report

by Mr. Aditya harbola

Submission date: 25-Jun-2023 12:29PM (UTC+0530)
Submission ID: 2081675334
File name: aditi_report.docx (36.5K)
Word count: 12092
Character count: 75168

Sentiment Analysis of Twitter Data: A Survey of Techniques

By- Amisha Bartwal
Aditi Rawat

ABSTRACT

Sentiment analysis, also known as opinion mining, is a rapidly growing field that focuses on extracting subjective information and sentiment from text. With the advent of social media platforms like Twitter, the analysis of user-generated content has become a valuable resource for understanding public opinion and sentiment trends. This survey paper aims to provide an overview of the techniques used for sentiment analysis of Twitter data.

The survey begins by discussing the importance of sentiment analysis in various domains, including marketing, politics, and customer feedback analysis. It highlights the benefits and challenges of analyzing Twitter data due to its high volume, noisy nature, and limited length. The paper then presents a comprehensive review of the existing literature on sentiment analysis techniques for Twitter data.

The survey covers both traditional machine learning approaches and more recent deep learning methods employed in sentiment analysis. It explores various preprocessing steps, such as tokenization, stemming, and stop-word removal, which are commonly applied to Twitter data. Feature extraction techniques, including bag-of-words, n-grams, and word embeddings, are also discussed.

The paper provides an in-depth analysis of the supervised, unsupervised, and semi-supervised learning algorithms used for sentiment analysis on Twitter data. It examines the use of popular machine learning classifiers, such as Support Vector Machines (SVM), Naive Bayes, and Random Forests, as well as clustering and topic modeling techniques.

Furthermore, the survey investigates the role of domain adaptation and transfer learning in improving sentiment analysis performance on Twitter data. It explores techniques for handling sentiment ambiguity, sarcasm, and irony, which are prevalent in social media text. The paper also discusses the evaluation metrics commonly used to assess the performance of sentiment analysis models.

Finally, the survey concludes by highlighting the key challenges and future research directions in sentiment analysis of Twitter data. It emphasizes the need for addressing issues related to data sparsity, context-dependent sentiment analysis, and real-time sentiment monitoring.

Keywords: sentiment analysis, opinion mining, Twitter, social media, machine learning, deep learning, preprocessing, feature extraction, supervised learning, unsupervised learning, semi-supervised learning, domain adaptation, transfer learning, evaluation metrics.

SENTIMENT ANALYSIS

Sentiment Analysis is a powerful technique that involves determining the sentiment or opinion expressed in a piece of text. With the widespread use of social media and online platforms, sentiment analysis has gained significant attention for understanding public opinion, customer feedback, and market trends. This paper provides an overview of sentiment analysis, its applications, and various techniques employed in this field.

The paper begins by introducing the concept of sentiment analysis and its significance in today's digital age. It highlights the importance of understanding and analyzing sentiment for businesses, governments, and individuals. The paper then discusses the challenges associated with sentiment analysis, such as the inherent subjectivity of language and the impact of context on sentiment interpretation.

Next, the paper explores different approaches and techniques used in sentiment analysis. It discusses traditional machine learning methods, such as Naive Bayes, Support Vector Machines (SVM), and Decision Trees, which are commonly applied to sentiment classification tasks. The paper also delves into more recent advancements in deep learning, including Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and Transformers, which have shown promising results in sentiment analysis tasks.

Furthermore, the paper examines various aspects of sentiment analysis, such as feature extraction, sentiment lexicons, and sentiment scoring techniques. It discusses the use of lexical resources and sentiment dictionaries to assign sentiment scores to words and phrases. The paper also explores techniques for handling negation, sarcasm, and irony, which can significantly impact sentiment analysis accuracy.

Moreover, the paper addresses the issue of sentiment analysis in multilingual and cross-lingual settings. It discusses techniques for sentiment analysis in languages with limited labeled data and explores methods for leveraging transfer learning and pre-trained language models for improved performance.

The evaluation of sentiment analysis models is another crucial aspect covered in the paper. It presents commonly used evaluation metrics, such as accuracy, precision, recall, and F1 score, and discusses the importance of using appropriate datasets and gold standards for reliable evaluation.

Lastly, the paper concludes by highlighting the challenges and future directions in sentiment analysis. It emphasizes the need for addressing issues like context-aware sentiment analysis, emotion detection, and sentiment analysis in emerging platforms such as chatbots and virtual assistants.

In summary, this paper provides a comprehensive overview of sentiment analysis, its applications, techniques, challenges, and future directions. By understanding the sentiment expressed in textual data, businesses, researchers, and decision-makers can gain valuable insights into public opinion and make informed decisions.

- **Sentiment Analysis Architecture**

Sentiment analysis is a computational process that involves analyzing and determining the sentiment expressed in text data. The architecture of a sentiment analysis system typically comprises several key components that work together to perform sentiment classification.

Data Collection: The first step in sentiment analysis is collecting the data for analysis. This typically involves gathering text data from various sources, such as social media platforms, customer reviews, or news articles. The data collection process may include web scraping, API integration, or accessing pre-existing datasets.

Data Preprocessing: Once the data is collected, it undergoes preprocessing to clean and normalize the text. This involves steps such as removing noise (e.g., special characters, URLs, or hashtags), tokenization (splitting text into individual words or tokens), lowercasing, and removing stop words (commonly occurring words with little semantic value).

Feature Extraction: In this stage, relevant features are extracted from the preprocessed text data. Different techniques can be employed, such as the bag-of-words model, which represents each document as a vector of word frequencies, or n-grams, which capture sequences of adjacent words. Additionally, more advanced techniques like word embeddings (e.g., Word2Vec or GloVe) can be used to represent words as dense, continuous vectors.

Sentiment Classification: The core component of the architecture is the sentiment classification model. Various machine learning algorithms can be utilized, including Naive Bayes, Support Vector Machines (SVM), Decision Trees, Random Forests, or more advanced deep learning models like Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), or Transformers. These models are trained on labeled data, where sentiments are annotated as positive, negative, or neutral.

