QUESTIONS CLASSIFIER

National School of Applied Science Al Hoceima
Digital transformation and artificial intelligence Field
Author: ACHBAB Mohammed
Supervision: KHAMJANE Aziz

ABSTRACT

Classifying questions as either personal or non-personal is essential for applications like privacy protection , or filter of questions because it may help chatbots to recognise whether the user is looking for specific information or just wants to engage in casual conversation. This paper focuses on creating machine learning and deep learning models that can accurately differentiate between these two types of questions. By scraping data from various web sources, I built a dataset of questions, each labeled as either personal (related to the individual's personal life) or non-personal (general knowledge), this experiments show that the model performs well, suggesting it could be useful in improving information security and making conversational AI systems more context-aware.

Introduction

In the digital age, the ability to differentiate between personal and non-personal questions is becoming increasingly important. Applications such as natural language processing, sentiment analysis, conversational AI, and privacy protection all benefit from understanding the nature of the questions posed. Personal questions, which often involve sensitive or private information, require careful handling to ensure user privacy and security. Conversely, non-personal questions, typically related to general knowledge or impersonal topics, do not carry the same privacy concerns.

The distinction between personal and non-personal questions is not always straightforward and can vary depending on context and phrasing. This complexity necessitates the development of robust classification models capable of accurately identifying the nature of a question. By leveraging machine learning techniques, we aim to build a model that can learn and recognize the subtle differences between these two types of questions.

To achieve this, we first constructed a dataset of questions, each labeled as personal or non-personal. The dataset was compiled through web scraping, which allowed us to gather a diverse set of questions from various online sources. This method provided a rich and varied dataset, crucial for training an effective classification model.

RELATED WORKS

The task of classifying questions into categories has been addressed in various studies, each contributing valuable insights and methodologies to the field. This section provides an overview of the relevant literature and highlights the advancements and challenges in question classification.
One of the foundational studies in this area was conducted by Li and Roth (2002), who introduced a framework for learning question classifiers using a hierarchical taxonomy. Their work laid the groundwork for subsequent research by demonstrating the effectiveness of machine learning algorithms in categorizing questions based on syntactic and semantic features.

DATASET DESCRIPTION:

1. Source of the Dataset

The dataset for this research was collected from diverse and publicly available websites via web scraping using BeautifulSoup, each hosting a rich repository of questions and answers. These websites include:

For the class 1 ***(the personal questions):***

1. https://www.signupgenius.com/groups/getting-to-know-you-questions.cfm
2. https://thepleasantconversation.com/questions-to-get-to-know-someone/

For the class 0 ***(non personal questions):***

1. https://blog.livereacting.com/100-fun-general-knowledge-quiz-questions/
2. https://bestlifeonline.com/general-knowledge-questions/
3. https://www.opinionstage.com/blog/trivia-questions/

These sources were chosen for their wide range of content, ensuring a comprehensive collection of both personal and non-personal questions.
The scraping process involved extracting text-based questions and categorizing them into personal and non-personal categories based on predefined criteria.
To maintain data quality and relevance, each website was carefully reviewed to ensure that the questions were appropriate for the classification task. The diversity of sources also helped in capturing a broad spectrum of question types, enhancing the robustness of the dataset for training and evaluating the classification model. Especially for the non personal questions.

2. Data Preprocessing

Data preprocessing was conducted in two primary phases: during the data scraping process and after reading the final dataset.

Phase 1: Scraping Data

During the scraping process, it was crucial to filter out sentences that were not questions. This step ensured that the dataset contained only relevant entries.

Phase 2: Basic Cleaning

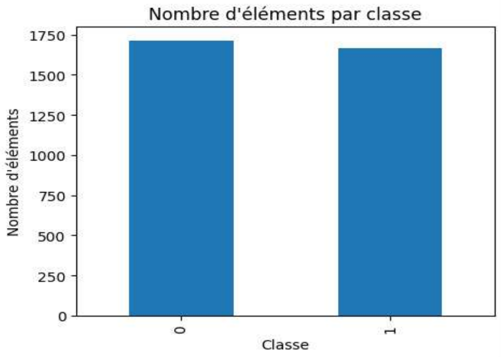
Once the final dataset was compiled, additional cleaning steps were necessary to further refine the data and avoid overfitting. The basic cleaning function was employed to perform the following operations:

- Removing Interrogation Marks: All interrogation marks were removed from the text data to maintain consistency.
- Eliminating Duplicates: Duplicate entries were identified and removed to ensure the uniqueness of each record.
- Removing Missing Values: Rows with missing values in the text data column were eliminated.
- Excluding Empty Strings: Rows containing empty strings in the text data column were also excluded.

