Exploring the Impact of Text Watermarking Algorithms on Text Quality and Classification Model Performance

Youssef Mohamed

September 2024

Abstract

This report examines the impact of watermarking algorithms on both the quality of text generated by large language models and the accuracy of text classification models. The watermarking technique embeds imperceptible signals into the generated text, which can be detected algorithmically. I assess the influence of this watermarking on text quality using BLEU, METEOR, and ROUGE scores, and find that it negatively affects the readability and fluency of the text. Additionally, I evaluate the impact of watermarking on downstream text classification tasks and observe a detrimental effect on classification accuracy. Our findings indicate that the watermarking algorithm compromises both text quality and the performance of classification models, highlighting the trade-offs involved in balancing model attribution with the preservation of text integrity and classification reliability.

1 Introduction

In the era of large language models (LLMs) such as ChatGPT, the ability to generate human-like text has led to both remarkable advancements and significant challenges. One critical concern is the potential misuse of machine-generated content for malicious purposes, including misinformation and academic dishonesty. To address these issues, text watermarking has emerged as a promising solution. Text watermarking involves embedding subtle, imperceptible signals into generated text to identify and trace its origin as synthetic, without affecting its readability or usefulness.

Text watermarking plays a crucial role in mitigating the risks associated with the widespread use of LLMs. By embedding hidden markers into the text, it becomes possible to detect and attribute machine-generated content, thereby safeguarding against misuse. This technology is vital for maintaining content integrity, enforcing accountability, and preventing the proliferation of synthetic data that could undermine the authenticity of information.

Various approaches have been proposed for text watermarking, each with its own advantages and challenges. These methods generally involve manipulating token probabilities or sequences in a way that introduces detectable patterns while keeping the watermarked text indistinguishable from human-generated content. Common techniques include probabilistic methods, where certain tokens are subtly promoted during generation, and embedding techniques, which insert specific patterns into the text.

While text watermarking is essential for detection and attribution, it is crucial to assess its impact on text quality. Effective watermarking should minimally affect the readability, coherence, and overall quality of the generated text. However, introducing watermarks can potentially degrade text quality, making it necessary to evaluate the extent of this impact to balance detection capabilities with text usability.

1.1 Approaches to Measure Text Quality

To assess the impact of watermarking on text quality, several evaluation metrics are employed. These include:

- **BLEU (Bilingual Evaluation Understudy)**: Measures the precision of n-grams in the generated text against reference texts, evaluating the quality of the translation or text generation.
- **ROUGE (Recall-Oriented Understudy for Gisting Evaluation)**: Focuses on recall by comparing n-grams, word sequences, and word pairs between generated and reference texts, providing insights into text overlap and relevance.

- **METEOR (Metric for Evaluation of Translation with Explicit ORdering)**: Evaluates text quality based on precision and recall, taking into account synonyms, stemming, and word order.

These metrics help quantify how watermarking affects the text's fidelity and coherence, ensuring that the quality remains acceptable despite the presence of hidden markers.

1.2 Testing the Impact of Watermarking on Text Classification

In addition to evaluating text quality, it is essential to understand how watermarking influences downstream tasks such as text classification. This report investigates the impact of watermarking on classification accuracy using two models: T5-small and Flan-T5-small. I test these models on two datasets, XSum and CNN/DailyMail, to determine how watermarking affects their performance. The results offer insights into the trade-offs between effective watermarking and the reliability of text classification, highlighting any potential drawbacks introduced by the watermarking process.

2 Datasets

In this study, we use two prominent datasets for evaluating text summarization and classification: the XSum dataset and the CNN/DailyMail dataset. Both datasets are widely used for tasks like abstractive summarization, making them suitable for testing the impact of text watermarking on generated summaries.

2.1 XSum Dataset

The Extreme Summarization (XSum) dataset is designed for evaluating abstractive single-document summarization systems. The primary objective is to generate a short, one-sentence news summary that answers the question "What is the article about?". The dataset contains 226,711 news articles, each paired with a one-sentence summary. These articles are collected from BBC news sources (spanning from 2010 to 2017) and encompass a wide range of domains such as News, Politics, Sports, Weather, Business, Technology, Science, Health, Family, Education, and Entertainment.