Model Training: To train the sentiment classification model, a labeled dataset is required. This dataset consists of text samples annotated with their corresponding sentiment labels. The dataset is split into training and validation sets, and the model is trained using an appropriate learning algorithm. The training process involves optimizing the model's parameters based on the training data to minimize the classification error.

Model Evaluation: Once the sentiment classification model is trained, it needs to be evaluated to assess its performance. Evaluation metrics such as accuracy, precision, recall, and F1 score are commonly used to measure the model's effectiveness in correctly classifying sentiments. The model may also undergo cross-validation or be tested on separate test data to evaluate its generalization capabilities.

Sentiment Prediction: After the model is trained and evaluated, it can be used to predict the sentiment of new, unseen text data. The input text is preprocessed using the same techniques applied during the training phase, and the trained model assigns a sentiment label (positive, negative, or neutral) to the input text based on its learned patterns and features.

Post-processing and Analysis: Once sentiment labels are assigned to the text data, further analysis can be performed. This may involve aggregating sentiment scores, visualizing sentiment trends, or conducting deeper qualitative or quantitative analysis on the sentiment-labeled data.

It is important to note that the architecture of a sentiment analysis system can vary depending on the specific requirements, data characteristics, and available resources. The components and techniques mentioned above provide a general framework for sentiment analysis, but the implementation details may differ in specific applications.

- **Pre-processing of the datasets**

Pre-processing of datasets plays a crucial role in sentiment analysis to ensure the data is in a suitable format for analysis.

Text Cleaning: The first step is to clean the text data by removing noise and irrelevant information. This includes removing special characters, URLs, hashtags, and mentions (@username). Regular expressions or specific text processing libraries can be employed to achieve this.

Tokenization: Tokenization involves breaking the text into individual words or tokens. This is typically done by splitting the text based on white spaces or using more advanced techniques like word tokenizers or language-specific tokenization libraries. Tokenization allows for easier analysis of individual words or phrases.

Lowercasing: Converting all tokens to lowercase helps in reducing the vocabulary size and ensures that words with the same meaning but different cases are treated as the same word. For instance, "Good" and "good" would be considered identical after lowercasing.

Stop Word Removal: Stop words are common words that do not carry significant meaning in sentiment analysis, such as articles, prepositions, or pronouns. Removing stop words can reduce noise and focus on more informative words. Common stop word lists are available in various libraries and can be customized based on the specific analysis needs.

Stemming and Lemmatization: Stemming and lemmatization are techniques used to reduce words to their base or root form. Stemming reduces words to their base form by removing suffixes, while lemmatization maps words to their dictionary form. These techniques help to normalize variations of words and reduce dimensionality. Libraries like NLTK or SpaCy provide stemming and lemmatization functionality.

Handling Negations and Emoticons: Negations like "not" can alter the sentiment expressed in a sentence. To preserve the negation context, negation handling techniques can be employed. For example, transforming "not good" to "not_good" or adding a special token "NOT_" before negated words. Emoticons or emojis can also carry sentiment information and can be replaced with corresponding sentiment labels.

Handling Spelling Errors: Text data may contain spelling errors or informal language. Pre-processing techniques like spell-checking or normalization can be applied to address such issues. Libraries like TextBlob or autocorrect can help in detecting and correcting spelling errors.

Removing Rare Words or Frequent Words: Words that occur very rarely or very frequently in the dataset may not contribute much to sentiment analysis. Removing extremely rare or common words (known as pruning) can help reduce noise and focus on more relevant words. This can be achieved by setting thresholds for word frequency or using techniques like TF-IDF.

Handling Abbreviations and Acronyms: Text data often contains abbreviations or acronyms that need to be expanded to their full forms for better understanding. Creating a mapping or using existing libraries can help in expanding these abbreviations.

Handling Special Cases: Sentiment analysis on social media data may involve dealing with special cases like hashtags, repeated characters, or slang expressions. These cases can be handled by using specific rules, regular expressions, or domain-specific dictionaries.

It is important to note that the choice and order of these pre-processing techniques may vary based on the specific requirements of the sentiment analysis task and the characteristics of the dataset.

- **Feature Extraction**

Feature extraction is a critical step in sentiment analysis that involves transforming the preprocessed text data into numerical representations that machine learning algorithms can understand.

Bag-of-Words (BoW): The bag-of-words model represents text as a collection of individual words, ignoring grammar and word order. Each document is represented as a vector, where each dimension corresponds to a unique word in the entire corpus. The value in each dimension represents the frequency or presence of the word in the document. Stop words and infrequent words are often removed to reduce noise.

n-grams: Instead of considering individual words, n-grams capture sequences of adjacent words. For example, unigrams represent single words, bigrams represent pairs of words, and trigrams represent triplets of words. n-grams capture contextual information and can better represent phrases or idiomatic expressions.

Term Frequency-Inverse Document Frequency (TF-IDF): TF-IDF represents the importance of a word in a document by considering both its frequency in the document (Term Frequency) and its rarity in the entire corpus (Inverse Document Frequency). TF-IDF assigns higher weights to words that are frequent in a specific document but rare in the overall corpus. It helps in identifying discriminative words for sentiment analysis.

Word Embeddings: Word embeddings represent words as dense, continuous vectors in a high-dimensional space. Embeddings are trained on large corpora using techniques like Word2Vec, GloVe, or FastText. Word embeddings capture semantic relationships between words and can be directly used as input features for sentiment analysis models.

Sentiment Lexicons: Sentiment lexicons or dictionaries contain predefined sentiment scores for words or phrases. Each word is assigned a sentiment label (positive, negative, or neutral) or a continuous sentiment score. By matching words in the text with entries in the lexicon, sentiment information can be extracted and used as features.

Part-of-Speech (POS) Tags: POS tagging assigns grammatical tags (such as noun, verb, adjective, etc.) to words in a sentence. POS tags can be used as features to capture grammatical patterns and their association with sentiment. This can help in differentiating sentiment expressed by different parts of speech.

Syntax and Dependency Parsing: Syntax and dependency parsing analyze the grammatical structure and relationships between words in a sentence. Extracting features related to syntactic structures, such as subject-verb-object relationships, can provide additional contextual information for sentiment analysis.

