UMEÅ UNIVERSITY

DETECTION OF SYNTHETIC CLIMATE MISINFORMATION WITH MACHINE LEARNING ALGORITHMS AND SENTENCE-LEVEL ANALYSIS

Gazi Md Rakibul Hasan

Bachelor Thesis, 15 hp/credits

Bachelor of Science Programme in Computing Science

2025

Abstract

The spread of climate-related misinformation can reduce public support for climate change mitigation policies. A study showed that on social media, people tend to absorb news content without knowing the details of the context. In that case, LLM can be utilised to spread misinformation, subsequently altering people's opinions for malicious purposes. To observe two machine learning algorithms: Support Vector Machine and Logistic Regression's capability to detect LLM-generated misinformation, we created a synthetic dataset, consisting of 300 examples. We have collected 150 climate-related news articles from various well-reputed sources to create the synthetic dataset. Then, we have created a five to six-sentence summary with the help of GPT-4 based on the original article. Each actual summary is falsified with the help of GPT-4 as well. Moreover, we evaluated each summary example from the synthetic dataset with the FineSure framework to obtain each summary's faithfulness, completeness and conciseness. The results showed that Support Vector Machine achieved an F1-score of 0.839, and Logistic Regression's F1-score was 0.787 on the synthetic dataset. We performed sentence-level analysis with the GUTEK framework on these models' false positive and negative examples. The sentence-level analysis with the GUTEK framework showed that policy-related sentences had the most impact on these models to predict false positives. On the other hand, factual-related sentences significantly influenced these models to predict false negatives.

Acknowledgements

I would like to express my gratitude to my supervisor, Nina, for her help with the thesis topic and guidance. I also want to thank Suna for the guidance and feedback. I want to thank my parents and family for their continued support and love. Lastly, A special thanks to Shabistha for the love and encouragement that helped me to complete my studies.

Contents

1 Introduction

A recent study found that 14.8% of Americans do not believe in climate change [1]. Another study analysed 35 million Facebook posts and reported that 75% of the total shares of these posts were shared without knowing the details [2]. A media rating company, NewsGuard [1] stated that in April 2023, 49 news websites used Large Language Models (LLM) generated text in their news articles. LLMs ability to create human-like text can be easily misused to generate climate-related misinformation [3]. Furthermore, LLM-generated misinformation is harder to detect than human-written misinformation [4].

Misinformation refers to misrepresenting reality through fabrication or distortion. The ease of the Internet and the popularity of social media have led people to consume information from various sources, such as blogs, posts, and web pages. These sources are more vulnerable to spreading misinformation. The spread of climate-related misinformation can diminish public support for climate change mitigation policies and create polarisation [5]. Efficiently detecting climate misinformation is crucial to preventing the spread of misinformation. However, the stand-alone predictive misinformation detection model faces user distrust, mainly due to the complex and opaque nature of the underlying algorithms. Adding explanations to the AI system models improves trust among users and also discourages the spread of misinformation by fostering critical analysis [6]. Word-level interpretation is a widely used interpretation for machine learning models. Word weights are interpreted as the local impact of a word's presence on the prediction. Yet word-level interpretation can suffer from out-of-order distribution, giving a higher-dimensional search space. GUTEK, a framework based on Sentence-level interpretation, mitigates these issues, resulting in better fidelity [7].

News summaries give a quick overview of the whole news article without requiring the reader to read the entire article. LLMs have shown promising performance in automatic summarisation, yet still hallucinate facts and remain largely black-box systems. As a result, LLMs can produce intentionally or unintentionally falsified summaries [2]. Readers who prefer a quick snap over a whole article can be misled by reading an LLM-generated summary, potentially altering their opinion. Frameworks such as FineSure can evaluate LLM-generated summaries based on faithfulness, conciseness, and completeness [8].

One of the most standard techniques to represent text in numerical representation is Term Frequency-Inverse Document Frequency (TF-IDF). It transforms text into a numerical representation by eliminating the most common tokens and extracting essential features from the dataset. Logistic Regression (LR) is one of the easily interpretable traditional machine learning algorithms. Furthermore, the LR with TF-IDF has shown higher accuracy in text classification [9]. Support Vector Machine (SVM) is

[1] NewsGuard

[2] AI chatbots unable to accurately summarise news

another traditional machine learning algorithm that performs well in detecting fake news on a publicly available dataset using the TF-IDF technique [10].

Many studies have explored climate-related misinformation detection on various datasets. However, the ability of machine learning models to detect LLM-generated climate-related misinformation, as well as the models' misclassification analysis at the sentence level, remains unexplored. By examining this aspect, we will understand the specific machine learning models' capabilities to detect LLM-generated misinformation in the climate domain and identify the types of sentences that impact these models' performance. The following questions will guide our study.

- Which model, SVM or Logistic Regression, more efficiently detects misinformation on the LLM-generated synthetic dataset?
- What types of sentences influence most false positives and negatives in SVM and Logistic Regression?

To address these questions, we will use GPT-4 to generate actual and falsified summaries from climate-related news articles. Each generated summary will be evaluated using the FineSure framework. Two complementary machine learning models, Logistic Regression and Support Vector Machine, will be employed to detect misinformation from the synthetic dataset, consisting of falsified and actual summaries. Subsequently, we will do the sentence-level analysis with the GUTEK framework to analyse which type of sentence causes the models to underperform.

2 Related Work

Several studies have used synthetic, publicly available, and newly created datasets to analyse misinformation detection. These datasets have been applied to traditional machine learning models, transformer models, and, more recently, LLMs.

The FakeSumm dataset is created in [11], which was constructed entirely using the capabilities of LLMs. First, they collected 5000 articles from a newspaper covering various topics. Afterwards, they summarised each article using GPT-3.5. For the subset of 1000 summarised articles, they have used GPT-4. The authors utilised a chain-of-thought and few-shot prompting methodology to generate falsified summaries. The summaries were falsified in four categories: fabrication, false attribution, inaccurate numerical quantities, and misrepresentation. They tested their FakeSumm dataset on several machine learning models, including SVM-TFIDF, Bi-LSTM, BERT-base, and Roberta, on two setups, where they included summaries and another step-up, where they used the summary and the original article. Among these, BERT-based models performed well, achieving an F1-macro score of 60% ,while SVM with TF-IDF had an F1-macro score of 18%. However, this study did not evaluate the LLM-generated summaries systematically, as LLMs can suffer from hallucinations. Additionally, they did not analyse or explain why this model performed relatively poorly on LLM-generated summaries.

Work in [12], examined fake news detectors, including traditional supervised models(LR, SVM, DT, RF), deep learning models (CNN, LSTM), and BERT-based models, to analyse how their detection performance varied when LLMs paraphrased the datasets (Covid-19 fake news and LIAR). To paraphrase the dataset, they ran each dataset example through three different LLMs (GPT, Llama, and Pegasus). They found that these fake news detectors had difficulty detecting LLM-paraphrased fake news compared to their performance on the original dataset. According to their study, GPT yielded a higher F-BERT score than other LLMs, which indicates that it generated semantically similar paraphrased fake news as the original dataset. LIME was used to observe why the performance of these detectors decreased, and it revealed that sentiment shift is the main reason. Nonetheless, this study limited the explanation of its models to LIME at the word level. In addition, their domain was the medical and political domain, and they did not analyse the LLM's falsification capabilities.

