Text Classification of SMS Messages: Spam vs Ham

Srinathi K
Department of Artificial Intelligence and Data Science
Dr. Mahalingam College of Engineering and Technology
Pollachi, India
srinathi486@gmail.com

***Abstract* - Text preprocessing plays a vital role in organizing and analyzing short text data such as SMS messages. This project presents a complete end-to-end pipeline for clustering and topic modeling of SMS data. The raw text undergoes essential preprocessing steps including lowercasing, tokenization, stopword removal and lemmatization to prepare it for further analysis. Following preprocessing, dimensionality reduction techniques are applied to simplify high-dimensional data for visualization and interpretation. To group semantically similar messages, multiple clustering algorithms are utilized, including KMeans, Agglomerative Clustering, DBSCAN, Spectral Clustering and Gaussian Mixture Models. The effectiveness of these clustering methods is evaluated by examining representative texts from each cluster to extract underlying themes. In addition, Latent Dirichlet Allocation (LDA) is employed for topic modeling, offering deeper insights into the principal topics within the SMS corpus. This comprehensive pipeline facilitates systematic exploration of short text clustering and topic modeling techniques, contributing to the identification of meaningful patterns and structural relationships in the data.**

***Index Terms* - SMS Classification, Spam Detection, Text Preprocessing, Text Representation, Clustering, Topic Modeling with LDA**

I. Introduction

The world is increasingly shaped by data, with individuals generating vast amounts of digital content daily, particularly in the form of short text messages such as SMS, tweets and chat logs. These micro-texts, despite their limited length, carry rich and valuable information. However, their brevity, informal language and unstructured nature present significant challenges for conventional text analysis methods. One fundamental approach to making sense of such data involves classifying or grouping it into meaningful categories or clusters. [12]

Clustering and classification represent some of the oldest and most intuitive cognitive processes performed by humans. To understand new concepts or phenomena, comparisons are instinctively made with known categories based on similarity, guided by learned patterns and rules. In the context of data science, this translates into the design of algorithms capable of automatically grouping similar items and distinguishing them from others.

This project addresses the challenge of organizing and interpreting unstructured SMS text data through unsupervised learning techniques. A comprehensive pipeline is developed that integrates clustering algorithms and topic modeling to reveal hidden structures within short text messages. By applying a range of text preprocessing methods, vector representation models and multiple clustering techniques including KMeans, DBSCAN and Agglomerative Clustering the impact of different combinations on the coherence and quality of discovered clusters is explored.

The objective is to identify methodologies that yield the most insightful and interpretable results, with potential applications in areas such as spam detection, sentiment analysis and automated content classification within short text domains.

II. Methodology

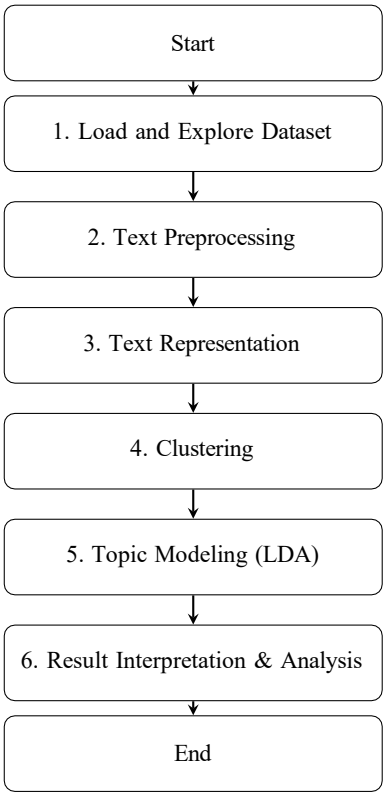
This section outlines the comprehensive methodology employed to develop and execute a text clustering and topic modeling pipeline for analyzing SMS messages. The primary objective is to uncover patterns, group similar messages and extract thematic insights from unstructured short text data.

The pipeline begins with data loading and inspection, where the SMS dataset is imported and essential columns are retained. The subsequent step involves text preprocessing, which includes lowercasing, tokenization, removal of punctuation and stopwords and lemmatization to standardize the text. These preprocessing steps ensure that the data is clean, consistent and suitable for analysis.