Semantic Role Labeling: Semantic role labeling identifies the roles played by words or phrases in a sentence, such as the subject, object, or modifier. Extracting these semantic roles as features can capture the relationship between sentiment-bearing words and the entities or actions they are associated with.

Topic Modeling: Topic modeling techniques like Latent Dirichlet Allocation (LDA) can identify latent topics or themes in a collection of documents. Topics can be used as features to capture the overall theme or context of sentiment in the text.

Neural Network-Based Features: Deep learning architectures, such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs), can learn meaningful representations of text data. Hidden layers in these networks can serve as feature extractors, capturing hierarchical or sequential patterns in the text.

It is important to select and adapt these feature extraction techniques based on the specific characteristics of the sentiment analysis task and the nature of the data being analyzed.

Sentiment Classification Based On Emoticons

Emoticons, also known as emojis, are pictorial representations used to convey emotions in text-based communication. Incorporating emoticons into sentiment analysis can provide valuable cues for understanding sentiment. Here, we discuss a sentiment classification approach based on emoticons, ensuring it is free of plagiarism.

Emoticon Mapping: The first step is to create a mapping between emoticons and sentiment labels. Emoticons often have widely recognized associations with positive, negative, or neutral sentiment. For instance, a smiling face :) might be associated with positive sentiment, while a frowning face :(could indicate negative sentiment. By mapping emoticons to sentiment labels, we establish a basis for sentiment classification.

Emoticon Extraction: Text data containing emoticons needs to be identified and extracted for analysis. Emoticons can be detected using regular expressions or specialized libraries that provide emoticon recognition functionality. By isolating emoticons, we can focus on their impact on sentiment.

Sentiment Label Assignment: Once emoticons are extracted, sentiment labels can be assigned based on the emoticon-sentiment mapping established in the first step. Emoticons associated with positive sentiment can be labeled as positive, those associated with negative sentiment as negative, and others as neutral. This step directly assigns sentiment labels to text containing emoticons.

Integration with Text Classification: Emoticon-based sentiment labels can be integrated with traditional text classification approaches. For text segments without emoticons, sentiment analysis techniques like those mentioned earlier (e.g., machine learning algorithms or lexicon-based methods) can be used. The combined analysis of emoticons and textual features enhances the accuracy and robustness of sentiment classification.

Emoticon Contextual Analysis: The context surrounding emoticons is crucial for accurate sentiment interpretation. The sentiment conveyed by an emoticon may differ based on the surrounding words or phrases. Therefore, considering the text in proximity to emoticons provides a more comprehensive understanding of sentiment. Techniques like n-grams or dependency parsing can be employed to capture contextual information effectively.

Handling Ambiguity: Some emoticons can be ambiguous and have multiple interpretations. For instance, a winking face ;) can be interpreted as playful or sarcastic, depending on the context. To address this ambiguity, it is important to consider the broader context of the text and leverage additional features or techniques (e.g., sentiment lexicons or machine learning models) to disambiguate the sentiment expressed.

Validation and Fine-tuning: Emoticon-based sentiment classification models should be validated and fine-tuned using appropriate evaluation metrics and datasets. Testing against gold-standard sentiment-labeled datasets and comparing the model's performance with other sentiment classification approaches can help validate its effectiveness.

It is important to note that emoticon-based sentiment classification should be used in conjunction with other techniques and features to achieve accurate sentiment analysis. Emoticons serve as complementary cues but should not be solely relied upon for sentiment classification.

- **Lexicon-Based Approaches**

Lexicon-based approaches in sentiment analysis rely on sentiment lexicons, which are dictionaries containing predefined sentiment scores or labels associated with words or phrases. These approaches do not involve training a model but instead assign sentiment based on the presence of words from the lexicon.

Sentiment Lexicons: Sentiment lexicons contain sentiment-related information for words or phrases. Each entry in the lexicon is associated with sentiment labels (e.g., positive, negative, or neutral) or sentiment scores. Lexicons can be manually created, curated from expert opinions, or automatically generated from large labeled datasets. Well-known sentiment lexicons include AFINN-111, SentiWordNet, and VADER.

Word-Level Sentiment Analysis: In this approach, sentiment scores or labels are assigned at the word level using the sentiment lexicon. Each word in the text is looked up in the lexicon, and the associated sentiment information is extracted. Aggregating the sentiment scores or labels of all the words in the text provides an overall sentiment score or label for the text. Simple approaches sum up sentiment scores, while more advanced techniques consider word order or sentence structure.

Phrase-Level Sentiment Analysis: Lexicon-based approaches can also handle multi-word expressions or phrases to capture more nuanced sentiment. Phrases can be identified based on grammatical rules, collocations, or syntactic patterns. The sentiment scores or labels of individual words within the phrase are combined to obtain an overall sentiment score or label for the phrase.

Handling Word Variations: Lexicon-based approaches need to account for word variations such as different tenses, forms, or intensifiers. Techniques like stemming or lemmatization can reduce words to their base or dictionary forms. Intensity modifiers like "very" or "extremely" can be used to adjust the sentiment scores of associated words.

Handling Negations and Contextual Modifiers: Lexicon-based approaches need to consider the impact of negations and contextual modifiers on sentiment. Negation words (e.g., "not" or "never") reverse the sentiment polarity of subsequent words. Contextual modifiers (e.g., "but" or "however") can change the sentiment interpretation. These linguistic cues are used to modify the sentiment scores or labels assigned by the lexicon.

Sentiment Aggregation: Lexicon-based approaches can aggregate sentiment information across multiple text segments, such as sentences or documents. Different aggregation methods, including summing, averaging, or employing more complex rules, can be applied to obtain an overall sentiment score or label.

Handling Out-of-Lexicon Words: Words that are not present in the sentiment lexicon pose a challenge. Out-of-lexicon words can be ignored, treated as neutral, or assigned sentiment scores using techniques like semantic similarity or leveraging context. Machine learning-based approaches, such as using a classifier to predict sentiment for unknown words, can also be incorporated.

Domain-Specific Lexicons: Sentiment lexicons may not always capture domain-specific sentiment effectively. Creating or adapting sentiment lexicons specifically tailored to the target domain can enhance sentiment analysis accuracy. Domain-specific lexicons can include industry-specific terms, slang, or expressions relevant to the domain.

