_explanation" & "answer_type_reason_explanation" are highly correlated. On the other hand, "question_fact_seeking" & "question_opinion_seeking", "answer_type_instruction" & "answer_type_reason_explanation" are anti-correlated. The exploration of the features of the dataset, mainly the labeled features shows that answers obtained for a specific question in terms of the instructions, procedure, and reason explanation were the results of the same from the question's structures. The anti-correlation also showed that the users who asked questions seeking for facts do not seek any opinion, instead those questions with a high fact seeking look for answers with high procedure type. It also shows that questions and answers with high instructions type are the total opposite of questions and answers with high reason explanations.

Below is the correlation matrix of the labeled features

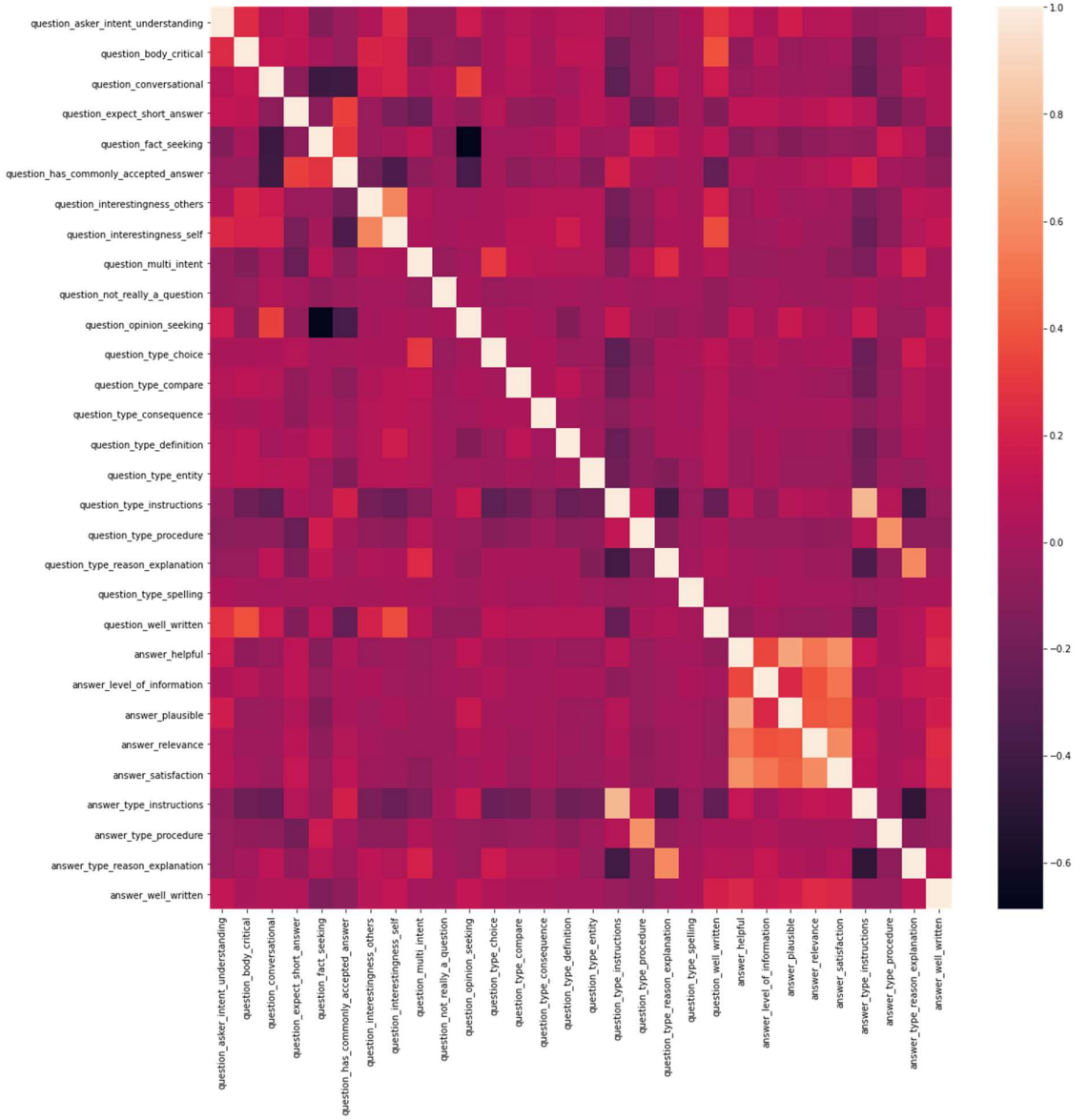


Figure 7:Labelled features correlation

4.2. Creating a database of labeled questions and answers for improved question answering systems. (OBJ 1)

Data pre-processing of our training dataset showed that, although the training dataset contains 6079 rows of questions, some questions were repeated. The training dataset is composed of only 3583 unique questions with 6079 unique answers. This means that questions in the dataset are repeated but no answer is repeated, therefore there is more than one answer to one question.

