Text Classification: How Machine Learning is Revolutionizing Text Categorization

Hesham Allam * , Lisa Makubvure , Benjamin Gyamfi , Graham Kwadwo , Kehinde Akinwolere

Posted Date: 16 December 2024

doi: 10.20944/preprints202412.1304.v1

Keywords: Text Categorization; Machine Learning; Automation; Document Representation; Dimensionality Reduction; Classifier Evaluation; Emerging Trends; Feature Selection; Data Preprocessing; Predictive Modeling

Article

Text Classification: How Machine Learning is Revolutionizing Text Categorization

Hesham Allam *, Lisa Makubvure, Benjamin Gyamfi, Kwadwo Graham and Kehinde Akinwolere

Center for Information & Communication Sciences (CICS), Ball State University, Indiana, USA
* Correspondence: Hesham.allam@bsu.edu

Abstract: The automated classification of texts into predefined categories has become increasingly prominent, driven by the exponential growth of digital documents and the demand for efficient organization. This paper serves as an in-depth survey of text classification and machine learning, consolidating diverse aspects of the field into a single, comprehensive resource—a rarity in the current body of literature. Few studies have achieved such breadth, and this work claims to provide a unified perspective, offering a significant contribution to researchers and the academic community. The survey examines the evolution of machine learning in text categorization (TC), highlighting its transformative advantages over manual classification, such as enhanced accuracy, reduced labor, and adaptability across domains. It delves into various TC tasks and contrasts machine learning methodologies with knowledge engineering approaches, demonstrating the strengths and flexibility of data-driven techniques. Key applications of TC are explored, alongside an analysis of critical machine learning methods, including document representation techniques and dimensionality reduction strategies. Moreover, this study evaluates a range of text categorization models, identifies persistent challenges like class imbalance and overfitting, and investigates emerging trends shaping the future of the field. It discusses essential components such as document representation, classifier construction, and performance evaluation, offering a well-rounded understanding of the current state of TC. Importantly, this paper also provides clear research directions, emphasizing areas requiring further innovation, such as hybrid methodologies, explainable AI (XAI), and scalable approaches for low-resource languages. By bridging gaps in existing knowledge and suggesting actionable paths forward, this work positions itself as a vital resource for academics and industry practitioners, fostering deeper exploration and development in text classification.

Keywords: Text Categorization (TC); machine learning; automation; document representation; dimension reduction; classifier evaluation; emerging trends

1. Introduction

The history of text categorization (TC) is a narrative of continuous evolution, driven by the growing need to efficiently manage and organize ever-increasing volumes of text data. Initially a manual process rooted in text and corpus linguistics, TC involved categorizing texts into predefined topics or genres [1–3]. [1,2] However, the digital revolution and exponential growth of textual data rendered manual methods impractical, necessitating the development of automated systems. Early approaches relied on handmade features and rule-based systems, which, while foundational, were limited by their rigidity and inability to adapt to new data. Subsequent advancements introduced statistical techniques, nature-inspired algorithms, and graph-based methods to enhance the flexibility and accuracy of text categorization [4].

The introduction of machine learning (ML) marked a turning point, with algorithms like k-Nearest Neighbors (KNN) and Support Vector Machines (SVM) offering improved scalability and accuracy by learning directly from data. These methods utilized feature selection techniques to

 doi:10.20944/preprints202412.1304.v1

address challenges such as high-dimensional feature spaces and scalability. This shift represented a significant improvement in classification performance and adaptability [1,5,6]. More recently, deep learning has revolutionized TC, enabling the development of models capable of capturing intricate semantic relationships in text. Techniques like recurrent neural networks (RNNs), convolutional neural networks (CNNs), and transformer-based models (e.g., BERT) have dramatically improved performance. Additionally, semantic methods such as ontology-based classification and latent semantic indexing have enhanced contextual understanding of text data [7].

Despite its progress, TC remains a field at the crossroads of ML and information retrieval (IR), sharing features with related areas like text mining and knowledge extraction. This overlap has led to fragmented literature, inconsistent terminology, and a lack of standardized frameworks [1–3]. Challenges include ambiguous definitions of terms like "automatic text classification," which variously refer to assigning predefined categories, creating new categories, or clustering texts [8,9]. Furthermore, the field lacks comprehensive resources such as dedicated textbooks or journals, hindering the consolidation of knowledge and impeding newcomers [10].

However, these gaps present opportunities for advancement. By developing systematic methodologies, standardizing terminologies, and centralizing resources, researchers can unify the field and enhance its applicability. The absence of structured guidance also underscores the potential for innovative contributions, such as creating frameworks that bridge theory and practice or addressing evolving challenges like multilingual classification, noisy data handling, and explain ability in models.

This research paper presents an extensive survey of text classification and machine learning, offering a unified framework that consolidates best practices from ML, natural language processing (NLP), and information retrieval (IR). It introduces a comprehensive taxonomy of text classification techniques, encompassing traditional algorithms, modern ML approaches, and emerging trends in deep learning. Furthermore, the paper provides a detailed evaluation of methods using standardized datasets and metrics, making it a foundational resource for researchers and a practical guide for industry professionals.

The paper is planned as follows: Section 2 outlines the scope and role of TC, differentiating it from related tasks. Section 3 highlights the growing importance of text classification (TC) in organizing digital data, detailing its evolution with machine learning, key applications, methodologies, and critical techniques like document representation and dimensionality reduction. Section 4 explores machine learning techniques for text classification, covering supervised learning, classifier construction, feature selection, and advanced methods to enhance accuracy and adaptability. Section 5 examines document processing techniques, including the Vector Space Model (VSM), Bag-of-Words, lexical semantics, text tokenization, word stemming, stop word removal, and weighting schemes like TF-IDF and advanced alternatives. Section 6 discusses common methods for term selection. Section 7 discusses the evaluation of text categorization models, focusing on performance metrics, the F-Measure, and challenges associated with model evaluation. Section 8 addresses challenges in machine learning-based text classification, including overfitting, underfitting, class imbalance, feature space complexity, and linguistic issues like ambiguity and polysemy. Section 9 looks at major advances in deep learning for TC, transfer learning, and hybrid techniques that combine knowledge engineering and machine learning. Finally, section 10 concludes with future directions and ethical considerations, emphasizing the need for scalable, unbiased, and context-aware TC systems.

 doi:10.20944/preprints202412.1304.v1

2. Background

2.1. Overview of Text Categorization (TC) and Recent Research

Text categorization (TC), also known as text classification, is a core task in text mining and natural language processing (NLP). It involves assigning predefined categories to text documents, facilitating the management of large volumes of unstructured text data. This process enables efficient information retrieval and analysis, with applications in sentiment analysis, spam detection, and topic classification. By streamlining activities such as content filtering and subject identification, TC enhances productivity and supports decision-making.

Historically, TC was performed manually, which was suitable for small datasets but lacked scalability, consistency, and speed, especially with dynamic data like social media. The advent of automated TC, driven by machine learning (ML) and NLP, revolutionized the field by increasing speed, accuracy, and scalability while reducing human bias. Automated TC is now widely adopted in industries that process extensive and rapidly growing data streams [10].

The domain of text categorization has experienced remarkable advancements, propelled by continuous innovations in machine learning (ML) and natural language processing (NLP). Over the years, researchers have developed sophisticated techniques to enhance classification accuracy, scalability, and adaptability across diverse datasets and applications. These developments span traditional machine learning approaches, deep learning architectures, and hybrid models that blend the strengths of both paradigms. This evolution has not only improved the precision of text categorization systems but also expanded their relevance to areas such as sentiment analysis, spam detection, topic identification, and domain-specific classification.

One of the most impactful trends in recent research is the integration of transfer learning techniques, which allow models to influence knowledge from pre-trained language illustrations. This approach has significantly boosted the effectiveness of text categorization, particularly in handling low-resource languages and niche domains. Additionally, studies have explored the use of hybrid methodologies, combining rule-based systems with advanced machine-learning practices to address challenges in multilingual and domain-specific contexts. These approaches underscore the growing importance of adaptability and context-awareness in modern text categorization systems.

Recent studies in 2024 have placed a particular emphasis on leveraging pre-trained language models, domain-specific adaptations, and innovative clustering techniques. These advancements have demonstrated special effectiveness in managing complex, multidimensional datasets, enabling more nuanced and accurate classifications. The following table highlights key contributions from research conducted in 2024, providing an overview of cutting-edge methodologies and findings. A comprehensive extension of this table is available in the appendix, offering insights into earlier studies and broader trends. This focused summary of 2024 not only illustrates the state-of-the-art advancements but also lays the groundwork for future research, addressing emerging challenges and opportunities in the field.

Table 1. Key Contributions in Text Categorization Research. Table 1 Recent Studies on Text Classification.

Publication Type	Title	Year	Authors	Objectives	Insights	Practical Implications
Journal Article	Research on Intelligent Natural Language Texts Classification	2022	[11]	- Summarize and compare text classification methods. - Explore development direction of text classification research.	The paper summarizes previous studies on text classification, highlighting the rapid development of machine learning technologies and the diversification of research methods. It compares classification methods based on technical routes, text vectorization, and classification information processing for further research insights.	- Intelligent classification enhances efficient use of natural language texts. - Provides references for further research in text classification methods.
Journal Article	The Research Trends of Text Classification Studies (2000–2020): A Bibliometric Analysis	2022	[12]	- Evaluate the state of the arts of TC studies. - Identify publication trends and important contributors in TC research.	The study analyzes 3,121 text classification publications from 2000 to 2020, highlighting trends, contributors, and disciplines. It reveals increased interest in advanced classification algorithms, performance evaluation methods, and practical applications, indicating a growing interdisciplinary focus in text classification research.	- Recognizes recent trends in text classification research. - Highlights importance of advanced algorithms and applications.
Journal Article	A survey on text classification and its applications	2020	Xujuan et. Al [13]	- Overview of existing text classification technologies. - Propose research direction for text mining challenges.	Previous studies on text classification have proposed various feature selection methods and classification algorithms, addressing challenges such as scalability due to the massive increase in text data. These studies highlight the importance of effective information organization and management in diverse research fields.	- Important applications in real-world text classification. - Addresses challenges in text mining and scalability.
Journal Article	A Survey on Text Classification: From Traditional to Deep Learning	2022	[14]	- Review state-of-the-art approaches from 1961 to 2021.	The paper reviews state-of-the-art approaches in text classification from 1961 to 2021, highlighting traditional models and deep learning	- Summarizes key implications for text classification research.

doi:10.20944/preprints202412.1304.v1

				- Create a taxonomy for text classification methods.	advancements. It discusses technical developments, benchmark datasets, and provides a comprehensive comparison of various techniques and evaluation metrics used in previous studies.	- Identifies future research directions and challenges.
Book Chapter	Case Studies of Several Popular Text Classification Methods	2023	[15]	- Evaluate automatic language processing techniques for text classification. - Analyze and compare performance of various text classification algorithms.	The paper discusses various text classification methods, highlighting that deep learning models, particularly distributed word representations like word2vec and Glove, outperform traditional methods such as Bag of Words (BOW). Contextual embeddings like BERT also show significant performance improvements.	- Improved text classification methods for massive data analysis. - Enhanced performance using advanced feature extraction techniques.
Journal Article	Text Classification Using Deep Learning Models: A Comparative Review	2023	[16]	- Analyze deep learning models for text classification tasks. - Address gaps, limitations, and future research directions in text classification.	The paper conducts a literature review on various deep learning models for text classification, analyzing their gaps and limitations. It highlights previous studies' comparative results and discusses classification applications, guiding future research directions in this field.	- Guidance for future research in text classification. - Highlights challenges and potential directions in the field.
Journal Article	Survey on Text Classification	2020	[17]	- Classify documents into predefined classes effectively. - Compare various text representation schemes and classifiers.	Previous studies on text classification have utilized various techniques, including supervised learning with labeled training documents, Naive Bayes, and Decision Tree algorithms. Challenges include the difficulty of creating labeled datasets and the limited applicability of individual classifiers across different domains.	- Detailed information on text classification concepts and algorithms. - Evaluation of algorithms using common performance metrics.
Journal Article	The Text Classification Method Based on BiLSTM and Multi-Scale CNN	2024	[18]	- Overview of deep learning in text classification. - Analyze research progress and technical approaches.	Previous studies on text classification have transitioned from traditional machine learning methods to deep learning models, including attention mechanisms and pre-trained language	- Overview of deep learning text classification methods. - Analysis of labeled datasets for research support.

					models, highlighting significant progress and challenges in enhancing model performance and dataset quality across various domains.	
Journal Article	Research on Text Classification Method Based on NLP	2023	[19]	- Describe text classification concepts and processes. - Explore deep learning models for text classification.	Previous studies on text classification have explored various methods, including LSTM-based multi-task learning architectures, capsule networks, and hybrid models like RCNN, demonstrating advancements in feature extraction and improved performance in tasks such as sentiment analysis and spam recognition.	- Text classification methods are important for effectively classifying text-based data. - New ideas such as word embedding models and pre-training models have made great progress in text classification.
Book Chapter	A Comparative Study on Various Text Classification Methods	2020	[20]	- Analyze methods for efficient text classification. - Examine featurization techniques and their performance.	The paper does not provide a review of previous studies on text classification. Instead, it focuses on analyzing various text classification methods and featurization techniques, such as bag of words, Tf-Idf vectorization, and Word2Vec approaches.	- Analyzes efficient text classification methods for decision-making. - Discusses various featurization techniques for improved performance.
Journal Article	Evaluating text classification: A benchmark study	2024	[21]	- Investigate necessity of complex models versus simple methods. - Assess performance across various classification tasks and datasets.	The paper highlights a gap in existing literature, noting that previous research primarily compares similar types of methods without a comprehensive benchmark. This study aims to provide an extensive evaluation across various tasks, datasets, and model architectures.	- Simple methods can outperform complex models in certain tasks. - Negative correlation between F1 performance and complexity for small datasets.
Proceedings Article	Comparative Performance of Machine Learning Methods for Text Classification	2020	[22]	- Compare performance of machine learning and deep learning algorithms. - Explore scalability with larger data instances.	Previous studies on text classification primarily tested machine learning and deep learning methods with relatively small-sized data instances. This paper builds on that by comparing these methods' performance and scalability using a larger dataset of 6,000 instances across six classes.	- Deep learning outperforms traditional methods in text classification. - Scalability of methods for larger data instances explored.