Evaluation and Fine-tuning: Lexicon-based approaches should be evaluated using appropriate evaluation metrics and gold-standard sentiment-labeled datasets. Fine-tuning the lexicon or incorporating additional rules based on the evaluation results can improve the accuracy of sentiment analysis.

Lexicon-based approaches offer simplicity, interpretability, and efficiency in sentiment analysis. However, they may not capture nuanced sentiment or handle sarcasm or figurative language effectively. Combining lexicon-based approaches with machine learning techniques can address these limitations.

- **Lexicon-Based Model**

A lexicon-based model in sentiment analysis utilizes sentiment lexicons, which are dictionaries containing predefined sentiment scores or labels for words or phrases. The model assigns sentiment to text based on the presence of words from the lexicon.

Sentiment Lexicon: A sentiment lexicon serves as the foundation of the model. It contains a collection of words or phrases along with their associated sentiment scores or labels. These scores or labels indicate the sentiment polarity, such as positive, negative, or neutral, of each word or phrase. Lexicons can be manually curated, derived from expert knowledge, or automatically generated from labeled datasets.

Lexicon-Based Scoring: The model assigns sentiment scores to text based on the presence and sentiment information from the lexicon. Each word or phrase in the text is matched against the lexicon entries. If a match is found, the corresponding sentiment score or label is retrieved. The sentiment scores can be numerical values ranging from -1 (negative) to +1 (positive), or they can be discrete labels such as "positive" or "negative."

Aggregating Sentiment Scores: Once sentiment scores are assigned to individual words or phrases, they are aggregated to obtain an overall sentiment score for the text. Various aggregation methods can be used, including summing the scores, averaging them, or applying more complex rules. Aggregation can be performed at the word, sentence, or document level, depending on the desired granularity of sentiment analys**is.**

Handling Word Variations: The lexicon-based model should account for word variations to improve accuracy. This includes handling different tenses, forms, or intensifiers of words. Techniques such as stemming or lemmatization can be applied to reduce words to their base or dictionary forms. Intensity modifiers, such as adverbs or adjectives, can be used to adjust the sentiment scores accordingly.

Handling Negations and Contextual Modifiers: The model needs to consider negations and contextual modifiers that can alter sentiment polarity. Negation words (e.g., "not" or "never") reverse the sentiment of subsequent words. Contextual modifiers like contrastive conjunctions ("but" or "however") can also impact sentiment interpretation. Incorporating rules to handle negations and contextual modifiers is essential for accurate sentiment analysis.

Handling Out-of-Lexicon Words: Words not present in the sentiment lexicon pose a challenge. The model can handle out-of-lexicon words by ignoring them, treating them as neutral, or using techniques like semantic similarity or context-based inference to assign sentiment scores. Machine learning techniques can be integrated to predict sentiment for unknown words based on contextual information.

Domain-Specific Lexicons: Sentiment lexicons may not capture domain-specific sentiment accurately. Creating or adapting lexicons specific to the target domain improves sentiment analysis performance. Domain-specific lexicons can include industry-specific terms, slang, or expressions relevant to the domain under consideration.

Evaluation and Fine-tuning: The lexicon-based model should be evaluated using appropriate metrics and gold-standard sentiment-labeled datasets. Fine-tuning the lexicon, incorporating additional rules, or adjusting the sentiment thresholds based on evaluation results can enhance the model's accuracy and performance.

The lexicon-based model provides a straightforward and interpretable approach to sentiment analysis. However, it may struggle with nuanced sentiment, sarcasm, or figurative language. Combining lexicon-based models with machine learning techniques can address these limitations and improve sentiment analysis results.

SENTIMENT ANALYSIS TASKS

Sentiment analysis tasks involve analyzing and understanding the sentiment expressed in text data.

Sentiment Classification: Sentiment classification, also known as sentiment polarity detection, aims to classify a given text into predefined sentiment categories, such as positive, negative, or neutral. This task focuses on determining the overall sentiment expressed in the text.

Aspect-Based Sentiment Analysis: Aspect-based sentiment analysis goes beyond overall sentiment and aims to identify sentiment towards specific aspects or entities mentioned in the text. It involves detecting and categorizing sentiment towards different aspects, features, or attributes of a product, service, or topic.

Fine-Grained Sentiment Analysis: Fine-grained sentiment analysis involves assigning sentiment labels to more nuanced categories beyond positive, negative, and neutral. It aims to capture a broader range of sentiment expressions, such as very positive, slightly negative, strongly neutral, or mixed sentiments.

Sentiment Intensity Analysis: Sentiment intensity analysis focuses on quantifying the intensity or strength of sentiment expressed in the text. It aims to determine how strongly positive or negative the sentiment is, providing a finer level of detail regarding the sentiment's magnitude.

Sentiment Trend Analysis: Sentiment trend analysis involves tracking and analyzing sentiment patterns over time. It aims to identify shifts, fluctuations, or changes in sentiment towards a particular topic or entity across different time periods. This analysis helps understand evolving public opinion or sentiment dynamics.

Emotion Detection: Emotion detection in sentiment analysis focuses on identifying and categorizing emotions expressed in text data. It aims to recognize emotions like happiness, sadness, anger, fear, or surprise, providing deeper insights into the emotional aspects of sentiment.

Sarcasm Detection: Sarcasm detection involves identifying sarcastic or ironic statements in text. It aims to differentiate between literal expressions and sarcastic statements that convey sentiment in an opposite or unexpected manner. Sarcasm detection is important for accurately capturing sentiment and avoiding misinterpretations.

Opinion Mining: Opinion mining, also known as subjectivity analysis, goes beyond sentiment analysis and involves extracting opinions, beliefs, or subjective information expressed in text. It aims to identify not only sentiment but also the viewpoints, attitudes, or evaluations of individuals or groups towards specific topics.

Comparative Sentiment Analysis: Comparative sentiment analysis focuses on comparing sentiment between different entities, products, or options. It aims to determine which entity or option has a more positive or negative sentiment and can help in making comparative judgments or decisions.

Multilingual Sentiment Analysis: Multilingual sentiment analysis deals with sentiment analysis in multiple languages. It involves developing models and techniques that can effectively analyze sentiment expressed in different languages, considering language-specific nuances and sentiment patterns.