The dataset is split into three parts: 204,045 documents (90%) for training, 11,332 documents (5%) for validation, and 11,334 documents (5%) for testing.

2.2 CNN/DailyMail Dataset

The CNN/DailyMail dataset is an English-language dataset composed of over 300k news articles written by journalists from CNN and the Daily Mail. The dataset supports both extractive and abstractive summarization, with its original purpose focused on machine reading comprehension and abstractive question answering.

It includes human-generated abstractive summaries created from news stories on the CNN and Daily Mail websites, often structured as questions with one entity hidden and the story used as the corresponding passage from which to derive the answer. The dataset contains 286,817 training pairs, 13,368 validation pairs, and 11,487 test pairs. On average, the source documents in the training set contain 766 words across 29.74 sentences, while the summaries have 53 words spanning 3.72 sentences.

2.3 Comparison Between Datasets

The table below provides a comparison of the XSum and CNN/DailyMail datasets based on key characteristics:

Dataset Name	Number of Records	Description
XSum	226,711	Abstractive summarization of BBC news articles into a one-sentence summary
CNN/DailyMail	311,672	Abstractive and extractive summarization of CNN and Daily Mail news stories

Table 1: Comparison of the XSum and CNN/DailyMail Datasets

3 Evaluation Metrics

In this study, we utilize three key evaluation metrics—BLEU, ROUGE, and METEOR—to assess the impact of watermarking on text quality and classification tasks. These metrics provide a quantitative measure of how closely the watermarked text resembles the original text and its performance in downstream tasks.

3.1 BLEU (Bilingual Evaluation Understudy)

BLEU is a precision-based metric used to evaluate the similarity between a generated text and one or more reference texts by comparing n-grams (sequences of n words). It computes the geometric mean of modified n-gram precisions and includes a brevity penalty to handle overly short sentences. The BLEU score is mathematically defined as:

$$\text{BLEU} = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right), \tag{1}$$

where p_n is the modified precision for n-grams of size n, w_n is the weight assigned to the n-grams (typically uniform, i.e., $w_n = \frac{1}{N}$), and BP is the brevity penalty, calculated as:

$$BP = \begin{cases} 1 & \text{if } c > r \\ \exp(1 - \frac{r}{c}) & \text{if } c \leq r \end{cases} \tag{2}$$

Here, c is the length of the candidate text, and r is the length of the reference text. The brevity penalty discourages generating overly short sentences, which could artificially inflate precision scores. BLEU measures the overlap between generated and reference text in terms of n-grams. In the context of this study, BLEU evaluates how faithfully the watermarked text retains the original content while potentially introducing slight modifications.

3.2 ROUGE (Recall-Oriented Understudy for Gisting Evaluation)

ROUGE is a recall-based metric commonly used in summarization tasks. It measures the overlap between n-grams, word sequences, or word pairs in the generated and reference texts. The ROUGE-N metric is specifically used to measure the recall of n-grams. ROUGE-N is defined as:

$$\text{ROUGE-N} = \frac{\sum_{\text{gram}_n \in \text{Ref}} \min(\text{Count}_{\text{match}}(\text{gram}_n), \text{Count}_{\text{gen}}(\text{gram}_n))}{\sum_{\text{gram}_n \in \text{Ref}} \text{Count}_{\text{Ref}}(\text{gram}_n)}, \tag{3}$$

where $\text{Count}_{\text{match}}(\text{gram}_n)$ is the number of overlapping n-grams between the generated and reference text, and $\text{Count}_{\text{Ref}}(\text{gram}_n)$ is the total number of n-grams in the reference text. ROUGE primarily measures recall, making it useful for evaluating how much of the essential content from the reference text is preserved in the watermarked text. In this study, ROUGE helps assess whether watermarking impacts the retention of key information.