Journal Article	A Survey on Text Classification using Machine Learning Algorithms	2019	[23]	- Explore algorithms for automated text document classification. - Select best features and classification algorithms for accuracy.	Previous studies on text classification have explored various methodologies, including feature selection techniques like Document Frequency Thresholding and Information Gain, and classification algorithms such as K-nearest Neighbors and Support Vector Machines, highlighting the importance of efficient keyword prioritization for accurate categorization.	- Automated text classification improves efficiency in document handling. - Reduces reliance on expert classification for large text documents.
Dataset	Text Classification Data from 15 Drug Class Review SLR Studies	2023	[24]	- Automate citation classification in systematic reviews. - Reduce workload in systematic review preparation.	The paper references a study by Cohen et al. (2006) that focused on reducing workload in systematic review preparation through automated citation classification, providing a foundation for the datasets used in the current text classification research on drug class reviews.	- Automates citation classification in systematic reviews. - Reduces workload for researchers in drug class studies.
Proceedings Article	An Exploration of the Effectiveness of Machine Learning Algorithms for Text Classification	2023	[25]	- Explore effectiveness of machine learning algorithms for text classification. - Compare performance of various algorithms like SVM, KNN, CNN, RNN.	The paper does not provide specific details on previous studies in text classification. It focuses on evaluating and comparing the performance of various machine learning algorithms, such as decision trees, SVM, KNN, CNN, and RNN for text classification tasks.	- Machine learning improves text classification accuracy and efficiency. - Algorithms can handle complex and large datasets effectively.
Proceedings Article	A Comparative Text Classification Study with Deep Learning-Based Algorithms	2022	[26]	- Compare deep learning algorithms for text classification. - Optimize hyperparameters and evaluate word embeddings effectiveness.	The paper compares its results with previous studies in the literature, highlighting significant improvements in classification performance using deep learning algorithms and word embeddings. It specifically utilizes an open-source Turkish News benchmarking dataset for this comparative analysis.	- Improved text classification performance using deep learning algorithms. - Effective hyperparameter tuning enhances classification accuracy.

doi:10.20944/preprints202412.1304.v1

Proceedings Article	Classification Models of Text: A Comparative Study	2021	[27]	- Overview of classification process stages. - Survey and compare popular classification algorithms.	The paper does not provide specific details on previous studies in text classification. Instead, it focuses on the classification process, including preprocessing, feature engineering, dimension decomposition, model selection, and evaluation, while surveying and comparing popular classification algorithms.	- Text classification has implications in education, politics, and finance. - The paper provides a comparative study of popular classification algorithms.
Journal Article	Trends and patterns of text classification techniques: a systematic mapping study	2020	[28]	- Provide an overview of text classification research trends and gaps. - Analyze research patterns, problems, and problem-solving methods in text classification.	The paper systematically reviews ninety-six studies on text classification from 2006 to 2017, identifying nine main problems and analyzing research patterns, data sources, language choices, and applied techniques, highlighting significant trends and gaps in the field.	- Highlights trends and gaps in text classification research. - Identifies nine main problems in text classification area.
Journal Article	Research On Text Classification Based On Deep Neural Network	2022	[29]	- Design text representation and classification models using deep networks. - Improve text feature representation and classification accuracy.	The paper highlights that traditional text classification methods, such as the bag-of-words model and vector space model, face challenges like loss of context, high dimensionality, and sparsity, prompting a shift towards deep learning techniques for improved performance.	- Deep learning models improve text classification performance compared to traditional methods. - The BRCNN and ACNN models proposed in the paper show better text feature representation and classification accuracy.

doi:10.20944/preprints202412.1304.v1

 doi:10.20944/preprints202412.1304.v1

2.2. Approaches to Text Categorization

TC employs a variety of machine learning practices, broadly categorized into organized, unsupervised, and deep learning methods:

2.2.1. Supervised Learning

- In supervised learning, models are trained using labeled datasets to classify new documents into predefined categories. Popular algorithms for this purpose include Logistic Regression, Naive Bayes, Random Forest, Support Vector Machines (SVM), and AdaBoost. For example, Naive Bayes has demonstrated impressive accuracy, reaching up to 96.86% in certain applications [30].
- Deep learning techniques, such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, achieve high accuracy while requiring minimal feature engineering. For instance, LSTMs have demonstrated accuracy rates of up to 92% in specific tasks [31].

2.2.2. Unsupervised Learning

- Unsupervised learning techniques, including hierarchical clustering, k-means clustering, and probabilistic clustering, are used to group documents based on content similarity in cases where labeled data is not available [32].
- These techniques uncover inherent data structures and are instrumental in analyzing unlabeled datasets[4],

Advancements in TC leverage feature extraction and dimensionality reduction techniques like PCA and LDA, enhancing model performance. Deep learning models further refine TC by capturing linguistic subtleties such as tone and context, making them invaluable for tasks like sentiment analysis. These developments position TC as a vital tool across industries for deriving insights from text data and managing digital information environments [31].

Despite its benefits, TC faces challenges such as handling ambiguous or overlapping categories and requiring large labeled datasets for supervised learning. Algorithm selection also influences outcomes, with models like Naive Bayes and SVM performing differently across datasets and applications [33].

2.3. The Rise of Machine Learning in TC

Machine learning (ML) has significantly advanced TC by transitioning from rule-based systems to adaptive algorithms that learn from tagged input. Early ML models like Naive Bayes and SVM laid the groundwork for TC.

- **Naive Bayes**: Effective for large vocabularies due to its probabilistic approach.
- **SVM**: Achieves precision by mapping text into high-dimensional spaces, helping identify closely related themes.

The emergence of deep learning further transformed TC with models like CNNs and RNNs, capable of capturing local word dependencies and long-term correlations. Transformer models, such as BERT, have set new benchmarks by understanding bidirectional, long-distance interactions in text. These innovations enable tasks like sarcasm and emotion detection with minimal fine-tuning. Additionally, metrics like burstiness and perplexity enhance TC by identifying significant phrases and quantifying prediction uncertainty.

Applications of ML-driven TC span customer service, healthcare, finance, and more, enabling rapid and precise classification. This supports innovations in content moderation, personalized recommendations, and trend analysis [1].

 doi:10.20944/preprints202412.1304.v1

2.4. Benefits of Automated TC Over Manual Classification

Automated TC offers numerous advantages over manual processes:

- **Scalability and Efficiency**: Handles large datasets rapidly and consistently, unlike manual methods that are time-intensive and impractical for extensive collections.
- **Objectivity**: Applies standard criteria uniformly, eliminating human bias and ensuring reliable outcomes, crucial for domains like legal document classification.
- **Real-Time Processing**: Facilitates immediate classification, essential in industries like finance and journalism where timely decisions are critical [34].

Advanced ML techniques, such as burstiness and perplexity, improve TC's adaptability to complex and dynamic contexts. Automated systems are versatile, supporting tasks like thematic grouping, sentiment analysis, and customer feedback analysis. This versatility positions automated TC as a cornerstone for deriving actionable insights in today's data-driven world [35].

2.5. Types of TC Tasks

Text categorization assigns predefined categories to free-text documents, organizing them conceptually for efficient retrieval and management [5]. Applications include email filtering, topic labeling, and content organization for digital libraries.

TC tasks vary depending on the nature of the classification problem. Common types include:

- **Binary Classification:** This involves two classes, such as spam and non-spam emails, where each document belongs to one of the two categories [5].
- **Multi-Class Classification:** More than two classes are involved, and each document is assigned to only one class, such as classifying news articles into topics like politics, sports, or entertainment [5] [36].
- **Multi-Label Classification:** In this case, each document may belong to multiple categories simultaneously. For example, an academic paper may be categorized under multiple disciplines like biology and technology [5].
- **Hierarchical Classification:** Documents are classified into categories organized in a hierarchical format. This type is beneficial for large datasets with numerous categories [5].

2.5. Single-Label vs. Multi-Label Classification

Single-label classification refers to the assignment of only one class label to each document, whereas multi-label classification allows multiple labels to be assigned simultaneously.

a. **Single-Label Classification:** Often approached using binary classification methods, where documents are classified into distinct categories without overlap [37].
b. **Multi-Label Classification:** In this context, a document can belong to several categories, necessitating complex modeling to manage overlapping labels. Methods for handling multi-label classification often involve transforming the problem into multiple binary classification tasks [37].

2.7. Document-Pivoted vs. Category-Pivoted TC

Document-pivoted and category-pivoted text categorization represent two methodologies for organizing the classification process.

a. **Document-Pivoted Categorization (DPC):** This approach focuses on classifying a document by searching across all possible categories. It is generally simpler to implement and more efficient for practical applications [38] [39].
b. **Category-Pivoted Categorization (CPC):** In contrast, CPC classifies documents by first identifying the relevant category. This method is more complex, as it requires re-evaluating document classifications when new categories are added.

 doi:10.20944/preprints202412.1304.v1

2.8. Hard Categorization vs. Ranking

Hard categorization and ranking are two distinct approaches to classifying documents.

a. **Hard Categorization**: This method assigns each document to a single category, resulting in binary decisions about the classification of whether the text belongs to the category or not.
b. **Ranking**: In contrast, ranking categorization involves generating a list of categories ranked by their relevance to the document. This approach provides a more nuanced view of a document's classification, allowing for further decision-making processes based on category probability.

2.9. Machine Learning vs. Knowledge Engineering in TC

Both machine learning and knowledge engineering play significant roles in the development of text categorization systems.

2.9.1. Machine Learning

Machine learning algorithms automatically learn from data without prior programming to enable systems to adapt and make predictions or decisions according to patterns of interest in the data [40]. These algorithms can be managed, unsupervised, and reinforcement learning methods that improve categorization accuracy by finding and analyzing complex patterns, trends, and relationships in large datasets. This is important and beneficial when working with large, complex data to achieve a more granular and efficient categorization process.

Machine learning methods have been extensively employed in the field of Text Categorization for predictive modeling. Essentially, historical or pre-labeled data is used to train algorithms which are later applied to new, unseen data to categorize it effectively [41]. These techniques give rise to the development of machine learning that allows a TC system to extend beyond manual or rule-based approaches to text categorization, scalable and adaptable to large volumes of text [40]. In addition, iterative machine learning assures the system of continuous improvement to become increasingly effective with time in identifying subtle patterns of interest and accurately classifying data items into their respective categories or themes.

2.9.2. Knowledge Engineering

Knowledge engineering focuses on emulating human expert decision-making processes in specific domains [41]. Knowledge engineering involves creating systems that replicate the decision-making processes of human experts in specific fields. It focuses on capturing and representing expert knowledge, such as rules and reasoning, to develop systems that can analyze problems and provide accurate solutions. These systems are widely used in specialized domains to enhance problem-solving and decision-making capabilities. It involves creating expert systems that can utilize rules and data to facilitate complex problem-solving. In the context of TC, knowledge engineering systems integrate human expertise with machine learning outputs to enhance decision-making capabilities and ensure accuracy in categorizations [42].

3. Applications of Text Categorization

Text categorization is integral to many sectors, bringing with it a host of benefits that include better information management, enhanced customer engagement, and operational efficiency [43]. Text categorization is essential across various sectors, offering benefits like improved information management by organizing data effectively, enhanced customer engagement through personalized content delivery, and increased operational efficiency by automating classification processes.

3.1. Document Indexing for Information Retrieval Systems

Document indexing Treating documents with keywords or key phrases to facilitate retrieval in Boolean IR systems. In order to avoid inconsistencies in the tags assigned to the documents, controlled dictionaries or thematic thesauri like the MESH thesaurus for medicine are used [44] [45].

 doi:10.20944/preprints202412.1304.v1

Though manual indexing has been considerably replaced by automated indexing, it helps to manage large databases efficiently in research and library systems.

3.2. Role of Controlled Vocabulary and Thesauri

Control vocabulary helps standardize the terminology in certain fields and thus supports the consistent categorization of documents. The thesauri give a hierarchical and relational context to the terms, thereby making the search and retrieval processes in systems using TC more effective, particularly in large-scale document databases [46] [44].

3.3. Automated Document Organization and Archiving

For large document bases, TC automates the filing system for corporate records, patent filings, and other institutional archives. The tools can classify patents or group news stories by theme, which can cut down on manual classification workload [47] [44].

3.4. Use in Corporate and News Media

Corporations use TC to filter incoming information, such as routing relevant documents to specific departments. News agencies use TC to pre-categorize articles before publication, for example, placing content under "Politics" or "Lifestyle" [48] [44]. In high-volume environments, this is particularly important for facilitating streamlined operations and maintaining organized archives.

3.5. Text Filtering and Content Personalization

Content personalization in TC assists in tailoring content to user preferences through the classification of information that is stored according to user profiles. Applications such as personalized news feeds, customized email filtering, and targeted advertisements are performed with systems trained on filtering or promoting content based on precise thematic categories [49] . Content personalization in Text Categorization (TC) delivers user-specific content by classifying information based on individual profiles. Systems trained in thematic categories enable applications like personalized news feeds, email filtering, and targeted ads. This ensures users receive relevant and tailored experiences.

3.6. Newsfeeds, Email Filtering, and Spam Detection

A good example for Test Categorization is spam detection, which classifies the email content into spam or non-spam categories based on the keywords and patterns of the sender. Filters analyze text content to block unsolicited emails that provide relief to users from irrelevant content [50] [44].

3.7. Word Sense Disambiguation (WSD)

WSD disambiguates polysemous words for recognize the true sense of a term in context. It has been identified as one of the fundamental tasks in natural communication to handle requests and engines and machine translation. Categorization of word senses using WSD provides for more accurate keyword searching and indexing [51] [44].

3.8. Hierarchical Categorization of Web Content

Content categorization taxonomy organizes online information into a hierarchical structure, similar to those used in digital libraries or internet directories. Text categorization (TC) techniques classify websites into nested levels, such as "Technology" > "Artificial Intelligence" > "Machine Learning," enabling users to navigate vast online repositories with ease and efficiency. [52] [44].

4. Machine Learning Techniques in Text Categorization (TC)

Machine Learning (ML) has proven dangerous to the creation and progress in automated text categorization (TC) systems. “Text classification (TC) is the process of categorizing text documents

 doi:10.20944/preprints202412.1304.v1

based on their content". ML techniques improve TC by automatically learning from big datasets and refining categorization models, resulting in increased efficiency, accuracy, and flexibility to varied data sources [53].

This Section provides a comprehensive overview of machine learning techniques for text classification, focusing on several key areas. It explores supervised learning methods, emphasizing the strategies for training models using labeled datasets. The section also delves into classifier construction, offering insights into various types of algorithms and their design for effective text categorization. Additionally, it highlights the importance of feature selection and engineering, discussing techniques to identify and refine features that enhance model performance. Lastly, it examines advanced approaches to text categorization, showcasing cutting-edge machine learning methods that improve accuracy and adaptability in classification tasks.