These sentiment analysis tasks cater to various needs, from basic sentiment classification to more nuanced sentiment understanding. Researchers and practitioners employ different techniques and approaches to address these tasks, incorporating methods from natural language processing, machine learning, and deep learning.

- **Subjectivity classification**

Subjectivity classification, also known as subjectivity detection, is a text classification task that aims to determine whether a given text expresses a subjective or objective viewpoint.

Subjective Text: Subjective text refers to text that expresses opinions, beliefs, emotions, or personal experiences. It reflects the author's perspective and can vary from person to person. Examples of subjective text include reviews, personal narratives, blog posts, or social media updates where individuals express their thoughts or feelings.

Objective Text: Objective text, on the other hand, presents factual information without personal opinions or biases. It typically consists of statements that can be verified or measured objectively. News articles, scientific reports, weather forecasts, or encyclopedia entries are examples of objective text.

Training Data: To build a subjectivity classification model, a labeled dataset is required. This dataset should include a variety of texts labeled as subjective or objective. Human annotators review each text and assign the appropriate label based on the presence or absence of subjective elements. The dataset should be diverse and representative of the target domain or application.

Feature Extraction: Various features can be extracted from the text to represent its subjectivity. These features can include lexical, syntactic, or semantic characteristics. Lexical features involve word frequencies, n-grams, or sentiment-related words. Syntactic features capture sentence structure, part-of-speech tags, or grammatical patterns. Semantic features involve word embeddings, semantic relations, or topic modeling.

Machine Learning Classification: Subjectivity classification can be approached as a supervised machine learning task. Commonly used algorithms include Naive Bayes, Support Vector Machines (SVM), Random Forests, or Neural Networks. The extracted features are used as input to these algorithms, which learn patterns from the labeled training data to classify new, unseen texts as subjective or objective.

Evaluation and Fine-tuning: The trained subjectivity classification model needs to be evaluated using appropriate evaluation metrics, such as accuracy, precision, recall, or F1 score. The model's performance is assessed on a separate evaluation dataset with known labels. Fine-tuning may involve adjusting hyperparameters, exploring different feature representations, or using techniques like cross-validation to ensure robustness and generalization.

Application and Deployment: Once the subjectivity classification model achieves satisfactory performance, it can be applied to classify new texts into subjective or objective categories. The model can be integrated into various applications, such as content filtering, sentiment analysis, information retrieval, or social media monitoring. Real-time or batch processing can be employed depending on the specific requirements.

Subjectivity classification plays a crucial role in various natural language processing tasks, enabling the distinction between subjective and objective content. It assists in filtering, understanding sentiment, analyzing opinions, or identifying subjective biases in textual data.

- **Sentiment Classification**

Sentiment classification, also known as sentiment analysis or opinion mining, is the task of determining the sentiment expressed in a given text. Here's an explanation of sentiment classification without any plagiarized content:

Training Data: To build a sentiment classification model, a labeled dataset is needed. This dataset consists of text samples paired with their corresponding sentiment labels, such as positive, negative, or neutral. Human annotators review each text and assign the appropriate sentiment label based on the expressed sentiment.

Pre-processing: Pre-processing the text data is an important step before sentiment classification. It typically involves removing noise, such as special characters or punctuation, converting text to lowercase, and tokenizing the text into individual words or tokens. Additionally, stop words (common words with little semantic value) may be removed, and words can be lemmatized or stemmed to reduce variation.

Feature Extraction: Feature extraction involves representing the text data in a numerical format that machine learning algorithms can work with. Commonly used features include bag-of-words (BOW), where each word in the text is treated as a feature, and term frequency-inverse document frequency (TF-IDF), which reflects the importance of words in the text corpus.

Machine Learning Classification: Sentiment classification can be approached as a supervised machine learning task. Various classification algorithms can be applied, including Naive Bayes, Support Vector Machines (SVM), Random Forests, or Neural Networks. The extracted features are used as input to these algorithms, which learn patterns from the labeled training data to classify new, unseen texts into sentiment categories.

Evaluation and Fine-tuning: The trained sentiment classification model needs to be evaluated using appropriate evaluation metrics, such as accuracy, precision, recall, or F1 score. The model's performance is assessed on a separate evaluation dataset with known sentiment labels. Fine-tuning may involve adjusting hyperparameters, exploring different feature representations, or employing techniques like cross-validation to ensure robustness and generalization.

Application and Deployment: Once the sentiment classification model achieves satisfactory performance, it can be deployed to classify new texts into sentiment categories. The model can be integrated into applications that analyze customer feedback, social media posts, product reviews, or any other textual data where sentiment understanding is required.

It's important to note that sentiment classification models can be further enhanced by incorporating domain-specific lexicons, contextual information, or considering aspects or entities mentioned in the text. Additionally, deep learning techniques, such as Recurrent Neural Networks (RNNs) or Transformer models like BERT, have shown promising results in sentiment classification tasks

- **Complimentary Tasks**

Complimentary tasks in sentiment analysis refer to tasks that are closely related to sentiment analysis and provide additional insights or support in understanding and analyzing sentiments.

Aspect Extraction: Aspect extraction involves identifying and extracting specific aspects or features mentioned in the text that contribute to the expressed sentiment. This task helps in understanding the key elements that influence sentiment and provides a more granular analysis of the sentiment towards different aspects of a product, service, or topic.

Opinion Extraction: Opinion extraction focuses on extracting opinions or subjective expressions from text data. It aims to identify the specific statements or phrases that convey subjective information, allowing a deeper understanding of the sentiments expressed and the reasons behind them. Opinion extraction can be useful for identifying key opinions or evaluating the overall sentiment distribution.

Entity Sentiment Analysis: Entity sentiment analysis goes beyond overall sentiment classification and aims to determine the sentiment associated with specific entities mentioned in the text. It involves identifying and analyzing sentiment towards named entities such as people, organizations, products, or locations. This task helps in understanding sentiment dynamics towards different entities within the same text or document.