These preprocessing steps were vital to maintain the quality and integrity of the dataset, ensuring it was suitable for the subsequent classification tasks.

Balancing the Dataset

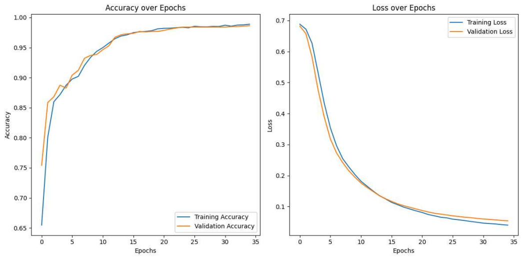
To address class imbalance, the dataset was balanced by adding additional rows to underrepresented classes. This step was essential to ensure that the model received a well-distributed training set, reducing the risk of bias towards any particular class.

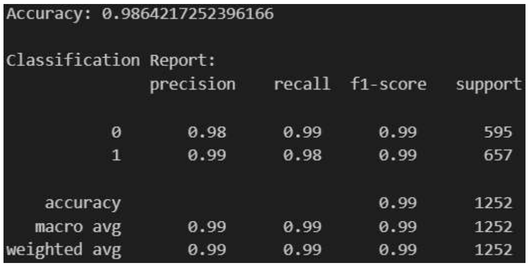


Models Architectures:

LSTM:

For this research, we employed a deep learning approach to classify questions as either personal or non-personal. The model architecture was built using Keras, a high-level deep learning API, with TensorFlow as the backend. The initial model was created using a Sequential approach, consisting of an Embedding layer, followed by an LSTM layer,and a Dense layer for classification. The model was compiled using the Adam optimizer with an optimized learning rate of and binary cross-entropy as the loss function because the question is either personal or not personal. As result for 35 epochs:





```
Accuracy: 0.9864217252396166

Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.99      0.99       595
           1       0.99      0.98      0.99       657

    accuracy                           0.99      1252
   macro avg       0.99      0.99      0.99      1252
weighted avg       0.99      0.99      0.99      1252
```

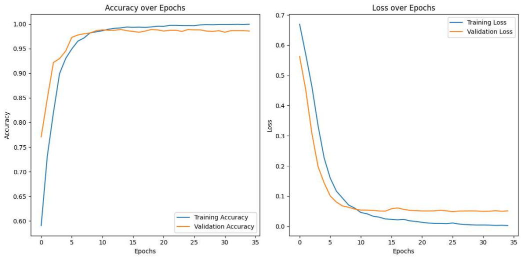
RNN(Recurrent Neural Network):

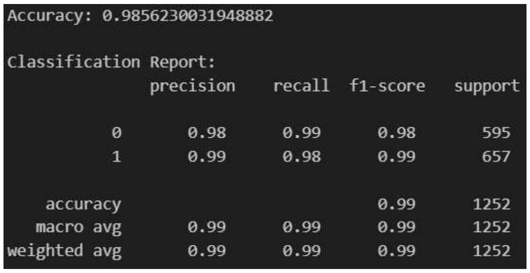
In addition to the LSTM model used for classifying personal and non-personal questions, we also employed a Recurrent Neural Network (RNN) for comparison. RNNs are effective for sequential data tasks, as they process sequences step-by-step and maintain an internal state that captures temporal dependencies in the input data. The RNN model was designed using Keras, with the following layers:

- Embedding Layer: This layer maps the input sequences (of length 200) into dense vectors of size 128. It was initialized with a vocabulary size of 5000.
- RNN Layer (SimpleRNN): A simple RNN layer with 128 units was used to capture the sequential nature of the questions. The model also included dropout layers (both regular and recurrent) with a rate of 0.2 to prevent overfitting and improve generalization.
- Dense Layer: The final dense layer uses a sigmoid activation function to produce the binary output (personal or non-personal question).

The RNN model was compiled with the Adam optimizer, with a customized learning rate, and the loss function was set to binary cross-entropy. The model was trained for 35 epochs with a batch size of 16, using a validation split to evaluate its performance on unseen data.

