Table of Contents

[PROJECT: IMPROVING RELATIONAL LANGUAGE UNDERSTANDING IN INTELLIGENT VIRTUAL AGENTS (IVAS) 2](#_Toc159523319)

[1.0 OVERVIEW 2](#_Toc159523320)

[1.1 Introduction 2](#_Toc159523321)

[1.2 Challenges 2](#_Toc159523322)

[1.3 Proposed Solutions 2](#_Toc159523323)

[1.4 Brief Conclusion 3](#_Toc159523324)

[2.0 BUSINESS UNDERSTANDING 3](#_Toc159523325)

[2.1 Problem statement 3](#_Toc159523326)

[2.2 Audiences / Stakeholders 3](#_Toc159523327)

[2.3 Objectives 3](#_Toc159523328)

[3. 0 DATA UNDERSTANDING AND PREPARATION 4](#_Toc159523329)

[3.1 Data Understanding 4](#_Toc159523330)

[3.1.1 Source of Data 4](#_Toc159523331)

[3.1.2 Dataset Details/ Descriptions 4](#_Toc159523332)

[3.2 Data Analysis 6](#_Toc159523333)

[3.2.1 Univariate Analysis 6](#_Toc159523334)

[3.2.2 Bivariate Analysis 7](#_Toc159523335)

[4.0 MODELING 8](#_Toc159523336)

[5.0 CONCLUSION AND RECOMMENDATION 9](#_Toc159523337)

[5.1 Conclusion 9](#_Toc159523338)

[5.2 Recommendations 9](#_Toc159523339)

[5.3 Next Steps 9](#_Toc159523340)

# PROJECT: IMPROVING RELATIONAL LANGUAGE UNDERSTANDING IN INTELLIGENT VIRTUAL AGENTS (IVAS)

# 1.0 OVERVIEW

In order to promote more efficient and natural human-agent interactions, this project aims to improve Intelligent Virtual Agents' (IVAs') relational language understanding skills. The goal of this project is to enhance the overall user experience and empower intelligent assistants (IVAs) to execute tasks more accurately and efficiently by enhancing their capacity to understand complex relationships expressed in language.

## Introduction

Intelligent virtual agents, or IVAs, have become essential parts of many applications, from virtual assistants to chatbots for customer service. However, IVAs' efficacy is frequently constrained by their capacity to accurately comprehend and react to human language, especially when it comes to intricate relational concepts. By concentrating on enhancing IVAs' relational language understanding, the project aims to overcome this difficulty and close the gap between machine comprehension and human communication.

## Challenges

Several challenges hinder IVAs' ability to understand relational language effectively:

1. Semantic Complexity: Relational language frequently contains intricate semantic structures and nuanced expressions that are difficult for conventional NLP models to understand.
2. Context Sensitivity: In order to fully comprehend relational language, one must take into account the conversation's larger context, which includes prior exchanges and contextual elements.
3. Real-Time Responsiveness: In order to preserve a natural conversation flow, IVAs must process language inputs quickly, which places limitations on their efficiency and computational resources.

## Proposed Solutions

This project proposes a comprehensive approach to address the challenges in relational language understanding:

1. Advanced Natural Language Processing (NLP) Models: Develop and improve neural network structures designed expressly to precisely capture and represent the semantics of relational languages.
2. Contextual adaptation: Enhance contextual sensitivity by implementing mechanisms that allow intelligent virtual assistants (IVAs) to modify their comprehension of relational language in response to contextual cues and user intent.
3. Optimize model architectures and algorithms to ensure real-time responsiveness and efficient utilization of computational resources.

## Brief Conclusion

In conclusion, the project on improving relational language understanding in IVAs represents a critical step towards advancing the capabilities of virtual agents to interact more intelligently and effectively with users. By addressing the challenges associated with relational language comprehension and proposing innovative solutions, the project aims to elevate the state-of-the-art in IVAs and unlock new opportunities for their integration across various domains and applications.

# 2.0 BUSINESS UNDERSTANDING

## 2.1 Problem statement

When interacting with customers, existing Intelligent Virtual Agents (IVAs) often find it difficult to decipher and react correctly to relational language cues. This shortcoming impairs customer satisfaction and experiences, as well as the IVAs' ability to build rapport, effectively handle customer needs, and cultivate positive relationships. Thus, there is a pressing need for processes and instruments that enable IVAs to recognise and react to relational language cues with precision, improving the calibre of customer service encounters and increasing customer satisfaction levels.