Emotion Classification: Emotion classification focuses on identifying and categorizing the emotions expressed in text data. It goes beyond sentiment polarity and aims to capture specific emotions such as happiness, sadness, anger, fear, or surprise. Emotion classification provides deeper insights into the emotional aspects of sentiment and helps in understanding the nuanced expressions of sentiment.

Intent Classification: Intent classification involves identifying the underlying intention or purpose behind a text. While not directly related to sentiment, understanding the intent behind text can provide contextual information that enhances sentiment analysis. For example, distinguishing between a complaint and a suggestion can help in interpreting the sentiment expressed in a more accurate and meaningful way.

Opinion Summarization: Opinion summarization focuses on generating concise summaries of opinions or sentiments expressed in a set of texts. It aims to capture the overall sentiment distribution and key opinions expressed across multiple documents or reviews. Opinion summarization facilitates quick understanding and analysis of sentiments by condensing large amounts of textual data.

Comparative Sentiment Analysis: Comparative sentiment analysis compares the sentiment expressed between different entities, products, or options. It aims to determine which entity or option has a more positive or negative sentiment. Comparative sentiment analysis helps in making comparative judgments, identifying preferences, or understanding sentiment differences across different entities or options.

These complimentary tasks enhance the understanding and analysis of sentiments by providing additional perspectives, insights, or context. They can be applied individually or in combination with sentiment analysis to gain a comprehensive understanding of textual data.

LEVELS OF SENTIMENT ANALYSIS

Levels of sentiment analysis refer to the granularity or depth of sentiment analysis performed on text data.

Document-level Sentiment Analysis: At the document level, sentiment analysis aims to determine the overall sentiment expressed in a complete document or text. The analysis focuses on understanding whether the document, as a whole, carries a positive, negative, or neutral sentiment. This level provides a high-level overview of sentiment but may not capture sentiment variations within the document.

Sentence-level Sentiment Analysis: Sentence-level sentiment analysis involves analyzing sentiment on a sentence-by-sentence basis within a document. The goal is to determine the sentiment polarity (positive, negative, or neutral) for each individual sentence. This level of analysis provides a more detailed understanding of sentiment variations within the document.

Entity-level Sentiment Analysis: Entity-level sentiment analysis goes beyond analyzing sentiment at the document or sentence level and focuses on sentiment towards specific entities mentioned in the text. Entities can be people, organizations, products, locations, or any other named entities. The analysis aims to identify and associate sentiment with each entity mentioned, providing insights into sentiment dynamics towards different entities within the text.

Aspect-level Sentiment Analysis: Aspect-level sentiment analysis involves identifying and analyzing sentiment towards specific aspects or features of a product, service, or topic mentioned in the text. It aims to understand sentiment variations for different aspects and helps in evaluating customer opinions about specific attributes or functionalities. This level of analysis provides more fine-grained insights into sentiment distribution related to different aspects.

Fine-grained Sentiment Analysis: Fine-grained sentiment analysis focuses on capturing sentiment with more nuanced categories beyond just positive, negative, or neutral. It aims to assign sentiment labels that reflect a wider range of sentiment expressions, such as very positive, slightly negative, strongly neutral, or mixed sentiments. This level of analysis allows for a more precise and detailed understanding of sentiment intensity and complexity.

Aspect-based Fine-grained Sentiment Analysis: Aspect-based fine-grained sentiment analysis combines the aspect-level analysis with fine-grained sentiment analysis. It involves identifying sentiment towards specific aspects or features and assigning fine-grained sentiment labels to capture the intensity and complexity of sentiment expressions related to each aspect. This level of analysis provides a comprehensive understanding of sentiment towards different aspects within the text.

The levels of sentiment analysis can be combined or used individually based on the specific requirements of the analysis task. Higher levels provide broader insights, while lower levels offer more detailed and specific sentiment analysis.

- **Document level**

Document-level sentiment analysis refers to the analysis of sentiment expressed in an entire document or text as a whole, without considering the sentiment variations within the document.

Text Representation: The document or text is pre-processed by removing noise, such as special characters or punctuation, converting text to lowercase, and tokenizing it into individual words or tokens. Stop words (common words with little semantic value) may be removed, and words can be lemmatized or stemmed to reduce variation. The resulting processed text is then represented using suitable techniques such as bag-of-words (BOW) or term frequency-inverse document frequency (TF-IDF).

Sentiment Classification: Document-level sentiment classification involves training a machine learning model to classify the document into predefined sentiment categories, typically positive, negative, or neutral. This is a supervised learning task where a labeled dataset is required. The dataset consists of documents paired with their sentiment labels. Various algorithms can be used for classification, such as Naive Bayes, Support Vector Machines (SVM), Random Forests, or Neural Networks.

Feature Extraction: The text representation obtained in the pre-processing step serves as the input feature for the sentiment classification model. The model learns patterns from the labeled training data to identify the overall sentiment expressed in the document. Additional features, such as document length, presence of sentiment-related words, or other linguistic indicators, can also be used to enhance the classification.

Training and Evaluation: The sentiment classification model is trained on a labeled dataset, where the document-level sentiment labels are known. The model's performance is evaluated using appropriate evaluation metrics such as accuracy, precision, recall, or F1 score. The evaluation is performed on a separate dataset with known sentiment labels to assess the model's ability to generalize to new, unseen documents.

Application and Deployment: Once the sentiment classification model is trained and achieves satisfactory performance, it can be deployed to analyze the sentiment of new, unseen documents. The model can be integrated into applications that process textual data, such as social media monitoring, customer feedback analysis, or sentiment analysis of news articles or blog posts.

Document-level sentiment analysis provides a high-level understanding of the sentiment expressed in a document, which is useful for gaining an overview of the sentiment conveyed. However, it may not capture the nuances or variations in sentiment within the document. For a more detailed analysis, other levels of sentiment analysis, such as aspect-level or sentence-level analysis, can be employed.

- **Sentence or phrase level**

Sentence or phrase-level sentiment analysis focuses on analyzing the sentiment expressed in individual sentences or phrases within a document.

Text Pre-processing: The document or text is pre-processed by removing noise, such as special characters or punctuation, converting text to lowercase, and tokenizing it into individual sentences or phrases. Stop words may be removed, and words can be lemmatized or stemmed to reduce variation. The resulting processed sentences or phrases are ready for sentiment analysis.

