TM4Spam: Supervised and Unsupervised Approaches for Spam Detection

Sara Borello[1] **and Keita Jacopo Viganò**[1]

[1]MSc Students in Data Science, University of Milano-Bicocca (Unimib)

This project explores the problem of spam detection in text messages through a dual approach combining supervised classification and unsupervised topic modeling. The classification task is addressed by implementing two pipelines: one based on traditional feature extraction using TF-IDF coupled with Logistic Regression, and another leveraging a Transformer-based architecture, DistilBERT, fine-tuned on the dataset. In parallel, BERTopic is applied to uncover latent topics within the corpus, providing a qualitative understanding of the thematic content in both spam and non-spam messages.

Contents

1 Introduction **1**

2 Pre-Processing **2**

3 Classification **3**

A Logistic regression as baseline model . 3

B DistilBert 4

4 Unsupervised Topic Modeling with BERTopic **6**

A Spam messages 7

B Non Spam messages 8

5 Conclusions **9**

6 Bibliography **10**

1 Introduction

The widespread diffusion of digital communication has made the phenomenon of spam an increasingly pressing issue in both technological and social domains. Unwanted messages, often automatically generated, not only degrade the quality of communication but can also carry harmful content such as scams, phishing attempts, or malware. For this reason, the automatic detection of spam is a critical task in the field of Natural Language Processing, with direct implications for the security and efficiency of digital communication systems. This work addresses the problem of automatic spam message classification by comparing traditional approaches based on feature extraction techniques such as TF-IDF with advanced neural models like DistilBERT, a lightweight, optimized version of BERT pretrained on large text corpora. The goal is to assess the performance differences between linear models and Transformer-based models, particularly in their ability to capture the contextual semantics of text. In parallel, the latent semantic structure of the text corpus was investigated through BERTopic, a modern unsupervised topic modeling technique that combines transformer-based embeddings with clustering algorithms to extract interpretable topics. This approach was employed to achieve a dual objective: first, to gain qualitative insights into the thematic composition of both spam and non-spam messages; second, to assess whether the discovered topics could support the classification task or enrich the broader understanding of the dataset.

Dataset Description

The dataset (1), available on the Hugging Face Datasets Hub, comprises a total of 42,619 preprocessed text messages and emails, specifically curated for spam detection tasks. Each entry includes a text body and a binary label indicating whether the message is classified as spam (1) or not (0). The corpus is multilingual and combines both large-scale automatically gathered content and a smaller manually verified subset.

- **text**: A string field containing the full content of each message or email. These texts vary in length and content, ranging from short spam mes-

sages to lengthy scam attempts, including phishing schemes, fraudulent offers, and legitimate personal communications. Messages are written in 43 different languages, with a predominance of English.

- **is_spam**: An integer field representing the binary label:
  - `1` denotes a spam or scam message
  - `0` denotes a legitimate (ham) message

A notable portion of the dataset is a manually collected and annotated subset of 1,040 messages, carefully curated to include both casual, conversational content and scam messages in approximately 10 languages. This subset is balanced across classes and was developed to enhance multilingual and few-shot learning experiments.

2 Pre-Processing

Missing Values. No missing values were detected in the dataset. Both variables, `text` and `is_spam`, are complete across all 42,619 entries. This ensures the integrity of the dataset and eliminates the need for imputation or row-wise deletion.

Language Filtering. An analysis of the language distribution reveals that the overwhelming majority of messages in the dataset are written in English. Specifically, more than 40,000 out of approximately 42,000 total entries are in English, while other languages such as Spanish, French, and German appear only marginally (Fig 1). Given this strong linguistic skew, and to simplify the preprocessing and modeling pipeline, we restrict our analysis to English-language messages only.