## 2.2 Audiences / Stakeholders

Our audiences and key stakeholders include:

1. Kenya Airways
2. Jambo Jet
3. Safaricom

## 2.3 Objectives

Our objectives include:

1. Develop algorithms and models to enable IVAs to comprehend relational language expressions comprehensively, including complex relationships and subtle nuances. (**MAIN**)
2. Enhance IVAs' ability to understand relational language in various contexts, adapting their interpretations based on the specific scenario and user intent.
3. Optimize the computational efficiency of the relational language understanding models to ensure real-time responsiveness of IVAs during interactions.

# 3. 0 DATA UNDERSTANDING AND PREPARATION

## 3.1 Data Understanding

### 3.1.1 Source of Data

The RSiCS dataset was collected from commercial customer service IVAs and the TripAdvisor airline forum (<https://nextit-public.s3-us-west-2.amazonaws.com/rsics.html)>).

### 3.1.2 Dataset Details/ Descriptions

The dataset contains 4 files and they include

1. x\_y\_align.csv

Captures alignment and agreement metrics between Annotator A (x) and Annotator B in a group for specific requests.

Columns:

* **Annotator A ID**: **x**
* **Annotator B ID**: Annotator that the alignment score with **x** is calculated against.
* **Group ID**: The group of 4 annotators that the compared users belong to.
* **Dataset ID**: Dataset **y** that the request originated from.
* **Request ID**: Unique ID of a request to allow joining between different files.
* **Text**: The original request text.
* **Annotator A Text**: The request text with selections from annotator A contained within [ and].
* **Annotator B Text**: The request text with selections from annotator B contained within [ and].
* **Length**: The character length (n) of the original request text in column 6.
* **Error**: The number of character positions (e) where the binary determination of A and B do not agree.
* **Alignment Score**: The alignment as calculated by align = (n - e) / n.
* **Agreement**: Whether or not A and B agree that *any* selection is necessary.

1. all\_data\_by\_threshold.csv

Merges selections and determines user intentions.

Columns:

* **Dataset ID**: Dataset that the request originated from.
* **Group ID**: The group of 4 annotators that the selections originated from.
* **Request ID**: Unique ID of a request to allow joining between different files.
* **MultiIntent**: 1 if at least one annotator flagged the text as containing more that one user intention, 0 otherwise.
* **Threshold**: The threshold (i) to merge selections by.
* **MergedSelections**: If at least i annotators marked a character as unnecessary then it will be contained within the selected portion denoted by [ and ].
* **Unselected**: All text from **MergedSelections** not contained by [ and ].
* **Selected**: All text from **MergedSelections** contained by [ and ].
* **Removed**: Amount of text removed from the original request by the merged selections: length(Selected) / n.

1. tagged\_selections\_by\_sentence.csv

Identifies relational language in user requests.

Columns:

* **Dataset ID**: Dataset that the request originated from.
* **Group ID**: The group of 4 annotators that the selections originated from.
* **Request ID**: Unique ID of a request to allow joining between different files.
* **Threshold**: The threshold (i) to merge selections by.
* **MergedSelections**: If at least i annotators marked a character as unnecessary then it will be contained within the selected portion denoted by [ and ].
* **Unselected**: All text from **MergedSelections** not contained by [ and ].
* **Selected**: All text from **MergedSelections** contained by [ and ].
* **Greeting**: If a greeting of some kind (*Hi*, *How are you*) is present in **Selected**
* **Backstory**: If self-exposure language is present in **Selected**. The user is telling the audience about themselves, their situation, what led them to contact the agent or ask their question.
* **Justification**: If justification language is present in **Selected**. The user is giving facts to build credibility that their request or statement is true. Also can be *why* they need resolution or a consequence if something is not resolved.
* **Rant**: If ranting is present in **Selected**. Excessive complaining or negative narrative.
* **Gratitude**: If some expression of gratitude to the audience for past or future help is present in **Selected**.
* **Other**: If some or all of the highlighted section does not contain any relational language in **Selected**. Could be additional facts the user gave but annotators determined was unnecessary to determine their intention, or a general question such as *Can you help?*.
* **Express Emotion**: If any emotional language not covered by **Rant** is present in **Selected**.

1. all\_multi\_intent.csv

Flags requests with multiple intentions for intent detection strategies.

Columns:

* **Dataset ID**: Dataset that the request originated from.
* **Group ID**: The group of 4 annotators that the selections originated from.
* **Request ID**: Unique ID of a request to allow joining between different files.
* **Text**: The original request texts.
* **Annotator *x*:** Will be 1 if annotator **x** believed more than one intent was present in the text, 0 otherwise.