This might mean that also all the answers in the training dataset are not correct. However, the assessment of the questions URL, the provided dataset, and the labeled data revealed something else. The questions URL information showed that the chosen answers are not necessarily the best-ranked answers of the question, the question rank towards a question is not recorded in our training dataset too. The labeled features are the ones decided without considering their ranks on the questions URL. Therefore, the answers might not be correct but all of them have been considered because each answer has been classified correctly towards the labeled features.

4.2. To apply a Ridge Logistic Regression algorithm to classify the questions and answers (OBJ 2)

Ridge regression modeling has been applied to our pre-processed data with data normalization and the word vectorization with TFIDF Vectorizer. Data Normalization such as removing stop words, punctuations, and numbers have been done within the TFIDF Vectorizer.

Cross-validation has been applied to the model to learn and measure the accuracy of our training dataset. K-folds has been used, with k=10. K=10 proved to give the best accuracy results on the model.

With a simple regression model like Ridge regression on our model, one had to run two different models. One from the question vectorized feature to train to the labeled question-related question and the answer vectorized feature to train the labeled answer related to the answers.

A Spearman rank correlation and ROC has been used to look at the performance of Ridge regression. The average spearman rank correlation to all feature labels is 0.37 and the AUC score is 0.77. Below is the representation of both the Spearman and the roc_auc_score.