Exploratory Analysis of Textual Features. To gain deeper insights into the characteristics of spam versus non-spam messages, we performed an exploratory analysis focusing on word usage and the presence of special symbols. Spam messages tend to include terms commonly associated with HTML content or marketing language, such as `http`, `font`, `nbsp`, and `td`, which suggest the presence of formatted templates, links, or automated content. In contrast, non-spam messages

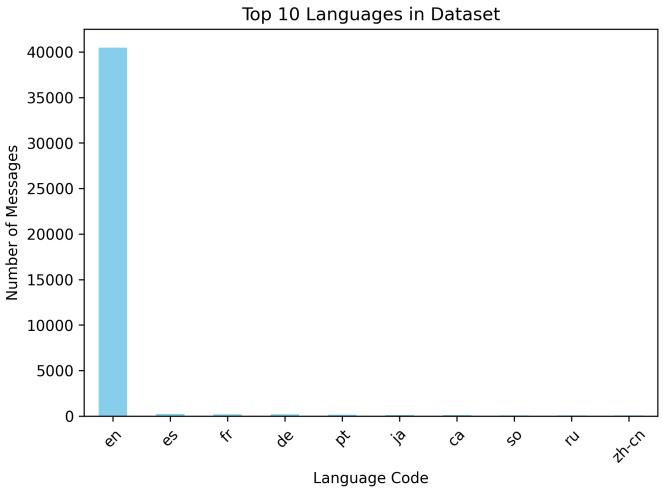

Fig. 1. Top 10 most frequent languages in the dataset

are more likely to contain terms indicative of internal communication or metadata, such as `ect`, `hou`, and `subject`.

In addition to token frequency, we also examined the use of special symbols and formatting cues within the texts. On average, spam messages contain a higher number of exclamation marks, dollar signs, URLs, and uppercase letters compared to non-spam messages. This pattern reflects a typical strategy employed in spam to emphasize urgency, promote offers, or draw attention through visual formatting.

Class Distribution. The dataset exhibits a moderately imbalanced class distribution, with `23,958` non-spam messages (label `0`) and `16,523` spam messages (label `1`). This corresponds to approximately **59% non-spam** and **41% spam**, as shown in Figure 2. Despite this difference, the imbalance is not severe enough to require the application of undersampling or oversampling techniques.

Modern machine learning models can handle such levels of imbalance effectively, especially when evaluation metrics beyond simple accuracy, such as precision, recall, F1-score, and ROC-AUC, are employed. Furthermore, the dataset split was performed using stratification to preserve this class distribution in both training and test sets. Therefore, the original data was retained without any resampling, in order to preserve the natural distribution and avoid introducing artificial bias.

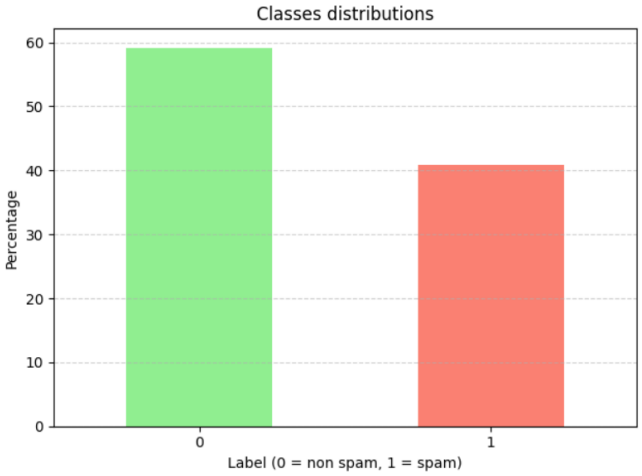

Fig. 2. Distribution of spam (1) and non-spam (0) messages

Train-Test Split. To evaluate model performance in a reliable and reproducible manner, the dataset was partitioned into training and test sets using an 80/20 stratified split. The stratification was applied with respect to the `is_spam` label to ensure that both subsets maintain the same class distribution as the original dataset. Specifically, the `train_test_split` function from `scikit-learn` was employed, with `test_size=0.2` and `stratify=df["is_spam"]`. The resulting splits were stored as separate CSV files for downstream processing: one containing 80% of the data for training, and the other 20% for testing. This setup helps prevent sampling bias and supports robust generalization assessment of the models.

