FAITHDIAL: A FAITHFUL BENCHMARK FOR INFORMATION SEEKING DIALOGUE

Jagath Sai Narayana Kakaraparty
UBIT: 50442123
jagathsa@buffalo.edu

Siddhi Satish Jadhav
UBIT: 50442859
sjadhav3@buffalo.edu

Briyana Rana
UBIT: 50442498
brana@buffalo.edu

I. INTRODUCTION

Information-seeking dialog systems are becoming more and more common in a variety of uses, including search engines, chatbots for customer support, and virtual aides. In order to comprehend user inquiries and respond with information that is pertinent and accurate, these systems depend on natural language processing techniques. However, despite their rising appeal, these systems can still suffer from a lack of confidence and dependability because their responses sometimes contain hallucinatory data.

Information that is hallucinatory can occur for several reasons, including inadequate knowledge bases, biases in algorithms, and mistakes in data sources. Due to limitations in its design or code, the system might occasionally purposefully provide incorrect information. Regardless of the reason, the presence of hallucinatory information in the system's answers may leave users perplexed and less inclined to believe it.

The goal is to create a system for information-seeking dialogue that is taught to identify and exclude information obtained through hallucinations from the response text. Information that is inaccurate or unreliable and may cause the user to become misinformed or confused is referred to as hallucinated information. Existing dialogue systems might not have adequate safeguards in place to stop the inclusion of such information in their replies, which can undermine confidence in the system's dependability. To increase the overall credibility and efficiency of the information-seeking dialogue system, the goal is to create a system that can precisely identify and filter out hallucinated information.

II. DATASET PREPROCESSING

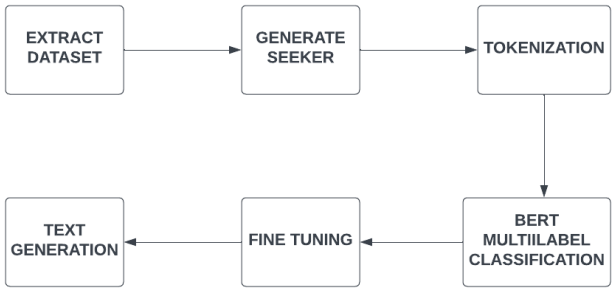
The dataset consisted of various fields out of which 'history', 'response', 'BEGIN' and 'VRM' were the fields that would be used in the analysis. 'history' field has multiple sentence utterances and hence was processed to create a new field 'seeker' which will consist of all the seeker utterances excluding the 'response' from it. Since the data in fields "BEGIN' and 'VRM' have multiple labels, we process then using 'MultiLabelBinarizer' library to encode them.

III. BASELINE ARCHITECTURE

The baseline architecture consists of two tasks:

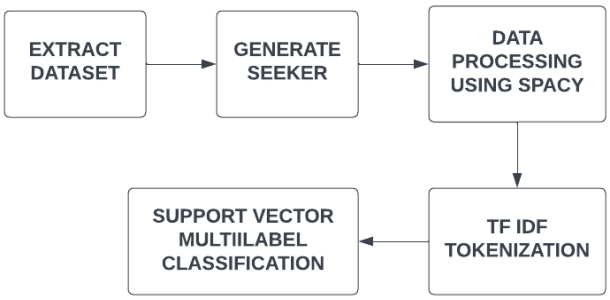
(i) Determining if the response is hallucinated or not along with VRM taxonomy classification. Hallucination-free dialogue generation.

A. **BERT**: To use the BERT-based method, we prepared the dialogue dataset by labeling each dialogue as "Entailment", "Hallucinated", "Not Hallucinated" & "Generic". The dataset should be split into train and test sets. Next, we loaded the pre-trained BERT model from a pre-trained model library, such as the Hugging Face Transformers library. Using the tokenizer, we have tokenized the input dialogue text into individual tokens and converted them into numerical representations that can be fed into the BERT model. The pre-trained BERT model is then fine-tuned on the dialogue dataset by adding additional layers and training the model to classify each dialogue as either "Entailment" or "Hallucinated". Finally, we have evaluated the performance of the BERT-based model using metric macro-average F1-score on the test set.