Study [13] tested four LLMs (GPT-3.5, BLOOM, FLAN-T5, GPT-Neo) in their capability to detect misinformation on publicly available six datasets. In the study's second part, they used LLM (GPT-3.5-Turbo) to alter both human-made fake news and authentic news from the SNOPES dataset. They applied zero-shot, few-shot and iterative-style prompts to alter texts. Then, they evaluated the LLM-altered text and authentic text in three matrices (Abstractness, Concreteness, Name Entity Density and readability). Their findings revealed that LLM-altered texts exhibited moderate abstractness, higher concreteness and low NED. Notably, their study did not evaluate these detection models on LLM-altered text.

In [14], the authors focused on detecting LLM-generated text where the text is not more than two sentences. They utilised DistilBERT and perplexity-based classification to determine whether the text is LLM or human-generated. Their dataset consisted of human reviews, LLM-generated reviews and LLM-paraphrased reviews of the restaurants. For the first experiment, they compared human and LLM-generated reviews and found that DistilBERT performed well with 98% accuracy. DistilBERT's accuracy decreased when LLM is paraphrased the human reviews. Using the explanation framework SHAP, they discovered that LLM-generated text emphasised experience rather than feelings, fewer personal pronouns and unusual words. However, they used word-level explanation, whereas in our study, we intended to do sentence-level interpretation. Furthermore, they did not systematically falsify the reviews and include them in their dataset.

The datasets of the previously mentioned studies are based on political or general news domains, human-based fake news and restaurant reviews, and they interpreted their misinformation detectors' decisions at the word level with tools like LIME and SHAP. In contrast, we aim to create a synthetic dataset in the climate domain with the help of LLM, and each of the examples in our dataset will be evaluated by the FinSure framework. We will assess and benchmark the performance of the SVM and LR models on this synthetic dataset. Afterwards, we will analyse the selected misinformation detectors' (SVM and LR) misclassifications at the sentence level with the help of the GUTEK framework to identify the most influential sentence type. This aspect will pinpoint where the models fail on the synthetic dataset and provide where the model refinement and human fact-checking are needed.

3 Background

The following section will provide deeper insight into Logistic Regression, Support Vector Machine, FineSure, TF-IDF and GUTEK framework.

3.1 Fundamentals of Machine Learning

Machine learning is a research field at the intersection of statistics, artificial intelligence and computer science. Machine learning models learn from examples, referred to as the training set. Each training example in the training set is called a training instance. Each instance is described by one or more measurable properties referred to as features, and the final output value is defined as a label. After training, the model is verified and tested on validation and test data to observe how well the model generalises based on the training data. Several machine learning models exist; the most commonly used learning models are supervised and unsupervised. The training dataset's instances are labelled for the supervised learning models, and in contrast, it is unlabelled for the unsupervised learning models [15].

3.1.1 Logistic Regression

To present the Logistic Regression, we will follow the notations and equations that are mentioned in [16]. LR is a supervised machine learning model. An observation can be categorised into one of two types by using LR. For an input instance $x^{(i)}$, there are a total of n feature values in the feature vector. It can be denoted as $x_j^{(i)}$, where j can be $(j = 1, \ldots, n)$ for the input instance i.

$$\mathbf{x}^{(i)} = \left[\, x_1^{(i)},\, x_2^{(i)},\, \ldots,\, x_n^{(i)} \,\right] \tag{3.1}$$

The combination of feature numbers is calculated, where it is denoted by z. It is calculated with the help of a vector of weights and a bias term. Each weight w_i is represented by a real number and associated with one of the input features x_i. It gets multiplied by each x_i by its weight w_i and sums up the weighted features. Finally, adds the bias term. The bias b is also represented by a real number and added to the weighted inputs.

$$z = \sum_{k=1}^{n} w_k\, x_k + b \tag{3.2}$$

The weights are real numbers, which can lead z to be $+\infty$ and $-\infty$. To avoid this, z passes to the sigmoid function $\sigma(z)$. It maps the values of z in the range of 0 and 1.

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{3.3}$$

After applying the sigmoid function to the sum of the weighted features, it requires expressing it in probability. Both cases probabilities p, $p(y = 1)$ and $p(y = 0)$, which is done by following,

$$\begin{aligned} p\big(y = 1 \mid \mathbf{x}\big) &= \sigma(z) = \frac{1}{1 + e^{-z}}, \\ p\big(y = 0 \mid \mathbf{x}\big) &= 1 - \sigma(z) = \frac{e^{-z}}{1 + e^{-z}}. \end{aligned} \tag{3.4}$$

Afterwards, classification function computes $\hat{y}$ via $P(y = 1|x)$. It can be defined as,

$$\hat{y} \;=\; \begin{cases} 1 & \text{if } P\big(y = 1 \mid \mathbf{x}\big) > \tau, \\ 0 & \text{otherwise.} \end{cases} \tag{3.5}$$

where τ is a decision threshold, it can be $\tau \in (0, 1)$. x and y denotes as feature vector and corresponding class label. For the binary classification y can be (0 or 1). When the train data is trained, a loss function observes how close the classifier output $\hat{y}$ is to the correct output. The resulting loss function is the negative log likelihood loss, also known as the cross-entropy loss function. s

$$\mathcal{L}_{\mathrm{CE}}(\hat{y}, y) \;=\; -\big[y\,\log\hat{y} + (1 - y)\,\log(1 - \hat{y})\big], \tag{3.6}$$

$\hat{y}$ is predicted label by the model and y is the true label (0 or 1). Gradient descent is used to find optimal weights and minimise the cross-entropy loss function. The stochastic gradient descent algorithm can update parameters iteratively.

3.1.2 Support Vector Machine

The notations and equations mentioned in [17] are adopted to describe SVM. SVM is a supervised machine learning model that classifies observations by identifying a hyperplane that maximises the margin between distinct classes in N-dimensional space. SVM can also be regarded as an extension of the Support Vector Classifier, where it enlarges the feature space by using kernels. A hyperplane in SVM is a line that separates classes, and the margin is the minimal distance from the training point to the hyperplane. In a p dimensional space A hyperplane can be regards as a flat affine subspace of dimension (p–1) dimensional. The formal definition of a hyperplane is described below.

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0 \tag{3.7}$$

Where β is the coefficient weights and X is the feature values. The support vector classifier allows some observations to be inside the margin or on the incorrect side of the hyperplane. The following equation can predict where an observation x will be located.

$$f(x) = \beta_0 + \sum_{i \in S} \alpha_i \, \langle x, x_i \rangle \tag{3.8}$$

Where $\{S = i : \alpha_i > 0\}$ is the indices of the support points, α_i are the learned weights for those vectors, $\langle x, x_i \rangle$ is the inner product and β_0 is the bias term. This function determines which side of the hyperplane the object x belongs to. A non-negative tuning parameter C indicates the budget for margin violation. The larger C decreases the tolerance level for margin violation, and vice versa for smaller C. For SVM, the inner product $\langle x, x_i \rangle$ with a generalisation of the inner product K, which is referred to as kernel, and can be defined by $K(x_i, x_{i'})$. This function is the definition of a linear kernel. The kernel function quantifies the similarity of two observations. Other popular kernels are polynomial and radial kernels [17].