Text Preprocessing. To prepare the data for text mining and classification, a preprocessing pipeline was applied to all messages in both training and test sets. This process involved multiple steps:

- **Lowercasing**: All text was converted to lowercase to ensure case-insensitive matching.
  Example: `"URGENT Response Required"` → `"urgent response required"`
- **Removal of noise**: Email addresses, URLs, numeric digits, and punctuation were removed using regular expressions and translation tables.
  Example: `"Contact us at support@example.com or visit http://help.com"` → `"contact us at or visit "`
- **Tokenization**: The text was split into individual words using the NLTK `word_tokenize` function.
  Example: `"urgent response required"` → `["urgent", "response", "required"]`
- **Stopword removal**: Common English stopwords were removed using the stopword list provided by NLTK. This step helps eliminate high-frequency words that carry little semantic meaning, thereby reducing noise and improving the quality of the text representation.
  Example: `["this", "is", "an", "email"]` → `["email"]`
- **Lemmatization and stemming**: Tokens were first lemmatized with `WordNetLemmatizer` (e.g., plural to singular), then stemmed using the `PorterStemmer` to reduce them to their base forms.
  Example: `"running"` → `"run"` → `"run"`

The cleaned output was stored in a new column named `clean_text`. For instance, the raw message `"KC resources nominated April 2000 resource deal"` was preprocessed into `"kc resourc nom april nom kc resourc deal"`.

3 Classification

A. Logistic regression as baseline model.

Text Representation using TF-IDF. To convert raw textual data into a numerical form suitable for machine learning algorithms, we adopted the Term Frequency–Inverse Document Frequency (TF-IDF) representation. TF-IDF is a widely used method in information retrieval and natural language processing that reflects how important a word is to a document relative to the entire corpus (2). Specifically, we applied `TfidfVectorizer` from the `scikit-learn` library, limiting the vocabulary to the top 1,000 most informative tokens. This transformation converts each message into a sparse feature vector where each component captures the TF-IDF

score of a corresponding term. Such a representation emphasizes rare but discriminative words while down-weighting overly frequent ones, thereby reducing noise and improving model interpretability.

Baseline Classification with Logistic Regression. As a baseline classifier, we employed logistic regression, a probabilistic linear model widely used for binary classification tasks (3). The logistic regression model estimates the probability that a given input belongs to the positive class (in our case, spam) by applying the sigmoid function to a linear combination of the input features. We integrated the TF-IDF vectorizer and the logistic regression classifier into a single pipeline using `Pipeline` from `scikit-learn`, enabling streamlined preprocessing and model fitting. Despite its simplicity, logistic regression is known to perform competitively on high-dimensional sparse data such as text (4), making it a strong and interpretable benchmark against which more complex models can be compared.

Evaluation of Logistic Regression. The logistic regression model achieved strong performance in the binary spam classification task. The overall accuracy on the test set is **90.4%**, indicating that the model correctly classified the vast majority of messages. Precision and recall scores are also balanced across the two classes. For the non-spam class (label `0`), the model achieved a precision of **0.896** and a recall of **0.948**, while for the spam class (label `1`), the precision was **0.917** and recall **0.841**. These results suggest that the model is slightly better at detecting legitimate messages (ham) than spam, though its ability to identify spam remains strong.

The confusion matrix (Figure 3) provides a clearer picture of the model's predictions. Out of 5,038 ham messages, 237 were misclassified as spam (false positives), and out of 3,486 spam messages, 313 were predicted as ham (false negatives). While these misclassifications are not negligible, they are acceptable given the complexity of the task and the simplicity of the model. Overall, the logistic regression serves as a solid baseline for future comparisons with more complex models.

B. DistilBert.

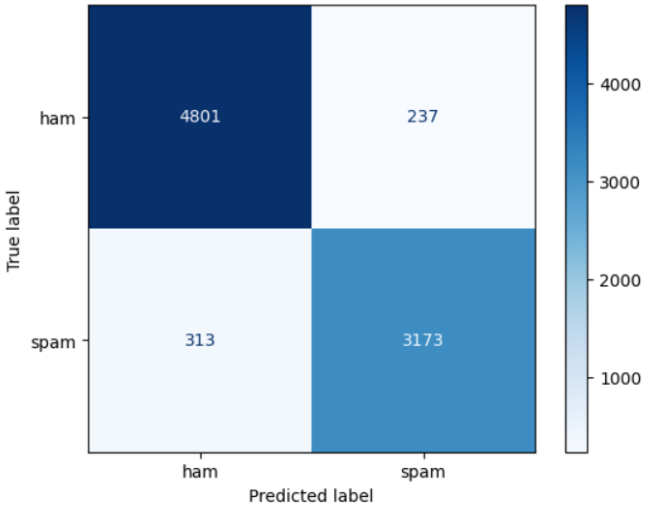

Fig. 3. Confusion matrix of logistic regression on test set

Text Representation with DistilBERT. For transformer-based classification, we used the pretrained `distilbert-base-uncased` model through the `AutoModelForSequenceClassification` class provided by Hugging Face Transformers (5). This model combines the DistilBERT encoder with a classification head for sequence-level prediction. Tokenization was performed using the corresponding pretrained tokenizer, which transforms each message into input IDs and attention masks using WordPiece subword encoding.