RNN result:





```
Accuracy: 0.9856230031948882

Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.99      0.98       595
           1       0.99      0.98      0.99       657

    accuracy                           0.99      1252
   macro avg       0.99      0.99      0.99      1252
weighted avg       0.99      0.99      0.99      1252
```

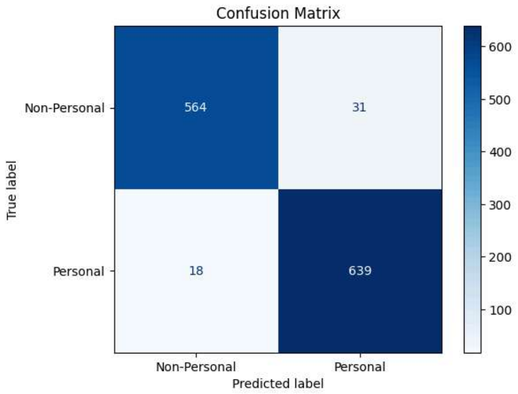
Naive Bayes Model

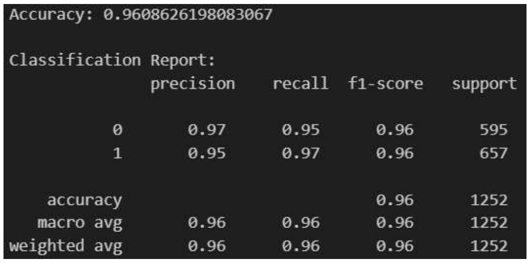
The Naive Bayes model is a probabilistic classifier based on Bayes' theorem, with an assumption of independence between features. Despite its simplicity, Naive Bayes is particularly well-suited for high-dimensional data, such as text classification. This makes it an ideal choice for our task of categorizing questions.

For our classification task, we employed the Multinomial Naive Bayes variant, which is commonly used for discrete data.

the result of this model was significant:

and here is the confusion matrix of our model:

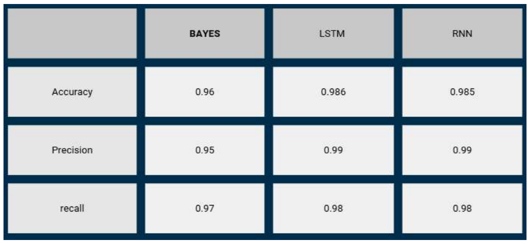




```
Accuracy: 0.9608626198083067

Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.95      0.96       595
           1       0.95      0.97      0.96       657

    accuracy                           0.96      1252
   macro avg       0.96      0.96      0.96      1252
weighted avg       0.96      0.96      0.96      1252
```

Choice of the model:

The choice of the final model was based on three key performance metrics: accuracy, precision, and recall. These metrics provide a comprehensive evaluation of the model's classification performance.



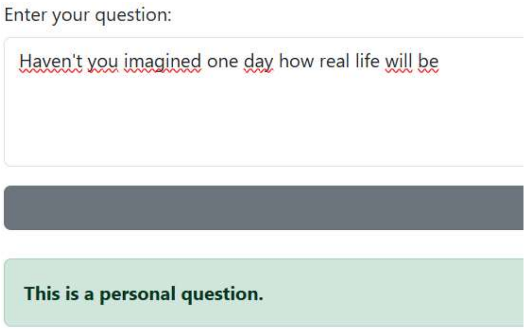
	BAYES	LSTM	RNN
Accuracy	0.96	0.986	0.985
Precision	0.95	0.99	0.99
recall	0.97	0.98	0.98

The LSTM model was selected as the final model because it demonstrated the best performance across these metrics, achieving the highest accuracy, with a slight improvement of 0.001 over the RNN model, along with stable precision and recall values.

Testing the models with sample questions:

During the testing phase of my project, I developed a Flask web application that integrates the WolframAlpha API to provide real-time answers to non-personal questions. My model worked here as a filter to di3scern non-personal questions and provide an answer using the WolframAlpha API.

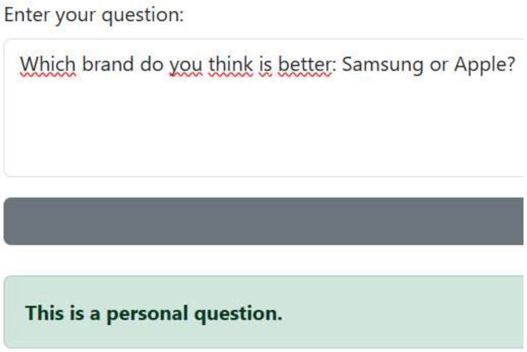
personal questions:



Enter your question:

Haven't you imagined one day how real life will be

This is a personal question.

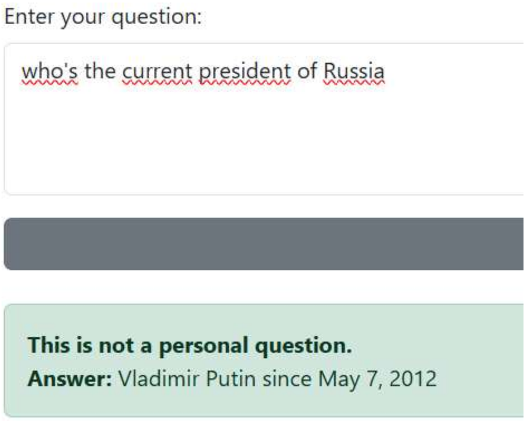


Enter your question:

Which brand do you think is better: Samsung or Apple?

