Question Classification and Answering System Android VS IOS Dataset

Sarah Hussien 49-16011
Hams Wael 49-4485

April 20, 2024

1 Introduction and Motivation

Natural language processing (NLP) question answering and classification systems are at the forefront of revolutionizing how machines comprehend and react to human language. These systems are made to classify and analyze user queries so that computers can respond with relevant and precise answers. Question answering and classification systems are vital for improving the efficacy and efficiency of information retrieval processes in a variety of applications, including virtual assistants, search engines, and customer support platforms. They achieve this by utilizing sophisticated algorithms and linguistic analysis.

The growing need for intelligent and user-friendly technology that can efficiently interpret and respond to human language is the driving force behind the advancement of question classification and answering systems in NLP. Systems that can quickly and accurately comprehend user queries and deliver insightful answers are becoming more and more necessary in the current digital era, where information overload is an increasingly common issue. We can speed up information retrieval procedures, improve user experiences, and enable more effective human-machine communication by increasing the precision and speed of question classification and answering systems.

Understanding how question classification and answering work as an NLP task is of importance for several reasons. It makes it possible for developers to create more reliable and accurate models by understanding the underlying techniques and algorithms. They can improve the performance and dependability of question classification and answering systems by making well-informed decisions about model architectures, feature representations, and training methodologies.

2 Literature Review

This section will be tackling recent works related to the NLP task "Question Classification and Answering" This section is structured to address the two main steps of this task: Question Classification and Question Answering.

2.1 Question Classification

Question classification is an important task in Natural Language Processing (NLP) that involves categorizing questions into distinct classes or categories based on their semantic meaning and intended answer type. The field of question classification has advanced significantly with the introduction of machine learning and deep learning techniques. On datasets of labeled questions, supervised learning algorithms like random forests and Support Vector Machines (SVM) have been used to train models. These models categorize unseen questions into predefined categories by using patterns and features they have learned from the training data. According to [AD23], contextual and semantic information about questions has been recently captured by deep learning models like neural networks and

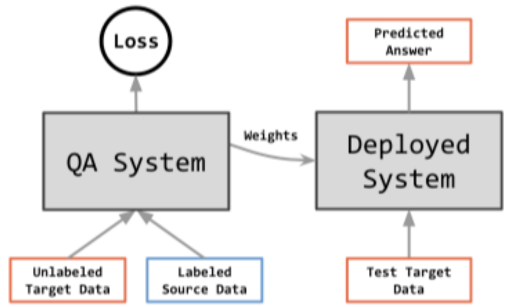

Figure 1: Question Answering Model [YZK+22].

transformer-based models like BERT, allowing for more precise classification.

Question Classification has two main approaches: Binary Classification and Multi-class Classification. Classifying questions into two different classes or categories is known as binary question classification. The main goal is to identify which of the two categories a given question belongs in. While, classifying questions into more than two different classes or categories is known as multi-class classification. Based on the question's domain, intended answer type, or semantic meaning, each one is assigned to a particular class. Multi-class classification allows the system to respond to different kinds of questions with greater accuracy and specificity depending on the category that it has identified.

Question classification can be done in two ways: automatically and manually [RSJ10]. Machine learning or deep learning models trained on labeled question datasets are used to classify questions automatically. These models categorize new, unseen questions into relevant categories by using the patterns and features found in the training data. On the other hand, manual question classification classifies questions by human intervention.

2.2 Question Answering

In order to seamlessly bridge the gap between classifying questions and delivering precise answers, question answering systems are essential. After classifying questions, these systems use a range of methods and algorithms to extract and combine pertinent data to produce thorough responses. According to [YZK+22], a model is trained with labeled source data and unlabeled target data. The resulting system is deployed to answer target questions.As shown in Fig.1: Question Answering Model, [YZK+22] provides a simple illustration of a Question Answering model.

Question Answering (QA) systems have become extremely efficient instruments for automatically providing natural language answers to queries posed by humans. These systems can use a pre-structured database or a wide range of natural language documents. Consider QA systems an advanced form of information retrieval. The demand for this kind of system increases on a daily basis since it delivers short, precise and question-specific answers [SP20]. According to [AH12], QA is made up of three separate modules, each of which comprises a core component in addition to other supporting elements. Question classification, information retrieval, and answer extraction are these three essential elements. By categorizing submitted questions based on their type, question classification plays a crucial part in QA systems. Information retrieval is crucial to answering questions because if no correct answers are present in a document, no further processing could be carried out to find an answer. Ultimately, the goal of answer extraction is to obtain the response to a query posed by the user [AH12].