No manual embedding extraction was needed, as the DistilBERT model handles the conversion of text to contextual representations and outputs classification logits directly. This architecture allows leveraging rich contextualized embeddings learned during large-scale pretraining without additional feature engineering or manual processing.

Fine-Tuning Configuration of DistilBERT. Fine-tuning was performed using the Trainer API from Hugging Face's transformers library, which simplifies the training workflow by integrating key components such as evaluation, checkpointing, and metric logging. The pretrained distilbert-base-uncased model was used as the backbone, with a classification head added for binary spam detection.

Training was carried out over 3 epochs, using a learning rate of 2×10^{-5} and a weight decay of 0.01, as commonly adopted in fine-tuning transformer-based

models (6). The optimizer used was AdamW (7). A batch size of 16 was used during training, while evaluation was performed with a batch size of 64. We adopted the `eval_strategy="epoch"` and enabled `load_best_model_at_end=True` based on validation accuracy, following recommendations from the Hugging Face documentation (5).

The training process was monitored using the Weights & Biases (W&B) platform. The table below summarizes training and validation metrics across the three epochs:

Epoch	Training Loss	Validation Loss	Accuracy
1	0.1005	0.0544	0.9851
2	0.0268	0.0530	0.9884
3	0.0079	0.0612	0.9886

Table 1. Training metrics for DistilBERT fine-tuning

The entire fine-tuning process took approximately 90 minutes and was executed on a Google Colab instance equipped with an NVIDIA T4 GPU (16 GB VRAM). As illustrated in Figure 4, GPU utilization remained close to 100% throughout the training phases, indicating efficient hardware usage. Toward the end, GPU usage dropped significantly as the training concluded and evaluation routines took over.

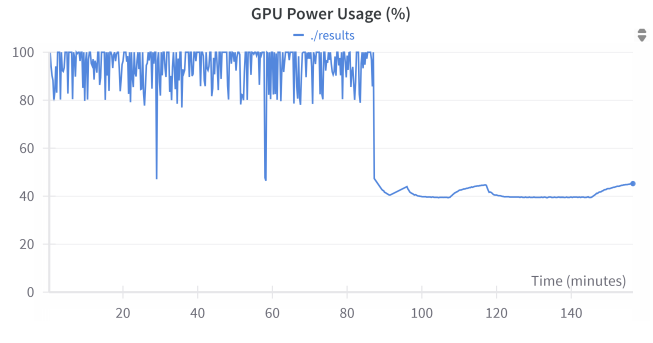

Fig. 4. GPU usage over time during DistilBERT training on Google Colab T4 GPU

Additional training logs and interactive visualizations are available on the W&B dashboard.

The resulting model was uploaded to Hugging Face and is publicly available at https://huggingface.co/cornualghost/tm4spam-distilbert.

Evaluation Results of DistilBERT. After training, the fine-tuned DistilBERT model was evaluated on the test set using standard classification metrics. The evaluation demonstrated outstanding performance, with an overall accuracy of 98.9%. Both classes, spam (label 1) and non-spam (label 0), were predicted with high precision and recall, confirming the model's robustness and generalization capabilities across categories.

The following table summarizes the detailed classification metrics:

Class	Precision	Recall	F1-score	Support
0 (non-spam)	0.990	0.990	0.990	5038
1 (spam)	0.986	0.986	0.986	3486
Accuracy			0.989	8524
Macro avg	0.988	0.988	0.988	8524
Weighted avg	0.989	0.989	0.989	8524

Table 2. Classification report for the fine-tuned DistilBERT model on the test set.