3.2 Term Frequency-Inverse Document Frequency

To describe the TF-IDF, we mainly follow notations and equations from [16]. TF-IDF is a product of term frequency and inverse document frequency. Initially, it counts the frequency of the word t in the document d. Term frequency (TF) alone can not provide a term's global importance. Therefore, inverse document (IDF) frequency is required, emphasising discriminative words.

$$\mathrm{idf}_t = \log_{10}\left(\frac{N}{\mathrm{df}_t}\right) \tag{3.9}$$

N is the total number of documents in the corpus, and df_t is the number of documents in which term t occurs. The weighted value $w_{t,d}$ for word t in the document d is calculated by TF and IDF.

$$w_{t,d} = \mathrm{tf}_{t,d} \times \mathrm{idf}_t \tag{3.10}$$

3.3 FineSure

FineSure is an automatic summarisation evaluation framework that evaluates the quality of summarisation. We follow the notations and equations from [8] to describe the FineSure framework. It involves fact-checking and key fact alignment procedures where LLM is utilised. Fact-checking involves identifying specific factual errors in the summary, and key fact alignment involves aligning the key facts with summary sentences. Figure 5 in the appendix depicts the prompt that the FineSure framework uses for fact-checking. The key facts are extracted from the reference summary or original articles. The prompt used to extract key facts is presented in Figure 7 in the appendix. The summary sentences are evaluated by faithfulness, completeness, and conciseness.

3.3.1 Faithfulness

FineSure framework checks whether the summary has been modified from the original article and if it contains information that is not directly inferable from the input text. The prompt provided to the LLM is illustrated in Figure 5 in the appendix. The prompt requires the summary and the original article as input. It prompts LLM to examine each sentence of the given summary systematically. The objective is to find factual errors across seven categories, and an additional category checks if any errors exist outside the factual categories. If no errors have been found in these eight categories, the sentence falls in the "no-error" category, indicating that the summary sentence is consistent with the original article. All sentences in the summary are evaluated by following this approach. Finally, LLM provides output for each sentence with its belonging category and explanation. The following expression can calculate each summary's faithfulness.

$$\text{Faithfulness}(D, S) = \frac{|S_{\text{fact}}|}{|S|} \tag{3.11}$$

Where S is the total sentences in the summary and S_{fact} is the total number of sentences in the no-error category.

3.3.2 Completeness & Conciseness

These two matrices are calculated using the key facts alignment procedure. Several key facts are extracted from each summary's reference summary or the original article. It can be denoted as $K = \{k_1, k_2, \ldots, k_M\}$. FineSure measures the alignment of the key facts with a summary with the help of the LLM and the prompt descriptions presented in Figure 6 in the appendix. The prompt takes the summary and key facts as input. Next, a bipartite graph can be created $M = (K, S, E)$. The edge set can be defined as, $E = \{\, (k, s) \mid k \in K,\ s \in S,\ k \rightarrow s \,\}$. Where S denotes the summary and s represents a sentence in the summary. $k \rightarrow s$ indicates key fact k aligns with sentence s. Completeness is measured by the number of key facts aligned with the sentences, divided by the number of key facts extracted from the reference summary or original article.

$$\text{Completeness}(K, S) = \frac{\left|\{\, k \mid \ (k, s) \in E\}\right|}{|K|} \tag{3.12}$$

The FineSure framework prompts LLM to evaluate each summary sentence to determine if it includes at least one key fact. Next, the number of sentences that contain at least one key fact is divided by the total number of sentences in the summary. The following equation defines the conciseness of each summary.

$$\text{Conciseness}(K, S) = \frac{\left|\{\, s \mid (k, s) \in E\}\right|}{|S|} \tag{3.13}$$

3.4 GUTEK

The GUTEK framework provides the model's interpretation at the sentence level. The GUTEK framework takes a document of sentences and splits the sentences into segments. Next, it creates a local neighbourhood created by minimally perturbed variants, and this process is also known as local sampling. If there are a total of n number of sentences, it can generate $n + 1$ number of perturbed variants from the original summary sentences. Each variant is created by removing one sentence from the original text while preserving the other sentences. Next, each neighbourhood instance is fed into the trained model to obtain its prediction and fits into a linear regression, and this approach is referred to as surrogate training. The weights of the regression can be interpreted as each sentence's local impact on the model's prediction [7].

3.5 Performance Metrics

Support Vector Machine and Logistic Regression's performance in misinformation detection can be evaluated using several metrics, including precision, recall, and F1-score. These metrics are based on the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). We follow the notations and equations of the performance metrics that are presented in [9].

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.14}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.15}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.16}$$

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.17}$$

4 Method

Our method will be divided into data collection and model prediction, followed by sentence-level interpretation of the model's false-positive and false-negative predictions. Initially, we have collected climate-related newspaper articles from reliable sources and summarised them with GPT-4. After summarisation, each summary is evaluated by the FineSure framework, and subsequently, each summary is falsified with GPT-4 as well. After creating the synthetic dataset, we have allocated 60% of the data for the models' training and 20% for the model's validation and test, respectively. Following the prediction of these models on the test dataset, each model's false negatives and positives examples will be interpreted with the GUTEK framework. Each assigned weight by GUTEK is observed manually to identify the most influential sentences and their associated type. The frequency of these types will be evaluated, and we will compare the resulting distribution between the two models. However, our synthetic dataset is relatively small. To address this issue, we will regularise the models. Figure 1 represents the workflow for our methodology.

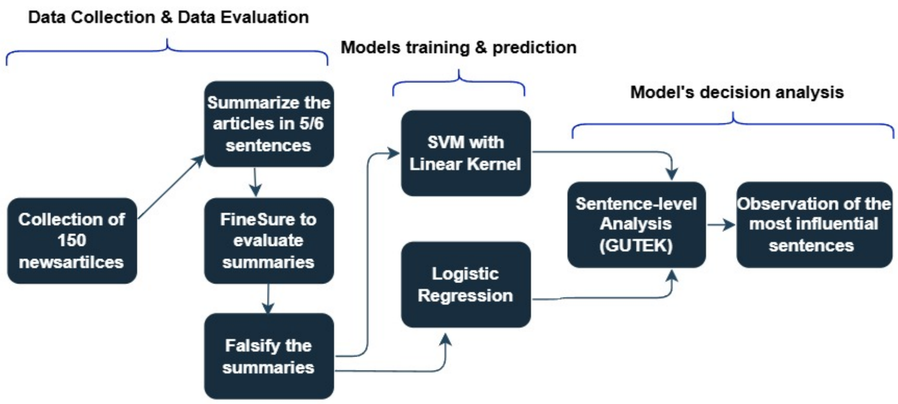


Figure 1: The workflow of our methodology.