Following preprocessing, the cleaned messages are transformed into numerical vectors using three text representation techniques: Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF) and sentence embeddings derived from transformer-based models. These vectorized representations capture both lexical and semantic features of the texts.

To enhance interpretability, dimensionality reduction techniques such as Principal Component Analysis (PCA) are applied. Clustering algorithms including KMeans, Agglomerative Clustering, DBSCAN, Spectral Clustering and Gaussian Mixture Models are then utilized to group similar messages. Each algorithm is applied to all three vector representations to compare clustering outcomes.

Finally, Latent Dirichlet Allocation (LDA) is employed for topic modeling to identify latent topics within the SMS corpus. Different topic numbers are tested and representative keywords from each topic are extracted for interpretation. This integrated pipeline facilitates the discovery of meaningful clusters and thematic structures in the SMS data.



A. Data Preprocessing

The dataset used for this project contains SMS messages labeled as either spam or ham (non-spam). It is loaded from a CSV file and cleaned by renaming columns (e.g., v1 to label and v2 to text). Only relevant columns are retained and null entries are removed. For efficient experimentation, the dataset is optionally sampled down to 100 messages using random sampling with a fixed seed to ensure reproducibility.

B. Text Preprocessing

Effective clustering and topic modeling begin with thorough text preprocessing. In this project, each SMS message undergoes a standardized cleaning pipeline to enhance the quality of text representations. All text is first converted to lowercase to ensure uniformity. Tokenization is performed using NLTK's word tokenize() function, which splits each message into individual word tokens. Non-alphabetic characters and punctuation are removed to retain only meaningful textual content. Common English stopwords such as "the," "is," and "at" are filtered out using NLTK's stopword corpus to eliminate frequently occurring but uninformative words. Lemmatization is then applied using WordNet's lemmatizer to reduce words to their base forms (e.g., "running" becomes "run"). All preprocessing steps are implemented in a preprocess text() function, which is applied to each message in the dataset. The resulting cleaned text is stored in a new column labeled cleaned text for use in downstream modeling tasks. [7]

C. Text Representation

To convert text into a numerical format suitable for machine learning models, three types of vector representations are used in this study. The first is the Bag of Words (BoW) [3] model, which represents each document by counting the frequency of each word in the corpus. This results in a high-dimensional sparse matrix, generated using the CountVectorizer() function. The second representation is Term Frequency-Inverse Document Frequency (TF-IDF) [14], which extends BoW by scaling word frequencies according to their rarity across documents, helping to down-weight common but less informative words. TF-IDF vectors are created using the TfidfVectorizer() method. Finally, to capture deeper semantic relationships between words and sentences, dense sentence embeddings are generated using pre-trained transformer models. These embeddings encode the contextual and semantic meaning of the entire message into fixed-length numerical vectors, making them particularly effective for understanding short texts such as SMS messages.

D. Clustering Algorithms

To uncover latent structures within the SMS dataset, five widely used clustering algorithms are applied across three types of vector representations: Bag of Words (BoW), TF-IDF and sentence embeddings and are evaluated at different cluster sizes ($k = 3, 5, 7$). KMeans clustering serves as a baseline method, partitioning the data into k clusters by minimizing intra-cluster variance. Agglomerative Clustering, a hierarchical technique, follows a bottom-up approach that iteratively merges the closest clusters based on linkage criteria. Gaussian Mixture Models (GMM) provide a probabilistic framework, assuming that the data originates from a mixture of multivariate Gaussian distributions. Spectral Clustering applies eigen-decomposition of a similarity graph to project the data into a lower-dimensional space before performing clustering. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) identifies dense regions of data points as clusters while designating isolated points as noise. These algorithms offer complementary perspectives on how messages group semantically, structurally, and spatially within vector space.