3.3 METEOR (Metric for Evaluation of Translation with Explicit ORdering)

METEOR is a metric that evaluates the quality of generated text based on a combination of precision, recall, and alignment, considering factors such as stemming and synonymy. Unlike BLEU, which relies solely on n-gram matching, METEOR aligns the generated text with the reference text at the word level, penalizing word order errors. The METEOR score is computed as:

$$\text{METEOR} = F_{\text{mean}} \times (1 - \text{Penalty}), \tag{4}$$

where F_{mean} is the harmonic mean of precision and recall:

$$F_{\text{mean}} = \frac{10 \cdot P \cdot R}{R + 9 \cdot P}, \tag{5}$$

with P and R representing precision and recall, respectively. The penalty term is based on the number of chunks (uninterrupted sequences of words in the correct order) in the alignment:

$$\text{Penalty} = 0.5 \times \left(\frac{\text{chunks}}{\text{matches}} \right). \tag{6}$$

The METEOR metric goes beyond simple n-gram matching by considering synonymy and stemming, making it more robust in evaluating semantic similarity between the watermarked and original text. In this study, METEOR helps determine whether the watermarked text preserves the original meaning and fluency.

3.4 Purpose of Metrics in the Study

Each of these metrics plays a unique role in evaluating the effects of watermarking on text quality:

- **BLEU** primarily measures the preservation of lexical choices in the watermarked text, indicating how closely the watermarked text matches the original in terms of word usage.
- **ROUGE** focuses on recall, helping evaluate how well the watermarked text retains the most important content from the original text.
- **METEOR** goes beyond word matching by accounting for synonyms, stemming, and word order, providing a deeper assessment of the text's semantic coherence and fluency.

By combining these metrics, we gain a comprehensive understanding of how watermarking impacts the readability, informativeness, and overall quality of the generated text. Additionally, these metrics allow us to measure the trade-offs between effective watermarking and the reliability of downstream tasks, such as text classification.

4 Models Used

In this study, we implemented several machine learning models to assess their performance in text classification tasks. The models employed include logistic regression with n-grams (bigrams and trigrams), Multinomial Naive Bayes, K-Nearest Neighbors, and a Long Short-Term Memory (LSTM) network. Each of these models utilizes different techniques for representing and learning from text data, and they serve distinct roles in our comparative analysis.

4.1 Logistic Regression with N-Grams

Logistic regression is a linear model commonly used for binary classification tasks. In this work, we used both bigram and trigram representations of text to capture the relationships between consecutive words. The text was vectorized using the Term Frequency-Inverse Document Frequency (TF-IDF) approach, which transforms the text data into numerical feature vectors based on word frequency while penalizing common words to reduce their importance.

The logistic regression model can be described mathematically as:

$$P(y = 1|X) = \frac{1}{1 + \exp(-X \cdot \beta)}, \tag{7}$$

where $P(y = 1|X)$ is the probability of the positive class given the feature vector X, and β is the vector of coefficients learned from the training data. The model fits a linear decision boundary by learning the optimal set of weights to minimize the log-loss:

$$\text{Log-Loss} = -\frac{1}{n} \sum_{i=1}^{n} \left[y_i \log(P(y_i)) + (1 - y_i) \log(1 - P(y_i)) \right], \tag{8}$$

where y_i are the true labels and $P(y_i)$ are the predicted probabilities for each sample. Logistic regression with n-grams enables the model to capture not only individual words but also the local context of words that appear consecutively, thereby improving the classification performance on text data.

4.2 Multinomial Naive Bayes

The Multinomial Naive Bayes (MNB) model is a probabilistic classifier that assumes conditional independence between features given the class label. In text classification, MNB models the likelihood of each word in the vocabulary given a class, making it particularly well-suited for tasks where word frequency is an important feature.

The conditional probability of a class label y given the feature vector X is defined using Bayes' theorem as:

$$P(y|X) \propto P(y) \prod_{i=1}^{n} P(x_i|y), \tag{9}$$

where $P(y)$ is the prior probability of the class, and $P(x_i|y)$ is the likelihood of word x_i given class y. The model makes predictions by selecting the class that maximizes this conditional probability. MNB assumes a multinomial distribution over word counts and is particularly effective when the features are word frequencies or counts, such as those produced by TF-IDF vectorization.