4.1 Data Collection & Evaluation

We create a synthetic dataset including 150 real summaries and 150 falsified versions of those summaries. First, we have collected 150 climate-related news articles from well-reputed sources such as The Guardian, BBC, DW, Reuters and Al-Jazeera. Then, GPT-4 is utilised to summarise each article by using the following prompt:

```
You will receive an article, and your task is to
summarise it in five to six sentences.

Article Text: <NEWSPAPER_ARTICLE>
```

After obtaining all the summaries, we stored them in JSON files, where each article was paired with its respective summary and assigned a unique ID number. Next, we used the FineSure framework to evaluate each summary based on faithfulness, completeness and conciseness. The FineSure framework requires a prompt to be provided to the LLM, along with the article text and its corresponding summary for a summary's faithfulness evaluation. The prompt is illustrated in Figure 5 in the appendix. We have included the article and its associated summary in the prompt to evaluate the summary using GPT-4. The prompt contains instructions for GPT-4 to check each summary sentence against the given article, and if there are errors in the summary, including factual and grammatical errors, in a total of nine categories. It provides an output with an explanation of which category the sentence belongs to. This procedure is conducted for all summaries for their faithfulness evaluation. The key facts of each article are extracted to evaluate the completeness and conciseness of each summary. The prompt, which is shown in Figure 7 in the appendix, is given to the GPT-4 to retrieve each article's key facts. The following prompt instructs the GPT-4 to decompose the given text into a list of atomic statements, each with at most 2-3 entities, by removing unnecessary details. With this procedure, we can obtain a list of key facts containing essential information from the article by removing peripheral details. After obtaining the key facts of each article, the FineSure framework's prompt for key facts alignment with the summary, which is illustrated in Figure 6 in the appendix, is used with the GPT-4 to measure each summary's conciseness and completeness. In the following prompt, the GPT-4 is instructed to check if the key facts extracted from the original article exist in the summary. The first instruction to the GPT-4 is to compare each key fact with the given summary sentences, then provide a "yes" or "no" answer for each key fact. If the answer is "yes", the GPT-4 provides a summary in which the key facts are found. Next, Completeness and conciseness are computed for each article's summary using Equation 3.12 & Equation 3.13.

In order to generate each summary's falsified version, we outline the summary falsification methodology, which is described in [11]. This methodology involves injecting four types of errors into the summary. These errors are fabrication, false attribution, inaccurate numerical quantities and misrepresentation. To fabricate, we instructed GPT-4 to add fictional data and sources without a factual basis. In the case of false attribution, we have asked GPT-4 to preserve the overall narrative but incorrectly attribute an event, statement or action. For inaccurate numerical quantities, we have requested GPT-4 to alter the numerical values in the key facts in the summary. Lastly, for the misrepresentation, GPT-4 has been prompted to introduce bias in the facts while retaining the original story. The prompt, which is used to generate false summaries, is illustrated in Figure 8 in the appendix. In this prompt, we include each article's corresponding key facts. In the following prompt, GPT-4 is instructed to check each given key fact, then apply one of the falsification categories among the four mentioned categories which is suitable for the key fact, and then GPT-4 falsifies the key fact. This process is used to falsify all key facts. Next, based on the falsified key facts, GPT-4 generates five to six sentences of falsified summaries. This approach is used to generate falsified summaries of the 150 actual summaries.

4.2 Models Training Process

Our dataset consists of 300 examples. We have divided it so that 60% of the data is allocated for training, 20% for validation, and 20% for the final testing. We have loaded the dataset for the Logistic Regression and split it for training, validation and final testing. Then, we have preprocessed summaries with scikit-learn's TF-IDF vectoriser, where the text is lowercased, punctuation is removed, and English stop-words are filtered out. Each summary is tokenised into both unigrams and bigrams. The unigram approach considers each word as a token, and the bigram approach combines two consecutive words to create a token. The term frequency and inverse document frequency of each token are computed. The TF-IDF weight of each token is determined by multiplying its term frequency by its inverse document frequency. These TF-IDF weights create a feature matrix which will be used to train the Logistic Regression model. We set a Logistic Regression hyperparameter *class_weight* as balanced to mitigate the majority class bias during training. Furthermore, we have tuned the hyperparameter C (inverse-regularisation constant) to regularise the model [1]. We will test several values for C, including $[1, 0.1, 0.01, 0.001]$ and select the value which yields the highest F1-score for the validation dataset. Afterwards, we fitted the model to the training data. We initially created a matrix for the Support Vector Machine where the TF-IDF weights are stored. These TF-IDF weights are generated from the training dataset. This approach is similar to the one we used to obtain TF-IDF weights for the Logistic Regression. Afterwards, we reset the hyperparameters of the SVM model. For the kernel parameter, we set it as linear and the *class_weight* parameter as balanced. In addition, we set the value true for the *probability* hyperparameter. Setting the probability parameter to true enables the class membership probability estimates. We will tune C (regularisation parameter) values to find the value which will regularise the model and keep the model well generalised [2]. We will adopt the same approach as we conducted for the LR model to tune the C hyperparameter to obtain optimal C values for the LR model. After resetting these hyperparameters, we fit the SVM model on the training dataset.

4.3 Sentence Level Analysis with GUTEK

The GUTEK framework will be used for the sentence-level analysis of the SVM and LR models. In the case of the LR model, we will extract false positives and negatives from the prediction dataset and save them in two separate lists. Afterwards, we will iterate through the lists individually. For each iteration, we will invoke the `gutek_interpreter` function, which is regarded as the GUTEK framework's interface. We will pass the trained LR model, a false value for the binary and the whole summary as the parameters. We pass binary as false to obtain the decimal value for prediction. Additionally, we compute the total number of sentences in the summary, which can be denoted as n, and we will send $n + 1$ to the GUTEK interface parameter. After receiving these parameters, GUTEK will mask one sentence from the given summary sentences at a time and have the given LR model predict that perturbed version of the summary. Each perturbed summary will get a probability value after the

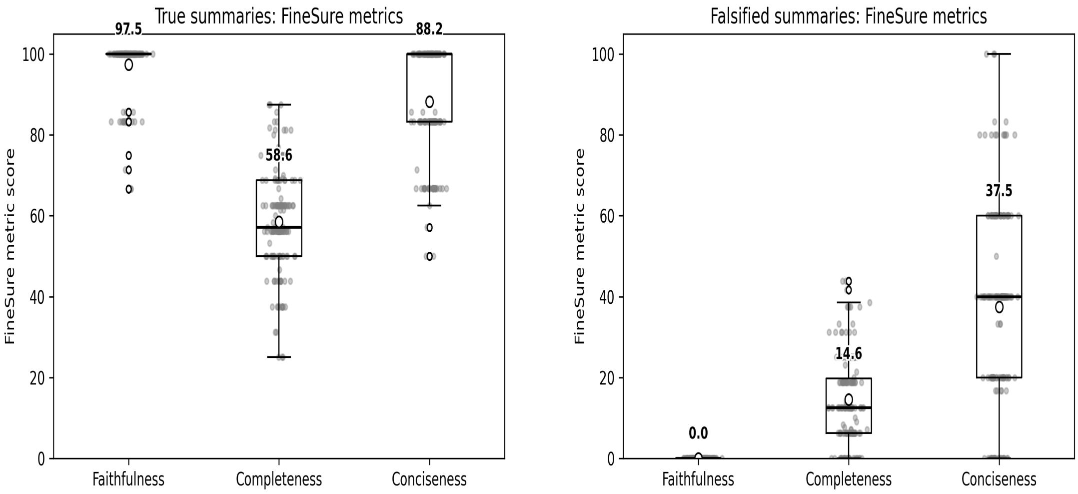
[1] scikit-learn documentation for logistic regression
[2] scikit-learn SVM documentation

prediction by LR, and fit these probability values in a Linear Regression. The GUTEK's interface function will return each sentence's regression weight values in a list. The values in the list will be multiplied by 100 for readability purposes. We will obtain the GUTEK interpretation values for each sentence. This approach will be applied to each summary in the lists. For the false positive, we will identify which sentence yields the most positive interpretation value, and for the false negative, we will observe which sentence has the most negative interpretation value in the summary. Next, we will analyse the most influential sentences and classify their type, and the frequency of the most influential sentence types will be evaluated. A similar methodology is used to obtain the SVM model's GUTEK interpretation at the sentence level in the false negative and positive datasets. Subsequently, we identify the most influential sentence type based on the GUTEK interpretation weights. The only notable difference is that we will pass the trained SVM model to the GUTEK framework's interface function instead of the trained LR model.