To complement the numerical metrics, we also present the confusion matrix of the model predictions in Figure 5. The matrix shows a highly symmetric and accurate classification pattern, with only 49 false positives (non-spam misclassified as spam) and 48 false negatives (spam misclassified as non-spam). These minimal errors indicate that the model makes very few critical mistakes, a desirable property in spam detection systems.

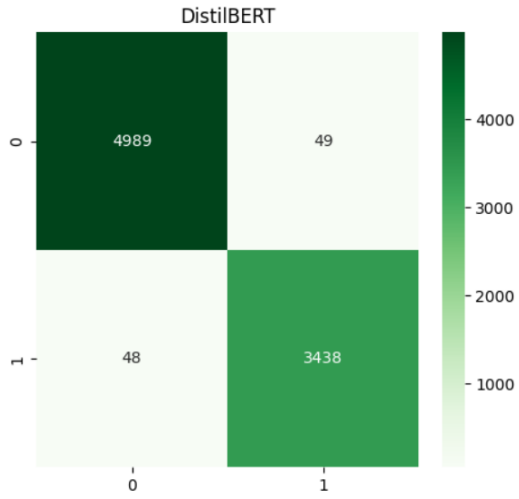

Fig. 5. Confusion matrix for the DistilBERT model on the test set.

These results confirm that the DistilBERT model, when fine-tuned appropriately, is highly effective for binary text classification tasks such as spam detection.

Comparative evaluation and final considerations for classification. To assess the performance of the implemented models, we compared Logistic Regression and DistilBERT across several evaluation metrics: precision, recall, F1-score, and accuracy. The table below summarizes the results obtained on the test set:

Model	Precision	Recall	F1	Accuracy
Logistic Regression	0.917	0.841	0.878	0.904
DistilBERT	0.986	0.986	0.986	0.989

Table 3. Comparison of Logistic Regression and DistilBERT performance on the test set.

As evident from the metrics, DistilBERT significantly outperforms Logistic Regression in every category. The improvement in recall for the spam class (label 1) is particularly noteworthy, indicating DistilBERT's superior ability to correctly identify unwanted messages, an essential requirement for any spam detection system. Moreover, DistilBERT shows a remarkable reduction in both false positives and false negatives, as confirmed by the confusion matrix analysis.

To further validate these findings, we also computed and compared the Area Under the ROC Curve (AUC) for both models. As shown in Figure 6, DistilBERT achieves an AUC of 0.999, compared to 0.965 for Logistic Regression. This indicates an almost perfect ability to discriminate between the two classes.

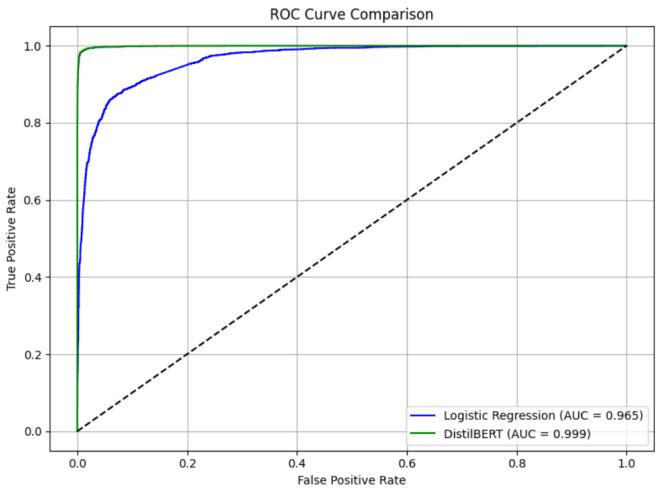

Fig. 6. ROC curve comparison between Logistic Regression and DistilBERT.

Overall, these results confirm that transformer-based models like DistilBERT, even in a distilled and lightweight form, offer substantial gains over traditional approaches such as logistic regression. Despite requiring more computational resources, DistilBERT demonstrates excellent generalization, robustness, and suitability for real-world deployment in spam detection pipelines.

4 Unsupervised Topic Modeling with BERTopic

To complement the supervised classification task with an exploratory analysis of the textual data, an unsupervised topic modeling approach was employed using BERTopic. This model leverages semantic embeddings and clustering techniques to identify latent themes across the document corpus.

The topic modeling pipeline follows the standard architecture introduced by BERTopic (8), combining semantic embeddings, dimensionality reduction, clustering, and topic representation.

