**ECC Technical Report**

Name: Rishwanth Mithra

University Mail ID: [rm23445@essex.ac.uk](mailto:rm23445@essex.ac.uk)

Personal Mail ID: [rishwanth.mithra22@gmail.com](mailto:rishwanth.mithra22@gmail.com)

Phone no: 07424809676

Contents

[Data Collection: 3](#_Toc170214714)

[Data Pre-processing: 3](#_Toc170214715)

[Building LDA Model for Topic Modelling: 4](#_Toc170214716)

[Abstractive Text Summarization : 4](#_Toc170214717)

# 

# Data Collection:

The initial step before cleaning the data is collecting it. For this report, we gathered the data from the online survey hosted by Essex County Council using Citizen Space, which allowed respondents to indicate whether they agreed or disagreed with each part of the draft strategy. Respondents were also provided with the opportunity to provide comments on each part of the draft strategy. The collected data typically includes raw text, which needs to be pre-processed.

# Data Pre-processing:

* **Dropping NaN values:** If the missing values are substantial and could significantly affect the analysis, we can choose to drop the rows or columns containing these values.
* **Lower Casing:** Converting text to lowercase is a fundamental step in text preprocessing text analysis. It ensures uniformity and consistency in the text data by eliminating variations due to case differences.
* **Remove Special Characters:** Special characters in text data can introduce noise and affect the performance of analysis models. Removing these characters helps in cleaning the text data, making it more suitable for analysis.
* **Removing stopwords:** Stopwords are common words such as "is", "and", "the", etc., that often do not contribute significantly to the meaning of the text. Removing these words helps in focusing on the more meaningful words. Stopwords can be identified using predefined lists provided by natural language processing (NLP) libraries such as NLTK, SpaCy, or custom lists. Using the NLTK library, we can easily remove stopwords from our text data.
* **Lemmatization:** Lemmatization is the process of converting words to their base or dictionary form (lemma). Unlike stemming, which may simply cut off prefixes or suffixes, lemmatization considers the context and converts the word to its meaningful base form. After lemmatization, it is important to verify that the text data has been properly lemmatized and that words are in their base form.
* **Vectorization:** Vectorization is a crucial step in preprocessing text data. It transforms text into numerical representations that can be fed into machine learning models. By integrating vectorization using Gensim's corpora module into the preprocessing pipeline, we can ensure that the text data is ready for analysis and that the models can effectively learn from the data.

# Building LDA Model for Topic Modelling:

* LDA (Latent Dirichlet Allocation) is indeed a powerful tool used for topic modeling in Natural Language Processing (NLP).
* We added the preprocessed data, known as the corpus, to the LDA model. LDA generates probabilities for the words, using which the topics are formed, and eventually, the topics are classified into documents.
* Based on these word-topic assignments, LDA refines its understanding of the topics themselves, adjusting the word probabilities within each topic
* Specifying the optimal number of topics can be challenging. We have used number of topics to be 15 because too few might miss important themes, while too many could lead to overly granular or uninterpretable topics.
* Overall, LDA is a valuable tool for uncovering latent thematic structures within a corpus.

# Abstractive Text Summarization:

* We pre-processed the raw data again by performing all the mentioned techniques in Data Pre-processing, and finally we tokenized the data in the form of sentence for text summarization.
* Using the topic modelling part, we tested all the sentences to determine the best score for a particular topic and assigned all the relevant sentences into a string for a particular topic.
* We have used pipeline function from transformers library which allows you to create a pre-configured object to perform a specific NLP task. pipeline function helps to summarize a piece of text into a shorter version that captures the main points.
* By using the pipeline function, we summarized the sentences to obtain a short summary.
* To further summarize the short summary into one action point, we have used an API Call method. Open llama 7B model from HuggingFace repository helps to call the application without using API tokens.
* We have developed a prompt by passing the template to summarize the text to one action point and produced the result.