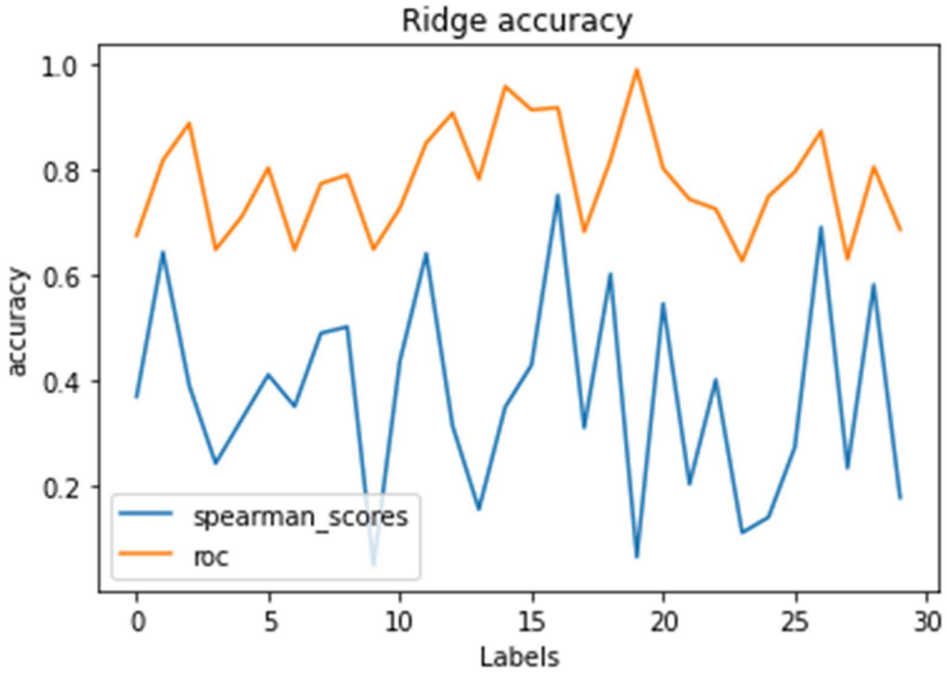


Figure 8:Ridge regression- Spearman and ROC scores

The above graph shows the spearman and the auc values of each label. It shows that most of the predictions on the classification (having 1 for those with more than 0.5 and 0 for those with values less than 0.5). This means that the labels were well classified but finding the right value computed with spearman correlation for each label was still low.

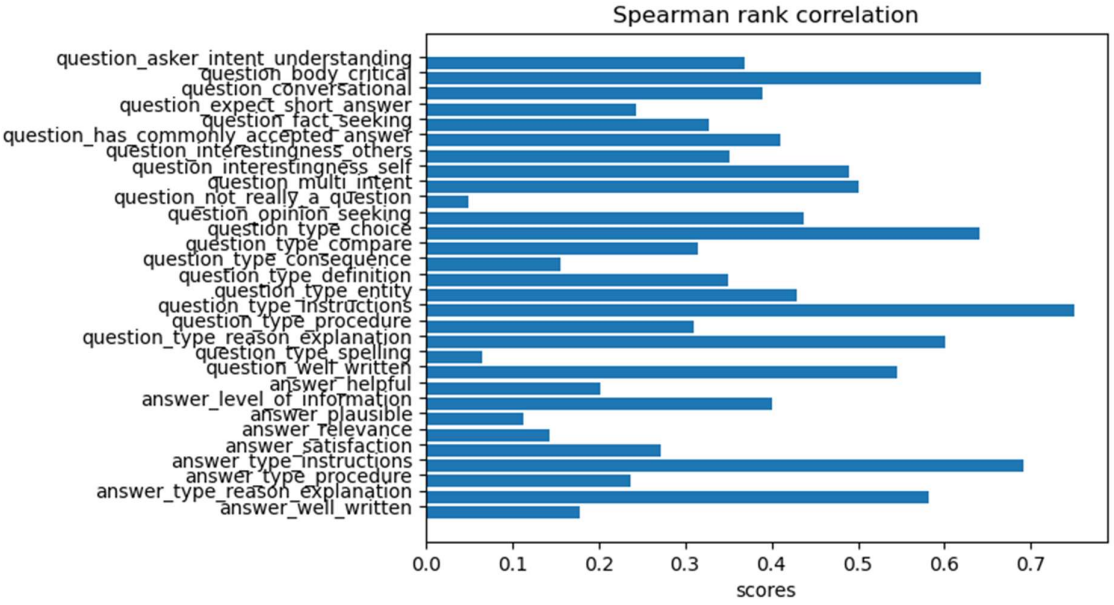


Figure 9: Ridge regression- Labels vs spearman scores

Looking closer to Spearman rank correlation for each label, labels with lower scores have imbalanced distributions in our training dataset, and labels with high accuracy like question type instructions have different values.

4.3. To apply Neural Network algorithms to classify the questions and answers (OBJ 3)

Long Short Term Memory (LSTM)

LSTM has been applied after data normalization and word vectorization with Universal Sentence Encoder. Questions and answers were passed in the embedding function, whereby question titles and body were put together. All the necessary data normalization processes were performed under the universal Sentence encoder. An LSTM model with a pytorch library was then defined with the below details.