- **Semantic Embedding**: Each document was converted into a dense vector using the `all-MiniLM-L6-v2` model from the `SentenceTransformers` library (9). This model provides a compact and efficient representation that captures contextual and semantic nuances of text, and is widely adopted for its balance between speed and accuracy.
- **Dimensionality Reduction**: The high-dimensional embeddings were projected into a five-dimensional space using UMAP (10) with the following parameters:
  - `n_components = 5`, to reduce complexity while maintaining semantic structure;
  - `metric = "cosine"`, to capture angular similarity between vectors;
  - `n_neighbors = 15`, to balance local and global manifold structure;
  - `min_dist = 0.0`, to allow denser packing of points in the reduced space.

  These settings are aligned with best practices reported in the BERTopic documentation (8).

- **Clustering**: Reduced embeddings were clustered using HDBSCAN (11), a hierarchical density-based clustering algorithm capable of detecting clusters with varying density and identifying outliers (assigned to Topic `-1`). Key parameters include:
  - `min_cluster_size = 30`, to ensure a minimum topic granularity;
  - `metric = "euclidean"`, as required by the internal distance computations of HDBSCAN.
- **Topic Representation**: For each cluster, representative terms were extracted using a `CountVectorizer` with:
  - `min_df = 15`, to filter out rare terms;
  - `stop_words = "english"`, to exclude common stopwords.

  The final topic descriptors were computed using class-based TF-IDF (c-TF-IDF), which enhances topic distinctiveness by comparing word frequencies within clusters against the global corpus distribution (8).

The model was instantiated by explicitly passing all components to BERTopic (embedding model, dimensionality reduction algorithm, clustering model and Topic Representation). After fitting the model with the precomputed embeddings, each document was assigned to a topic, and the most significant keywords per topic were extracted. This procedure enabled the discovery of coherent and interpretable themes within the spam and non-spam messages without requiring any labeled data.

A. Spam messages.

Analysis of Discovered Topics The topic modeling process resulted in the identification of 222 distinct topics (excluding the outlier topic `-1`). Each topic is characterized by a set of representative keywords derived from the c-TF-IDF scores and a collection of documents most closely associated with it.

Figure 7 presents a bar plot of the 20 most frequent topics based on the number of documents assigned to each. The most prominent topic (`Topic 0`) aggregates nearly 900 documents, significantly more than subsequent topics, which rapidly decay in size. This distribution suggests the presence of a few dominant themes (e.g., emails related to visual formatting or email order services) coexisting with a long tail of more specific or niche topics.

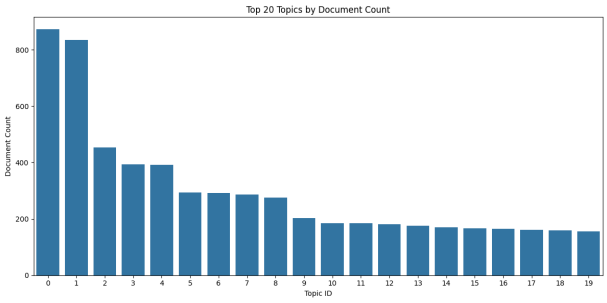

Fig. 7. Top 20 Topics by Document Count (excluding outliers).

Interpretation of the Most Representative Topics Among the topics extracted by the model, some are particularly semantically interpretable and coherent. Below, we highlight three especially illustrative topics:

- Topic 0 relates to adult content and pornography, as indicated by terms such as *girl*, *adult*, *sex*, *video*, and *membership*. This topic can be interpreted as *online adult content*.
- Topic 5 is closely associated with the financial and corporate domain, with keywords like *statement*, *company*, *stock*, *invest*, and *market*. It is interpreted as *corporate communications and financial investments*.
- Topic 10 displays a lexicon typical of fraudulent or spam messages involving prizes, including words like *prize*, *urgent*, *claim*, *award*, and *cash*. This topic clearly represents *scam or prize-based spam*.