4.1. Supervised Learning Techniques

Supervised learning is the most common strategy in TC, in which labeled data is used to "teach" models how to effectively categorize texts. This method divides datasets "into three subsets: training, test, and validation sets". "The training set is utilized to build the model; the validation set fine-tunes hyperparameters and evaluates model performance during development; and the test set is reserved for final evaluation to ensure generalizability" [54] [55].

Using different datasets reduces the risk of overfitting, which occurs when a model performs well on training data but poorly on a different data [56]. Furthermore, effective partitioning ensures balanced and representative data, which is critical for applications like as sentiment analysis and document categorization, where certain terms and contexts must be learned consistently [44].

4.2. Classifier Construction and Types of Algorithms

The choice of algorithm greatly influences the building of classifiers for TC, as it dictates the model's learning strategy, interpretability, and processing needs [57]. Rule-based systems, decision trees, Naive Bayes, and neural networks are among the most used TC algorithms, each with its strengths and shortcomings when dealing with text input.

a. **Rule-based Systems:** These classifiers use handmade rules, which are highly interpretable but less flexible for complicated or huge datasets [44]. Rule-based systems, on the other hand, continue to be useful in situations when plain, transparent decision-making is required.
b. **Decision Trees:** Decision trees divide data based on certain criteria, making them intuitive and interpretable but susceptible to overfitting. Decision trees are effective for small to medium-sized text corpora, but they may struggle with scalability and feature depth [53].
c. **Naive Bayes:** Naive Bayes is frequently used in TC due to its simplicity, efficiency, and resilience, especially in document categorization and spam filtering [58]. However, while the assumption of feature independence simplifies calculation, it can reduce efficiency when features are highly linked [57].

 doi:10.20944/preprints202412.1304.v1

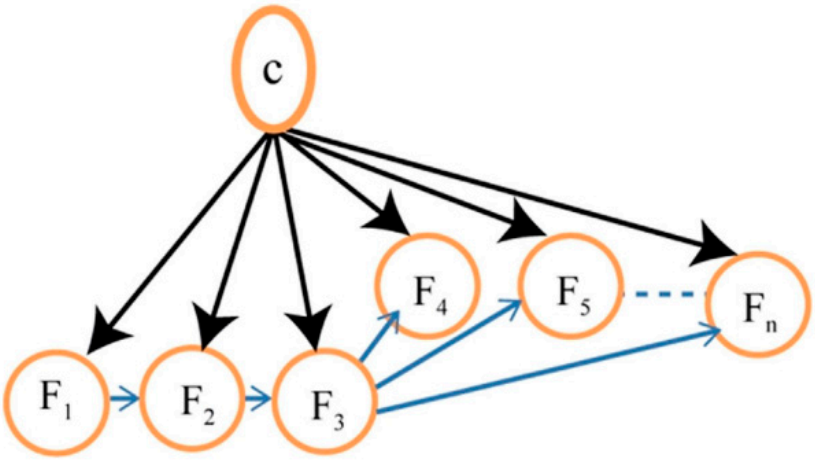

Figure 1. Naive Based Classification [59].

d. **Neural Networks:** "Neural networks, particularly deep learning models, have transformed TC by allowing them to learn sophisticated, hierarchical text representations. Although neural networks often need big datasets and significant computer resources, they provide unrivaled accuracy in capturing semantic meaning and contextual nuances" [60].

Each of these techniques is used depending on the use case, dataset features, and resource availability, emphasizing the importance of personalized ML approaches in TC.

4.3. Feature Selection and Engineering

Feature selection and engineering are critical in TC because they decide the data qualities the model learns from, which influences its overall performance [61]. Text classification features are often words, sentences, or semantic representations, therefore their selection is critical for increasing model accuracy and interpretability [56]. By focusing solely on important features, effective feature selection eliminates unnecessary data and computational expenses. "Term Frequency-Inverse Document Frequency (TF-IDF) and word embeddings are popular techniques for capturing the textual structure, context, and importance of words in texts". Furthermore, feature engineering approaches like stemming, lemmatization and n-gram analysis improve feature representation, hence enhancing classifier performance [44]. Furthermore, high-quality feature selection frequently results in improved generalization across domains and datasets, which is crucial for applications that require models to work in various languages or specialized disciplines [53].

4.4. Advanced Machine Learning Approaches to Text Categorization

The evolution of ML has seen a blend of traditional and advanced models enhancing TC accuracy and efficiency.

4.4.1. Traditional ML Techniques

- **Naive Bayes and Logistic Regression**: Offer simplicity and effectiveness in text classification, with Naive Bayes achieving up to 96.86% accuracy in specific datasets [30].
- **Support Vector Machines (SVM)**: Efficiently handle high-dimensional data and demonstrate strong performance with word embeddings.
- **Random Forest (RF)**: Achieves a mean accuracy of 99.98% when combined with Word2Vec embeddings [62].
- **K-Nearest Neighbors (KNN) and Decision Trees**: Useful for smaller datasets but less effective compared to SVM and RF [63].

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 December 2024 doi:10.20944/preprints202412.1304.v1

4.4.2. Deep Learning Approaches

- **Convolutional Neural Networks (CNNs)**: Capture spatial patterns in text, ideal for classification tasks.
- **Recurrent Neural Networks (RNNs)**: Particularly LSTMs and GRUs, excel at modeling sequential dependencies in text.
- **Transformer-Based Models**: Tools like BERT revolutionize TC by providing contextual embeddings, achieving over 97% accuracy in some applications [62].

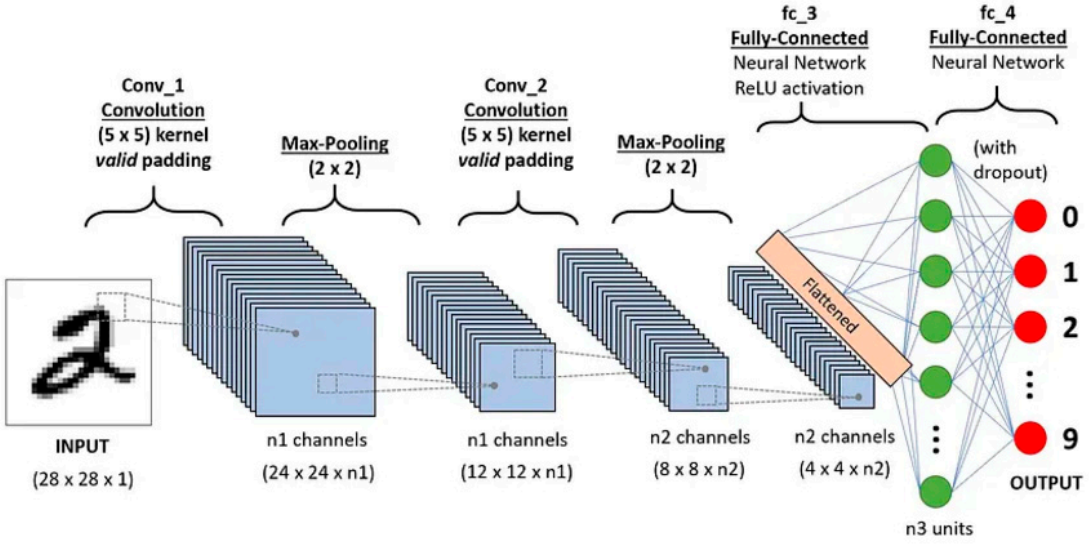

Figure 2. CNN Sequence to Classify Digits [64].

Figure 1 illustrates a Convolutional Neural Network (CNN) architecture designed for classifying handwritten digits, such as those found in the MNIST dataset. The process begins with an input image of size 28×28×128 \times 28 \times 128×28×1, representing a grayscale image of a digit. The first layer, **Conv_1**, applies convolution using a 5×55 \times 55×5 kernel with valid padding, producing n1n1n1 feature maps of size 24×24×n124 \times 24 \times n124×24×n1. This is followed by a **Max-Pooling** layer with a 2×22 \times 22×2 filter, reducing the feature map dimensions to 12×12×n112 \times 12 \times n112×12×n1.

Next, a second convolutional layer, **Conv_2**, uses another 5×55 \times 55×5 kernel with valid padding, resulting in n2n2n2 feature maps of size 8×8×n28 \times 8 \times n28×8×n2. This is again followed by max-pooling, which down samples the feature maps to 4×4×n24 \times 4 \times n24×4×n2. These outputs are then **flattened** into a one-dimensional vector and passed through two fully connected layers (**fc_3** and **fc_4**) with ReLU activations. Dropout is applied in the final fully connected layer to prevent overfitting.

The network outputs probabilities for 10 classes (digits 0 through 9) via softmax, effectively predicting the digit represented in the input image. This architecture highlights the sequential feature extraction and classification process of CNNs.

4.4.3. Hybrid and Ensemble Methods

- **Model Combinations**: Traditional classifiers paired with similarity measures like cosine similarity enhance performance [65].
- **Ensemble Learning**: Combines diverse models to boost robustness and accuracy in TC tasks [66].

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 December 2024 doi:10.20944/preprints202412.1304.v1

5. Document Representation Techniques

Document representation techniques translate text documents into structured formats that machine learning models can understand, to retain as much semantic information as feasible. Effective representation strategies support correct text classification, enhancing models' ability to recognize and analyzes key patterns in textual data [67].

This section explores document processing techniques, providing a detailed examination of foundational and advanced methods. Topics include the Vector Space Model (VSM) and its applications, the evolution from Bag-of-Words to more sophisticated approaches, and techniques in Lexical Semantics and Text Tokenization for understanding textual content. It also highlights Word Stemming and Stop Word Removal as essential preprocessing steps, along with a discussion on Weighting Schemes such as Term Frequency-Inverse Document Frequency (TF-IDF) and other innovative weighting strategies to enhance text analysis and classification.

5.1. Vector Space Model (VSM)

The Vector Space Model (VSM) serves as a fundamental tool for document representation in text categorization. It represents documents as vectors within a multidimensional space, where each dimension corresponds to a distinct term from the corpus.[68]. VSM enables the calculation of document similarity using metrics such as cosine similarity, which is useful in applications like clustering, search, and categorization.

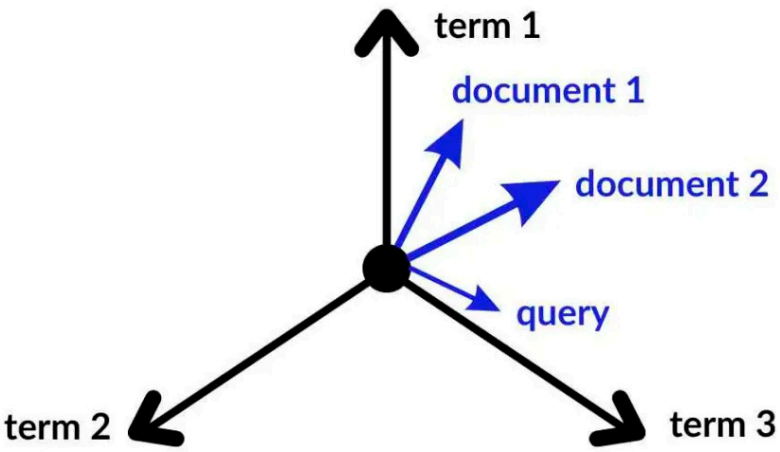

Figure 3. How the Victor Space Model Works [69].

5.2. Bag-of-Words and Beyond

"The Bag-of-Words (BoW)" approach in VSM relates to a simple but effective strategy that depicts each text as a gathering of individual phrases, disregarding word order but capturing the frequency of each term. BoW is computationally efficient and successful for many classification applications, but it has drawbacks, such as neglecting word order and semantic nuances [44]. To address these limitations, BoW extensions such as n-grams and distributed representations have arisen, which better capture word context and relationships by taking term sequences into account or employing embeddings [70]. These methods increase the semantic depth of document representation, making them appropriate for more complicated text analysis tasks.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 December 2024 doi:10.20944/preprints202412.1304.v1

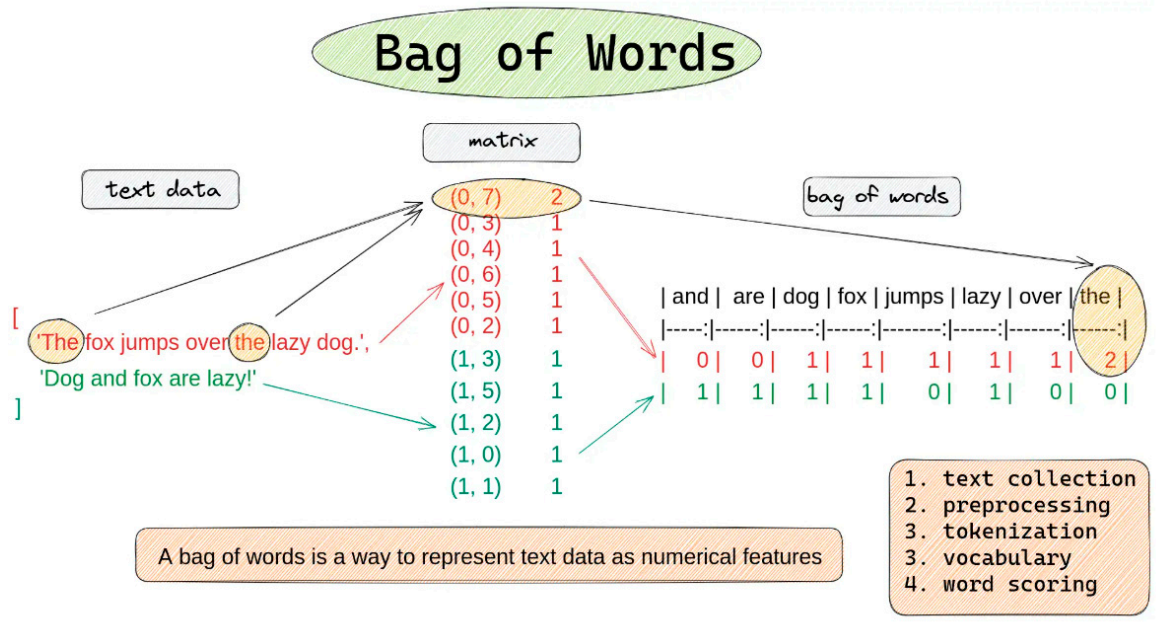

Figure 4. How Bag of Words Model Works [71].

5.3. Lexical Semantics and Text Tokenization

Lexical semantics, when paired with tokenization, divides text into meaningful units while preserving the document's fundamental information. Tokenization breaks down text into smaller components, typically words or phrases, allowing algorithms to handle text as discrete tokens rather than continuous strings [67].