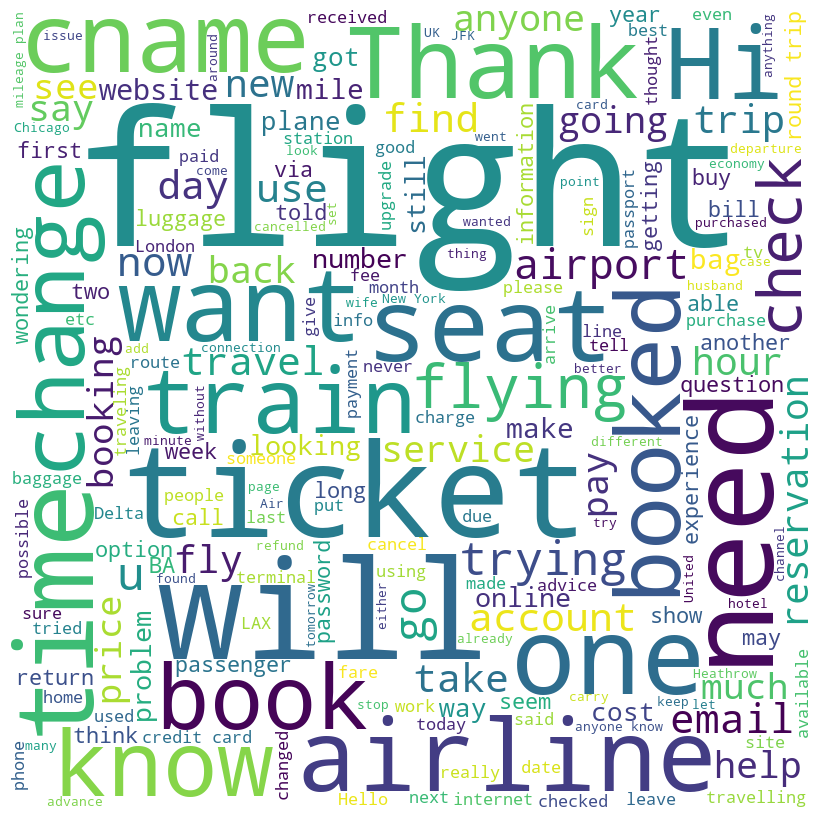
## 3.2 Data Analysis

The following steps were followed in preparing the data;

* Importing the necessary libraries.
* Loading the dataset from the CSV format it was stored in.
* Exploring the dataset.
* Cleaning datasets (Filling null values, removing duplicates, dropping unnecessary columns).
* Creating a new data frame with the necessary columns for our project.
* Exploratory Data Analysis (EDA).