Table 3: LSTM model summary

Layer type	**Description**
Input_1 (Input Layer)	For the question title and body inputs
Input_2(Input Layer)	For the question body inputs
Lambda_1	Operating function on question title data to vectors
Lambda_2	Operating function on question body data to vectors
Dense_1	Activation function with ReLu
Output Dense	With sigmoid activation function

The accuracy was calculated on Spearman rank correlation and performed with an accuracy score of 0.38. However, some of the folds showed that the accuracy could go higher, as shown in the below figure.

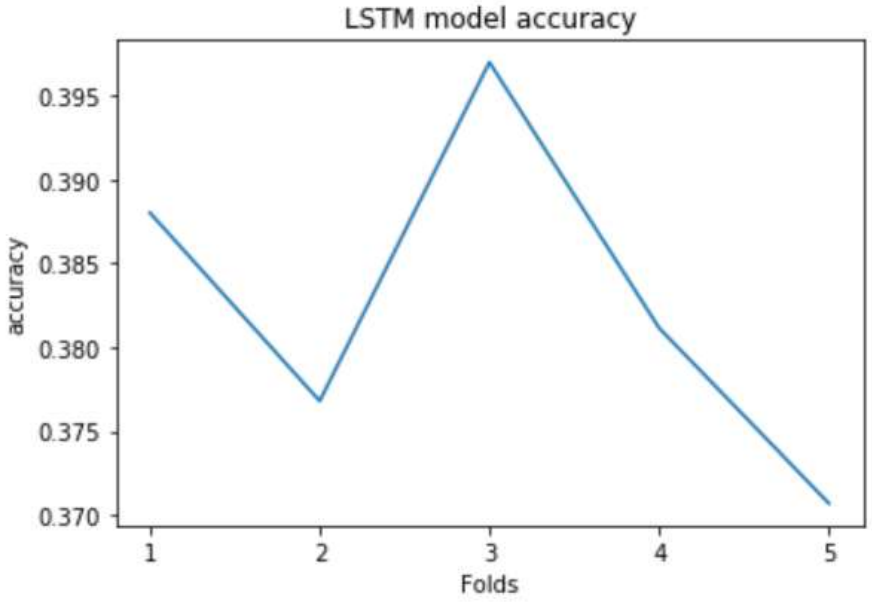


Figure 10: LSTM model accuracy per k-folds

Therefore, the LSTM model results were then optimized by using scikit-learn transformers called TransformerMixin. The results from our model have been passed through the transformer and reached an accuracy of 0.40.

Keras with Neural Network

Deep learning with Keras has been implemented with the word vectorization with Universal Sentence Encoder. Universal Sentence Encoder being a pre-trained, enables one to apply it and give better results. A Keras model was defined for applying the Universal Sentence Encoder for a transfer learning. Below are the model details.

Table 4: Keras with NN model summary

Layer type	**Description**	**Output shape**	**Parameter #**
Input_1 (Input Layer)	For the question title inputs	(None,1)	0
Input_2(Input Layer)	For the question body inputs	(None,1)	0
Input _3 (Input Layer)	For the answer inputs	(None,1)	0
Lambda_1	Operating function on question title data to vectors	(None,512)	0
Lambda_2	Operating function on question body data to vectors	(None,512)	0
Lambda_3	Operating function on answer data to vectors	(None,512)	0

Concatenate_1	Concatenating data from Lambda_1, Lambda_2, Lambda_3	(None,1536)	0
Dense_1	Activation function with Sigmoid	(None,256)	393472
Dropout_1	Dropout regularization layer of 0.4	(None,256)	0
Batch normalization_1	Standardization of the inputs	(None,256)	1024
Dense_2	Activation function with lr=0.001	(None,64)	16448
Dropout_2	Dropout regularization layer of 0.4	(None,64)	0
Batch normalization_2	Standardization of the inputs	(None,64)	256
Output Dense	With sigmoid activation function	(None,30)	1950

Reduce Learning Rate On Plateau, which is Keras callback has been also used on our training model to reduce the learning rate once the learning stagnates.

The accuracy on deep learning is 0.57. Below is the graph of accuracies on training and validation sets.