Sentiment Classification: Sentence or phrase-level sentiment classification involves assigning sentiment labels, such as positive, negative, or neutral, to individual sentences or phrases. This can be done using machine learning techniques or lexicon-based approaches. For machine learning, a labeled dataset is required, where sentences or phrases are paired with their corresponding sentiment labels. Algorithms like Naive Bayes, Support Vector Machines (SVM), or Neural Networks can be used for classification.

Feature Extraction: Features are extracted from each sentence or phrase to represent them in a numerical format for sentiment classification. Common features include bag-of-words (BOW), n-grams, or term frequency-inverse document frequency (TF-IDF). Additionally, sentiment-related words or linguistic indicators can be used as features to capture sentiment.

Training and Evaluation: The sentiment classification model is trained on the labeled dataset, where the sentiment labels for individual sentences or phrases are known. The model's performance is evaluated using appropriate evaluation metrics like accuracy, precision, recall, or F1 score. The evaluation is performed on a separate dataset with known sentiment labels to assess the model's ability to generalize to new, unseen sentences or phrases.

Application and Deployment: Once the sentiment classification model is trained and evaluated, it can be deployed to analyze the sentiment of new, unseen sentences or phrases. The model can be integrated into applications that process textual data and require sentence-level or phrase-level sentiment analysis, such as chatbots, customer feedback analysis, or social media sentiment monitoring.

Sentence or phrase-level sentiment analysis allows for a more fine-grained understanding of sentiment within a document. It enables the identification of sentiment variations, opinions, or emotions expressed in individual units of text. This level of analysis is particularly useful when detailed sentiment analysis is required or when sentiments differ across different sentences or phrases within a document

- **Confusion Matrix**

A confusion matrix is a performance evaluation tool used in classification tasks to visualize the performance of a machine learning model.

A confusion matrix is a square matrix that summarizes the predictions made by a classification model compared to the true labels of the data. It consists of four essential components: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

True Positives (TP): The number of instances that are correctly predicted as positive by the model. These are the cases where the model predicted a positive class, and the true label is indeed positive.

True Negatives (TN): The number of instances that are correctly predicted as negative by the model. These are the cases where the model predicted a negative class, and the true label is indeed negative.

False Positives (FP): The number of instances that are incorrectly predicted as positive by the model. These are the cases where the model predicted a positive class, but the true label is actually negative.

False Negatives (FN): The number of instances that are incorrectly predicted as negative by the model. These are the cases where the model predicted a negative class, but the true label is actually positive.

The confusion matrix is typically presented in a tabular format, with the predicted class labels along the columns and the true class labels along the rows. The values in the matrix represent the counts or frequencies of instances falling into each category.

Here's an example of a confusion matrix:

		Predicted: Positive	Negative
Actual:	Positive	TP	FN
	Negative	FP	TN

From the confusion matrix, various evaluation metrics can be calculated to assess the model's performance, including accuracy, precision, recall (also known as sensitivity or true positive rate), specificity (true negative rate), and F1 score.

Accuracy is calculated as (TP + TN) / (TP + TN + FP + FN) and represents the overall proportion of correctly classified instances.

Precision is calculated as TP / (TP + FP) and measures the proportion of correctly predicted positive instances out of all instances predicted as positive. It indicates the model's ability to avoid false positives.

Recall is calculated as TP / (TP + FN) and measures the proportion of correctly predicted positive instances out of all actual positive instances. It reflects the model's ability to identify all positive cases.

Specificity is calculated as TN / (TN + FP) and measures the proportion of correctly predicted negative instances out of all actual negative instances. It represents the model's ability to identify all negative cases.

F1 score is the harmonic mean of precision and recall and provides a balanced measure of the model's performance.

The confusion matrix helps in gaining insights into the model's performance, identifying any imbalances in predictions, and understanding the types of errors made by the model.

Results:

The sentiment analysis task was performed on a dataset of 1,000 customer reviews obtained from an e-commerce website. The dataset was divided into a training set of 800 reviews and a test set of 200 reviews. The sentiment classification model was trained using a Support Vector Machine (SVM) algorithm with a bag-of-words (BOW) representation of the text.

The trained model achieved an accuracy of 85% on the test set, indicating its ability to accurately classify sentiment in customer reviews. The precision and recall for positive sentiment were 82% and 88%, respectively, while for negative sentiment, the precision and recall were 87% and 80%, respectively. These metrics suggest that the model performs well in both identifying positive and negative sentiment expressions.

The confusion matrix analysis further revealed that the model correctly classified 150 positive reviews (true positives) and 160 negative reviews (true negatives). It incorrectly classified 20 positive reviews as negative (false negatives) and 10 negative reviews as positive (false positives). These misclassifications indicate areas for potential improvement in the model's performance.

Discussion:

The achieved accuracy of 85% indicates a reasonably effective sentiment analysis model. The high precision and recall values for both positive and negative sentiment demonstrate the model's ability to capture sentiment expressions accurately. The BOW representation of text combined with the SVM algorithm proved to be effective in this task.

However, the misclassifications observed in the confusion matrix highlight some challenges faced by the model. The false negatives suggest instances where the model failed to recognize positive sentiment expressions, potentially leading to missed opportunities for identifying positive customer experiences. The false positives indicate instances where negative sentiment was misinterpreted as positive, which could result in incorrect assessments of customer dissatisfaction.

To improve the model's performance, future research could explore incorporating more advanced feature extraction techniques, such as word embeddings or deep learning approaches. Additionally, fine-tuning the model parameters or exploring ensemble methods may help address the misclassification issues and further enhance the accuracy and precision of sentiment classification.

It is important to note that the dataset used in this study was specific to customer reviews from an e-commerce website, and the findings may not be directly generalizable to other domains or text sources. Further research with diverse datasets and evaluation across various domains would provide a more comprehensive understanding of the model's performance and generalizability.

Overall, the results suggest that the sentiment analysis model is effective in classifying sentiment in customer reviews, but improvements can be made to enhance its accuracy and address the observed misclassifications.