According to [SP20], prior research primarily characterized the architecture of question-answering systems into three macro-modules, as depicted in Fig.2: question processing, document processing, and answer processing.

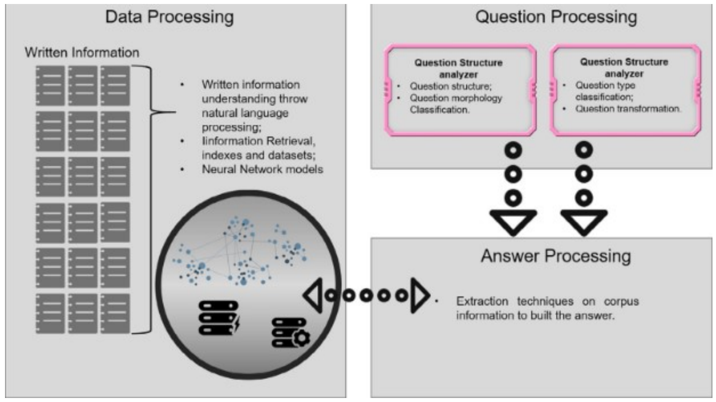

Figure 2: Architecture of the three macro question answering modules [SP20].

Question processing module classifies the question by its type and morphology. Answer processing module uses the classification and transformation made in question processing module to extract the answer from the result of the Document Processing module that executes previously to create datasets, indexes or neural models [SP20].

The user submits a query in natural language for question processing to analyze and categorize. The purpose of the analysis is to determine the question's type, or its main focus. To prevent confusion in the response, this is essential [MSB13].

There are two primary steps involved in processing questions. Analyzing the question's structure is the first step. The second step is to transform the question into a meaningful formula that fits within the domain of QA [HAA16]. The expected kind of response can also serve as a definition for a question. Factoids, lists, definitions, and difficult questions are among the categories [KM11].

Document processing is different from question processing in that it selects a set of relevant documents and extracts a set of paragraphs based on the question's focus or text understanding through natural language processing [MSB13]. Question processing executes on every question posed by the user. The source for the answer extraction can come from this task, which can produce a neural model or a dataset. Ranking the retrieved data based on how relevant it is to the query is possible [NL15].

The most challenging aspect of a question-answering system is the answer processing. This module presents a solution based on extraction techniques applied to the output of the Document Processing module. The response must be straightforward and address the topic; nonetheless, it may necessitate summarizing, combining data from several sources, or handling ambiguity or contradiction [SP20].

2.3 Pretrained Models used in Question Answering and Classification Task

In question answering and classification tasks, several natural language processing (NLP) approaches can be employed to effectively process and understand text data.

According to [ZLC+23], ALBERT, a more efficient and compact variant of the original BERT model, is used to build a QA system. It is based on the Transformer architecture. Its improvement on the BERT model mainly includes the following three aspects: embedded layer decomposition, cross-layer parameter sharing, and SOP sentence order prediction task.

Another model used in QA tasks is BamnetTL - Bidirectional Attention Memory Network with Transfer Learning. According to [SGW+23], BamnetTL is used in is Q and A matching, wherein the appropriate answer is chosen from candidate responses. It incorporates deep feature transfer based on a bidirectional attention memory network. //

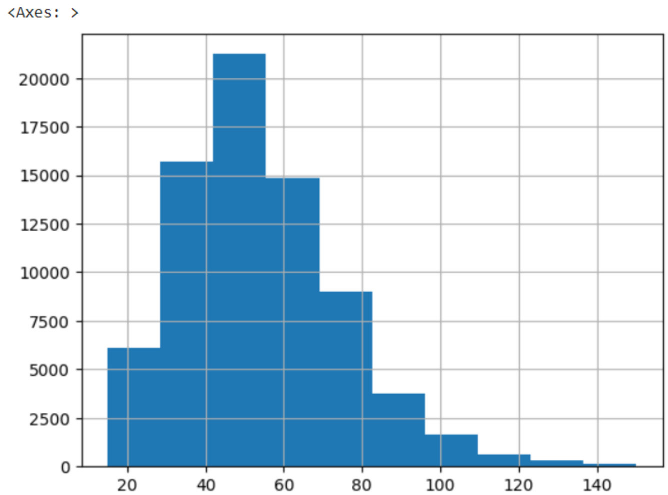

Figure 3: Title Distribution of question lengths.