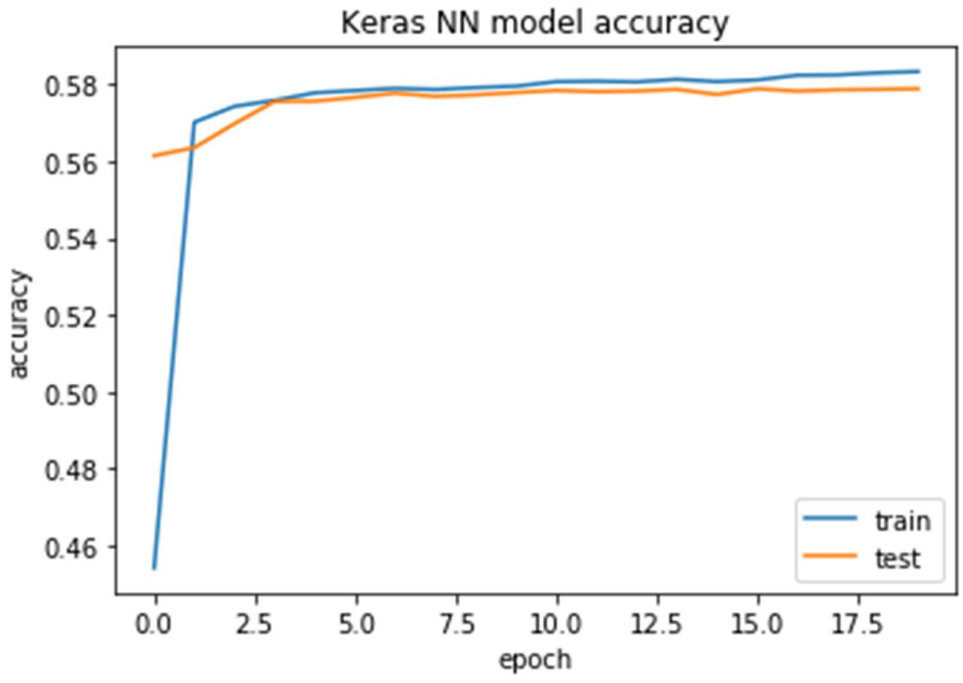


Figure 11: Keras accuracy- train & validation

The training set accuracy increased as the epochs increased and have been stabilized at a certain epoch by Reduce Learning Rate On Plateau. The testing accuracy is also good and also kept on increasing as epochs increased too.

4.4. To compare the Ridge Logistic regression algorithm with the Neural Network algorithms (OBJ 4)

Based on the above observations, the results showed that Neural Network algorithms were performing better on our text data than the Ridge Logistic Regression. However, comparing the ridge regression performance with the LSTM model showed a rise in accuracy of only 0.02.

The use of the regression model on text data that was to be performed on multiple inputs required to be done twice and differently. However, the use of Neural networks allowed to pull all the inputs at once, they were defined on different layers.

Although both LSTM and Keras with Neural network both used the Universal Sentence Encoder, the optimized performance of the LSTM model was 0.40 but the Keras with Neural Network reached up to 0.58. This is a considerable difference between the two models. Therefore, Keras deep learning libraries' models perform well on transfer learning better than the used PyTorch libraries on LSTM. The LSTM model was also expensive in terms of computing (steps to get to the final result) compared to the Keras with Neural Network.

5. CONCLUSION AND RECOMMENDATION

Conclusion

All the used models have been able to give the output, which was supposed to classify every question on 21 different labels and every answer on 9 different labels.

The highest accuracy was the one given by Keras with Neural Network model which used Universal Sentence Encoder and Reduce Learning Rate On Plateau that Keras offers.

Universal Sentence Encoder, the used word vectorizer managed to normalize the data more than all the long steps went through with ridge regression in its pre-processing stage.

Though the highest, but the accuracy of the Keras with Neural Network is still low on the cross-validation. However, its ranking compared to the leaderboard of Kaggle training is only making a difference of 0.10 on the testing data accuracy score. It has been also observed that the distribution of data of each label contributed to their classifications.