Topic Coherence and Diversity Evaluation

Topic Coherence and Diversity Evaluation To assess the quality of the extracted topics, two intrinsic evaluation metrics were computed: *topic coherence* and *topic diversity*. The *topic coherence* score, computed using the c_v metric, was 0.5938. This metric evaluates how

Topic	Term	Weight
0	girl	0.0372
	adult	0.0290
	sex	0.0230
	video	0.0225
	site	0.0216
	free	0.0207
	membership	0.0205
	date	0.0192
	movi	0.0190
	http	0.0189
5	statement	0.0146
	compani	0.0144
	stock	0.0142
	invest	0.0129
	trade	0.0105
	expect	0.0095
	industri	0.0094
	investor	0.0092
	materi	0.0092
	market	0.0091
10	prize	0.0972
	urgent	0.0746
	txt	0.0730
	claim	0.0712
	award	0.0702
	ur	0.0701
	cash	0.0481
	holiday	0.0446
	draw	0.0436
	mobil	0.0411

Table 4. Top keywords and their weights for Topics 0, 5, and 10

semantically coherent the top keywords within each topic are, based on their co-occurrence in the original corpus. Formally, c_v coherence combines a sliding window, normalized pointwise mutual information (NPMI), and the cosine similarity between context vectors of top words (12). A value close to 1 indicates that the topic keywords frequently appear together in similar contexts. While the coherence score obtained suggests reasonably good interpretability, there may still be room for improvement through fine-tuning of clustering parameters or preprocessing.

Additionally, the *topic diversity* was assessed by computing the average Jaccard distance between the top 10 keywords of each topic pair. The Jaccard distance between two sets A and B is defined as:

$$d_J(A,B) = 1 - \frac{|A \cap B|}{|A \cup B|}$$

The resulting *Jaccard Diversity Score* was 0.9941, indicating that the vast majority of topics are lexically distinct, with minimal keyword overlap. This high diversity implies that the model successfully identifies a broad set of semantically unique themes across the dataset, enhancing the richness of the topic modeling output.

B. Non Spam messages.

Analysis of Discovered Topics in Non-Spam Messages

The topic modeling procedure applied to non-spam messages yielded 111 distinct topics (excluding the outlier topic `-1`). Each topic was described by its top keywords computed via class-based TF-IDF (c-TF-IDF), along with a representative set of documents.

Figure 8 shows the 20 most frequent topics among the non-spam messages. The most dominant topics, such as `Topic 0` and `Topic 1`, are each associated with over 400 documents. These likely represent broad themes commonly found in legitimate communications, such as scheduling messages, company newsletters, or meeting arrangements. After these, the document frequency per topic decreases gradually, suggesting a relatively even distribution of recurring professional or organizational topics across the corpus.

This smoother decay, compared to the sharper drop observed in the spam corpus, suggests that non-spam messages are more evenly distributed across a wide variety of real-world communication contexts, without a few dominant spammy patterns overwhelming the distribution.

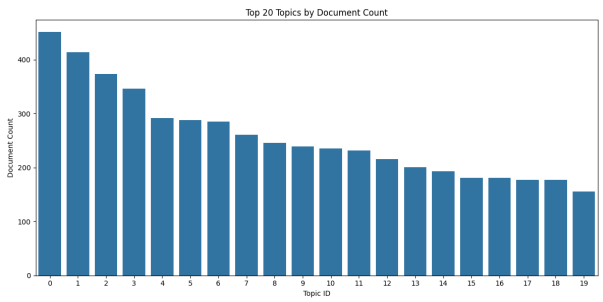

Fig. 8. Top 20 Topics by Document Count in Non-Spam Messages (excluding outliers).

Interpretation of the Most Representative Topics (Non-Spam Messages) Among the topics extracted from non-spam messages, several exhibit strong semantic coherence and interpretability. Below, we summarize three particularly meaningful topics:

Topic	Term	Weight
2	interview	0.0690
	vinc	0.0293
	resum	0.0264
	kaminski	0.0247
	research	0.0183
	thank	0.0173
	pm	0.0148
	shirley	0.0137
	pleas	0.0135
	forward	0.0128
3	meet	0.0679
	prc	0.0261
	attend	0.0230
	pleas	0.0188
	eb	0.0169
	thank	0.0163
	dinner	0.0163
	pm	0.0158
	vinc	0.0157
	th	0.0153
10	nation	0.0221
	state	0.0182
	peopl	0.0177
	world	0.0169
	govern	0.0167
	war	0.0164
	polit	0.0159
	american	0.0148
	unit	0.0135
	attack	0.0119