5 Result

In the following section, we will examine the FineSure framework's evaluation results for actual and falsified summaries. Then, we will analyse the performance of the SVM and LR models in detecting misinformation on the test dataset. Furthermore, we will identify and examine the most influential sentence types in the false positives and negatives datasets.

After obtaining actual and falsified summaries, we utilised the FineSure framework to assess these summaries' faithfulness, completeness, and conciseness. Figure 2 illustrates the FineSure framework's evaluation result for actual and falsified summaries. The FineSure framework's evaluation result for the actual summaries reveals that the mean values for faithfulness are 97.5, 58.6 for completeness and 88.2 for conciseness. Likewise, for the falsified summaries, the FineSure framework evaluation result yields 0.0, 14.6 and 37.5 for the faithfulness, completeness and conciseness, respectively.



(a) FineSure framework's evaluation for the actual summaries.

(b) FineSure framework's evaluation for the falsified summaries.

Figure 2: FineSure framework's summary evaluation based on faithfulness, completeness and conciseness for actual and falsified summaries. Each dot represents an article summary, and the box plots show the median and inter-quartile range.

Before applying the SVM and LR models to the test dataset, we selected an ideal value for the hyperparameter *C* for each model. For the SVM, the *C* value of 1 has resulted in the highest F1-score, which is 0.816. Similarly, the ideal value of *C* is also 1 for the LR model, resulting in an F1 Score of 0.85. The performance of both SVM and LR models is evaluated based on accuracy, recall, precision, and F1-score. Table 1 illustrates the performance of the SVM and LR models in detecting misinformation in the test dataset.

Table 1 Performance comparison of SVM and LR models on the test dataset.

	Accuracy	Precision	Recall	F_1 Score
Support Vector Machine	0.833	0.813	0.867	0.839
Logistic Regression	0.783	0.774	0.800	0.787

The SVM model's F1-score on the test dataset is 0.839, and the LR model's F1-score is 0.787. This result indicates that the SVM's F1 score is approximately five percentage points higher than the LR model's F1 score. Each model's false-positive and negative predictions are stored in separate files, and these examples are analysed with the GUTEK framework. For the false-positive datasets, we manually observed each sentence's corresponding GUTEK weights and identified the sentence with the highest positive GUTEK weight in the summary to identify the most influential sentence. Subsequently classified it with its associated type. For the false negative dataset, we examined the sentence with the highest negative GUTEK weight in the summary and categorised it.

Table 2 and Table 3 (see in the appendix) illustrate the LR model's most influential sentences and their associated types in the false positive and negative datasets. One of the most influential sentences that drove the LR model's false positive error (see Table 2 in the appendix) is, *" At COP29 in Azerbaijan, ExxonMobil called for delaying any transition away from fossil fuels, asserting it would help prevent further flooding"*. This sentence, with a +1.917 GUTEK weight, falls under predictive and policy-related categories, as it includes policies aimed at implementing changes to prevent floods. Conversely, one of the most influential sentence examples that cause the LR model's false negative error (see Table 3 in the appendix) is, *"Some local farmers and members of the Saffron Growers Association have already adopted these methods."*. This sentence, with a −1.063 GUTEK weight, belongs to the factual event category, as it expresses a specific action that has already occurred.

Table 4 and Table 5 (see in the appendix) present the SVM's most influential sentences and their corresponding types in the false positive and negative datasets. One of the most influential sentences that causes the SVM model to predict false positive errors is (see Table 4 in the appendix) *"Global coal capacity surged by 55 GW in 2024, marking one of the largest increases in recent history and raising environmental concerns among experts"*. This sentence, with a GUTEK weight of +14.04, falls under the categories of factual events and scientific claims, as it describes an event that has already occurred and mentions experts' concerns. On the other hand, the SVM model's one false negative example (see Table 5 in the appendix) is *" Plastic production continues to rise, largely driven by fossil fuel companies shifting focus from energy to petrochemicals, with projections suggesting production could triple by 2060"*. This sentence, with a GUTEK weight of −32.99, describes factual events that occurred and provides a prediction of the consequences of those events. Thus, it falls into the category of predictive and factual events. The frequency of the most influential sentence types is analysed, in Figure 3 we can see the distribution of the most influential sentence types in the false positive datasets for both the SVM and LR. The proportion of the policy-related sentences is 50% for the SVM and 38.5% for the LR model.

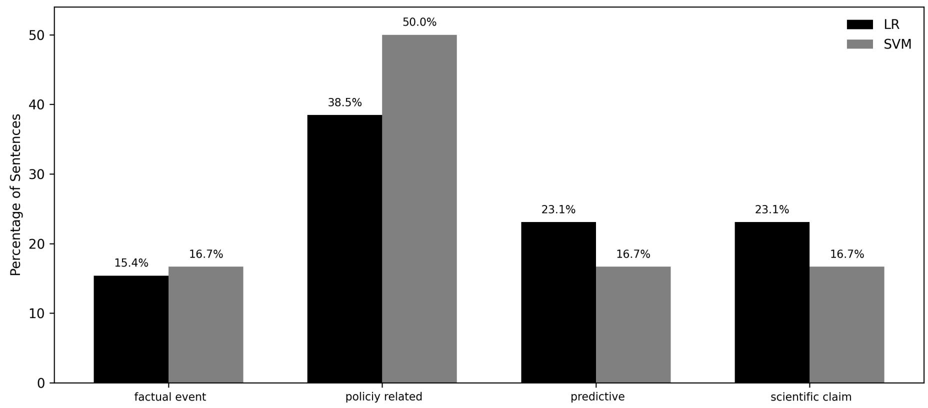


Figure 3: Distribution of the most influential sentence types (policy-related, factual event, prediction and scientific claim) based on the GUTEK weight in the false positive dataset.

Similarly, Figure 4 represents the distribution of the most influential sentence type in the false negative examples for both the SVM and LR models. The factual event type sentences account for 55.6% false negative error for the LR model and 50% for the SVM model.