E. Topic Modeling with LDA

To uncover latent themes within the SMS dataset, Latent Dirichlet Allocation (LDA) is applied to the Bag of Words (BoW) representation of the text. LDA models operate with different numbers of topics specifically 5, 7 and 10 to explore the interpretability and granularity of the resulting topics. For each model, the top 10 most representative words are extracted per topic to facilitate interpretation. An automatic theme detection mechanism further supports this process by mapping keywords to intuitive labels (for example, keywords such as "free" or "win" may indicate a topic related to advertisements or promotions). To understand how the model assigns topics to individual documents, the top two dominant topics, along with their associated probabilities, are identified for selected messages. This approach provides meaningful insights into the thematic composition of the SMS data and characterizes the types of messages present within the collection.

III. Results

The complete pipeline is executed on a sampled subset of 100 SMS messages, yielding insightful results at various stages of analysis. The following summarizes the key observations from preprocessing, vectorization, clustering and topic modeling.

A. Text Preprocessing Results

The text preprocessing stage, consisting of lowercasing, tokenization, stopword removal and lemmatization, significantly reduces the noise and length of the original SMS messages. This process enhances the clarity of key textual features, making the data more suitable for analysis. For example, a message such as "FREE entry into our 250 weekly competition. . . " is simplified to "free entry weekly competition," retaining only the most informative words. By removing irrelevant tokens and standardizing word forms, the preprocessing step plays a crucial role in improving the quality and effectiveness of subsequent clustering and topic modeling tasks.

B. Text Representation Results

The Bag of Words (BoW) model serves as a foundational approach to text representation by transforming each SMS message into a numerical vector based on word frequency, disregarding grammar and word order. In this study, **BoW** generates a sparse matrix of approximately (100, 500), representing 100 preprocessed messages and around 500 unique terms. Despite its simplicity and high dimensionality, **BoW** proves effective for models that rely on literal word occurrences. It functions as a straightforward baseline for evaluating the performance of more advanced methods, such as TF-IDF and sentence embeddings. By capturing exact word presence, **BoW** offers an interpretable and computationally efficient means of representing text for clustering and topic modeling tasks. [6]

TF-IDF (Term Frequency–Inverse Document Frequency) provides a numerical representation that reflects the importance of a word in a document relative to a collection of documents (corpus). It combines two components: term frequency (TF), which measures how often a word appears in an individual document, and inverse document frequency (IDF), which down-weights words that are common across many documents. This adjustment highlights distinctive terms while reducing the influence of frequently occurring but uninformative words. TF-IDF enhances the discriminative power of features compared to raw frequency counts, making it particularly useful for text clustering and retrieval tasks. [9]

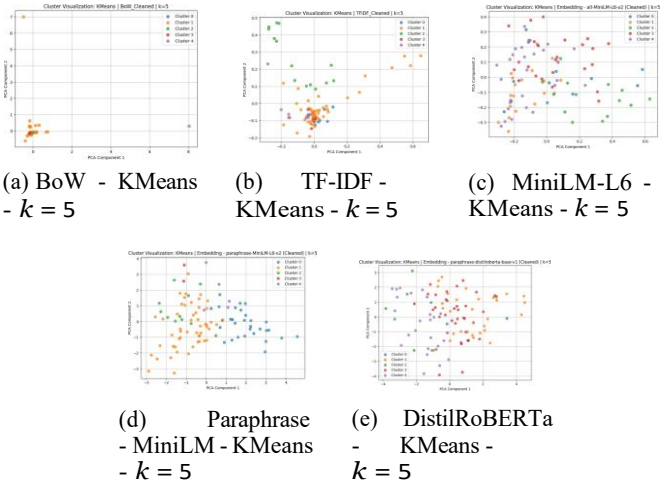
Sentence embeddings generate dense vector representations that capture the semantic and contextual meaning of entire messages. Unlike BoW and TF-IDF, which rely on word frequency and weighting schemes, sentence embeddings are derived from transformer-based models trained to understand language at the sentence level. In this study, three models are employed to produce these embeddings: **all-MiniLM-L6-v2** and **paraphrase-MiniLM-L6-v2**, both generating vectors of shape (100, 384), and **paraphrase-distilroberta-base-v1** [13], which produces vectors of shape (100, 768). These embeddings prove particularly effective for short messages, as they retain nuanced meaning and semantic similarity even when word choices vary significantly between texts. [10]

C. Clustering Results

Clustering is applied to all text representations using five algorithms across multiple cluster sizes (3, 5, and 7). Principal Component Analysis (PCA) reduces the high-dimensional vectors to two dimensions for visualization. The resulting clusters are analyzed to uncover underlying themes and patterns within the SMS data.