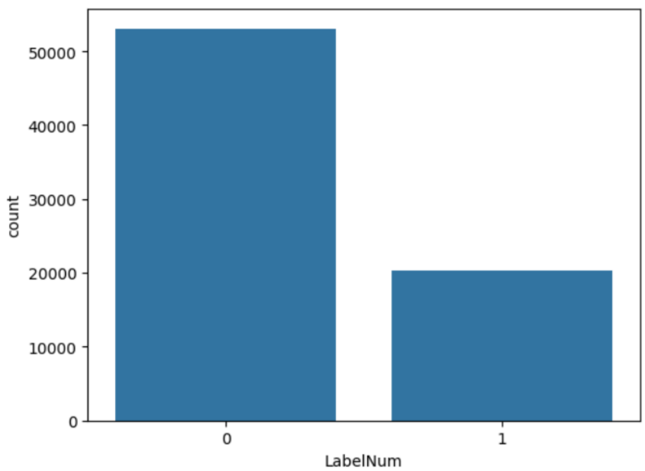

Figure 4: Count of Android (0) vs iOS (1) Questions .

For question classification, [AD23] presents an approach that combines the strengths of Electra (Transformer-based model, GloVe, and LSTM models). It also mentioned that BERT, RoBERTa, and DistilBERT are well-known models used in question classification task.

2.4 Data Analysis and Insights over the Dataset

After analyzing Android VS IOS Dataset, we came up with the following insights regarding the classification of questions either as Android or IOS.

We started by inspecting the distribution of question lengths in order to gain a better understanding of the dataset which will help us in determining the appropriate architecture to be applied on the dataset. For example, if the majority of questions are relatively short, a simpler model might be sufficient. In our dataset, the questions' length are approximately between 40 and 60, thus, we will be using a simple model.

Moreover, we inspected the distribution of class labels. We plotted a count plot that showed that the dataset is imbalanced as it's obvious that android class is exceeding the IOS by approximately 30,000 records.

Following that, we visualized the most frequent words in both classes by visualizing wordcloud by using

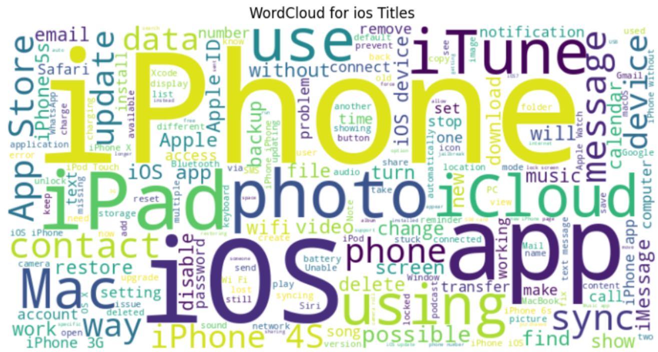

Figure 5: 'WordCloud for iOS Titles'.

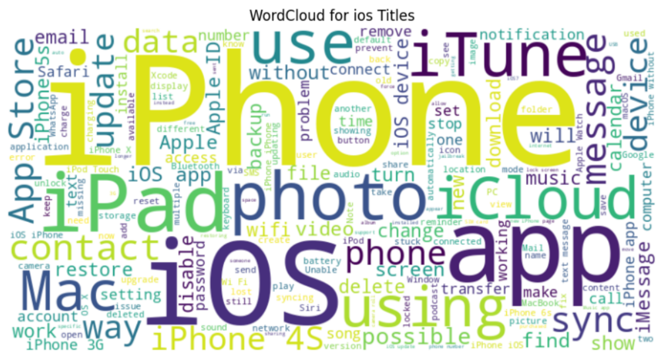

Figure 6: 'WordCloud for Android Titles'.

the attribute 'Titles'. For example, questions including the words 'iphone', 'ios', and 'ipad' are most likely to be classified as 'IOS or 1 (Label Number)'. While questions including the words 'android' and 'app' could be classified as 'Android or 0 (Label Number)'. However, we noticed that some questions in the dataset may exhibit overlap between classes, making classification challenging. For example, the word 'app' is considered a frequent word in both classes as visualized in the wordcloud. This insight can guide the inclusion of additional features to differentiate similar questions and avoid ambiguous classification.