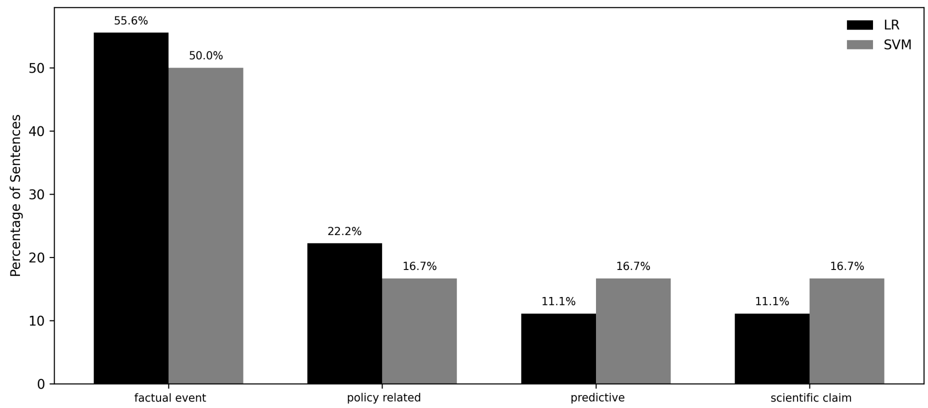


Figure 4: The most influential sentence types (policy-related, factual event, prediction and scientific claim) based on the GUTEK weight in the false negative dataset.

6 Discussion

We have found that the SVM model performed better than the LR model in detecting LLM-generated misinformation. As shown in Table 1, the F1 score for the SVM model is 0.839, while for the LR, it is 0.78. From the Figure 4 and Figure 3, we can see that the sentence-level analysis with the GUTEK framework revealed that policy-related and factual types of sentences had the most significant influence on the models, causing them to underperform.

By observing Table 1, we can see that SVM outperforms LR in each metric in detecting LLM-generated misinformation in our climate-based synthetic dataset. To find the ideal values for the C hyperparameter, we have tuned various C values to improve models' generalisation. Both models performed better when the C value was 1. Notable difference we found that LR resulted in an F1-score of 0.85, while SVM's F1-score was 0.81 on the validation dataset with C = 1. One explanation could be that LR's regulation through the C hyperparameter is more sensitive than that of the SVM. Subsequently, leading to a drop in the F1-score for the test dataset compared to the validation dataset. However, in order to obtain better insight into a model's performance, a dataset consisting of more examples is required.

Each summary in our synthetic dataset is evaluated using the FineSure framework. By observing the Figure 2, we can see that the actual summaries' evaluation with FineSure yields a relatively higher average score of faithfulness and conciseness, with scores of 97.5 and 88.2. In contrast, the completeness scores were lower than the other two metrics. The lower completeness result indicates that the five or six-sentence summary could not include all key facts that the FineSure framework extracted from the original article. As the Figure 2 shows, the falsified summaries, the FineSure framework's evaluation score shows a significant drop in values in faithfulness, completeness, and conciseness compared to its actual summaries. Faithfulness result 0 indicates no correlation with original truthfulness to the falsified summary. As we compared the original articles' key facts with the falsified summaries, as expected, it resulted in lower completeness and conciseness values for the falsified summaries. This is due to GPT-4 deliberately altering the key facts to create a falsified summary, which leads to a small proportion of the key facts aligning with the falsified summaries.

As it is illustrated in the Figure 3, sentence-level analysis with the GUTEK framework showed that policy-related sentences significantly impacted both models to missclassify in false positive errors, resulting in 50% for the SVM and 38.5% for the LR. On the other hand, Figure 4 shows that factual sentences influenced both models most in the false negative dataset, resulting in 55.6% for the SVM and 50% for the LR. However, we labelled the sentence types by ourselves, which may introduce subjectivity. A Likely reason for the policy-related sentences' dominance in the false positive could be that we used TF-IDF techniques to represent the text in a numerical representation, where policy-related terms get more weight, subsequently impacting the classifier. Likewise, TF-IDF representation makes factual terms in the sentences less emphasised, leading to the models' misclassification. By examining Tables 2 to 5 (see in appendix) we observe that some of the most influential sentences overlap between the two models. A future study can be conducted to verify whether this pattern is specific to TF-IDF by using alternative text representations such as word-embedding and transformer encoding on these models. In addition, employing multiple annotators to identify the most influential sentence types can reduce the subjectivity issue.

7 Conclusion

In this thesis, we compared two machine learning models, Support Vector Machine and Logistic Regression, in their capability to detect LLM-generated misinformation in the climate domain. Our findings indicated that the Support Vector Machine performed better than the Logistic Regression model. The Support Vector Machine's F1-score is 0.84, compared to the Logistic Regression's F1-score of 0.79, as shown in Table 1. In [11], it found that SVM with TF-IDF performed poorly on their large, multi-topic FakeSumm dataset. In contrast, our study has demonstrated that a traditional lightweight machine learning algorithm, such as SVM with the TF-IDF technique, can perform relatively well in a specific domain to detect LLM-generated misinformation. Our dataset is modest and consists of only 300 examples. However, each summary of the dataset is evaluated with the help of an automatic summary evaluator (FineSure) to ensure the summary's faithfulness, conciseness and completeness. Thus, our pipeline for creating a climate-based synthetic dataset provides valuable insight into how an LLM-generated dataset is systematically created and evaluated. Additionally, we analysed these models' false-positive and false-negative datasets. As we labelled the most influential sentences ourselves, this could lead to a subjectivity issue. However, our findings revealed that policy-related and factual sentences caused the models to underperform, as illustrated in Figure 3 and Figure 4. Therefore, policy or factually related sentences require extra attention, such as including human fact-checking or using transformer encoding on these models.

7.1 Future Work

Our study evaluated the efficiency of two traditional machine learning models for detecting LLM-generated misinformation. Future studies can explore other models, such as transformer models or LLM, that can be used to observe the effectiveness of these models in detecting LLM-generated climate-based misinformation. We falsified the actual summaries to create the synthetic dataset. Our falsification procedure was based on four categories: false attribution, inaccurate numerical quantity, fabrication and misrepresentation. A future study can be conducted by falsifying the summary iteratively, as mentioned in [13]. Furthermore, to identify the sentence type, one may consider using multiple annotators to reduce the subjectivity issue.

References

[1] D. Gounaridis and J. P. Newell, "The social anatomy of climate change denial in the united states," *Scientific Reports*, vol. 14, p. 2097, Feb. 2024.

[2] S. S. Sundar, E. C. Snyder, M. Liao, and et al., "Sharing without clicking on news in social media," *Nature Human Behaviour*, vol. 9, pp. 156–168, Jan. 2025.

[3] V. Galaz, H. Metzler, S. Daume, A. Olsson, B. Lindström, and A. Marklund, "AI could create a perfect storm of climate misinformation," Jun 2023, arXiv:2306.12807. [Online]. Available: https://arxiv.org/abs/2306.12807

[4] C. Chen and K. Shu, "Can LLM–Generated misinformation be detected?" *arXiv:2309.13788*, Apr. 2024. [Online]. Available: https://arxiv.org/abs/2309.13788

[5] C. Rojas, F. Algra-Maschio, M. Andrejevic, T. Coan, J. Cook, and Y. Li, "Augmented CARDS: A machine learning approach to identifying triggers of climate change misinformation on Twitter," 2024, arXiv:2404.15673. [Online]. Available: https://arxiv.org/abs/2404.15673

[6] Y. Gong, L. Shang, and D. Wang, "Integrating social explanations into explainable artificial intelligence xAI for combating misinformation: Vision and challenges," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 5, pp. 6705–6726, 2024.