KMeans Clustering

KMeans Clustering is a widely used partitioning algorithm that divides the dataset into k non-overlapping clusters by minimizing intra-cluster variance. Each data point is assigned to the nearest centroid based on Euclidean distance, and centroids are iteratively updated as the mean of the points assigned to each cluster. In this study, KMeans performs consistently well across various text representations, including Bag of Words (BoW), TF-IDF and sentence embeddings. It produces well-separated and balanced clusters, effectively uncovering distinct themes such as promotional content, casual conversations, and relationship-based messages. While KMeans is computationally efficient and scalable to large datasets, it remains sensitive to the initial placement of centroids, which can influence the final clustering outcome. [4]



(a) BoW - KMeans - $k = 5$ (b) TF-IDF - KMeans - $k = 5$ (c) MiniLM-L6 - KMeans - $k = 5$

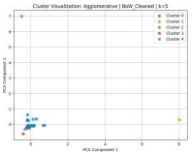
(d) Paraphrase - MiniLM - KMeans - $k = 5$ (e) DistilRoBERTa - KMeans - $k = 5$

Fig. 1: KMeans clustering results using BoW, TF-IDF and sentence embedding representations with $k = 5$. PCA projections illustrate the separability and structure of the resulting clusters.

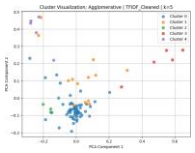
Agglomerative Clustering

Agglomerative Clustering is a hierarchical clustering technique that constructs a nested cluster tree in a bottom-up manner. Initially, each data point is treated as an individual cluster. At each iteration, the two closest clusters are merged based on a specified distance metric, such as Euclidean distance. This process continues until either a predefined number of

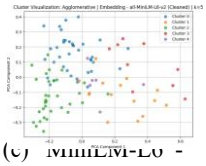
clusters is formed or all data points are grouped into a single cluster. In this study, Agglomerative Clustering demonstrates strong performance, particularly with TF-IDF representations. Its hierarchical nature facilitates clearer interpretation of the resulting clusters, revealing meaningful groupings within the SMS data. This method proves especially effective for uncovering layered or nested structures within short messages. [5]



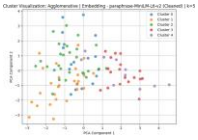
(a) BoW - Agglomerative - $k = 5$



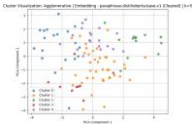
(b) TF-IDF - Agglomerative - $k = 5$



(c) MiniLM-L6 - Agglomerative - $k = 5$



(d) Paraphrase-MiniLM - Agglomerative - $k = 5$

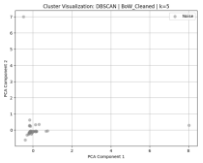


(e) DistilRoBERTa Agglomerative - $k = 5$

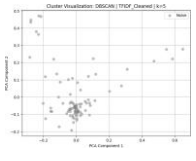
Fig. 2: Agglomerative Clustering results on different text representations with $k = 5$. PCA projections highlight the structure and separation of the clusters.