5.4. Word Stemming and Stop Word Removal

Many TC applications rely heavily on stemming and stop word removal to improve document representation. Stemming reduces words to their base forms, grouping variations of the same term to prevent repetition in representations. For example, "running," "ran," and "runner" are all derived from "run." This simplification enables models to concentrate on key meanings, increasing efficiency and relevance in text analysis [72].

Stop word deletion entails removing common terms like "the," "is," and "and" which often add little to document classification. By omitting these keywords, models reduce computational complexity while increasing accuracy by focusing on more informative words. These preprocessing techniques are especially beneficial in fields where separating important phrases from popular ones is critical to accurate categorization [44].

5.5. Weighting Schemes

Weighting methods add importance to phrases inside a document, allowing the model to better discover key traits related to specific categories. Accurate weighting distinguishes phrases that carry significant information from those that do not, which improves classification results [73].

5.6. Term Frequency-Inverse Document Frequency (TF-IDF)

One of the most widely used weighting methods in text categorization (TC) is Term Frequency-Inverse Document Frequency (TF-IDF). This metric evaluates a word's significance within a text by considering its frequency within the document and its distribution across the entire corpus. The Term Frequency (TF) component highlights terms that occur frequently in a single document, while the Inverse Document Frequency (IDF) component downscales the weight of terms that are common

 doi:10.20944/preprints202412.1304.v1

across multiple documents. This approach ensures a more balanced representation of term importance [74]. TF-IDF has proven effective in various applications, such as document retrieval and categorization, by prioritizing unique and contextually significant terms [67].

5.7. Other Weighting Schemes

In addition to TF-IDF, various weighting techniques such as entropy weighting and BM25 have been investigated to better capture word significance across different contexts [75]. Entropy weighting, for example, assesses each term's informational contribution across categories, minimizing the impact of highly predictable phrases [76]. The BM25 technique, an extension of TF-IDF, provides an improved strategy for document retrieval tasks by integrating parameters that account for document length and frequency saturation, hence improving performance in big text corpora [77].

These weighting techniques address a wide range of text processing needs while also improving document representation flexibility, making them useful for TC applications that must deal with heterogeneous datasets and complex language patterns.

6. Dimension Reduction in Text Categorization

Dimensionality Reduction (DR) is a vital process in text categorization, aimed at addressing the challenge of high-dimensional feature spaces that often characterize text datasets. These datasets can consist of thousands or even millions of unique words, making the feature space complex and computationally intensive. DR techniques help by eliminating noisy or irrelevant terms, thereby enhancing training efficiency and model interpretability without compromising critical information[78]. Additionally, DR mitigates overfitting, a common issue where models are excessively tailored to the training data, hindering their ability to generalize to new, unseen data [1].

Methods such as Principal Component Analysis (PCA) and Latent Semantic Analysis (LSA) are commonly employed to lower dimensionality while retaining the structural and relational integrity of the text data. This streamlined representation facilitates faster processing and more accurate predictions, making dimensionality reduction an indispensable component of the machine-learning pipeline.

6.1. Importance of Dimensionality Reduction

Reducing dimensionality offers several key advantages:

1. **Improved Efficiency**: Streamlines computational demands, particularly during training and testing phases.
2. **Enhanced Interpretability**: Simplifies understanding by focusing on the most significant features.
3. **Reduced Overfitting**: Ensures the model learns generalizable patterns rather than noise specific to the training dataset.

These benefits collectively enable the creation of robust and reliable machine-learning models, empowering practitioners to derive meaningful insights from complex datasets. Practices such as PCA and t-Distributed Stochastic Neighbor Embedding (t-SNE) have proven effective in maintaining essential information while reducing dimensions, thereby improving model performance and the extraction of insights.

Dimensionality reduction techniques focus on identifying discriminative features, which are then weighted and fed into classifiers to construct models. During the testing phase, test documents are pre-processed and represented using the same methods applied during training. This ensures consistency in data handling, leading to more reliable predictions and deeper insights into underlying patterns within the dataset.

 doi:10.20944/preprints202412.1304.v1

6.2. Dimensionality Reduction in Support Vector Machines (SVMs)

Support Vector Machines (SVMs) benefit significantly from dimensionality reduction, particularly when handling high-dimensional data. The optimization process in SVMs relies on the dual formulation of soft margin SVMs, which transforms the primal optimization problem into a dual problem. This approach leverages kernel functions to efficiently handle non-linear classifications.

Kernel Functions in SVM

1. **Linear Kernel**

The linear kernel, represented as:

$$K(x, x\_i) = < x, x\_i >$$

This calculates the dot product of two feature vectors x and x_i. This kernel is straightforward and works well for linearly separable data, where a linear decision boundary can effectively separate the classes.

2. **Polynomial Kernel**

The polynomial kernel is expressed as:

$$K(x, x\_i) = [< x, x\_i > + \backslash beta]\hat{}d$$

where d is the degree of the polynomial and β is a constant. This kernel enables the SVM to capture more complex relationships between data points by considering polynomial interactions of features. The degree d determines the level of complexity in the model, with higher degrees capturing more intricate patterns.

3. **Gaussian RBF Kernel**

The Gaussian RBF kernel is given by:

$$K(x, x\_i) = \backslash exp(-\backslash gamma\ ||x - x\_i||\hat{}2)$$

where γ is a parameter that controls the kernel's flexibility and sensitivity to differences between data points. It maps the data into an infinite-dimensional space, allowing the SVM to create highly non-linear decision boundaries. The parameter γ determines how closely the model fits the data, with larger values resulting in tighter fits around individual data points and smaller values producing smoother decision boundaries [79].

While kernel functions mitigate the impact of high feature space dimensions on computational complexity, the dimensionality of the input space still influences kernel evaluations, especially for large datasets. The optimal hyperplane is derived using the equation:

$$f(x, \alpha^*, b)$$

This hyperplane is calculated using support vectors, kernel functions, and bias terms, enabling precise classification. Dimensionality reduction enhances SVM efficiency by reducing the computational overhead required for training and testing [79].

6.3. Text Representation in Dimensionality Reduction

In text categorization, documents are typically represented as a term-document matrix A=(aij)A = (a_[65])A=(aij), where:

- **Rows**: Represent terms.
- **Columns**: Represent documents.
- **Entries (aija_[65]aij)**: Indicate the frequency or presence of term iii in document jjj.

This matrix serves as the basis for clustering and classification tasks, with dimensionality reduction techniques applied to enhance efficiency and accuracy. By leveraging this representation, models can focus on essential features, enabling better performance in high-dimensional spaces.

 doi:10.20944/preprints202412.1304.v1

6.4. Common Methods for Term Selection

6.4.1. Document Frequency

This approach selects terms that appear frequently across documents, as frequent terms may have more importance for classification. However, common terms across all documents (like stop words) are typically excluded. The calculation involves determining the number of text within a collection that contains a specific feature, which may include words, phrases, n-grams, or custom-derived attributes. The counting approach employs a binary method: each time a feature is present in a document, its Document Frequency (DF) is incremented by one. However, this conventional DF metric focuses solely on the presence or absence of a feature in a text without accounting for the significance or relevance of that feature within the document itself. [80]. While the Document Frequency (DF) metric effectively quantifies the presence of features across a collection of documents, its binary nature overlooks the contextual importance of those features within individual documents. DF's simple presence/absence counting overlooks feature frequency and relevance variations within a document, leading to an incomplete representation of feature importance in tasks like text classification.

To address these limitations, the Term Frequency-Inverse Document Frequency (TF-IDF) metric is frequently employed, as it evaluates both the occurrence of a feature within a document and its distribution across the dataset. This results in a more accurate assessment of a feature's significance. TF-IDF is particularly useful for minimizing irrelevant terms in tasks like text summarization and classification [81]. As a commonly used feature weighting method in the vector space model, TF-IDF is widely applied in text mining and information retrieval. It effectively emphasizes the importance of a term within a document collection, treating all documents equally in its computation[82].

6.4.2. Chi-Square Test

This method evaluates the independence of a term from the document class, selecting terms that show a significant relationship with the target labels. A proposed goodness of fit test with an approximate chi-square distribution chi-square tests are used to assess many classes of comparison such as tests of independence and tests of homogeneity [83,84]. Tao and Chang also use the chi-square test to cluster web query schema [85]. The chi-square test, initially introduced by Pearson, has become a widely used statistical tool for assessing relationships between categorical variables, such as testing independence or homogeneity. Its application in clustering tasks, like grouping web query schemas, demonstrates the versatility of the chi-square test beyond traditional statistical analysis. By comparing observed and expected frequencies, the test helps uncover patterns or associations that might not be immediately apparent in raw data. In the context of web queries, chi-square tests can be used to cluster web query schemas based on their content. For example, if a search engine wants to categorize search queries into topics like sports, technology, or health, the chi-square test can assess the relationship between query words and the topics, helping to improve search accuracy. The experiments show that the proposed method improves the performance of text categorization techniques using Chi-Square ($\chi 2$) for feature selection with the F-measure of 92.20% [86].

6.4.3. Mutual Information

This approach evaluates how much information a term contributes to predicting the class label, prioritizing terms that offer the greatest value for classification. Terms are ranked based on their predictive significance, which can be assessed using techniques like document frequency, information gain, mutual information, or the χ^2-test. [86]. The core idea is that the most effective terms are those that exhibit the greatest variation in distribution between positive and negative examples across different categories. These techniques evaluate a term's ability to distinguish between categories effectively. [87]. Document frequency evaluates how frequently a term appears, while information gain measures its significance in predicting a category. Mutual information examines the association between a term and a category, and the chi-square test determines their

 doi:10.20944/preprints202412.1304.v1

independence. These methods aid in identifying the most important terms, enhancing Support Vector Machine (SVM) training by concentrating on critical features and patterns.

6.4.4. Term Clustering

This technique groups terms that are semantically similar, reducing redundancy in the feature set. By clustering similar terms together, the model can focus on clusters rather than individual terms, improving efficiency. Term clustering phrases derived from syntactic meta-features and indexed based on document or document group co-occurrence are typically of higher quality compared to indexing methods that rely solely on individual syntactic phrases, single indexing words, or word clusters. [88]. Term clustering differs from term selection in that it focuses on grouping terms that are synonymous or nearly synonymous, whereas term selection primarily aims to eliminate non-informative terms. [44]. The relationships identified within clusters are often incidental rather than the intended systematic connections originally sought. [88]. Optimization techniques have a wide range of applications, including clustering and categorizing text documents, engineering, image processing, speech recognition, pattern recognition, weather forecasting, route optimization, wireless sensor networks, and job scheduling, among others. [89]. Clustering terms based on syntactic relationships and co-occurrence patterns improves document indexing by capturing nuanced meaning and context. This approach reflects how words work together, rather than treating them as isolated terms. For example, in legal document retrieval, clustering terms like "contract terms" or "legal agreement" enhances search relevance and accuracy.

6.4.5. Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a linear dimensionality decrease method that projects data onto the most significant axes, known as principal components. It is a statistical technique designed to reduce dimensionality while minimizing the loss of variance from the original dataset. PCA identifies the directions of maximum variance within the term-document matrix, allowing for a reduction in the number of features while preserving the majority of the data's variance. This approach is especially valuable for managing sparse or high-dimensional datasets. It achieves this by transforming the initial correlated quantitative variables into new, uncorrelated variables known as principal components. [90]. PCA reduces dimensionality by calculating the covariance matrix to identify eigenvectors (principal components) that capture the highest change in the data. These components transform correlated features into uncorrelated ones, simplifying analysis and eliminating redundancy. PCA is widely used for visualization, improving machine learning performance, and handling high-dimensional datasets.

6.5. Comparison of Dimensionality Reduction Methods

6.5.1. Term Extraction Techniques

Document Frequency (DF) measures how often a term appears in a document collection. Terms that appear in a lot or fewer documents may not provide useful distinguishing information. This technique is widely used for filtering out common for example, stop words or rare terms. One of the widely used weighting schemes like TF-IDF (Term Frequency –Inverse Document Frequency) is used to convert a document into structured format [91].

The Chi-Square test measures the dependence between two categorical variables, such as the occurrence of a term and its corresponding document category. It identifies terms that are strongly associated with specific categories, aiding in feature selection. A higher χ^2 value signifies a stronger relationship between the term and the category. This test is computationally efficient and is commonly used to examine the independence of categorical variables or assess how well a sample aligns with the distribution of a known population (goodness of fit).[92].

Mutual Information (MI) quantifies the dependency between two variables (terms). Text analysis measures how much information one term provides about another, capturing both frequency and context. High mutual information values suggest that the term is informative and

 doi:10.20944/preprints202412.1304.v1

relevant to the target classification task. Estimating Mutual Information (MI) accurately is a complex task, and using it as an objective in representation learning often leads to highly entangled representations because of its invariance under arbitrary invertible transformations. However, despite these difficulties, MI-based methods have repeatedly proven to be highly effective in practical scenarios. [93].

6.5.2. Term Extraction Techniques

Term clustering organizes terms by analyzing their co-occurrence patterns or contextual similarities. By grouping terms that frequently appear together or share similar meanings, this approach facilitates the efficient identification of key features. It also helps to uncover semantically related term groups while minimizing redundancy in the feature set.

Principal Component Analysis (PCA) is a statistical technique used for dimensionality reduction. It transforms a large set of correlated features into a smaller set of uncorrelated features, called principal components, while retaining as much of the original variance (information) as possible. This method is especially useful for high-dimensional text data, as it captures and emphasizes the most significant variations within the dataset.

7. Evaluation of Text Categorization Models

Text categorization is a key task in Natural Language Processing (NLP). The aim of text categorization methods is to associate one (or more) of a given set of categories to a particular document [94]. Evaluating the performance of text categorization models is crucial for understanding their effectiveness and ensuring they perform well in real-world applications. Evaluating the performance of a text categorization model involves the use of various metrics. This section discusses the evaluation of text categorization models, focusing on performance metrics, the F-Measure, and challenges associated with model evaluation.

