# Chapter 3: Methodology

The methodology section of this research outlines the steps and procedures followed to achieve the research objectives. This section is divided into four main sections, namely Dataset, Software Environment and Libraries, Proposed Pipeline, and Evaluation. In the Dataset section, we described the cable damage guideline dataset used in this research. The Software Environment and Libraries section outlined the tools, libraries, and software environments used for the development of the proposed chatbot. The Proposed Pipeline section detailed the steps and processes of the chatbot development, including data preprocessing, natural language processing, model selection, training, testing and evaluation, and deployment. Finally, the Evaluation section presented the metrics and evaluation techniques used to measure the performance of the chatbot. This methodology section provides a clear understanding of the approach taken in the research and serves as a guide for replicating and extending the research.

Graphical user interface

Description automatically generated

Figure 3. 1: Overall Pipeline of the Methodology

Diagram

Description automatically generated

Figure 3. 2: Detail Methodology workflow.

## Dataset

The dataset used in this study is designed to guide the approach for deciding on rectification of damages noticed during the installation of 132kV cables without jeopardizing the quality and long-term performance of the cable system. The purpose of the dataset is to provide a reference during the execution of cable laying projects for the recommended rectification and repair methods to be adopted for various possible damages on 132kV cables.

The dataset considers optimum solutions that do not impact the project's timeline and long-term performance, utilize technological advancements to determine the extent of damages and its effects, practices followed by various HV cable manufacturers and other utilities in the region, and international best practices or publications on HV cable maintenance procedures.

The dataset includes a table that outlines the standard practices to be followed for resolving technical issues of 132kV cables at the site without jeopardizing the quality of work or long-term performance of the system. The table provides a detailed description of defects that are outcomes of utility and OEM investigations and offers a range of repair methods for different types of damages, including cracks, punctures, cuts, and more. The general notes section of the dataset provides additional information on conducting factory quality audits, penalties for damages due to contractor negligence, and the installation of additional joints.

The dataset is useful for developing a chatbot that can assist in the decision-making process for repairing damages to 132kV cables during installation. The dataset should be preprocessed and organized in a way that can be easily loaded into the chatbot's backend system before being incorporated into the chatbot's knowledge base. This may entail developing a database or knowledge graph to store information in a systematic manner. Based on the user's queries, the chatbot can then access this knowledge base to provide recommendations and help. The chatbot may also be trained on the dataset using machine learning techniques, deep learning technique during the integration phase to increase its accuracy and performance.

## Software Environment and Libraries

Software environment and libraries play a crucial role in the development of chatbots. In this section, we describe the software environment and libraries that will be used in the proposed pipeline. The development of the chatbot will be carried out using the Python programming language, which is widely used for natural language processing tasks. Python provides a large number of libraries for natural language processing, including NLTK (Natural Language Toolkit), spaCy, and Hugging Face.

NLTK is a popular library for natural language processing and provides various tools and resources for tasks such as tokenization, stemming, and part-of-speech tagging. It also includes a large collection of text corpora and lexical resources, making it a valuable resource for natural language processing.

spaCy is another widely used library for natural language processing that provides efficient algorithms for tokenization, named entity recognition, and dependency parsing. It also includes pre-trained models for various languages, making it easy to perform natural language processing tasks without the need for extensive training data.

Hugging Face library provides access to pre-trained state-of-the-art language models, such as GPT-2, BERT, RoBERTA, XLNet ..etc which can be fine-tuned for specific tasks, including chatbot development. Additionally, the library includes various pre-processing tools, such as tokenizers, that can facilitate input and output formatting for chatbot interactions. The Hugging Face library also provides an API that can be used to host the chatbot on a server and integrate it into various applications.

In addition to these libraries, the proposed pipeline also uses various deep learning frameworks such as TensorFlow and PyTorch for model development and training. These frameworks provide a wide range of tools and resources for building and training neural networks, making them well-suited for developing chatbots using generative models.

Overall, the combination of these software environments and libraries provides a powerful and flexible platform for the development of chatbots. They offer a range of tools and resources for natural language processing and deep learning, making it possible to build chatbots with sophisticated language understanding and generation capabilities.

## Proposed Pipeline

The study's scope is centered on building a chatbot system that can assist users in resolving technical challenges linked to 132kV cables during installation without compromising job quality or system long-term performance. The suggested pipeline includes multiple processes, such as data preprocessing, model selection, training, and assessment, and it employs cutting-edge deep learning algorithms and generative models to generate natural language responses to user queries. The chatbot's accuracy and usefulness in resolving technical difficulties linked to 132kV cables during installation will be evaluated.

### Data preprocessing using Natural Language Processing Techniques

Data preprocessing is a crucial step in any machine learning pipeline as it directly affects the quality of the model's predictions. In the case of our cable damage guideline dataset, preprocessing involves cleaning and transforming the raw data into a format suitable for training our chatbot. The first step is to remove any irrelevant information from data that might not be providing any valuable information for prediction. We also need to convert the data into a structured format, such as a table or CSV file, so that it can be easily loaded into the machine learning model. Once the data is cleaned and structured, we can proceed with text preprocessing, which involves techniques such as tokenization, stemming, and lemmatization to standardize the text and make it more machine-readable. We may also need to remove stop words and perform feature engineering to extract relevant features from the text data using NLP techniques.