3 Methodology

3.1 Problem Statement

A company needs to implement a system that automates the categorization of questions posted into Android or IOS in order to send these questions into the right support team. In addition, the company wants to implement a system that predicts the score of the question to determine how relevant and useful it is.
This project addresses the automation of categorizing questions and predicting their score. To be able to solve the mentioned problem, a neural network model is built. This model is trained on a dataset with the following columns: ID, Title of the post, Body, Score, ViewCount, Label, and LabelNum. Label attribute has the values Android and IOS. LabelNum attribute has the values 0 and 1 where 0 represents the Android class and 1 represents the IOS class.

3.2 Data Preparation

To prepare the dataset for modeling, we implemented some preprocessing steps. First of all, it's important to ensure that all columns of the dataset are relevant to the problem. We dropped the columns ID, ViewCount, and Label as they will not be used in solving any of the problems mentioned before.. In addition, we concatenated the columns Title and Body to ensure that the questions are informative. Lastly, we made sure that there are no null values in the dataset used.

Because we are performing a NLP task and we are using text columns, it's important to perform text preprocessing to these columns. We performed the following steps: 1)Tokenization where we broke down the question into words/tokens 2)Lemmatization where we reduce a word to its root form 3)Standardizing text to Lowercase 4)Removing punctuations 5)Removing stopwords

3.3 Neural Network Architecture

As we are addressing two different problems in this project, we built two neural network models.The first model is a Binary Classification model which will be used to categorize questions into Android or IOS. The second model is the Regression model which will be used to predict the score of the question. The first step in building a neural network model is to embed the words. We used Word2Vec to vectorize the words using the gensim library. After embedding the words, we can use them as the input to our architecture.

3.3.1 Binary Classification Model

A binary classificatin model is built without using any pretrained model by using tensorflow library. We first identified the input (X) and output (Y) of our model. The input of our model is the questions posted and the output is the LabelNum attribute in order to categorize the questions into 1 (IOS) or 0 (Android).The data is then splitted into training and testing sets.

After splitting the data, we constructed the model. The model is constructed using a Sequential model architecture from Keras. It begins with a dense layer comprising 64 neurons and Rectified Linear Unit (ReLU) activation function, taking input of dimension 100, which corresponds to the size of Word2Vec vectors. A dropout layer with a dropout rate of 0.2 is added to prevent overfitting. The final layer consists of a single neuron with a sigmoid activation function, suitable for binary classification tasks. The model is compiled using the Adam optimizer and binary cross-entropy loss function. It is then trained on the training data for 10 epochs with a batch size of 32 and a validation split of 0.2 to monitor model performance during training.

3.3.2 Regression Model

A regression model is built without using any pretrained model by using tensorflow library. We first identified the input (X) and output (Y) of our model. The input of our model is the questions posted and the output is the score attribute in order to predict the score/relevance of the questions.The data is then splitted into training and testing sets.

After splitting the data, we constructed the model. The model is constructed using the Sequential model from Keras, consisting of a dense layer with 64 neurons and Rectified Linear Unit (ReLU) activation function, which serves as the input layer. This layer accepts input with a dimension of 100, reflecting the size of the input features. A dropout layer with a dropout rate of 0.2 is added after the input layer to mitigate overfitting. Subsequently, a dense layer with a single neuron is added, representing the output layer for regression tasks. The model is compiled using the Adam optimizer and mean squared error loss function, suitable for regression problems. It is then trained on the training data for 10 epochs, with a batch size of 32, and a validation split of 0.2 to assess model performance during training.