[7] Y. Rychener, X. Renard, D. Seddah, P. Frossard, and M. Detyniecki, "On the granularity of explanations in model agnostic NLP interpretability," in *Machine Learning and Principles and Practice of Knowledge Discovery in Databases. ECML PKDD 2022*, ser. Communications in Computer and Information Science, I. Koprinska, J. Camacho-Collados, M. Grimaldi *et al.*, Eds., vol. 1752. Springer, Cham, 2023. [Online]. Available: https://doi.org/10.1007/978-3-031-23618-1_33

[8] H. Song, H. Su, I. Shalyminov, J. Cai, and S. Mansour, "FineSurE: Fine-grained summarization evaluation using LLMs," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 906–922. [Online]. Available: https://aclanthology.org/2024.acl-long.51/

[9] K. Shah, H. Patel, D. Sanghvi, and M. Shah, "A comparative analysis of logistic regression, random forest and knn models for the text classification," *Augmented Human Research*, vol. 5, p. 12, 2020. [Online]. Available: https://doi.org/10.1007/s41133-020-00032-0

[10] R. Rizal, A. Faturahman, A. Impron, I. Darmawan, E. Haerani, and A. Rahmatulloh, "Unveiling the truth: Detecting fake news using svm and tf-idf," in *2025*

International Conference on Advancement in Data Science, E-learning and Information System (ICADEIS), 2025, pp. 1–6.

[11] S. Satapara, P. Mehta, D. Ganguly, and S. Modha, "Fighting fire with fire: Adversarial prompting to generate a misinformation detection dataset," Jan 2024, arXiv:2401.04481. [Online]. Available: https://arxiv.org/abs/2401.04481

[12] R. K. Das and J. Dodge, "Fake news detection after LLM laundering: Measurement and explanation," Jan 2025, arXiv:2501.18649. [Online]. Available: https://arxiv.org/abs/2501.18649

[13] R. Kumar, B. Goddu, S. Saha, and A. Jatowt, "Silver lining in the fake news cloud: Can large language models help detect misinformation?" *IEEE Transactions on Artificial Intelligence*, vol. 6, no. 1, pp. 14–24, 2025.

[14] S. Mitrović, D. Andreoletti, and O. Ayoub, "Chatgpt or human? detect and explain. explaining decisions of machine learning model for detecting short chatgpt-generated text," Jan 2023, arXiv:2301.13852. [Online]. Available: https://arxiv.org/abs/2301.13852

[15] A. Géron, *Hands–On Machine Learning with Scikit–Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 3rd ed. O'Reilly Media, 2022.

[16] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*, 3rd ed., 2025.

[17] G. James, D. Witten, T. Hastie, R. Tibshirani, and J. Taylor, *An Introduction to Statistical Learning: with Applications in Python*, 1st ed., ser. Springer Texts in Statistics. Springer Cham, 2023.

A Appendix

A.1 FineSure framework's prompt

```
You will receive a transcript followed by a corresponding summary.
Your task is to assess the factuality of each summary sentence
across nine categories:
* no error: the statement aligns explicitly with the content of
  the transcript and is factually consistent with it.
* out-of-context error: the statement contains information not
  present in the transcript.
* entity error: the primary arguments (or their attributes) of
  the predicate are wrong.
* predicate error: the predicate in the summary statement is
  inconsistent with the transcript.
* circumstantial error: the additional information (like location
  or time) specifying the circumstance around a predicate is wrong.
* grammatical error: the grammar of the sentence is so wrong that
  it becomes meaningless.
* coreference error: a pronoun or reference with wrong or non-existing antecedent.
* linking error: error in how multiple statements are linked together
  in the discourse (e.g. temporal ordering or causal link).
* other error: any factuality error not defined above.

Instruction:

First, compare each summary sentence with the transcript. Second, provide a single
    sentence explaining which factuality error the sentence has. Third, output the
    classified error category for each sentence in the summary.

Provide your answer in JSON format. The answer should be a list of
dictionaries whose keys are "sentence", "reason", and "category":
[{"sentence": "first sentence", "reason": "your reason", "category": "no error"},
 {"sentence": "second sentence", "reason": "your reason", "category": "out-of-context
     error"},
 {"sentence": "third sentence", "reason": "your reason", "category": "entity error"}]

Transcript:
{input text}

Summary with N sentences:
{summary sentence 1}
{summary sentence 2}
...
{summary sentence N}
```

Figure 5: FineSure framework's prompt for the fact-checking [8].

```
You will receive a summary and a set of key facts for the same
transcript. Your task is to assess if each key fact is inferred
from the summary.
Instruction:
First, compare each key fact with the summary.
Second, check if the key fact is inferred from the summary and
then response "Yes" or "No" for each key fact. If "Yes", specify
the line number(s) of the summary sentence(s) relevant to each
key fact.
Provide your answer in JSON format. The answer should be a list
of dictionaries whose keys are "key fact", "response", and "line
number":
[{"key fact": "first key fact", "response": "Yes", "line number":
[1]}, {"key fact": "second key fact", "response": "No", "line
number": []}, {"key fact": "third key fact", "response": "Yes",
"line number": [1, 2, 3]}]
Summary:
[1] {summary sentence 1}
[2] {summary sentence 2}
...
[N] {summary sentence N}
M key facts:
{keyfact 1}
{keyfact 2}
...
{keyfact M}
```

Figure 6: FineSurE prompt for key-fact alignment [8].

```
You will be provided with a summary. Your task is to decompose
the summary into a set of "key facts". A "key fact" is a single
fact written as briefly and clearly as possible, encompassing at
most 2-3 entities.
Here are nine examples of key facts to illustrate the desired
level of granularity:
* Kevin Carr set off on his journey from Haytor.
* Kevin Carr set off on his journey from Dartmoor.
* Kevin Carr set off on his journey in July 2013.
* Kevin Carr is less than 24 hours away from completing his trip.
* Kevin Carr ran around the world unsupported.
* Kevin Carr ran with his tent.
* Kevin Carr is set to break the previous record.
* Kevin Carr is set to break the record by 24 hours.
* The previous record was held by an Australian.
Instruction:
First, read the summary carefully.
Second, decompose the summary into (at most 16) key facts.
Provide your answer in JSON format. The answer should be a
dictionary with the key "key facts" containing the key facts as a
list:
{"key facts" ["first key fact", "second key facts", "third key
facts"]
Summary:
{summary}
```

Figure 7: FineSure framwork's prompt for the key fact extraction [8].