### Model selection

Model selection is a crucial step in chatbot development, as it determines the overall performance and functionality of the chatbot. The selection process involves considering various factors, such as the chatbot's objectives, the type of conversation it will handle, the available resources, and the performance metrics. In general, there are three types of models commonly used in chatbot development: rule-based, retrieval-based, and generative models, refer to point 2.1.

When selecting a model for chatbot development, it is important that we consider the trade-off between accuracy and complexity. Rule-based models are the simplest and require the least number of resources but are limited in their ability to handle complex conversations. Generative models, on the other hand, are the most complex and resource-intensive but can handle a wider range of conversations and produce more human-like responses. Additionally, the performance metrics used to evaluate the chatbot's performance should be considered, including accuracy, precision, recall, F1 score, BLEU score, ROUGE score, and perplexity etc.

In our proposal, we plan to leverage generative models as they allow for the creation of complex conversations. As we have a limited dataset, we will need to fine-tune pre-trained models and evaluate their performance. We will explore various state-of-the-art generative models such as GPT-3, T5, and BART. These models are based on transformer architecture and have demonstrated superior performance in language generation tasks. We will also consider the use of simpler models such as LSTM and GRU, which have been widely used in chatbot development. The model selection process will be guided by the performance of these models on our specific dataset and the complexity of the conversations we aim to generate.

### Training, Testing and Evaluation

The next step in our proposed pipeline is training the selected model on the preprocessed data in the form of questions and answers that are relevant to the domain. This involves feeding the data into the model and adjusting its parameters to optimize its performance on our specific task. This step is to teach the model how to identify relevant information and generate accurate responses to questions. During training, the model's parameters are adjusted to minimize the difference between the predicted answer and the ground truth answer in the dataset. The training process is iterative and may require multiple rounds of adjustment and retraining to achieve the desired performance metrics.

After the model is trained, it is important to test its performance and evaluate its effectiveness.

The evaluation of the chatbot system will be done using various metrics. These metrics will be used to measure the performance of the system in terms of how well it can understand user queries and provide relevant responses. Precision is a measure of how many of the responses provided by the chatbot were actually correct. Recall is a measure of how many of the correct responses were provided by the chatbot. F1 score is a measure of the overall accuracy of the system. Accuracy is a measure of how many of the responses provided by the chatbot were correct.

To evaluate the performance of the chatbot, a set of test data will be used such as word2vec, Glove, Fast text, Seq2seq with Single Unidirectional LSTM without and with attention Architectures, GRU, BERT Base , BERT Large, ROBERTa, DistiBERT, transformer etc.

In order to evaluate our system, we will use the following metrics:

* BLEU (Bilingual Evaluation Understudy) - measures the n-gram overlap between generated text and reference text. It is commonly used to evaluate machine translation systems, but can also be used for other text generation tasks.
* ROUGE (Recall-Oriented Understudy for Gisting Evaluation) - measures the overlap between generated text and reference text using various similarity measures such as n-gram overlap, longest common subsequence, and skip-bigrams. It is commonly used to evaluate text summarization systems.
* METEOR (Metric for Evaluation of Translation with Explicit ORdering) - a metric that combines multiple measures such as n-gram overlap, word alignment, and synonyms to evaluate machine translation systems.
* CIDEr (Consensus-based Image Description Evaluation) - a metric that is commonly used to evaluate image captioning systems. It measures the similarity between generated captions and human-generated captions using a consensus-based approach.
* Perplexity - a metric that is commonly used to evaluate language models. It measures how well a language model can predict a sequence of words. Lower perplexity values indicate better performance.
* F1-score - a metric that is commonly used to evaluate text classification systems. It measures the harmonic mean of precision and recall and is used to evaluate how well a model can classify text into different categories.
* Human evaluation - sometimes the best way to evaluate the quality of generated text is to have human evaluators rate it. This can be done using crowd-sourcing platforms such as Amazon Mechanical Turk.

Overall, the evaluation of the chatbot system will be a continuous process. As the system is used more and more, additional data will be gathered, and the system will be fine-tuned to improve its performance. The goal is to create a chatbot system that is accurate, efficient, and user-friendly, and that can meet the needs of a wide range of users.

### Deployment

In the deployment phase, the trained model is deployed in a production environment to interact with users through the chatbot interface. This involves integrating the model into the chatbot platform and configuring it to receive input from users and generate appropriate responses. The deployment process also includes testing the model in a real-world setting to ensure that it performs well and meets the desired performance metrics. Ongoing monitoring and maintenance of the deployed model is also important to ensure its continued performance and accuracy. This may involve updating the model periodically with new data or retraining it with updated algorithms to improve its performance.