4.3 K-Nearest Neighbors (KNN)

The K-Nearest Neighbors (KNN) algorithm is a non-parametric, instance-based learning method that classifies a new sample based on the majority label of its k nearest neighbors in the feature space. The distance between data points is typically measured using the Euclidean distance or other similarity measures. In this study, KNN was applied to the TF-IDF vectorized text data.

Mathematically, the distance between two samples x_i and x_j can be calculated using the Euclidean distance:

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2}. \tag{10}$$

The class of a new data point x_{new} is determined by the majority vote among the k nearest neighbors:

$$y_{\text{new}} = \arg\max_{y} \sum_{i=1}^{k} 1(y_i = y), \tag{11}$$

where $1(\cdot)$ is the indicator function that returns 1 if the condition is true. KNN is computationally efficient for small datasets and provides a flexible, interpretable approach to classification, although it can struggle with high-dimensional data.

4.4 Long Short-Term Memory (LSTM) Network

The Long Short-Term Memory (LSTM) network is a type of recurrent neural network (RNN) designed to capture long-range dependencies in sequential data, such as text. LSTM cells mitigate the vanishing gradient problem by using a series of gates—input, forget, and output gates—that control the flow of information through the network.

The state of an LSTM cell at time step t is computed as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \tag{12}$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \tag{13}$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C), \tag{14}$$
$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t, \tag{15}$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \tag{16}$$
$$h_t = o_t \odot \tanh(C_t), \tag{17}$$

where x_t is the input at time step t, h_{t-1} is the previous hidden state, C_t is the cell state, f_t, i_t, and o_t are the forget, input, and output gates, respectively, and σ is the sigmoid activation function. The LSTM model is particularly powerful for tasks that involve sequences of data, such as text, as it can learn to retain or forget information over long sequences.

In this study, we implemented an LSTM network to learn the sequence patterns in the input text. The embedding layer maps words to dense vectors, which are then processed by the LSTM layer. The final output is fed through a fully connected layer with a softmax activation function to perform classification.

4.5 Purpose of Model Comparison

By applying both traditional machine learning models (e.g., logistic regression, Naive Bayes, KNN) and deep learning models (LSTM), we aim to understand how different approaches perform on the task of text classification, particularly in the context of watermarked text. Each model brings unique strengths: logistic regression and Naive Bayes are effective for linear and probabilistic decision boundaries, KNN offers a non-parametric perspective, and LSTM can capture temporal dependencies in text sequences. Comparing these models enables us to evaluate the trade-offs between interpretability, computational cost, and classification accuracy.

5 Experiments and Results

In this section, we describe the experimental setup and the results obtained from evaluating the summaries generated by both watermarked and non-watermarked models. The evaluation metrics used to assess the quality of the generated summaries are BLEU, METEOR, ROUGE-1, ROUGE-2, and ROUGE-L. These metrics are commonly used for summarization tasks to compare the similarity between the generated summaries and the reference summaries.

5.1 Experiment Setup

The experiments were conducted using two pre-trained models: `FLAN-T5-small` and `T5-small`, each of which was tested in both its watermarked and non-watermarked versions. We performed the evaluation on two widely used text summarization datasets:

- **XSum:** A dataset for abstractive summarization that contains 226,711 news articles and corresponding one-sentence summaries.
- **CNN/DailyMail:** A dataset containing 311,672 news articles paired with multi-sentence summaries.

For each model, we generated summaries for the test sets of both datasets. The generated summaries were evaluated using the following metrics:

- **BLEU**
- **METEOR**
- **ROUGE-1, ROUGE-2, ROUGE-L**

Each experiment was repeated for both watermarked and non-watermarked models, allowing us to compare the impact of watermarking on the quality of generated summaries.

5.2 Results

We report the average scores for BLEU, METEOR, ROUGE-1, ROUGE-2, and ROUGE-L across the test sets of both datasets for all model configurations. The results are summarized in Table 2 for the XSum dataset and Table 3 for the CNN/DailyMail dataset.