7.1. Metrics for Performance Evaluation

These metrics are used to assess the model's effectiveness in accurately classifying text into the appropriate categories. Various evaluation measures are commonly employed, such as recall, precision, accuracy, error rate, F-measure or break-even point, micro-average and macro-average for binary classification, and 11-point average precision for ranking categories. [95].

a. **Accuracy**

Accuracy is the ratio of correctly foreseen instances to the total instances in the dataset. While simple and widely used, it can be misleading in imbalanced datasets.

$$\text{Accuracy} = \frac{True\ Positive + True\ Negatives}{Total\ Samples}$$

In text categorization (such as classifying documents into multiple categories or topics), we evaluate model performance using metrics like accuracy (how often the model predicts the correct category) or error rate (how often the model is wrong). However Yang [95].points out key issues when applying these metrics to certain datasets. As a result, a simplistic algorithm that rejects all documents for every category would achieve a global average error rate of 1.3% and a global average accuracy of 98.7%, whether measured on a micro or macro scale, as both values would be identical.[55]. This does not imply that a trivial rejector classifier is effective; rather, it highlights that accuracy or error alone may not be reliable metrics for evaluating the performance or utility of a classifier in text categorization, especially when the number of categories is large, and each document is associated with only a small number of categories on average [95]. A trivial classifier refers to a model that generates basic, non-informative predictions. Selecting an appropriate performance evaluation metric becomes especially critical when dealing with advanced machine learning methods, such as neural networks, to ensure meaningful and accurate assessments of their predictive capabilities. [96]. In this context, the trivial approach refers to a classifier that rejects all documents for every category. Alternatively, a predictor-rejector formulation involves learning both a predictor

 doi:10.20944/preprints202412.1304.v1

and a rejector, each derived from distinct families of functions, while explicitly considering the cost of abstaining from making a prediction. [97]**.** In simpler terms, this model consistently predicts that no categories are assigned to any document, earning it the label of a "rejector classifier." Despite failing to perform any meaningful classification, the rejector classifier could achieve a global accuracy of 98.7%, primarily because many documents in the dataset have very few assigned categories, making them irrelevant to most. While accuracy can be a reliable metric when positive and negative examples are balanced, it becomes misleading in imbalanced scenarios. For instance, if negative examples significantly outnumber positive ones, a system that assigns no documents to any category can still achieve an accuracy value close to 1, even though it provides no useful[97].

b. **Precision**

Precision (also called Positive Predictive Value) measures the accuracy of positive predictions. It is the ratio of true positives to the total predicted positives.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positives} \times 100 \text{ [98]}$$

Precision plays a critical role in scenarios where the cost of false positives is significant, such as spam detection, where misclassifying a legitimate email as spam can lead to undesirable outcomes. In the field of information retrieval, precision refers to the percentage of retrieved documents that are relevant, while recall represents the percentage of relevant documents successfully retrieved from the total set of relevant documents [99]. Studies have reported impressive results, with recall and precision averaging around 90% on a small subset (3%) of a specific corpus[44]. It is noted that micro-averaged scores (recall, precision, and F1) are predominantly influenced by the classifier's performance on frequently occurring categories, whereas macro-averaged scores are more impacted by performance on less common categories [95]. Precision becomes especially important in high-cost error cases, such as spam detection, where the misclassification of non-spam emails as spam can have significant repercussions.

c. **Recall (Sensitivity)**

Recall (also called Sensitivity or True Positive Rate) actions how well the model identifies all relevant instances. It is the ratio of true positives to the total actual positives.

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negatives} \times 100 \text{ [98]}$$

Recall is crucial in situations where the cost of false negatives is high, such as in medical diagnostics, where failing to detect a positive case could have serious consequences. Recall is defined as the ratio of correctly identified positive cases to the total number of actual positives. This measure evaluates the system's ability to identify true positives, with average performance sometimes assessed across different recall thresholds for all test documents [95]. It is particularly significant in cases where missing a positive diagnosis could result in severe outcomes, emphasizing the importance of capturing all relevant instances.

F-Measure

The F1-measure serves as the harmonic mean of precision and recall [98], providing a balanced evaluation of both metrics. It is especially valuable in scenarios with an uneven class distribution, where balancing false positives and false negatives is critical, such as in text classification tasks. The F1-measure is calculated as follows:

$$\text{F-Measure} = \frac{Precision \times Recall}{Precision + Recall}$$

The F-Measure is commonly employed when achieving a balance between precision and recall is important, such as in text classification tasks where it is necessary to minimize both false positives and false negatives. However, designing an appropriate significance test can be challenging, as the method’s performance is often summarized into a single metric, like the break-even point or the optimized F1 score [95]. Additionally, optimizing predictions to maximize the F1-measure is not always feasible by merely ranking labels based on their relevance and selecting the highest-ranked ones[100].

 doi:10.20944/preprints202412.1304.v1

Table 2. Keyword Statistics.

Assign Mean	Corr. Mean	P	R	F
8.6	3.6	41.5	46.9	44.0

Table 1: Table 1 summarizes the mean number of assigned (Assign.) keywords and correct (Corr.) keywords per document, as well as the precision (P), recall (R), and F-measure (F) achieved when extracting 312 keywords per document [101].

d. **Breakeven Point (BEP)**

The Breakeven Point represents the point where precision and recall are equal, providing insight into the trade-off between these metrics. Typically, BEP values are interpolated because exact matches of precision and recall are rare. When precision and recall values are significantly different, BEP may yield results that the system cannot achieve. Additionally, the point where precision equals recall is not always meaningful or desirable from the user's perspective [97].

BEP = where Precision = Recall

This also means that the BEP score of a system is always equal or less than the optimal value of F1 of that system [95]. The BEP score is a more lenient metric than F1, meaning it cannot exceed the optimal F1 score, which balances precision and recall.

7.2. Validation Techniques

Effective validation techniques are critical to evaluate how well a model performs on unseen data. The rapid growth of digital text data has necessitated the development of new methods for text processing and classification [89].

a. **k-Fold Cross-Validation**

k-Fold cross-validation is a widely-used method for assessing a classification algorithm's performance or comparing multiple algorithms. It splits the dataset into k. subsets (folds) of approximately equal size. Each fold serves as a testing set once, while the remaining k−1k-1k−1 folds are used for training[102]. This approach ensures that all data points are used for both training and validation, making it particularly useful for small to medium datasets.

In k-Fold cross-validation, the dataset is randomly partitioned into k-folds, with each fold being used once as a test set. As k increases, the evaluation becomes more stable by averaging the results over more models. However, increasing k. also requires training more models, making it important to choose an appropriate k. value [103]. This method is especially useful in fields like healthcare, where it helps assess classification model performance with limited datasets [104].

Train-Test Split

A simpler validation approach is the Train-Test Split, which divides the dataset into two parts: a training set for developing the model and a test set for evaluating its performance. A common split ratio is 80% training and 20% testing, although this may vary. Train-Test Split is often used in meta-learning, where models are adapted to specific tasks using one subset of data and evaluated on another [105]. While the Train-Test Split trains a single model, cross-validation improves generalization by training multiple models on different data subsets. K-Fold cross-validation is one of the most popular approaches for addressing the limitations of small datasets, as it allows every data point to be used in both training and validation by rotating folds between training and testing phases [106].

7.3. Challenges in Model Evaluation

a. There are only a few lexical databases for a small number of languages, hence knowledge-based systems can be developed only for those languages. Knowledge-based systems are mostly specific in nature for certain languages and subjects, so they cannot easily be used for other languages. These systems can be costly to maintain since languages keep changing. They are

 doi:10.20944/preprints202412.1304.v1

also not available for some subjects. [89]. Knowledge-based systems rely on lexical databases, which are limited to a few languages and domains, making them costly and hard to adapt. Researchers are urged to develop these resources for under-represented languages to expand system usability.

b. Building and implementing a deep learning-based system can be highly resource-intensive, as training such systems requires expensive hardware and significant computational power, which must be accounted for. [89].
c. The meaning relationships of the words in a text document give problems in text categorization, hence making it hard to create a system. Unorganized text data is a tough job for getting meaning relationships to make text categorization systems. [89].

8. Challenges in Machine Learning-Based Text Classification

This section examines the challenges in machine learning-based text classification, addressing critical issues such as overfitting and underfitting, which impact model generalization; class imbalance, which skews classification results; feature space complexity, which complicates model training and interpretation; and linguistic challenges like ambiguity and polysemy, which hinder accurate text understanding and categorization.

8.1. Overfitting and Underfitting in TC

Overfitting and underfitting pose major challenges to the quality of classification models. Overfitting occurs when a model learns excessively, including noise, leading to excellent performance on training data but poor generalization to unseen data. Both overfitting and underfitting can cause training errors that significantly impact the reliability of deep learning-based communication systems [107]. Regularization, dropout layers, and data augmentation are techniques that help to prevent overfitting by balancing model complexity and lowering sensitivity to certain parameters. The process of this problem is called generalization, and generalization mainly solves the problem of overfitting [108]. Underfitting happens when a model is overly simplistic in capturing relevant data patterns, resulting in poor performance on both training and test data. Underfitting in TC might occur because of the use of basic algorithms or insufficient feature extraction. This lack of depth inhibits the model from comprehending linguistic complexity and themes. To combat underfitting, increase model complexity, use advanced topologies such as transformers or pre-trained models, and include a wide range of data points. To solve these concerns, regularization, and dropout avoid overfitting while adding layers or pre-trained models prevents underfitting. These changes allow TC models to generalize more successfully, resulting in accurate classification across a wide range of text formats [109].

8.2. Class Imbalance in TC

The class imbalance problem in text categorization (TC) occurs when certain categories dominate a dataset, while others are underrepresented. This imbalance might cause machine learning models to favor majority classes, resulting in biased predictions. This is especially troublesome in applications such as spam detection or sentiment analysis, where minority classes are important. Addressing class imbalance is critical to ensuring TC models' robustness and fairness. The first challenge is multi-class imbalance: the Rapidly Intensifying (RI) and Extraordinarily Intensifying (EI) classes have significantly fewer training samples in comparison with the Neutral and Weakening classes [110]. Class imbalance is frequently the result of natural data distribution. Sports and politics, for example, may have significantly more data than specialty fields such as environmental news, particularly in user-generated content or real-time applications. An imbalance in class distribution skews models toward the majority class, reducing their ability to generalize effectively across different scenarios. To address this, data-level methods like SMOTE (Synthetic Minority Over-sampling Technique) are used to balance the dataset by oversampling minority classes and undersampling majority classes. However, classifiers trained under increasingly imbalanced conditions and evaluated similarly may show a deceptive improvement in classification accuracy.

 doi:10.20944/preprints202412.1304.v1

[110]. Algorithmic approaches modify the learning process by allocating higher weights to minority classes, with techniques such as boosting and bagging being useful. Advanced models such as BERT and GPT, through fine-tuning and cost-sensitive learning, aid in minority class recognition in highly skewed datasets [111].

8.3. Complexity in Feature Space

Gaining a deeper understanding of the distribution of patterns within the feature space can provide valuable insights into the difficulty and complexity of various classification tasks. [112]. The feature space in text categorization (TC) refers to the structured dimensions or variables used to process text input for machine learning. Because text is unstructured and contains words, sentences, and syntax, encoding it numerically results in a high-dimensional feature space. This intricacy can make it difficult for models to train successfully, resulting in significant computing costs and the danger of overfitting. To address these difficulties, feature selection and dimensionality reduction approaches can assist manage feature space complexity while retaining critical information [40].

Classifiers trained on datasets with increasing levels of class imbalance and evaluated under the same conditions often exhibit an artificially inflated classification accuracy, which can be misleading. [112]. The high complexity of text data raises computational demands and makes it difficult to differentiate relevant aspects. Feature space, like the physical Universe, is very sparsely populated [112]. Sparse data points in a large feature space can make generalization difficult and increase training time. Simple representations, such as bag-of-words, may fail to express linguistic nuances, especially when dealing with polysemy and synonyms. A higher dimensional feature space is required to cope with this more complex situation [113]. Feature engineering is critical to make text data more manageable and understandable. Methods like term frequency-inverse document frequency (TF-IDF) and n-grams help models identify important terms and phrase structures. Word embeddings, including Word2Vec, GloVe, and fastText, provide compact, dense representations that enhance generalization across related concepts. More advanced embeddings, such as BERT and GPT, go further by generating contextualized representations that capture the meanings of words based on their surrounding context. [114].

Dimensionality reduction techniques such as Principal Component Analysis (PCA), Singular Value Decomposition (SVD), and autoencoders condense the feature space by preserving only the most significant features. This not only enhances model interpretability but also reduces training time, making the models more efficient. Modern embedding models such as BERT and GPT improve TC by incorporating contextual nuances, increasing model accuracy for complex languages. While these developments improve TC, they also raise interpretability concerns. Deep learning models and embeddings are frequently viewed as "black boxes," which is especially troublesome in industries requiring explanation, such as healthcare or finance. Attention mechanisms and explainable AI (XAI) tools help to emphasize significant elements while balancing feature complexity and interpretability, allowing practitioners to make educated decisions in complicated language processing tasks [115].

8.4. Ambiguity and Polysemy in Language

Ambiguity and polysemy provide substantial issues in natural language processing (NLP), particularly in tasks such as text categorization. Ambiguity occurs when a term or phrase has many meanings, such as "bank" referring to a financial organization or a riverbank. Polysemy is a type of ambiguity in which words have multiple related meanings, such as "run" for physical exercise or executing a program. These phenomena hamper model performance because they require context for accurate interpretation, which standard models struggle with. Ambiguity creates confusion in TC, where context is critical for accurate classification [116]. For example, a headline like "local bank raises funds" requires contextual expertise to discern between financial and non-financial issues. Simple models frequently misclassify such scenarios, and even neural models such as transformers can fail when contextual cues aren't apparent or need cultural knowledge, emphasizing the importance of advanced context handling strategies [117].

 doi:10.20944/preprints202412.1304.v1

Polysemy is especially difficult since static word embeddings cannot record multiple meanings across contexts. Words like "light" can relate to either brightness or weight, depending on the context. Contextual embeddings, such as those used in BERT and GPT, address this by dynamically modifying meanings based on surrounding words, although complex phrases and nuanced interpretations continue to pose issues. Multilingual NLP complicates TC by varying ambiguity and polysemy across languages. Some languages use morphology to resolve ambiguity, while others rely significantly on context, which complicates operations like machine translation. To deal with these challenges, multilingual models such as mBERT are trained on a variety of datasets, although linguistic diversity still presents limits [118].

There are several ways for dealing with ambiguity and polysemy. Domain-specific models improve context and reduce misclassification, while auxiliary tasks such as part-of-speech tagging help clarify meaning. Ensemble models, which incorporate predictions from many models, improve overall performance. Although effective, these techniques are computationally expensive, demonstrating that ambiguity and polysemy remain key issues in NLP [119].

Categorization (TC)

9. Advancements and Emerging Trends in Text

Recent advances in TC reflect a paradigm shift away from traditional machine learning methods and toward deep learning and hybrid methodologies. These advancements enable better feature extraction, contextual comprehension, and flexibility across languages and domains, broadening the scope of TC's practical applications. This section explores deep learning approaches for text categorization, focusing on the application of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) for various text classification tasks. It also highlights the transformative impact of transfer learning and pre-trained language models, such as BERT and GPT, in advancing text categorization with contextual understanding and reduced training requirements.