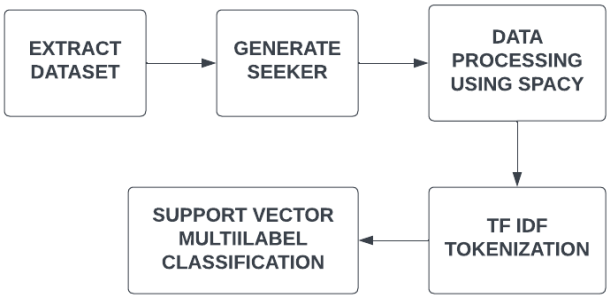


B. **SVM**: We will represent each sentence as a sequence of words. The SVM classifier can then use the number vectors created from each phrase as input. We weigh each word in the statement according to its significance using methods like TF-IDF. After each phrase has been transformed into a number vector, the SVM can be fed these vectors. To determine a judgment boundary between the positive and negative data, we will train a linear SVM classifier. The SVM will identify a hyperplane that has the greatest range of separation between the positive and negative data. We will modify the SVM's settings during training to improve its performance on a test set. Once the SVM has learned the decision boundary, we can use it to predict the labels of new input sentences. The output label will consist of both BEGIN and VRM taxonomy labels. As

mentioned above macro average F1 score will be considered to evaluate the model's performance.

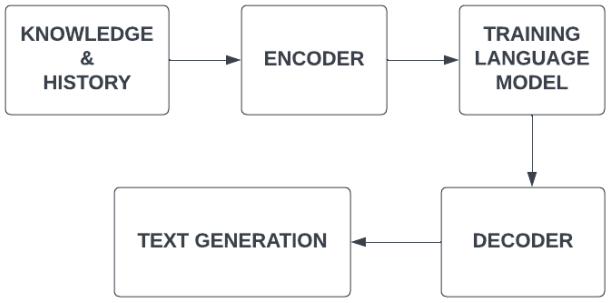


C. NAÏVE BAYES: To establish a judgment limit between the positive and negative data, we will employ a Naive Bayes classifier. The Naive Bayes classifier chooses the class with the greatest chance by calculating the probability of each class given the characteristics. The feature extraction layer creates a feature vector for each phrase after passing the incoming sentences through it. The Multinomial Naive Bayes classifier then receives these feature vectors and learns to categorize the phrases as "Entailment", "Hallucinated", "Not Hallucinated" & "Generic" based on their content. Finally, the classifier generates the final categories. The model application will determine the specifics of the feature extraction and Multinomial Naive Bayes training. But this design model gives a broad summary of the operation of the fundamental Multinomial Naive Bayes classifier.



(ii) Text Generation

For the task of text generation, we relied on state-of-the-art models such as GPT2. GPT2 is a transformer model which is specially used for Casual Language modelling and hence we have decided to use it for text generation. It is a model that is pretrained on a very large English data corpus. For Hallucination free dialog generation, we encode the 'history' utterances along with 'knowledge' utterances and we use the GPT2 fast tokenizer for this purpose. GPT2 makes use of attention masks internally so that the model is trained by predicting the masked word and hence being efficient. We initialize a new model for training instead of relying on the pretrained GPT2 model and train our data on this model. We check the training and testing loss for the model we trained. We push this model to HuggingFace repository for further usage. We then generate results manually.



IV. RESULTS

As the process of classification is done on both BEGIN and VRM taxonomy the result for each classifier varies largely using the same models on both taxonomies. The difference between both classifications can be demonstrated using the results of implementation of all the above classifiers are shown below:

A. BEGIN TAXONOMY: Multi labeled taxonomy to identify if the data is "Entailment", "Hallucinated", "Not Hallucinated" & "Generic".

- **BERT:**

	precision	recall	f1-score	support
0	1.00	0.74	0.85	100
1	0.00	0.00	0.00	0
2	1.00	0.67	0.80	100
3	0.00	0.00	0.00	0
micro avg	0.97	0.70	0.82	200
macro avg	0.50	0.35	0.41	200
weighted avg	1.00	0.70	0.83	200
samples avg	0.97	0.70	0.79	200

- **SVM:**

	precision	recall	f1-score	support
0	0.74	1.00	0.85	74
1	0.00	0.00	0.00	2
2	0.67	1.00	0.80	67
3	0.00	0.00	0.00	2
micro avg	0.70	0.97	0.82	145
macro avg	0.35	0.50	0.41	145
weighted avg	0.69	0.97	0.80	145
samples avg	0.70	0.97	0.79	145