Model	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	METEOR
flan-t5-small (summary)	9.87E-05	0.0989	0.0227	0.0654	0.0378
flan-t5-small (watermarked)	6.19E-05	0.0847	0.0185	0.0596	0.0314
t5-small (summary)	9.87E-05	0.0989	0.0227	0.0654	0.0378
t5-small (watermarked)	4.73E-05	0.0684	0.0143	0.0486	0.0265

Table 2: Evaluation Metrics for XSum Dataset

Model	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	METEOR
flan-t5-small (summary)	0.00172	0.1508	0.0854	0.1096	0.0618
flan-t5-small (watermarked)	6.44E-06	0.0554	0.0269	0.0458	0.0199
t5-small (summary)	0.00172	0.1508	0.0854	0.1096	0.0618
t5-small (watermarked)	1.62E-05	0.0598	0.0315	0.0495	0.0220

Table 3: Evaluation Metrics for CNN/DailyMail Dataset

In the conducted experiments, a consistent reduction in performance across all evaluation metrics (BLEU, ROUGE-1, ROUGE-2, ROUGE-L, and METEOR) was observed when comparing the watermarked and non-watermarked summaries generated by both the flan-t5-small and t5-small models for the XSum and CNN/DailyMail datasets. For the XSum dataset, the BLEU score for the watermarked flan-t5-small model decreased from 9.87E-05 (non-watermarked) to 6.19E-05 (watermarked), reflecting a notable reduction in quality. Similar reductions were observed in ROUGE-1 (from 0.0989 to 0.0847), ROUGE-2 (from 0.0227 to 0.0185), ROUGE-L (from 0.0654 to 0.0596), and METEOR (from 0.0378 to 0.0314). These results are visually represented in Figures 1 and 2, which depict the comparison of BLEU and METEOR scores, respectively, between the watermarked and non-watermarked models for XSum.

For the CNN/DailyMail dataset, a similar pattern was observed. The BLEU score for the watermarked flan-t5-small model dropped from 0.00172 to 6.44E-06, with corresponding decreases in ROUGE-1 (from 0.1508 to 0.0554), ROUGE-2 (from 0.0854 to 0.0269), ROUGE-L (from 0.1096 to 0.0458), and METEOR (from 0.0618 to 0.0199). These results are illustrated in Figures 3 and 4, which show the BLEU and METEOR score reductions between watermarked and non-watermarked summaries for the CNN/-DailyMail dataset. The t5-small model also demonstrated similar reductions, with more pronounced effects, particularly in BLEU and ROUGE scores. These observations confirm that the watermarking process consistently degrades the quality of the generated summaries, with a more significant impact observed in the CNN/DailyMail dataset compared to XSum.

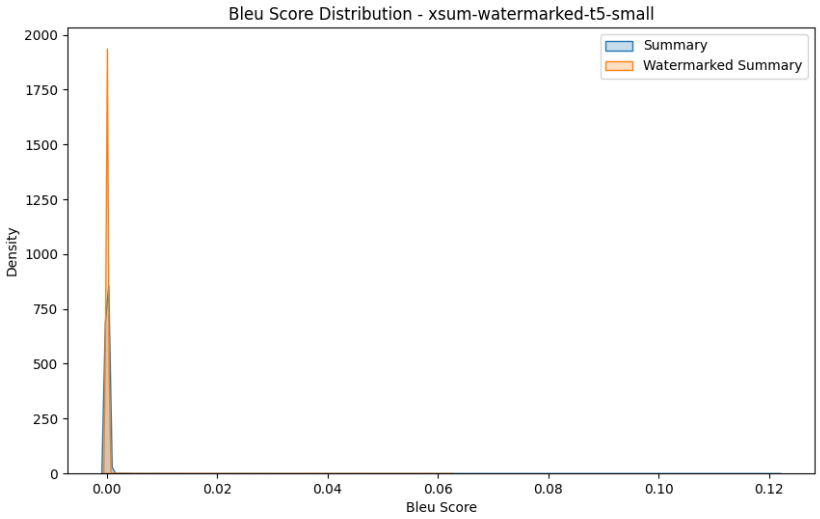

Figure 1: BLEU score comparison for watermarked and non-watermarked summaries (XSum)