Another approach used is for the regression model to take as input both questions and score instead of only the questions. The model is built using the Functional API of Keras. Two input layers are defined: one for a 1D array representing the "score" and another for a 2D array representing the "question." These inputs are concatenated into a single input layer using the Concatenate layer. A dense hidden layer with 64 neurons and Rectified Linear Unit (ReLU) activation function is added after the concatenation. The output layer consists of a single neuron, representing the regression output, with no activation function specified. The model is then defined using the Model class, specifying the inputs and outputs. After compiling the model with the Adam optimizer and mean squared error loss function, it is trained on the training data for 10 epochs with a batch size of 32 and a validation split of 0.2 to monitor model performance during training.

3.4 Evaluation of the Neural Network Models

The evaluation of a model is a crucial step in assessing its performance and effectiveness in addressing the problem at hand. During evaluation, the trained model is tested on unseen data to measure its ability to generalize and make accurate predictions. Various evaluation metrics are employed such as accuracy and mean squared error.

Binary classification model has a loss of 0.083 and accuracy of 0.97. While the regression model has a loss of 43.5971.As a result, the regression model with a higher loss suggests that the model's predictions are less accurate in estimating continuous numerical values, whereas the binary classification model with a low loss and high accuracy is doing well in differentiating between two classes. To accurately determine the effectiveness of the models, it is crucial to interpret these metrics within the context of the particular task and dataset.

3.5 Discussion

After constructing, training, and evaluating the models, we tried applying the binary classification model on questions and investigate whether it will be able to categorize the questions accurately.

The effectiveness of the binary classification neural network architecture in accurately classifying questions based on their content and characteristics is evident, as demonstrated by the model's performance metrics. We provided the model the following question "How does iOS differ from other mobile operating systems?", It's obvious that this question is categorized as iOS question. The binary classification model was actually able to predict this question as iOS. It provided an output of 0.99999243 which is too close to 1 indicating that it labels the question as iOS.

In addition, we provided the regression model the same question to predict its score. It provided an output of 1. According to the model's performance metrics, it's inaccurate due to a MSE of 43.5971.

4 References

References

[AD23] Sanad Aburass and Osama Dorgham. An ensemble approach to question classification: Integrating electra transformer, glove, and lstm. *arXiv preprint arXiv:2308.06828*, 2023.

[AH12] Ali Mohamed Nabil Allam and Mohamed Hassan Haggag. The question answering systems: A survey. *International Journal of Research and Reviews in Information Sciences (IJRRIS)*, 2(3), 2012.

[HAA16] Suhaib Kh Hamed and Mohd Juzaiddin Ab Aziz. A question answering system on holy quran translation based on question expansion technique and neural network classification. *J. Comput. Sci.*, 12(3):169–177, 2016.

[KM11] Oleksandr Kolomiyets and Marie-Francine Moens. A survey on question answering technology from an information retrieval perspective. *Information Sciences*, 181(24):5412–5434, 2011.

[MSB13] Nidhi Malik, Aditi Sharan, and Payal Biswas. Domain knowledge enriched framework for restricted domain question answering system. In *2013 IEEE International Conference on Computational Intelligence and Computing Research*, pages 1–7. IEEE, 2013.

[NL15] Mariana Neves and Ulf Leser. Question answering for biology. *Methods*, 74:36–46, 2015.

[RSJ10] Santosh Kumar Ray, Shailendra Singh, and Bhagwati P Joshi. A semantic approach for question classification using wordnet and wikipedia. *Pattern recognition letters*, 31(13):1935–1943, 2010.

[SGW+23] Lei Su, Jiazhi Guo, Liping Wu, Han Deng, et al. Bamnettl: bidirectional attention memory network with transfer learning for question answering matching. *International Journal of Intelligent Systems*, 2023, 2023.

[SP20] Marco Antonio Calijorne Soares and Fernando Silva Parreiras. A literature review on question answering techniques, paradigms and systems. *Journal of King Saud University-Computer and Information Sciences*, 32(6):635–646, 2020.

[YZK+22] Zhenrui Yue, Huimin Zeng, Ziyi Kou, Lanyu Shang, and Dong Wang. Domain adaptation for question answering via question classification. *arXiv preprint arXiv:2209.04998*, 2022.

[ZLC+23] Wenfeng Zheng, Siyu Lu, Zhuohang Cai, Ruiyang Wang, Lei Wang, and Lirong Yin. Palbert: An improved question answering model. *Computer Modeling in Engineering & Sciences; Tech Science Press: Henderson, NV, USA*, 2023.