This work also has been limited by the lack of more features to train on such as ranking the best results of a question for a better answer extraction while ranking them.

Recommendations

The above study had many questions and answer labels which would mislead the machine to learn from all the labels as some questions are not easy to rate by a human being in a general way, but would be subjective to some to give a certain rank. Also, some features are repeated in other features but in another way which would be more beneficial while classifying the questions and answers for better performance namely "question_not_really_a_question" by "question_opinion_seeking" or "question_type_choice". Some of the features data were also unbalanced in terms of distribution. For future research, I would recommend conducting a labeled feature selection (the output) to only focus on the features that would improve the performance of question answering systems which would also increase the accuracy of models.

Keras with neural network and Universal Sentence Encoder are powerful libraries for such work but they still need more improvements. One of the most used and shown by the best leaderboard on this Kaggle competition was Bert models to work better in Natural Language

Processing on both the pre-processing stage and also on learning. BERT (Bidirectional Encoder Representations from Transformers) models were launched in 2018 by Google AI language. The best algorithms built under the competition were implemented with BERT. I would recommend to use it for better results on text categorization tasks.

The study's feature columns selected are only text data, namely the question title, question body, and answer to be able to classify a question or an answer. As observed, many questions and answer users are not unique and reply to the same kinds of answers. For future research, I recommend considering user additional users' information on while learning the model for better performances.

REFERENCES

A. Chandra Obula Reddy, D. K. (2017). A Survey on Types of Question Answering System. *IOSR Journal of Computer Engineering (IOSR-JCE)* , 19-23.

Ahmad Abdellatif, K. B. (2020). Challenges in Chatbot Development: A Study of Stack Overflow Posts. *Research Gate*.

Anjali Saini, P. Y. (2017). A Survey on Question –Answering System. *International Journal of Engineering and Computer Science*, 20453-20457.

Bolanle Ojokoh, E. A. (2019). A Review of Question Answering Systems. *Journal of Web Engineering*, 717–758.

Cuelogic Insights. (2016, June 23). *Role of Python in Artificial Intelligence (AI)*. Retrieved from Cuelogic Insights: https://www.cuelogic.com/blog/role-of-python-in-artificial-intelligence

Das, R. K. (2018). *Build Better Chatbots.* Bangalore, India: Apress.

David Ferrucci, E. B.-C. (2010). Building Watson:An Overview of the DeepQA Project. *AI MAGAZINE*, 59-79.

Deepa Yogish, P. M. (2016). A Survey of Intelligent Question Answering. *International Journal of Advanced Research in Computer and Communication Engineering*, 536-540.

DeepAI. (n.d.). *Named-entity recognition*. Retrieved from DeepAI: https://deepai.org/machine-learning-glossary-and-terms/named-entity-recognition

Endicott, M. L. (2016, 08 23). *What are the differences between a question answering system and a chatbot?* Retrieved from https://meta-guide.com/: https://meta-guide.com/quora/what-are-the-differences-between-a-question-answering-system-and-a-chatbot

Feuerriegel, B. K. (2018). Adaptive Document Retrieval for Deep Question Answering. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 576–581.

G. Suresh Kumar, G. Z. (2015). Concept relation extraction using Naïve Bayes classifier for ontology-based question answering systems. *Journal of King Saud University - Computer and Information Sciences*, 13-24.

Glen, S. (2018, July 31). *Primary Data & Secondary Data: Definition & Example*. Retrieved from Statistics How To: https://www.statisticshowto.com/primary-data-secondary/

Heba Kurdi, S. A. (2014). Development And Evaluation Of A Web Based Question Answering System For Arabic. *International Journal on Natural Language Computing (IJNLC)*, 57-67.