DBSCAN Clustering

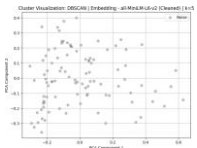
DBSCAN Clustering (Density-Based Spatial Clustering of Applications with Noise) is a density-based clustering algorithm that groups data points based on the concepts of proximity and density. It defines clusters as regions where a minimum number of points (minPt) fall within a specified neighborhood radius (epsilon). Points located in dense regions are assigned to clusters, while those in sparse regions are classified as noise or outliers. DBSCAN is particularly effective at discovering clusters of arbitrary shape and handling noisy data. However, selecting appropriate values for epsilon and minPts remains challenging, especially when working with datasets exhibiting varying densities. In this study, DBSCAN performs best when applied to dense vector representations such as sentence embeddings but shows limited effectiveness with sparse features like Bag of Words or TF-IDF. Despite these limitations, DBSCAN proves valuable for identifying outliers and detecting noise within the SMS dataset. [2]



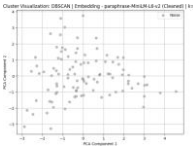
(a) BoW - DBSCAN



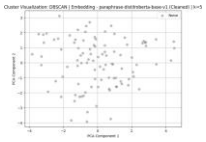
(b) TF-IDF - DBSCAN



(c) MiniLM-L6 - DBSCAN



(d) Paraphrase-MiniLM - DBSCAN

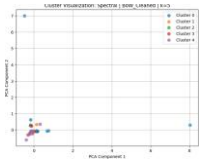


(e) DistilRoBERTa -DBSCAN

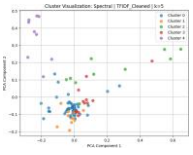
Fig. 3: This figure presents DBSCAN clustering results on BoW, TF-IDF and sentence embedding representations. PCA projections display the distribution of clusters and identification of noise points.

Spectral Clustering

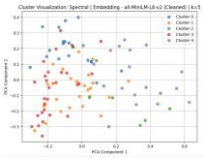
Spectral Clustering is a graph-based clustering algorithm that leverages the relationships between data points to identify meaningful clusters. It constructs a similarity graph where each data point is represented as a node and edges capture the similarity between pairs of points, typically based on inverse distance or another affinity measure. The algorithm applies eigen-decomposition to the graph Laplacian to project the data into a lower-dimensional spectral space, where standard clustering techniques such as KMeans are employed. Spectral Clustering is particularly effective in detecting non-spherical and complex-shaped clusters that traditional algorithms may fail to capture. However, its performance remains sensitive to the choice of the similarity function and the tuning of hyperparameters. Additionally, the matrix operations involved can be computationally intensive for large datasets. In this study, Spectral Clustering produces well-separated and coherent clusters when applied to sentence embeddings, particularly when the affinity matrix is appropriately configured. [8]



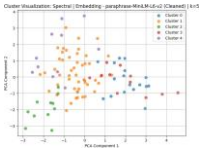
(a) BoW - Spectral - $k = 5$



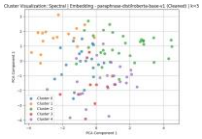
(b) TF-IDF - Spectral - $k = 5$



(c) MiniLM-L6 - Spectral - $k = 5$

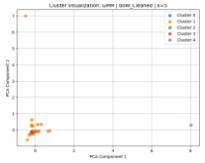


(d) Paraphrase-MiniLM - Spectral - $k = 5$

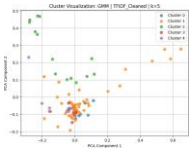


(e) DistilRoBERTa - Spectral - $k = 5$

Fig. 4: This figure presents Spectral Clustering results on BoW, TF-IDF and sentence embedding representations with $k = 5$. PCA projections illustrate the separability and structure of the clusters.

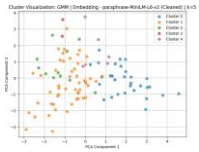


(a) BoW - GMM - $k = 5$

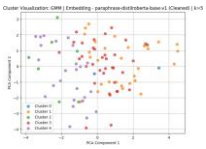


(b) TF-IDF - GMM - $k = 5$

(c) MiniLM-L6 - GMM - $k = 5$



(d) Paraphrase-MiniLM - GMM - $k = 5$



(e) DistilRoBERTa - GMM - $k = 5$

Fig. 5: This figure presents Gaussian Mixture Model (GMM) clustering results on BoW, TF-IDF and sentence embedding representations with $k = 5$. PCA projections visualize the distribution and overlap of the resulting clusters.