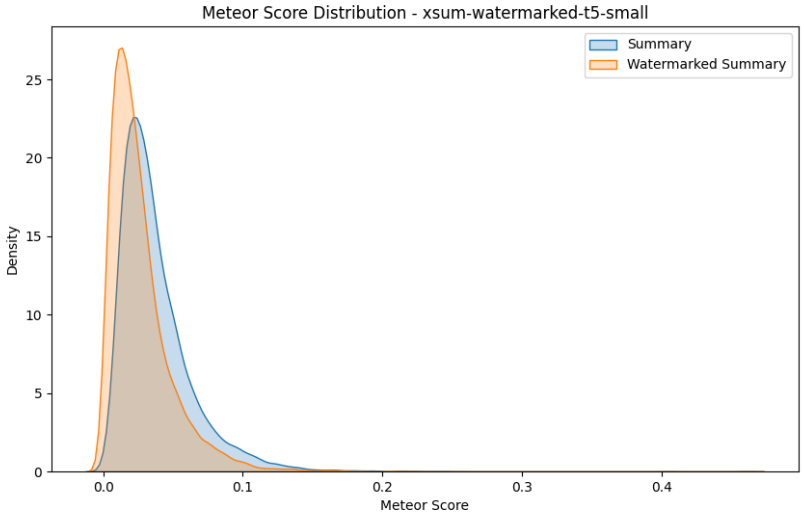

Figure 2: METEOR score comparison for watermarked and non-watermarked summaries (XSum)

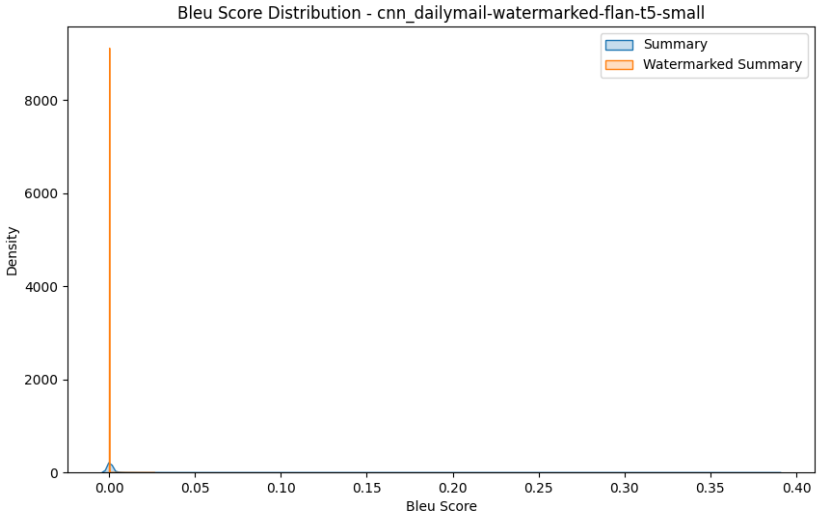

Figure 3: BLEU score comparison for watermarked and non-watermarked summaries (CNN/DailyMail)

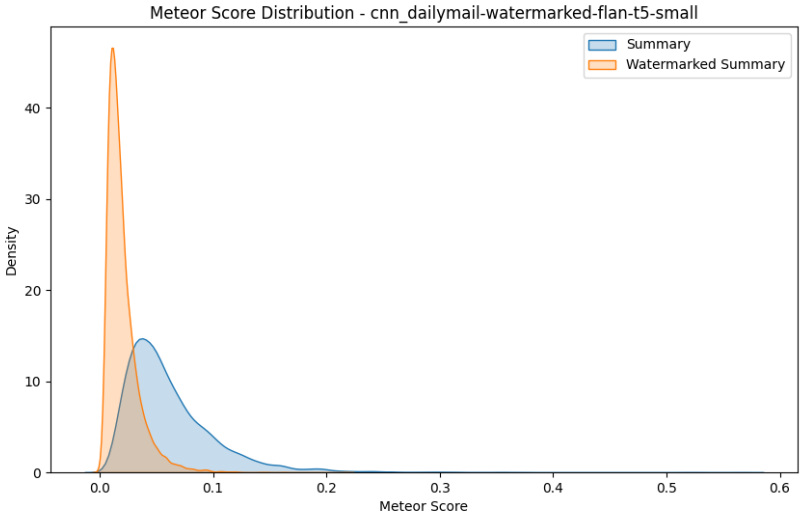

Figure 4: METEOR score comparison for watermarked and non-watermarked summaries (CNN/DailyMail)