Kaggle. (2020, September). Retrieved from Kaggle: https://www.kaggle.com/c/google-quest-challenge

Keras + Universal Sentence Encoder = Transfer Learning for text data. (2018). Retrieved from Dlology: https://www.dlology.com/blog/keras-meets-universal-sentence-encoder-transfer-learning-for-text-data/

Li Deng, Y. L. (2018). *Deep Learning in Natural Language Processing.* Springer Publishing Company.

Mahapatra, S. (2019, January 14). *Use-cases of Google's Universal Sentence Encoder in Production*. Retrieved from towards data science: https://towardsdatascience.com/use-cases-of-googles-universal-sentence-encoder-in-production-dd5aaab4fc15

Manvi Breja, S. K. (2018). Why-type Question classification in Question Answering System. *CEUR Workshop Proceedings*.

Mark A. Greenwood, H. S. (n.d.). *A Pattern Based Approach to Answering Factoid, List and Definition Questions.* Retrieved from staffwww: http://staffwww.dcs.shef.ac.uk/people/M.Greenwood/nlp/pubs/riao04.pdf

Moradi, A. (2020, December 3). *Feature scoring metrics in word-document matrix*. Retrieved from towards data science: https://towardsdatascience.com/feature-scoring-metrics-in-word-document-matrix-eb35b38c029e

Pai, A. (2020, May 26). *What is Tokenization in NLP? Here's All You Need To Know.* Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2020/05/what-is-tokenization-nlp/

Pooja, V. K. (2012). An Efficient Passage Ranking Technique for A QA System. *International Journal of Computer Science & Information Technology (IJCSIT)*, 65-76.

R.Mervin. (2013). An Overview of Question Answering System. *International Journal Of Research In Advance Technology In Engineering (IJRATE)*, 11.

Rahuljha. (2020, June 29). *LSTM Gradients.* Retrieved from towards data science: https://towardsdatascience.com/lstm-gradients-b3996e6a0296

Rosy Madaa, A. D. (2012). A comparative study of web based and IR based question answering systems. *An International online open access peer reviewed journal*, 188-194.

Shervin Minaee,Nal Kalchbrenner,Erik Cambria,Narjes Nikzad Khasmakhi. (2020). Deep Learning Based Text Classification: A Comprehensive Review. *Research Gate*.

Sujeevan Aseervatham,Anestis Antoniadis, Eric Gaussier,Michel Burlet,Yves Denneulin. (2012). A Sparse Version of the Ridge Logistic Regression for Large-Scale Text Categorization. *HAL archives ouvertes*, 2.

Tampa Bay Times. (2014, August 26). *IBM computer Watson wins Jeopardy challenge, surprising no one*. Retrieved from Campa Bay Times:

https://www.tampabay.com/archive/2011/02/17/ibm-computer-watson-wins-jeopardy-challenge-surprising-no-one/

Taylor, J. (2019, June 20). *What Are Question-Answering Systems?* Retrieved from Lucidworks: https://lucidworks.com/post/what-are-question-answering-systems/

What is Machine Learning? A Definition. (2020, May 6). Retrieved from expert.ai: https://www.expert.ai/blog/machine-learning-definition/

Wikipedia. (2020, May 13). *ELIZA*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/ELIZA

Wikipedia. (2020, May 31). *Question-answering*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Question_answering

Xuequan, M. (2019, 04 16). *http://www.xinhuanet.com/.* Retrieved from Xinhua: http://www.xinhuanet.com/english/2019-04/16/c_137979616.htm

APPENDIX: PLAGIARISM TEST

ORIGINALITY REPORT

16%	12%	6%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.theseus.fi Internet Source	1%
2	www.riverpublishers.com Internet Source	1%
3	www.safaribooksonline.com Internet Source	1%
4	en.wikipedia.org Internet Source	1%
5	Submitted to essex Student Paper	1%
6	Submitted to University of Western Sydney Student Paper	1%
7	P Lakshmi Prasanna, Dr D.Rajeswara Rao. "Text classification using artificial neural networks", International Journal of Engineering & Technology, 2017 Publication	1%
8	www.tensorflow.org Internet Source	1%