Table 5. Top keywords and their weights for Topics 2, 3, and 10 (non-spam)

- Topic 2 refers to job interviews and academic contexts, as suggested by terms such as *interview*, *resum*, *kaminski*, *research*, and *shirley*. This topic can be interpreted as *recruitment or academic interviews*.
- Topic 3 is associated with internal meetings or events, with keywords like *meet*, *attend*, *pleas*, *dinner*, and *thank*. It likely corresponds to *corporate meetings or social gatherings*.
- Topic 10 is centered on geopolitical or governmental discussions, including terms such as *nation*, *state*, *govern*, *polit*, and *war*. This topic can be interpreted as *political or international affairs*.

Topic Coherence and Diversity Evaluation (Non-Spam Messages) To evaluate the quality of the topics extracted from the non-spam portion of the corpus, we computed two intrinsic metrics: topic coherence and topic diversity. The *topic coherence*, measured using the c_v metric, yields a score of 0.5714. This value reflects the semantic consistency of the top keywords within each topic, based on their distributional co-occurrence in the non-spam documents. Although slightly lower than in the spam case, this score still indicates a fair level of interpretability, with potential for further refinement via improved preprocessing or parameter tuning.

The *topic diversity* was computed using the Jaccard distance across the top 10 keywords of each topic pair. The resulting *Jaccard Diversity Score* is 0.9906, suggesting very high lexical dissimilarity among topics. This indicates that the model has effectively discovered a wide range of semantically distinct themes within the non-spam subset, enhancing the overall descriptive power and reducing redundancy among topics.

5 Conclusions

This study has explored the task of spam detection in text messages through a twofold approach: supervised classification and topic modeling.

On the classification front, we compared two distinct modeling strategies. The baseline pipeline using TF-IDF with Logistic Regression achieved solid results, demonstrating that even simple linear models can perform well on high-dimensional textual data. However, the Transformer-based architecture, DistilBERT, substantially outperformed the baseline across all metrics, achieving a higher test accuracy and minimizing both false positives and false negatives. This confirms the effectiveness of contextualized word embeddings and transfer learning in capturing nuanced language patterns critical for distinguishing between spam and legitimate content.

In parallel, the application of BERTopic enabled an unsupervised exploration of latent themes within the cor-

pus. While the analysis of spam messages revealed meaningful and practically relevant patterns, such as adult content, financial scams, and prize-based deception, topic modeling of non-spam messages offered limited added value. In real-world applications, understanding the thematic structure of spam is far more critical for detection, filtering, and auditing purposes. Although we included topic modeling of non-spam messages for exploratory interest, such analysis would not typically be necessary or prioritized in operational settings.

Overall, the integration of classification and topic modeling proved mutually reinforcing. While classification ensures operational performance in filtering unwanted messages, topic modeling offers interpretability, insight, and context-awareness, valuable features for auditing automated systems or improving user trust. Future developments may include cross-lingual spam classification to address the multilingual nature of the dataset, leveraging the manually annotated subset for few-shot or zero-shot learning scenarios. Additionally, the integration of topic distributions as auxiliary features in supervised models could be explored to improve interpretability and potentially enhance classification performance through a hybrid modeling approach.

6 Bibliography

1. Fred Zhang. All-scam-spam dataset. https://huggingface.co/datasets/FredZhang7/all-scam-spam, 2023. Accessed: 2025-06-21.
2. Juan Ramos. Using tf-idf to determine word relevance in document queries. *Proceedings of the first instructional conference on machine learning*, 242(1):133–142, 2003.
3. David W Hosmer, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*. John Wiley & Sons, 2013.
4. Andrew Y Ng and Michael I Jordan. Discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, 14:841–848, 2002.
5. Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. Transformers: State-of-the-art natural language processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020.
6. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL-HLT*, 2019.
7. Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.
8. Maarten Grootendorst. Bertopic: Neural topic modeling with class-based tf-idf. *Information*, 13(1):31, 2022.
9. Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.
10. Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.
11. Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Advances in Knowledge Discovery and Data Mining*, pages 160–172. Springer, 2013.
12. Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*, pages 399–408. ACM, 2015. doi: 10.1145/2684822.2685324.