9.1. Deep Learning for Text Categorization

With its capacity to identify complex patterns in high-dimensional data, deep learning has transformed text classification by allowing models to learn directly from raw text with minimal feature engineering. “Deep learning algorithms, particularly Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) currently have exhibited significant promise in text classification tasks, excelling at capturing local and sequential relationships”.

9.2. CNNs and RNNs for TC Tasks

CNNs and RNNs are two of the most popular TC architectures due to their distinct ability to process and comprehend textual data. CNNs, which have typically been employed in image processing, have been adapted for text classification by applying convolutional filters on word embeddings or n-gram representations. This technique recognizes local word patterns and is especially beneficial for short text categorization, such as sentence-level sentiment analysis [120]. CNNs' hierarchical feature extraction technique finds relevant phrases and concepts, making them ideal for context-dependent document classification tasks [121].

“RNNs, notably Long Short-Term Memory (LSTM) networks, have also proven useful for TC because of their sequential character, allowing them to effectively model dependencies across phrases and paragraphs” [122]. Recurrent Neural Networks (RNNs) are a type of neural network architecture which is mainly used to detect patterns in a sequence of data [123]. The sequential learning capabilities of these models is especially useful in TC tasks that need large documents with complicated language structures.

9.3. Transfer Learning and Pre-Trained Language Models

Transfer learning, particularly through pre-trained language models, represents a significant advancement in text classification (TC). By leveraging knowledge from vast and diverse text corpora,

 doi:10.20944/preprints202412.1304.v1

it reduces the reliance on extensive labeled datasets, thereby enhancing the accessibility of text classification for low-resource languages and niche domains

9.3.1. Use of BERT, GPT, and Similar Models

Pre-trained language models like BERT (Bidirectional Encoder Representations from Transformers), GPT (Generative Pre-trained Transformer), and related architectures have revolutionized text classification (TC). These models are pre-trained on extensive corpora and can be fine-tuned with minimal additional training for specific tasks, setting new benchmarks in performance and efficiency. BERT, for example, uses a bidirectional attention mechanism to record the context of words from both left and right contexts, resulting in more nuanced understanding in TC applications" [124]. BERT's deep bidirectional methodology makes it particularly successful for context-dependent tasks like sentiment analysis and topic classification.

GPT, on the other hand, employs a unidirectional transformer architecture, excelling at producing coherent, contextually relevant text and doing well on tasks requiring text production or completion [125]. For TC, GPT and its descendants, such as GPT-3, have demonstrated exceptional performance in few-shot and zero-shot classification scenarios, decreasing reliance on labeled data and facilitating fast knowledge transfer between languages and domains [126].

The introduction of these models significantly improved TC capabilities, allowing classifiers to function with minimum task-specific input while maintaining high levels of accuracy. Their efficacy across a variety of TC applications demonstrates transfer learning's promise for dealing with complicated and developing text collections.

9.3.2. Hybrid Approaches Combining Knowledge Engineering and ML

Hybrid approaches that integrate knowledge engineering with machine learning are gaining traction, effectively bridging the gap between rule-based systems and data-driven methods. A SWOT analysis of the ten most frequently cited algorithms from the collected database highlights the strengths and weaknesses of traditional algorithms while uncovering the opportunities and challenges that hybrid methods aim to address [127]. These methods incorporate human-defined rules and domain expertise into machine learning models, enhancing the interpretability and robustness of text classification (TC) systems.

In recent years, other hybrid physics-ML models have been developed, extending beyond residual modeling. A simple method to integrate physics-based and ML models involves using the output of a physics-based model as input for an ML algorithm [128]. Within hybrid TC systems, knowledge engineering is often applied to create initial feature sets or rules that feed into machine learning algorithms. For instance, domain-specific ontologies or taxonomies can guide feature selection, enabling the model to capture critical semantic details relevant to the categorization task. This approach is particularly effective in specialized fields such as healthcare or legal document categorization, where domain expertise is crucial for achieving accurate classification [35].

10. Future Directions and Research Opportunities

10.1. Multi-Language and Cross-Cultural Text Classification

10.1.1. Importance of Cross-Language Communication

In today's interconnected global landscape, seamless cross-language communication is essential. As language diversity persists as a barrier, domains like multilingual translation and text summarization are reaching a critical juncture, requiring innovative automated solutions [129]. Text classification models, which often rely on large-scale labeled datasets, are typically tailored for specific languages and cultural contexts. This limitation underscores the growing demand for systems capable of addressing linguistic and cultural diversity in an increasingly interconnected world [130].

 doi:10.20944/preprints202412.1304.v1

10.1.2. Advancements in Multilingual NLP

New multilingual datasets featuring conversations in Chinese, English, Korean, and Japanese provide a robust foundation for developing powerful conversational AI systems [130]. Pre-trained models like BERT have expanded their capabilities to include multilingual versions such as mBERT and XLM-R. These models enable simultaneous processing of diverse linguistic inputs, enhancing cross-language text classification [131].

10.1.3. Cross-Lingual Transfer Learning

Cross-lingual transfer learning, facilitated by both social and machine translation, plays a pivotal role in multilingual text classification. Many multilingual datasets are generated through professional translations, while machine translation is frequently employed to translate training or test sets. Despite these advancements, challenges remain, such as the lack of standardized multilingual datasets annotated under consistent guidelines, particularly for intent detection and slot filling tasks[132,133].

10.1.4. Cultural Sensitivity in Text Classification

Text classification systems must navigate cultural nuances, including idiomatic expressions, societal norms, and sentiment variations across regions. For example, positive or neutral sentiment expressions can differ significantly between cultures, affecting sentiment analysis accuracy. Translators must ensure cultural appropriateness, preserving the natural tone and relevance for the target audience.

10.1.5. Future Research Directions

Universal Multilingual Models

Developing generalized models capable of learning across multiple languages with minimal reliance on labeled data is a critical research direction. Universal multilingual models such as XLM-R and mBERT have laid the groundwork, but further advancements are needed to enhance their adaptability to low-resource languages and diverse linguistic contexts. By leveraging transfer learning, cross-lingual embeddings, and domain adaptation, these models can facilitate effective communication and analysis across linguistic and cultural barriers.

Low-Resource Languages

Addressing data scarcity in low-resource languages remains a significant challenge. Techniques such as unsupervised learning, self-supervised approaches, and domain-specific transfer learning can mitigate these limitations. For instance, multilingual pre-trained models can be fine-tuned for precise low-resource languages, enabling their inclusion in broader applications and ensuring global inclusivity. Integrating machine translation and text classification duties could also enhance the usability of these models in multilingual environments.

Enhanced Language Identification

Future text categorization systems must incorporate advanced language identification techniques to process user-generated content that often includes multiple languages. Methods such as combining deep learning with linguistic rules can improve accuracy in detecting and processing code-switching and mixed-language texts. This capability is essential for applications in social media monitoring, global marketing, and multilingual customer support, where accurate language identification is critical [134]

Cultural Awareness in Models

Embedding cultural sensitivity into text categorization models is vital for improving their classification accuracy and relevance in diverse contexts. Cultural nuances, idiomatic expressions, and societal norms influence language usage and sentiment expression, which models must understand to perform effectively. Incorporating cultural awareness into training data and leveraging cross-cultural embeddings can enhance the adaptability and inclusivity of these systems.

 doi:10.20944/preprints202412.1304.v1

Integration with Real-Time and Multimodal Systems

The integration of text categorization with real-time processing and multimodal systems is another promising research avenue. Real-time categorization systems must handle dynamic data streams with minimal latency while maintaining accuracy. Combining text with visual and audio inputs, such as in social media content analysis, could provide richer contextual understanding and enhance classification outcomes. Edge computing and incremental learning techniques can support this shift toward dynamic, real-time systems.

Ethical AI and Bias Mitigation

Addressing ethical concerns, such as bias in training data and algorithms, will remain a priority for future research. Developing transparent and explainable AI (XAI) models will foster trust and accountability in text categorization systems, particularly in sensitive applications like recruitment, healthcare, and legal analytics. Ethical frameworks must be incorporated into model design to ensure fair and unbiased outcomes across diverse demographic and cultural contexts.

Hybrid and Explainable Models

Combining machine learning with rule-based systems offers a promising avenue for creating interpretable and robust text categorization models. Hybrid models can balance precision and transparency, making them more suitable for high-stakes applications. Explainable AI approaches will play a crucial role in enabling users to recognize and trust the decision-making processes of these copies.

By addressing these research directions, the next generation of text categorization systems can achieve greater inclusivity, adaptability, and ethical integrity. These advancements will not only refine technical performance but also ensure that text categorization technologies remain relevant and impactful in an increasingly interconnected and data-driven world.

10.2. Real-Time Text Categorization Applications

10.2.1. The Need for Real-Time Classification

Real-time text categorization enables immediate processing and classification of newly generated content, bypassing the need for batch operations. This capability is critical for applications such as social media monitoring, content filtering, and customer support, where real-time decision-making is essential [135].

10.2.2. Scalability and Speed in Real-Time Systems

The high volume and rapid generation of content on social media and news platforms demand systems that are both fast and scalable. For instance, integrating report texts with tweets containing relevant links has been shown to improve classification outcomes in real-time environments [136].

10.2.3. Incremental Learning for Dynamic Content

Real-time systems thrive in dynamic environments by employing incremental learning techniques. These approaches allow models to continuously adapt to new data, enhancing their robustness in ever-changing contexts. Lifelong learning frameworks provide methods for task-incremental, domain-incremental, and class-incremental learning, bridging the gap between natural and artificial intelligence [137].

10.2.4. Reducing Latency While Preserving Accuracy

Reducing latency is a critical requirement for real-time systems. Techniques such as knowledge distillation, edge computing, and model optimization help minimize computational demands while maintaining high accuracy, even on resource-constrained devices [50]

 doi:10.20944/preprints202412.1304.v1

10.2.5. Future Research Opportunities

High-Throughput Systems

Developing models capable of processing large-scale, real-time data streams with minimal latency remains a top priority. Future systems must leverage advanced technologies such as edge computing, model distillation, and parallel processing to handle massive volumes of data without sacrificing accuracy. High-throughput systems can play a critical role in applications like live news categorization, stock market analysis, and emergency response, where rapid decision-making is essential.

Dynamic Adaptation

Real-time content is highly dynamic, with patterns and trends shifting quickly. To maintain relevance and accuracy, it is essential to enhance models to adapt to these changes. Incremental learning techniques, which enable models to update and evolve without requiring complete retraining, offer a particularly effective solution. These methods can be combined with continual learning frameworks to create systems that seamlessly adjust to new topics, terms, and contexts over time.

Applications in Diverse Domains

Real-time text categorization offers significant potential across various fields:

- **Social Media Analytics**: Identifying trends, sentiment, and emerging topics in real-time.
- **Spam Detection**: Filtering spam messages or malicious content as they are generated.
- **Fraud Prevention**: Monitoring financial transactions or communications for suspicious patterns.
- **Customer Support Chatbots**: Providing instant, context-aware responses to user queries.

Real-Time Multimodal Integration

Combining text with other data modalities, such as images, videos, and audio, presents an exciting research direction. For instance, analyzing text alongside accompanying visuals in social media posts could provide richer insights into user intent and sentiment. Multimodal approaches will be critical for applications like live event monitoring and personalized content delivery, where a holistic understanding of data is necessary.

Latency-Aware Optimization

Reducing latency while preserving accuracy is a critical challenge for real-time systems. Research into lightweight model architectures, optimized inference algorithms, and energy-efficient processing will be essential to support latency-sensitive applications. Techniques like knowledge distillation and pruning can make models more efficient without compromising performance, particularly for deployment on resource-constrained devices.

Scalability for Global Applications

With the growing global nature of data, scalable systems capable of processing multilingual and culturally diverse content in real-time are needed. Advances in cross-lingual embeddings, transfer learning, and domain adaptation will enable models to handle diverse data streams efficiently. This scalability is particularly important for global platforms that deal with multilingual user bases, such as international social media networks and e-commerce platforms.

Context-Aware Personalization

Future systems should aim to provide personalized categorizations by incorporating user preferences, location, and historical interactions. Context-aware models can improve the relevance and utility of real-time classifications in applications like targeted marketing, personalized news feeds, and adaptive recommendation systems.

Ethical Considerations and Transparency

As real-time systems are deployed in sensitive areas, ensuring ethical AI practices and transparency will be crucial. Future research should focus on developing frameworks for bias detection and mitigation, as well as explainable AI techniques that allow users to understand how

 doi:10.20944/preprints202412.1304.v1

real-time categorization decisions are made. This is particularly important for applications involving content moderation and public safety.

10.3. Integration with Other NLP Tasks

10.3.1. Expanding the Scope of NLP Integration

Integrating various NLP tasks such as named entity recognition (NER), parsing, sentiment analysis, and information extraction into text classification can significantly improve system performance. These tasks enable models to derive deeper insights from textual data, supporting more complex applications.

10.3.2. Named Entity Recognition (NER)

NER identifies entities like names, locations, and organizations within the text, enhancing classification accuracy for domain-specific tasks such as medical or legal document analysis. This task is critical for structured data extraction in applications like information retrieval and question answering [138,139].

10.3.3. Parsing Techniques

Parsing systems analyze sentence structure and relationships between words, aiding models in understanding both syntactic and semantic nuances. These insights enable more accurate distinctions between text types, such as formal articles versus informal blog posts [139].

10.3.4. Information Extraction (IE)

IE techniques automatically identify structured data within unstructured text. This functionality is particularly useful in applications like legal document analysis and automated data entry, where structured outputs are crucial[140].

10.3.5. Multi-Task Learning Frameworks

Multi-task learning involves training models to handle several NLP tasks simultaneously, leading to richer feature representations and improved overall performance. For example, integrating text summarization and sentiment analysis within a single model can yield more nuanced outcomes [63,141].

10.4. Advancing Multimodal Text Classification

10.4.1. Combining Modalities for Comprehensive Analysis

Multimodal classification combines textual data with other data types, such as images, videos, or audio, providing a holistic understanding of user-generated content. For example, social media platforms can analyze both text and accompanying images to classify posts more effectively.

10.4.2. Practical Applications

- **E-Commerce**: Platforms can integrate sentiment analysis and NER to classify product reviews, extract brand mentions, and monitor customer feedback in real-time.
- **Social Media**: By combining text-based sentiment analysis with image-based emotion detection, platforms can enhance their content moderation and analytics capabilities.