- **MULTINOMIAL NAÏVE BAIYES:**

	precision	recall	f1-score	support
0	0.73	0.95	0.82	74
1	0.00	0.00	0.00	2
2	0.68	1.00	0.81	67
3	0.00	0.00	0.00	2
micro avg	0.70	0.94	0.81	145
macro avg	0.35	0.49	0.41	145
weighted avg	0.68	0.94	0.79	145
samples avg	0.71	0.95	0.78	145

B. VRM TAXONOMY: Multi labeled taxonomy to understand the context of the conversation or the response. The labels are as follows: "Acknowledgement", "Advisement", "Disclosure", "Edification", "Question".

- **BERT**

	precision	recall	f1-score	support
0	0.28	0.32	0.30	25
1	0.00	0.00	0.00	0
2	1.00	0.55	0.71	100
3	0.76	0.59	0.66	80
4	0.00	0.00	0.00	0
micro avg	0.67	0.54	0.60	205
macro avg	0.41	0.29	0.33	205
weighted avg	0.82	0.54	0.64	205
samples avg	0.71	0.54	0.58	205

- **SVM**:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	29
1	0.00	0.00	0.00	2
2	0.55	1.00	0.71	55
3	0.62	1.00	0.77	62
4	0.00	0.00	0.00	16
micro avg	0.58	0.71	0.64	164
macro avg	0.23	0.40	0.30	164
weighted avg	0.42	0.71	0.53	164
samples avg	0.58	0.78	0.64	164

- **MULTINOMIAL NAÏVE BAIYES**:

	precision	recall	f1-score	support
0	0.42	0.38	0.40	29
1	0.00	0.00	0.00	2
2	0.58	0.78	0.67	55
3	0.64	0.66	0.65	62
4	0.00	0.00	0.00	16
micro avg	0.58	0.58	0.58	164
macro avg	0.33	0.36	0.34	164
weighted avg	0.51	0.58	0.54	164
samples avg	0.60	0.62	0.57	164

Comparing the macro-average f1-score, the value for each model is shown in the below table:

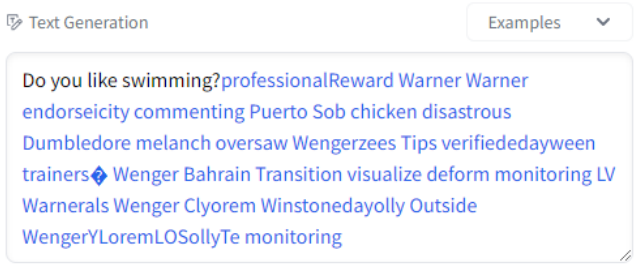
	BEGIN	VRM
BERT	0.35	0.33
SVM	0.41	0.30
MULTINOMIAL NB	0.41	0.34

The Faith Dial dataset is a conversational dataset that consists of dialogues between humans and a conversational agent on various topics related to faith and religion. These dialogues can be quite complex, with multiple turns and nuanced language, making it challenging to classify the dialogue's sentiment accurately. BERT is a great option for sentiment analysis on the Faith Dial dataset because it can catch context and comprehend the meaning behind the text. Furthermore, BERT is adaptable to various domains and languages, making it a flexible option for a range of NLP jobs. SVM and Multinomial Naive Bayes, on the other hand, are conventional models that use feature engineering and statistical presumptions to categorize text. These models can excel at straightforward categorization tasks, but they might have trouble handling the intricate Faith Dial dataset.

C. TEXT GENERATION:
The model used is a fine – tuned version of GPT2 on an unknown dataset. The following results can be obtained on the evaluation set:

- Train Loss: 10.9520
- Validation Loss: 10.9647

Example of the text generated by the pretrained model:



V. ANALYSIS OF BASELINE RESULTS AND AREAS OF IMPROVEMENT.

On observing the results (macro F1 score) form all the models for multi-label classification it is safe to assuming the BERT would be best suited for the current application as not only does it classifies the data but also learns the context of the conversation to obtained perfect the results. Increasing the size of the dataset would definitely be an area from improvement. We observe that the GPT2 model is not able to correctly predict a sentence. For future improvement, we will build our own vocabulary by training the tokenizer on the dataset given instead of relying on the pre-trained tokenizer.