9	medium.com Internet Source	1%
10	rss.sciencedirect.com Internet Source	1%
11	www.analyticsinsight.net Internet Source	1%
12	www.kaggle.com Internet Source	<1%
13	Submitted to Lebanese American University Student Paper	<1%
14	Amit Mishra, Sanjay Kumar Jain. "A survey on question answering systems with classification", Journal of King Saud University - Computer and Information Sciences, 2016 Publication	<1%
15	Submitted to Acacia Learning Student Paper	<1%
16	Submitted to Higher Education Commission Pakistan Student Paper	<1%
17	aitopics.org Internet Source	<1%
18	Submitted to University of Missouri, Kansas City Student Paper	<1%

19	Submitted to Laureate Higher Education Group Student Paper	<1%
20	Submitted to University of Sydney Student Paper	<1%
21	www.cuelogic.com Internet Source	<1%
22	Submitted to University of Westminster Student Paper	<1%
23	Submitted to BAC International Study Centre Student Paper	<1%
24	Youzheng Wu, Chiori Hori, Hideki Kashioka, Hisashi Kawai. "Leveraging social Q&A collections for improving complex question answering", Computer Speech & Language, 2015 Publication	<1%
25	Zewdie Mossie, Jenq-Haur Wang. "Vulnerable community identification using hate speech detection on social media", Information Processing & Management, 2020 Publication	<1%
26	Rohini P. Kamdi, Avinash J. Agrawal. "Keywords based Closed Domain Question Answering System for Indian Penal Code Sections and Indian Amendment Laws",	<1%

	International Journal of Intelligent Systems and Applications, 2015 Publication	
27	towardsdatascience.com Internet Source	<1%
28	Sumathi S., Indumathi S., Rajkumar S.. "chapter 9 Medical Reports Analysis Using Natural Language Processing for Disease Classification", IGI Global, 2020 Publication	<1%
29	computer-trading.com Internet Source	<1%
30	Weishan Zhang, Wuwu Guo, Xin Liu, Yan Liu, Jiehan Zhou, Bo Li, Qinghua Lu, Su Yang. "LSTM-based Analysis of Industrial IoT Equipment", IEEE Access, 2018 Publication	<1%
31	"Natural Language Processing of Semitic Languages", Springer Science and Business Media LLC, 2014 Publication	<1%
32	"Advanced Data Mining and Applications", Springer Science and Business Media LLC, 2017 Publication	<1%
33	monkeylearn.com Internet Source	

		<1%
34	Issa Annamoradnejad, Mohammadamin Fazli, Jafar Habibi. "Predicting Subjective Features from Questions on QA Websites using BERT", 2020 6th International Conference on Web Research (ICWR), 2020 Publication	<1%
35	Submitted to King Abdulaziz University Student Paper	<1%
36	dspace.jaist.ac.jp Internet Source	<1%
37	deepai.org Internet Source	<1%
38	Submitted to Queen Mary and Westfield College Student Paper	<1%
39	R Menaha, A Udhaya Surya, K Nandhni, M Ishwarya. "Question answering system using web snippets", 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2017 Publication	<1%
40	"Pattern Recognition", Springer Science and Business Media LLC, 2020 Publication	<1%

41	towardsmachinelearning.org Internet Source	<1%
42	"10th International Conference on Robotics, Vision, Signal Processing and Power Applications", Springer Science and Business Media LLC, 2019 Publication	<1%
43	Dipanjan Sarkar, Raghav Bali, Tushar Sharma. "Practical Machine Learning with Python", Springer Science and Business Media LLC, 2018 Publication	<1%
44	Mario Linares-Vasquez, Bogdan Dit, Denys Poshyvanyk. "An exploratory analysis of mobile development issues using stack overflow", 2013 10th Working Conference on Mining Software Repositories (MSR), 2013 Publication	<1%
45	Serge Linckels, Christoph Meinel. "E-Librarian Service", Springer Science and Business Media LLC, 2011 Publication	<1%