10.4.3. Future Research Directions in Multimodal Text Classification

The addition of multiple facts modalities in text classification opens up new opportunities for advancing the field. Beyond the current applications in e-commerce and social media, innovative research can explore the following directions:

 doi:10.20944/preprints202412.1304.v1

10.4.4. Dynamic Multimodal Fusion Techniques

Future research should focus on developing advanced techniques for dynamically fusing multimodal data. This includes creating adaptive models that can weigh the importance of text, images, videos, and audio based on the context of the task. For instance, a news categorization system might prioritize textual content for breaking news and image content for photojournalism.

10.4.5. Temporal Multimodal Analysis

Investigating the temporal aspects of multimodal data, such as analyzing how user sentiment evolves over time across different modalities, could be a valuable direction. This is particularly relevant for applications like campaign monitoring, where text, images, and videos are generated sequentially and provide evolving narratives.

10.4.6. Real-Time Multimodal Interaction

Building systems capable of real-time multimodal interaction presents an exciting challenge. For instance, integrating live video feeds with chat-based textual input can enhance virtual events, online education, and telemedicine. These systems would need to process and classify data across modalities simultaneously, ensuring high responsiveness and accuracy.

10.4.7. Cross-Modal Transfer Learning

Future work could explore cross-modal transfer learning, where knowledge from one modality (e.g., textual embeddings) is transferred to another (e.g., image features) to improve performance. This approach can be particularly effective in domains where one modality has abundant labeled data while another is scarce.

10.4.8. Domain-Specific Multimodal Solutions

Developing domain-specific multimodal frameworks tailored to fields like healthcare, finance, or legal analysis can drive significant progress. For instance:

- **Healthcare**: Analyzing patient notes alongside medical images for enhanced diagnostic accuracy.
- **Finance**: Integrating financial reports (text) with market trend graphs (visuals) to improve investment decision-making.
- **Legal Analysis**: Combining contract text with associated diagrams or annotations to classify clauses efficiently.

10.4.9. Augmented Reality (AR) and Virtual Reality (VR) Integration

As AR and VR applications grow, research could focus on integrating multimodal text classification into these environments. For example, AR systems could analyze spoken words, gestures, and textual annotations in real time to assist users in educational or professional contexts.

10.4.10. Emotion and Context Detection

Future systems could explore more nuanced emotion and context detection by combining textual sentiment analysis with facial expressions, voice tones, and visual cues. This could significantly enhance applications in customer service, mental health analysis, and human-computer interaction.

10.4.11. Energy-Efficient Multimodal Models

Multimodal classification systems are computationally intensive. Research into energy-efficient architectures, such as low-power neural networks and efficient hardware accelerators, can make these systems more accessible for real-world deployment, especially on mobile and edge devices.

 doi:10.20944/preprints202412.1304.v1

10.4.12. Interactive Multimodal Systems

Interactive systems that allow users to provide real-time feedback on classifications can improve model accuracy and adaptability. For instance, a system analyzing tweets and images could adjust its categorization based on user input, ensuring more accurate classifications.

10.4.13. Multimodal Anomaly Detection

Expanding research to include anomaly detection in multimodal data streams can enhance applications like fraud detection, cybersecurity, and disaster response. For example, detecting inconsistencies between textual content and visual evidence can flag potentially fraudulent activities.

By pursuing these directions, multimodal text classification can evolve into a more versatile, context-aware, and impactful tool, enabling transformative applications across industries and societal domains.

Table 2. A summary of future research in text categorization.

Research Focus	Future Research Direction	Potential Applications
Universal Multilingual Models	Develop generalized models for multilingual text classification with minimal labeled data.	Cross-cultural communication, multilingual customer support, and global content moderation.
Low-Resource Languages	Use transfer learning, domain adaptation, and unsupervised methods to address data scarcity.	Language preservation, text analysis in underserved regions, and niche domain categorization.
Enhanced Language Identification	Improve techniques for detecting and processing multiple languages in text.	Multilingual user-generated content analysis and global social media monitoring.
Cultural Awareness in Models	Embed cultural sensitivity to improve classification relevance across diverse contexts.	Sentiment analysis, cross-border marketing, and international public opinion tracking.
High-Throughput Systems	Develop systems capable of processing large-scale, real-time data streams with minimal latency.	Live news categorization, stock market monitoring, and emergency response systems.
Dynamic Adaptation	Enhance models to adjust to shifting patterns and evolving content in real-time.	Social media analytics, adaptive spam filtering, and customer sentiment tracking.
Multimodal Integration	Combine text with other modalities (images, videos, audio) for holistic content analysis.	Social media content moderation, e-commerce review analysis, and multimedia news classification.
Temporal Multimodal Analysis	Analyze how user sentiment or trends evolve over time using multiple data types.	Campaign monitoring, real-time sentiment tracking, and user behavior analysis.

 doi:10.20944/preprints202412.1304.v1

Real-Time Systems	Optimize latency and computational efficiency for real-time applications.	Chatbots, fraud detection, and personalized content delivery.
Cross-Modal Transfer Learning	Enable knowledge transfer between text and other data modalities for enhanced classification.	Healthcare diagnostics, financial trend analysis, and multimedia content categorization.
Domain-Specific Frameworks	Design tailored models for specific industries like healthcare, finance, and legal analysis.	Medical text categorization, contract clause extraction, and investment report analysis.
AR/VR Integration	Integrate text categorization into augmented and virtual reality systems.	Interactive learning environments, immersive customer support, and AR-based real-time text translation.
Emotion and Context Detection	Combine multimodal inputs for nuanced emotion and context understanding.	Mental health monitoring, sentiment-based recommendations, and adaptive marketing strategies.
Interactive Multimodal Systems	Develop systems allowing real-time user feedback to refine classification accuracy.	Live content moderation, chatbot systems, and collaborative filtering in e-commerce.
Ethical Considerations and Bias Mitigation	Focus on identifying and mitigating biases in training data and algorithms.	Recruitment systems, content moderation for sensitive topics, and legal document categorization.
Explainable AI and Hybrid Models	Combine rule-based systems with ML for interpretability and transparency.	Regulatory compliance, healthcare decision support, and consumer trust-building.
Energy-Efficient Architectures	Research architectures that optimize resource usage for text categorization.	Mobile applications, edge computing, and sustainable AI deployment in resource-constrained settings.
Anomaly Detection	Develop methods to detect inconsistencies across multimodal data streams.	Fraud detection, cybersecurity monitoring, and disaster response systems.
Real-Time Multilingual Systems	Extend real-time systems to handle multiple languages dynamically.	Global event monitoring, real-time multilingual chatbots, and international e-commerce platforms.

This table provides a synthesized overview of the key future research directions in text categorization, reflecting advancements in multilingual, multimodal, real-time, and ethical AI practices, along with their applications across various domains.

11. Conclusions

The field of text categorization (TC) has experienced significant evolution, becoming a foundational component in natural language processing (NLP) and machine learning (ML). Transitioning from manual classification to scalable, ML-driven methods has revolutionized the ability to process, organize, and analyze large-scale textual data across various domains. Advances

 doi:10.20944/preprints202412.1304.v1

such as supervised learning, feature engineering, and dimensionality reduction have greatly improved the accuracy and efficiency of TC systems, making them critical for applications like sentiment analysis, spam detection, and domain-specific categorization. Despite these achievements, challenges like overfitting, class imbalance, language complexity, and high computational demands remain, underscoring the need for innovations in model interpretability and robustness.

Deep learning techniques, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and pre-trained models like BERT and GPT, have expanded the capabilities of TC by enabling advanced language understanding and contextual analysis. However, their dependence on large datasets and high computational power limits their practicality, especially for low-resource languages and real-time applications. Addressing these constraints requires a focus on developing efficient learning techniques, hybrid approaches, and explainable AI (XAI) solutions. Combining machine learning with knowledge engineering can result in interpretable and reliable models, while integrating TC with other NLP tasks, such as text summarization, named entity recognition (NER), and sentiment analysis, has the potential to create more intelligent and context-aware systems.

The future of TC lies in its ability to adapt to the demands of an increasingly interconnected and data-driven world. Multilingual and cross-cultural applications, real-time systems, and multimodal integration are poised to shape the next wave of advancements in the field. These developments will not only enhance the scalability and precision of TC systems but also democratize access to AI technologies, fostering inclusivity and global applicability. Furthermore, addressing ethical concerns, such as bias mitigation and transparency, will be critical to building trust and ensuring equitable outcomes in high-stakes applications like recruitment, healthcare, and legal analytics.

By refining technical performance and enhancing real-world relevance, text categorization systems are positioned to play a pivotal role in information retrieval, data mining, and decision-making. The integration of advanced algorithms, ethical frameworks, and interdisciplinary approaches will drive innovation, enabling TC systems to overcome existing challenges while unlocking unprecedented opportunities across industries. As researchers and practitioners collaborate to push the boundaries of what is possible, the future of TC promises transformative impacts on how we process, understand, and derive value from textual data.

References

1. Joachims, T. and F. Sebastiani, *Guest editors' introduction to the special issue on automated text categorization.* Journal of Intelligent Information Systems, 2002. **18**(2-3): p. 103.
2. Knight, K., *Mining online text.* Communications of the ACM, 1999. **42**(11): p. 58-61.
3. Pazienza, M.T., *Information extraction*. 1999: Springer.
4. Sebastiani, F., *Text categorization: Advances and challenges.* Computational Linguistics,, 2024. **50**(2): p. 205–245.
5. Yang, Y. and T. Joachims, *Text categorization.* Scholarpedia, 2008. **3**(5): p. 4242.
6. Lewis, D.D. and P.J. Hayes, *Special issue on text categorization.* Information Retrieval Journal, 1994. **2**(4): p. 307-340.
7. Manning, C. and H. Schütze, Foundations of Statistical Natural Language Processing. 1999: MIT Press.
8. Paaß, G., Document classification, information retrieval, text and web mining. Handbook of Technical Communication, 2012. **8**: p. 141.
9. Larabi-Marie-Sainte, S., M. Bin Alamir, and A. Alameer, *Arabic Text Clustering Using Self-Organizing Maps and Grey Wolf Optimization.* Applied Sciences, 2023. **13**(18): p. 10168.
10. Dhar, V., The evolution of text classification: Challenges and opportunities. AI & Society, 2021. **36**(1): p. 123-135.
11. Chen, Y. and X.-M. Zhang, *Research on Intelligent Natural Language Texts Classification.* International Journal of Advanced Computer Science and Applications, 2021.
12. Haoran, Z. and L. Lei, The Research Trends of Text Classification Studies (2000–2020): A Bibliometric Analysis. SAGE Open, 2022.
13. Xujuan, Z., et al., A survey on text classification and its applications. 2020.
14. Qian, L., et al., *A Survey on Text Classification: From Traditional to Deep Learning.* ACM Transactions on Intelligent Systems and Technology, 2022.
15. Arsime, D., Case Studies of Several Popular Text Classification Methods. 2022.
16. Zulqarnain, M., et al., *Text Classification Using Deep Learning Models: A Comparative Review.* Cloud Computing and Data Science, 2024: p. 80-96.

17. Leena, B. and K.V. Satish, *Survey on Text Classification.* 2020.
18. Zhaowei, Z., et al., The Text Classification Method Based on BiLSTM and Multi-Scale CNN. 2024.
19. Mengnan, W., *Research on Text Classification Method Based on NLP.* Advances in Computer, Signals and Systems, 2022.
20. Samarth, K., et al., A Comparative Study on Various Text Classification Methods. 2019.
21. Manon, R., et al., *Evaluating text classification: A benchmark study.* Expert Systems with Applications, 2024.
22. Bello, A.M., et al., Comparative Performance of Machine Learning Methods for Text Classification. 2020.
23. Harshitha, C.P., et al. A Survey on Text Classification using Machine Learning Algorithms. 2019.
24. Paweł, C., Text Classification Data from 15 Drug Class Review SLR Studies. 2022.
25. Ankita, A., et al. An Exploration of the Effectiveness of Machine Learning Algorithms for Text Classification. 2023.
26. Ömer, K. and A. Özlem. A Comparative Text Classification Study with Deep Learning-Based Algorithms. 2022.
27. Tiffany, Z., Classification Models of Text: A Comparative Study. 2021.
28. Maw, M., et al., Trends and patterns of text classification techniques: a systematic mapping study. Malaysian Journal of Computer Science, 2020.
29. Dea, W.K., *Research On Text Classification Based On Deep Neural Network.* International Journal of Communication Networks and Information Security, 2022.
30. Dawar, I., et al., Text Categorization using Supervised Machine Learning Techniques. 2023.
31. Kowsari, K., et al., *Text classification algorithms: A survey.* Information, 2019. **10**(4): p. 150.
32. Quazi, S. and S.M. Musa, Performing Text Classification and Categorization through Unsupervised Learning. 2023.
33. Karathanasi, L.C., et al., *A Study on Text Classification for Applications in Special Education.* International Conference on Software, Telecommunications and Computer Networks, 2021.
34. Kadhim, A.I., Survey on supervised machine learning techniques for automatic text classification. Artificial intelligence review, 2019. **52**(1): p. 273-292.
35. Ittoo, A. and A. van den Bosch, *Text analytics in industry: Challenges, desiderata and trends.* Computers in Industry, 2016. **78**: p. 96-107.
36. Shen, D., Text Categorization. 2009.
37. Sajid, N.A., et al., Single vs. multi-label: The issues, challenges and insights of contemporary classification schemes. Applied Sciences, 2023. **13**(11): p. 6804.
38. Chen, R., W. Zhang, and X. Wang, *Machine learning in tropical cyclone forecast modeling: A review.* Atmosphere, 2020. **11**(7): p. 676.
39. Wang, Z., et al., A review on the application of machine learning methods in tropical cyclone forecasting. Frontiers in Earth Science, 2022. **10**: p. 902596.
40. Gasparetto, A., et al., A survey on text classification algorithms: From text to predictions. Information, 2022. **13**(2): p. 83.
41. Shortliffe, E.H., B.G. Buchanan, and E.A. Feigenbaum, *Knowledge engineering for medical decision making: A review of computer-based clinical decision aids.* Proceedings of the IEEE, 1979. **67**(9): p. 1207-1224.
42. Ali, M., et al., A data-driven knowledge acquisition system: An end-to-end knowledge engineering process for generating production rules. IEEE Access, 2018. **6**: p. 15587-15607.
43. Gupta, D., Applied analytics through case studies using Sas and R: implementing predictive models and machine learning techniques. 2018: Apress.
44. Sebastiani, F., *Machine learning in automated text categorization.* ACM computing surveys (CSUR), 2002. **34**(1): p. 1-47.
45. Fuhr, N. and G. Knorz. Retrieval test evaluation of a rule based automatic indexing (AIR/PHYS). in Proceedings of the 7th annual international ACM SIGIR conference on Research and development in information retrieval. 1984.
46. Borko, H. and M. Bernick, *Automatic document classification.* Journal of the ACM (JACM), 1963. **10**(2): p. 151-162.
47. Larkey, L.S. A patent search and classification system. in Proceedings of the fourth ACM conference on Digital libraries. 1999.
48. Hayes, P.J. and S.P. Weinstein. CONSTRUE/TIS: A System for Content-Based Indexing of a Database of News Stories. in IAAI. 1990.
49. Androutsopoulos, I., et al. An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages. in Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval. 2000.
50. Drucker, H., D. Wu, and V.N. Vapnik, *Support vector machines for spam categorization.* IEEE Transactions on Neural networks, 1999. **10**(5): p. 1048-1054.
51. Gale, W.A. and K. Church, A program for aligning sentences in bilingual corpora. Computational Linguistics. 1993.