Model	Unwatermarked Accuracy	Watermarked Accuracy
Bigram-logistic regression	0.83	0.70
Trigram-logistic regression	0.78	0.66
Multinomial Naive Bayes	0.81	0.62
K-Nearest Neighbors	0.72	0.51
LSTM	0.87	0.74

Table 4: Model accuracy comparison on XSum dataset with and without watermarking.

Model	Unwatermarked Accuracy	Watermarked Accuracy
Bigram-logistic regression	0.83	0.57
Trigram-logistic regression	0.78	0.45
Multinomial Naive Bayes	0.81	0.48
K-Nearest Neighbors	0.72	0.42
LSTM	0.87	0.60

Table 5: Model accuracy comparison on CNN/DailyMail dataset with and without watermarking.

Analysis: From the results shown in Tables 4 and 5, it is evident that watermarking significantly reduces the accuracy across all models for both the XSum and CNN/DailyMail datasets. The LSTM model consistently performs the best in both watermarked and unwatermarked settings, but still experiences a considerable drop in accuracy—approximately 13% for the XSum dataset and 27% for the CNN/DailyMail dataset. Simpler models, such as K-Nearest Neighbors and Multinomial Naive Bayes, exhibit larger performance reductions, with K-Nearest Neighbors suffering the most, particularly in the CNN/DailyMail dataset (a 30% reduction in accuracy).

Watermarking introduces a signal into the generated text that diminishes the fidelity of the summaries, thereby making it more difficult for the models to classify or extract meaningful information. This effect is more pronounced for simpler models like Naive Bayes and K-Nearest Neighbors, which rely on straightforward features such as word counts or proximity, compared to more advanced models like LSTM that utilize sequential dependencies and patterns. Nevertheless, even the more complex LSTM model experiences a significant performance hit, highlighting the trade-off between applying watermarking techniques and maintaining model accuracy.

6 Conclusion

This report has presented an evaluation of several models across two datasets, XSum and CNN/DailyMail, to assess the impact of watermarking on text summaries. The models tested included Bigram and Trigram logistic regression, Multinomial Naive Bayes, K-Nearest Neighbors, and LSTM. Results demonstrate that watermarking introduces a significant reduction in the accuracy of all models. In particular, simpler models like K-Nearest Neighbors and Naive Bayes suffered the most from the addition of watermarking, with reductions as high as 30% on some datasets. The more advanced LSTM model, while still the best performer, also saw noticeable drops in accuracy, indicating that no model was immune to the disruptive effects of watermarking.

The trade-off between applying watermarking techniques and maintaining model accuracy is clear: while watermarking can provide benefits for text tracking and identification purposes, it also comes with the cost of reduced performance in downstream tasks such as classification and summarization. Future work could focus on mitigating this degradation through more robust watermarking techniques or fine-tuning models to better handle watermarked content.

Ultimately, the results underline the importance of carefully considering the impact of watermarking when using it in machine learning pipelines, especially when model performance is critical for real-world applications.

References

[1] Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I., Goldstein, T. (2023, July). "A watermark for large language models." In *International Conference on Machine Learning* (pp. 17061-17084). PMLR.

[2] Ganesan, K. (2018). "Rouge 2.0: Updated and improved measures for evaluation of summarization tasks." *arXiv preprint arXiv:1803.01937*. https://arxiv.org/abs/1803.01937.

[3] See, A., Liu, P. J., Manning, C. D. (2017). "Get To The Point: Summarization with Pointer-Generator Networks." In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1073–1083). Vancouver, Canada: Association for Computational Linguistics. https://www.aclweb.org/anthology/P17-1099. DOI: 10.18653/v1/P17-1099.

[4] Narayan, S., Cohen, S. B., Lapata, M. (2018). "Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization." *ArXiv*, abs/1808.08745. https://arxiv.org/abs/1808.08745.