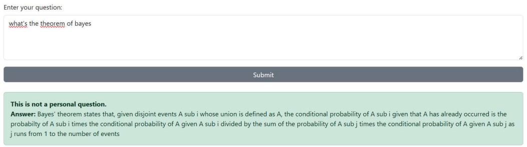
This is a personal question.

non personal questions:


Enter your question:

who's the current president of Russia

This is not a personal question.
Answer: Vladimir Putin since May 7, 2012


Enter your question:

what's the theorem of bayes

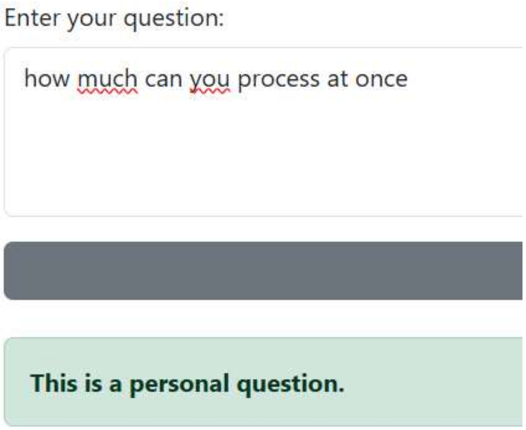
This is not a personal question.
Answer: Bayes' theorem states that, given disjoint events A sub i whose union is defined as A, the conditional probability of A sub i given that A has already occurred is the probabilty of A sub i times the conditional probability of A given A sub i divided by the sum of the probability of A sub j times the conditional probability of A given A sub j as j runs from 1 to the number of events

Limits of My Project

1- Sensitivity to Minor Text Variations Due to the absence of a spell-checker :

for example i entered a personal question than i made a simple error ("you" is misspelled as "yo")

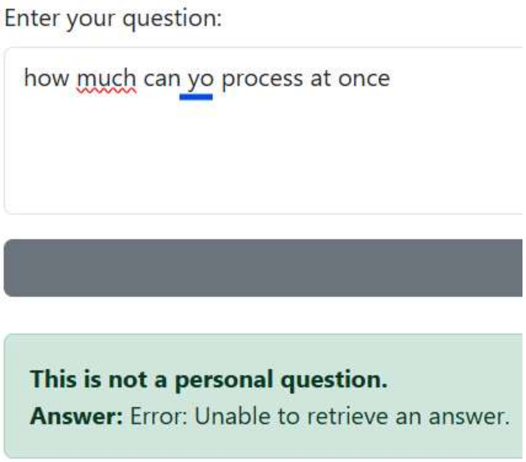
correct question:


Enter your question:

how much can you process at once

This is a personal question.

misspelled question:


Enter your question:

how much can yo process at once

This is not a personal question.
Answer: Error: Unable to retrieve an answer.

The reason why the model classified the personal question as non-personal is that the tokenizer treated "yo" as a different token from the token ‘you’, leading it to classify the question as non-personal instead of personal. This highlights the sensitivity of the model to spelling errors and underscores the importance of accurate text input for proper classification.

2-Limited Dataset Diversity:

The dataset primarily consists of general and personal questions collected through web scraping. However, it may not represent all linguistic variations, cultural contexts, or domain-specific terminologies. This limitation could affect the generalizability of the model when applied to new data from diverse sources.

Conclusion

In this study, a model was developed to classify questions as personal or non-personal, and it demonstrated promising performance. The model showed good accuracy in distinguishing between the two categories, highlighting its potential for real-world applications. However, several challenges were encountered, particularly in the data collection phase. The main difficulty stemmed from the limited number of questions available on the websites used for scraping, which resulted in a relatively small dataset. This constraint posed challenges for model generalization and may have impacted its ability to fully capture the nuances of different question types. Despite these limitations, the model achieved satisfactory results, indicating that with a larger and more diverse dataset, its performance could be further improved.

REFERENCES:

1. Li, X., & Roth, D. (2002). Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1* available at: https://aclanthology.org/C02-1150.pdf
2. Ittycheriah, M. Franz, W-J Zhu, A. Ratnaparkhi, and R.J. Mammone. 2001. IBM's statistical question answering system. In Proceedings of the 9th Text Retrieval Conference, NIST**.** available at: https://trec.nist.gov/pubs/trec9/papers/ibm_qa.pd