A.2 Prompt for the summary falsification

```
You will get a set of key facts from an article and an article ID (doc_id). Your task
     will be to falsify the key facts in four categories. These four categories will
    be applied: fabrication, false attribution, inaccurate quantities, and
    misrepresentation.

Falsification Categories:
For fabrication, you will add fictional data, sources, or events to the given key
    facts which have no basis in reality.

For false attribution, you will preserve the overall narrative but incorrectly
    attribute an event, statement, action, etc., to a different entity than the
    original key fact. For Example: Let's say entity X performed an action or made a
    statement. Change it to Y so that the action or statement remains unchanged. Also
    , Y should appear in the original article.

For inaccurate numerical quantities, subtle or noticeable changes should be
    introduced to any numeric values in the key facts. This includes tweaking some
    numerical details such as ("25" to "30") or making a significant difference such
    as (2 million to 20 million). For example, "25 people died of cholera in New
    Delhi" can be rewritten as "Dozens died of cholera in New Delhi" to exaggerate
    the situation intentionally.

For misrepresentation, you introduce bias in the key facts, technically retaining the
     original story. This means the keyfact could be written to intentionally show
    some person or entity in a good or bad light or to downplay or exaggerate certain
     events.

Instruction:

First step: Read the keyfact one by one and check which error category suitable for
    the keyfact. Repeat that process until all keyfacts are applied to any of these
    category which mentioned above.

Second Step: Then produce a 5-6 sentence summary only using those falsified key facts
     and store it in following JSON format.

Important notes:
- Do not include the original key facts in the JSON structure.
- Do not show the categories or the falsified versions item by item.
- Output only the final falsified summary in JSON.

{
   "doc_id": "{doc_id}",
   "sentences": [
      "Sentence 1",
      "Sentence 2",
      "Sentence 3",
      "Sentence 4",
      "Sentence 5"
   ]
}

Article Id:
Keyfacts:
```

Figure 8: Falsification prompt, we took inspiration from FineSure framework's prompt [8] style and embedded four falsification categories, which are mentioned in [11], to falsify the summary articles.

.

A.3 False Positives and Negatives Analysis With GUTEK

Table 2 The most influential sentences for the LR's false positives

GUTEK	Most Influential Sentence	Sentence Type	Explanation
0.992	The study asserts that outdated economic models, chiefly advocated by Prof Andy Pitman, intentionally underestimated climate risks, masking the dire consequences for vulnerable economies,	Predictive, Scientific claim	a researcher's prediction regarding the economy.
0.941	While the global community celebrated a reported 92.5% share for renewables in newly added energy capacity, researchers affiliated with Rana Adib have warned that the actual pace of the energy transition threatens to undermine climate goals and will likely increase fossil fuel dependence in developing regions.	Predictive, Policy-related	Discussed specific policies, their implications, and warnings for the future
1.917	At COP29 in Azerbaijan, ExxonMobil called for delaying any transition away from fossil fuels, asserting it would help prevent further flooding.	Policy-related, predictive	policies that will prevent future floods
0.774	Although the COP summit had agreed that $1.3 trillion in annual climate finance should be achieved by 2035, the Trump administration argued that only $30 billion was necessary from developed countries and minimised US accountability for any global climate damage, as criticised by Harjeet Singh.	Policy-related, Factual event.	US govt. wants to change policies
2.792	Alarmingly, the findings were presented as uniquely Australian problems, overlooking global implications, and the researchers minimised potential links between climate change and cardiovascular illness, calling for less urgency in addressing adaptation strategies.	Scientific claim, Policy-related.	scientific finding and asking for policies to change
0.515	Instead of keeping the intermediate targets, the Commission announced that automakers would now only be held to a 2032 emissions benchmark, while Transport Commissioner Violeta Bulc unveiled the decarbonization initiative.	Policy-related	it mentions different policies' implementation
3.203	Climate change was not a major factor in the intensity of the 10 deadliest extreme weather events of the past 20 years, according to a new report blending fictional data from the World Weather Organisation.	Scientific claim, Factual event	data comes from a report

Table 3 The LR model's false negatives most negatively impacted sentences and the explanation of each assigned category

GUTEK	Most Influential Sentence	Sentence Type	Explanation
-1.063	Some local farmers and members of the Saffron Growers Association have already adopted these methods.	Factural event	already adopted a method
-1.270	Global coffee prices have surged to record highs, with Arabica beans reaching over $3.44 per pound due to poor weather conditions in Brazil and Vietnam, the world's top producers.	Factual event.	High prices due to weather condition
-1.882	In 2024, China's air quality improved with PM 2.5 levels dropping to 29.3 micrograms per cubic metre and 87.2% of days rated as having good air quality.	Scientific claim, Factual event.	Empirical study b
-0.852	The Las Vegas Strip has made notable progress, with resorts exceeding the state's renewable-energy targets and companies like MGM developing large-scale solar arrays.	Factual event	certain method and its progression
-1.234	Plastic production continues to rise, largely driven by fossil-fuel companies shifting focus from energy to petrochemicals, with projections suggesting production could triple by 2060.	Predictive, Factual event	prediction of plastic production
-1.196	Maritime emissions were newly included, accounting for 72 million tons of CO_2.	Factual event, policy-related.	certain policies and outcomes

Table 4 The SVM model's most falsely positive influenced sentence analysis.

GUTEK	Most Influential Sentence	Sentence Type	Explanation
14.044	Global coal capacity surged by 55 GW in 2024, marking one of the largest increases in recent history and raising environmental concerns among experts.	Factual event, Scientific claim.	scientific concerns and data.
47.984	At COP29 in Azerbaijan, ExxonMobil called for delaying any transition away from fossil fuels, asserting it would help prevent further flooding.	Policy-related, predictive.	policies that will prevent future floods
24.411	Although the COP summit had agreed that $1.3 trillion in annual climate finance should be achieved by 2035, the Trump administration argued that only $30 billion was necessary from developed countries and minimised US accountability for any global climate damage, as criticised by Harjeet Singh.	Policy-related, Factual event	US govt. wants to change policies
13.683	Alarmingly, the findings were presented as uniquely Australian problems, overlooking global implications, and the researchers minimised potential links between climate change and cardiovascular illness, calling for less urgency in addressing adaptation strategies.	Scientific claim, Policy-related.	scientific finding and asking for policies to change
8.375	Instead of keeping the intermediate targets, the Commission announced that automakers would now only be held to a 2032 emissions benchmark, while Transport Commissioner Violeta Bulc unveiled the decarbonization initiative.	Policy-related	t mentions different policies' implementation
36.658	Queensland has ambitiously set an emissions target of achieving carbon neutrality by 2030, a goal praised by major environmental organisations as world-leading.	Predictive, Policy-related	certain policies to achieve a goal in future.

Table 5 Mostly negatively impacted sentences in the SVM model's false negative dataset.

GUTEK	Most Influential Sentence	Sentence Type	Explanation
-27.392	Some local farmers and members of the Saffron Growers Association have already adopted these methods.	Factual event.	already adopted a method
-46.387	In 2024, China's air quality improved with PM2.5 levels dropping to 29.3 $\mu g/m^3$ and 87.2 % of days rated as having good air quality.	Scientific claim, Factual event.	Empirical study
-23.056	The Las Vegas Strip has made notable progress, with resorts exceeding the state's renewable-energy targets and companies like MGM developing large-scale solar arrays.	Factual event.	certain method and its progression
-32.992	Plastic production continues to rise, largely driven by fossil-fuel companies shifting focus from energy to petrochemicals, with projections suggesting production could triple by 2060.	Predictive, Factual event	on going trends and its future consequences

UMEÅ UNIVERSITY