Gaussian Mixture Models (GMM)

Gaussian Mixture Models (GMM)is a probabilistic clustering technique that models data as a combination of multiple Gaussian (normal) distributions. Each cluster is characterized by its own mean and variance, allowing for flexible modeling of elliptical shapes in the feature space. Unlike hard clustering methods such as KMeans, GMM employs soft clustering, assigning each data point a probability of belonging to each cluster rather than a single fixed label. This approach proves particularly effective for capturing uncertainty and overlapping group structures within the data. However, GMM remains sensitive to the choice of initialization and the number of components, and may overfit when too many components are specified. The method also assumes that the underlying data is generated from Gaussian distributions, an assumption that may not always hold true. In this study, GMM performs well on semantic sentence embeddings, capturing subtle overlaps in meaning among SMS messages. Proper tuning of parameters, including the number of components and the covariance type, is essential for producing meaningful and interpretable clusters. [11]

D. LDA Topic Modeling Results

Latent Dirichlet Allocation (LDA) is a probabilistic model that treats each document as a mixture of topics and each topic as a distribution over words. When applied to the SMS data, LDA reveals meaningful topic groupings by analyzing word co-occurrence patterns. Most messages are associated with one or two dominant topics. Representative keywords from each topic are extracted to facilitate interpretation of the themes, highlighting key patterns across the dataset.

Latent Dirichlet Allocation (LDA) with five topics reveals distinct thematic groupings within the SMS dataset. Topic 1, characterized by keywords such as love, grl, think, let and know, relates to relationships. Topic 2, with words like free, care, mobile, reply and finish, reflects offers and promotions. Topic 3, including terms such as ok, said, minute and friend, represents casual conversation. Topic 4, identified by keywords such as call, claim, line, contact and prize, corresponds to marketing or spam-related content. Finally, Topic 5, containing words like meet, night, tomorrow, dinner and time, is associated with planning or social activities. These topic groupings highlight the effectiveness of LDA in uncovering meaningful themes within short text messages.

IV. Discussion

The results of this project highlight the importance of combining both traditional and modern NLP techniques for understanding short text data such as SMS messages. Preprocessing steps, including stopword removal and lemmati-

zation, significantly improve the quality of input texts. These enhancements enable vectorization methods such as TF-IDF and sentence embeddings to more effectively capture semantic patterns.

Clustering outcomes demonstrate that embedding-based representations particularly those generated by transformer models such as MiniLM and DistilRoBERTa outperform traditional methods like Bag of Words (BoW) and TF-IDF in producing coherent and semantically meaningful clusters. Visualizations using Principal Component Analysis (PCA) further validate the distinctness and separability of clusters, especially when applied to high-dimensional sentence embeddings.

Topic modeling using Latent Dirichlet Allocation (LDA) offers valuable insights by summarizing the main themes present in the SMS corpus. LDA performs best with BoW inputs and the incorporation of automatic theme labeling significantly enhances the interpretability of results. The topics uncovered such as marketing promotions, casual conversations and relationship-based messages align closely with the real-world content and context of SMS communication.

Overall, the pipeline illustrates the effectiveness of integrating diverse NLP components to extract meaningful structure from unstructured text data. This approach provides a solid foundation for downstream tasks such as spam detection, customer sentiment analysis and categorization of social media content. [1]

REFERENCES

[1] DM Blei, AY Ng, and MI Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3, 2003.

[2] Martin Ester, Hans-Peter Kriegel, Jo¨rg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

[3] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[4] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[5] Anil K Jain and Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[6] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer, 1998.

[7] Christopher Manning and Hinrich Schutze. *Foundations of statistical natural language processing*. MIT press, 1999.

[8] Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14, 2001.

[9] Juan Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. New Jersey, USA, 2003.

[10] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[11] Douglas Reynolds. Gaussian mixture models. In *Encyclopedia of biometrics*, pages 827–832. Springer, 2015.

[12] Muhammad Salman, Muhammad Ikram, and Mohamed Ali Kaafar. Investigating evasive techniques in sms spam filtering: A comparative analysis of machine learning models. *IEEE Access*, 12:24306–24324, 2024.

[13] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[14] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.