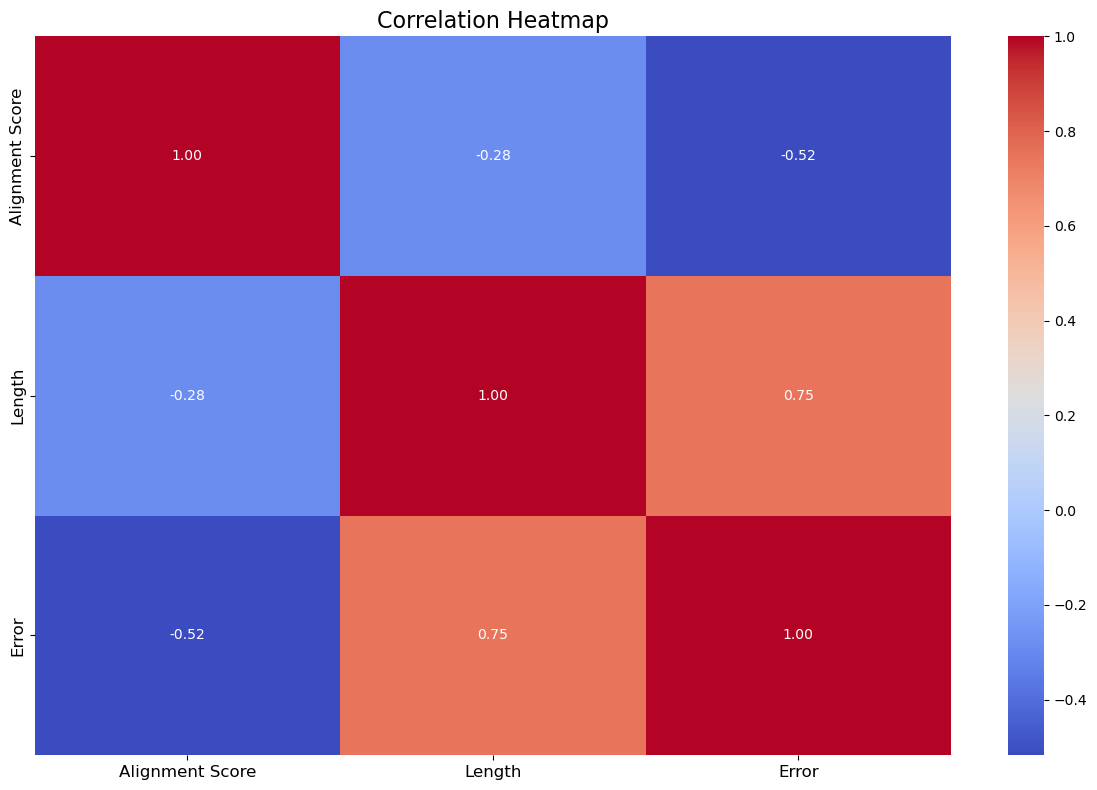
### 3.2.1 Univariate Analysis

A word cloud visualization to represent the frequency of words in the text data, providing a quick and visually appealing way to understand the most common words in the text.



### 3.2.2 Bivariate Analysis

A heatmap to visualize the correlation between pairs of numerical variables (Alignment Score, Length, and Error).



1. Alignment Score vs. Length: Negative correlation of -0.293, indicating a slight decrease in alignment score as text length increases.
2. Alignment Score vs. Error: Strong negative correlation of -0.856, showing that higher errors correspond to lower alignment scores.
3. Length vs. Error: Moderate positive correlation of 0.619, suggesting longer texts tend to have higher error rates.

# 4.0 MODELING

RASA was used for modelling and the following files utilized

1. NLU.yml - Contains training data for the Natural Language Understanding (NLU) model, including examples of user messages labelled with intents and entities.

2. Config.yml- specifies the configuration for the machine learning models used in the Rasa project, including settings for the NLU and dialogue management components.

3. stories.yml - Defines conversational flows by mapping out sequences of user inputs (intents) and corresponding bot responses. It's a collection of example conversations.

4. Domain. yml - defines the domain of the assistant, including intents, entities, actions, responses, and slots. It essentially outlines what the assistant can understand and do.

5. credentials.yml - Stores credentials and connection information for external services that the assistant interacts with, such as databases, APIs, or messaging platforms.

6. actions.py - Defines custom actions that the bot can perform, such as calling APIs, querying databases, or executing specific tasks based on user requests.

7. rules.yml - Contains rules for defining specific conversation paths or behaviours based on predefined conditions. These rules can help guide the dialogue in certain situations.

8. endpoints.yml - Specifies the endpoints for the Rasa project, including the location of the model server and other services like the action server and tracker store.

# 5.0 CONCLUSION AND RECOMMENDATION

## 5.1 Conclusion

In conclusion, the development of algorithms and models to enhance relational language understanding in Intelligent Virtual Agents (IVAs) represents a significant step forward in improving the quality and effectiveness of human-computer interactions. We have made substantial progress in enabling IVAs to comprehend complex relationships and subtle nuances in relational language expressions, adapt to various contexts, and ensure real-time responsiveness during interactions.

## 5.2 Recommendations

1. Model Fine-tuning: Continue fine-tuning the models on additional datasets to further improve their performance and adaptability to diverse contexts.
2. Integration of Multimodal Inputs: Explore the integration of multimodal inputs, such as text, speech, and visual cues, to enhance the understanding of relational language and enrich user interactions.
3. Continuous Monitoring and Feedback: Implement mechanisms for continuous monitoring of IVA interactions and gathering user feedback to iteratively refine the models and enhance user satisfaction.
4. Collaboration with Domain Experts: Collaborate with domain experts to incorporate domain-specific knowledge and refine the models' understanding of relational language within specific domains.

## 5.3 Next Steps

1. Further Research: Conduct further research on advanced NLP techniques, such as dynamic context modelling and commonsense reasoning, to enhance the understanding of complex relational language expressions.
2. Prototype Development: Develop prototypes incorporating the enhanced relational language understanding models and conduct extensive testing in real-world scenarios.
3. Deployment and Scaling: Deploy the optimized models in production environments and ensure scalability to handle large-scale interactions with users.
4. Continuous Improvement: Establish processes for continuous monitoring, evaluation, and improvement of the deployed models based on real-world usage and feedback.