52. Chakrabarti, S., et al., *Automatic resource compilation by analyzing hyperlink structure and associated text.* Computer networks and ISDN systems, 1998. **30**(1-7): p. 65-74.
53. Kowsari, K., et al., Text classification algorithms: A survey. Information 10, 4 (2019), 150. 2019.
54. Mohammad, S.M., Sentiment analysis: Detecting valence, emotions, and other affectual states from text, in Emotion measurement. 2016, Elsevier. p. 201-237.
55. Yang, Y. and X. Liu. A re-examination of text categorization methods. in Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. 1999.
56. Forman, G., An extensive empirical study of feature selection metrics for text classification. J. Mach. Learn. Res., 2003. **3**(Mar): p. 1289-1305.
57. Aggarwal, C.C. and C. Zhai, *An introduction to text mining*, in *Mining text data*. 2012, Springer. p. 1-10.
58. McCallum, A. and K. Nigam. A comparison of event models for naive bayes text classification. in AAAI-98 workshop on learning for text categorization. 1998. Madison, WI.
59. Luo, X., Efficient English text classification using selected machine learning techniques. Alexandria Engineering Journal, 2021. **60**(3): p. 3401-3409.
60. Young, T., et al., *Recent trends in deep learning based natural language processing.* ieee Computational intelligenCe magazine, 2018. **13**(3): p. 55-75.
61. Guyon, I. and A. Elisseeff, *An introduction to variable and feature selection.* Journal of machine learning research, 2003. **3**(Mar): p. 1157-1182.
62. Mondal, S., et al. Cancer Text Article Categorization and Prediction Model Based on Machine Learning Approach. 2023.
63. Agarwal, A., et al. An Exploration of the Effectiveness of Machine Learning Algorithms for Text Classification. 2023.
64. Saha, S., A comprehensive guide to convolutional neural networks — the ELI5 way. 2018.
65. Ali, S.I.M., et al., Machine learning for text document classification-efficient classification approach. IAES International Journal of Artificial Intelligence, 2024.
66. Valluri, D., S. Manne, and N. Tripuraneni. Custom Dataset Text Classification: An Ensemble Approach with Machine Learning and Deep Learning Models. 2023.
67. Manning, C.D., *Introduction to information retrieval*. 2008, Cambridge university press.
68. Salton, G., A. Wong, and C.-S. Yang, *A vector space model for automatic indexing.* Communications of the ACM, 1975. **18**(11): p. 613-620.
69. Van Otten, N., Vector Space Model Made Simple With Examples & Tutorial In Python. Spot Intelligence, 2023.
70. Mikolov, T., Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013. **3781**.
71. DataScienctyst. *How to Create a Bag of Words in Pandas Python*. Data Scientyst 2023 2023/01/10 [cited 2024 Novebmer 24]; Available from: https://datascientyst.com/create-a-bag-of-words-pandas-python/.
72. Lovins, J.B., *Development of a stemming algorithm.* Mech. Transl. Comput. Linguistics, 1968. **11**(1-2): p. 22-31.
73. Ramos, J. Using tf-idf to determine word relevance in document queries. in Proceedings of the first instructional conference on machine learning. 2003. Citeseer.
74. Salton, G. and C. Buckley, *Term-weighting approaches in automatic text retrieval.* Information processing & management, 1988. **24**(5): p. 513-523.
75. Robertson, S. and H. Zaragoza, *The probabilistic relevance framework: BM25 and beyond.* Foundations and Trends® in Information Retrieval, 2009. **3**(4): p. 333-389.
76. Wang, T., et al. Entropy-based term weighting schemes for text categorization in VSM. in 2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI). 2015. IEEE.
77. Jones, K.S., S. Walker, and S.E. Robertson, *A probabilistic model of information retrieval: development and comparative experiments: Part 2.* Information processing & management, 2000. **36**(6): p. 809-840.
78. Said, D.A., *Dimensionality reduction techniques for enhancing automatic text categorization.* Cairo: Faculty of Engineering at Cairo University Master of science, 2007.
79. Murty, M. and R. Raghava, Kernel-based SVM, in Support vector machines and perceptrons: Learning, optimization, classification, and application to social networks. 2016. p. 57-67.
80. Li, B., et al. Weighted document frequency for feature selection in text classification. in 2015 International Conference on Asian Language Processing (IALP). 2015. IEEE.
81. Christian, H., M.P. Agus, and D. Suhartono, *Single document automatic text summarization using term frequency-inverse document frequency (TF-IDF).* ComTech: Computer, Mathematics and Engineering Applications, 2016. **7**(4): p. 285-294.
82. Peng, T., L. Liu, and W. Zuo, *PU text classification enhanced by term frequency–inverse document frequency-improved weighting.* Concurrency and computation: practice and experience, 2014. **26**(3): p. 728-741.
83. Magnello, M.E., Karl Pearson, paper on the chi square goodness of fit test (1900), in Landmark Writings in Western Mathematics 1640-1940. 2005, Elsevier. p. 724-731.
84. Greenwood, P.E. and M.S. Nikulin, *A guide to chi-squared testing*. Vol. 280. 1996: John Wiley & Sons.

85. Chen, Y.-T. and M.C. Chen, *Using chi-square statistics to measure similarities for text categorization.* Expert systems with applications, 2011. **38**(4): p. 3085-3090.
86. Meesad, P., P. Boonrawd, and V. Nuipian. A chi-square-test for word importance differentiation in text classification. in Proceedings of international conference on information and electronics engineering. 2011.
87. Wang, G. and F.H. Lochovsky. Feature selection with conditional mutual information maximin in text categorization. in Proceedings of the thirteenth ACM international conference on Information and knowledge management. 2004.
88. Lewis, D.D. An evaluation of phrasal and clustered representations on a text categorization task. in Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval. 1992.
89. Dhar, A., et al., *Text categorization: past and present.* Artificial Intelligence Review, 2021. **54**(4): p. 3007-3054.
90. Lhazmir, S., I. El Moudden, and A. Kobbane. Feature extraction based on principal component analysis for text categorization. in 2017 international conference on performance evaluation and modeling in wired and wireless networks (PEMWN). 2017. IEEE.
91. Bafna, P., D. Pramod, and A. Vaidya. Document clustering: TF-IDF approach. in 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT). 2016. IEEE.
92. Franke, T.M., T. Ho, and C.A. Christie, *The chi-square test: Often used and more often misinterpreted.* American journal of evaluation, 2012. **33**(3): p. 448-458.
93. Tschannen, M., et al., On mutual information maximization for representation learning. arXiv preprint arXiv:1907.13625, 2019.
94. Cardoso-Cachopo, A. and A.L. Oliveira. An empirical comparison of text categorization methods. in International Symposium on String Processing and Information Retrieval. 2003. Springer.
95. Yang, Y., An evaluation of statistical approaches to text categorization. Information retrieval, 1999. **1**(1): p. 69-90.
96. Baldi, P., et al., Assessing the accuracy of prediction algorithms for classification: an overview. Bioinformatics, 2000. **16**(5): p. 412-424.
97. Ruiz, M.E. and P. Srinivasan, *Hierarchical text categorization using neural networks.* Information retrieval, 2002. **5**: p. 87-118.
98. Guo, G., et al., *Using k nn model for automatic text categorization.* Soft Computing, 2006. **10**: p. 423-430.
99. Lewis, D.D. Evaluating text categorization i. in Speech and Natural Language: Proceedings of a Workshop Held at Pacific Grove, California. 1991.
100. Wang, B., et al. A pipeline for optimizing f1-measure in multi-label text classification. in 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA). 2018.
101. Hulth, A. and B. Megyesi. A study on automatically extracted keywords in text categorization. in Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics. 2006.
102. Wong, T.-T., Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. Pattern recognition, 2015. **48**(9): p. 2839-2846.
103. Moss, H.B., D.S. Leslie, and P. Rayson, *Using JK fold cross validation to reduce variance when tuning NLP models.* arXiv preprint arXiv:1806.07139, 2018.
104. Marcot, B.G., and Hanea, A. M. , What is an optimal value of k in k-fold cross-validation in discrete Bayesian network analysis? 2009–2031. doi:10.1007/s00180-020-00999-9, 2021. **3**: p. 36.
105. Bai, Y., et al. How important is the train-validation split in meta-learning? in International Conference on Machine Learning. 2021. PMLR.
106. Vabalas, A., et al., Machine learning algorithm validation with a limited sample size. PloS one, 2019. **14**(11): p. e0224365.
107. Zhang, H., L. Zhang, and Y. Jiang. Overfitting and underfitting analysis for deep learning based end-to-end communication systems. in 2019 11th international conference on wireless communications and signal processing (WCSP). 2019. IEEE.
108. Bu, C. and Z. Zhang. Research on overfitting problem and correction in machine learning. in Journal of Physics: Conference Series. 2020. IOP Publishing.
109. Dogra, V., et al., *A Complete Process of Text Classification System Using State-of-the-Art NLP Models.* Computational Intelligence and Neuroscience, 2022. **2022**(1): p. 1883698.
110. Hachiya, H., et al., Multi-class AUC maximization for imbalanced ordinal multi-stage tropical cyclone intensity change forecast. Machine Learning with Applications, 2024. **17**: p. 100569.
111. Liu, Y., H.T. Loh, and A. Sun, *Imbalanced text classification: A term weighting approach.* Expert systems with Applications, 2009. **36**(1): p. 690-701.
112. Nagy, G. and X. Zhang, Simple statistics for complex feature spaces, in Data Complexity in Pattern Recognition. 2006, Springer. p. 173-195.
113. Le, P.Q., et al., Representing visual complexity of images using a 3d feature space based on structure, noise, and diversity. Journal of Advanced Computational Intelligence Vol, 2012. **16**(5).

 doi:10.20944/preprints202412.1304.v1

114. Mars, M., From word embeddings to pre-trained language models: A state-of-the-art walkthrough. Applied Sciences, 2022. **12**(17): p. 8805.
115. Sinjanka, Y., U.I. Musa, and F.M. Malate, Text Analytics and Natural Language Processing for Business Insights: A Comprehensive Review. vol.
116. Bashiri, H. and H. Naderi, Comprehensive review and comparative analysis of transformer models in sentiment analysis. Knowledge and Information Systems, 2024: p. 1-57.
117. Yadav, A., A. Patel, and M. Shah, A comprehensive review on resolving ambiguities in natural language processing. AI Open, 2021. **2**: p. 85-92.
118. Seneviratne, I.S., Text Simplification Using Natural Language Processing and Machine Learning for Better Language Understandability. 2024, Ph. D. thesis, The Australian National University.
119. Garg, R., et al., Potential use-cases of natural language processing for a logistics organization, in Modern Approaches in Machine Learning and Cognitive Science: A Walkthrough: Latest Trends in AI, Volume 2. 2021, Springer. p. 157-191.
120. Kim, Y., Convolutional neural networks for sentence classification In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1746-1751. ACL. 2014.
121. Johnson, R. and T. Zhang, Effective use of word order for text categorization with convolutional neural networks. arXiv preprint arXiv:1412.1058, 2014.
122. Yang, Z., et al. Hierarchical attention networks for document classification. in Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies. 2016.
123. Schmidt, R.M., Recurrent neural networks (rnns): A gentle introduction and overview. arXiv preprint arXiv:1912.05911, 2019.
124. Devlin, J., Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.
125. Radford, A., et al., *Language models are unsupervised multitask learners.* OpenAI blog, 2019. **1**(8): p. 9.
126. Brown, T.B., *Language models are few-shot learners.* arXiv preprint arXiv:2005.14165, 2020.
127. Azevedo, B.F., A.M.A. Rocha, and A.I. Pereira, Hybrid approaches to optimization and machine learning methods: a systematic literature review. Machine Learning, 2024: p. 1-43.
128. Willard, J., et al., Integrating scientific knowledge with machine learning for engineering and environmental systems. ACM Computing Surveys, 2022. **55**(4): p. 1-37.
129. Banu, S. and S. Ummayhani, *Text summarisation and translation across multiple languages.* Journal of Scientific Research and Technology, 2023: p. 242-247.
130. Orosoo, M., et al. Enhancing Natural Language Processing in Multilingual Chatbots for Cross-Cultural Communication. in 2024 5th International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV). 2024. IEEE.
131. Liang, L. and S. Wang, Spanish Emotion Recognition Method Based on Cross-Cultural Perspective. Frontiers in psychology, 2022. **13**: p. 849083.
132. Artetxe, M., G. Labaka, and E. Agirre, *Translation artifacts in cross-lingual transfer learning.* arXiv preprint arXiv:2004.04721, 2020.
133. Schuster, S., et al., Cross-lingual transfer learning for multilingual task oriented dialog. arXiv preprint arXiv:1810.13327, 2018.
134. Zhou, X., et al., A survey on text classification and its applications. 2020.
135. Yu, M., et al., Deep learning for real-time social media text classification for situation awareness–using Hurricanes Sandy, Harvey, and Irma as case studies, in Social Sensing and Big Data Computing for Disaster Management. 2020, Routledge. p. 33-50.
136. Demirsoz, O. and R. Ozcan, *Classification of news-related tweets.* Journal of Information Science, 2017. **43**(4): p. 509-524.
137. Van de Ven, G.M., T. Tuytelaars, and A.S. Tolias, *Three types of incremental learning.* Nature Machine Intelligence, 2022. **4**(12): p. 1185-1197.
138. Yan, H., et al., A unified generative framework for various NER subtasks. 2021.
139. Mohit, B., Named entity recognition, in Natural language processing of semitic languages. 2014, Springer. p. 221-245.
140. Bui, D.D.A., G. Del Fiol, and S. Jonnalagadda, *PDF text classification to leverage information extraction from publication reports.* Journal of biomedical informatics, 2016. **61**: p. 141-148.
141. Lu, Y., R. Dong, and B. Smyth. Why I like it: multi-task learning for recommendation and explanation. 2018.