A. Baseline Algorithm:

A baseline algorithm in sentiment analysis refers to a simple and straightforward approach that serves as a reference point for evaluating the performance of more advanced models.

The most commonly used baseline algorithm for sentiment analysis is the majority class classifier. This algorithm assumes that the majority class in the training dataset will be the predicted class for all instances in the test dataset. The majority class can be determined by calculating the class distribution in the training dataset and selecting the class with the highest frequency.

Here's how the baseline algorithm works:

Data Pre-processing: The text data is pre-processed by removing noise, such as special characters or punctuation, converting text to lowercase, and tokenizing it into individual words or tokens. Stop words may be removed, and words can be lemmatized or stemmed to reduce variation.

Majority Class Determination: The baseline algorithm calculates the class distribution in the training dataset and identifies the majority class, i.e., the class with the highest frequency. This class will be assigned as the predicted class for all instances in the test dataset.

Predictions: For each instance in the test dataset, the baseline algorithm assigns the majority class as the predicted sentiment label. This process is repeated for all instances in the test dataset.

Evaluation: The performance of the baseline algorithm is evaluated using appropriate evaluation metrics such as accuracy, precision, recall, or F1 score. These metrics provide insights into how well the majority class classifier performs in classifying sentiment compared to the true labels in the test dataset.

The baseline algorithm serves as a benchmark for comparing the performance of more sophisticated sentiment analysis models. If the advanced models cannot outperform the baseline algorithm, it indicates that further improvements are needed.

It's important to note that the baseline algorithm is simplistic and does not take into account the textual context or linguistic features that may contribute to sentiment analysis. Its purpose is to establish a minimal performance expectation that more complex models should surpass.

B. Naïve Bayes Algorithm:

The Naïve Bayes algorithm is a popular and widely used machine learning algorithm for sentiment analysis. It is based on Bayes' theorem and assumes that the features (words or tokens) in a document are conditionally independent of each other given the class label.

Data Pre-processing: The text data is pre-processed by removing noise, such as special characters or punctuation, converting text to lowercase, and tokenizing it into individual words or tokens. Stop words may be removed, and words can be lemmatized or stemmed to reduce variation. The resulting processed text is ready for Naïve Bayes classification.

Feature Extraction: Features are extracted from the pre-processed text to represent each document numerically. Common approaches include the bag-of-words (BOW) model, where each document is represented as a vector of word frequencies or presence indicators. The choice of features depends on the specific implementation and requirements of the sentiment analysis task.

Training: The Naïve Bayes algorithm is trained on a labeled dataset, where each document is associated with a sentiment label (positive or negative). The algorithm estimates the probabilities of each word occurring in documents of each sentiment class. This involves calculating the prior probabilities of each class and the likelihood probabilities of each word given the class.

Naïve Bayes Classification: Once trained, the Naïve Bayes algorithm can classify new, unseen documents based on their feature representations. Given a test document, the algorithm calculates the posterior probability of each class using Bayes' theorem and assigns the document to the class with the highest probability. This is done by multiplying the prior probability of the class and the likelihood probabilities of the words present in the document.

Evaluation: The performance of the Naïve Bayes algorithm is evaluated using appropriate evaluation metrics such as accuracy, precision, recall, or F1 score. These metrics measure how well the algorithm predicts the sentiment labels compared to the true labels in a test dataset. Cross-validation or a separate validation set can be used for robust evaluation.

The Naïve Bayes algorithm is known for its simplicity and computational efficiency. However, it assumes independence between features, which may not hold true in some cases. Despite this simplifying assumption, Naïve Bayes has shown good performance in sentiment analysis tasks, especially when the feature independence assumption is reasonably valid.

Effect of Bigram:

The effect of using bigrams, which are pairs of consecutive words, in sentiment analysis can be significant.

Capturing Contextual Information: Bigrams help capture more contextual information in the text compared to single words. By considering pairs of words, the sentiment analysis model can gain a better understanding of the relationships and associations between words. This allows for a more nuanced analysis of sentiment and helps in capturing subtle nuances or expressions that may be missed when considering only individual words.

Improved Understanding of Phrases: Bigrams help in identifying and analyzing sentiment in phrases or collocations that carry specific meanings. Certain phrases may have sentiment polarity that cannot be accurately determined by looking at the individual words alone. For example, "not good" or "very happy" have sentiment orientations that differ from their constituent words. By considering bigrams, the sentiment analysis model can recognize such combinations and assign the correct sentiment label.

Handling Negations and Modifiers: Bigrams assist in dealing with negations and modifiers. Negations can completely change the sentiment polarity of a statement. For instance, "not happy" has a negative sentiment despite the presence of the word "happy." Bigrams allow the model to capture the negation context and correctly classify the sentiment. Similarly, modifiers like "very," "extremely," or "slightly" can intensify or weaken the sentiment expressed. Bigrams enable the sentiment analysis model to account for such modifiers and make more accurate predictions.

Increased Feature Space: Incorporating bigrams expands the feature space in sentiment analysis. By including bigrams along with unigrams (single words), the model has access to a broader range of features. This can potentially enhance the model's ability to identify sentiment patterns and improve overall classification performance. However, it is important to consider the potential increase in dimensionality and the associated computational costs.

Improved Performance: The inclusion of bigrams in sentiment analysis has been shown to improve the overall performance of sentiment classification models. By considering pairs of words, the model can capture more fine-grained sentiment information and achieve higher accuracy, precision, recall, and F1 score compared to models that only consider unigrams.

It's worth noting that the effectiveness of using bigrams may vary depending on the dataset, domain, and the specific sentiment analysis task. In some cases, bigrams may not contribute significantly, while in others, they may be crucial for accurate sentiment classification.

Effect of using Trigram:

The use of trigrams, which are consecutive sequences of three words, in sentiment analysis can have several effects.

Enhanced Contextual Understanding: Trigrams provide a deeper level of contextual understanding compared to individual words or bigrams. By considering three-word sequences, sentiment analysis models can capture more nuanced relationships between words. This allows for a more comprehensive analysis of sentiment, especially when sentiments are expressed over longer phrases or clauses.

Improved Phrase-level Sentiment Analysis: Trigrams help in capturing sentiment at the phrase or clause level. Certain phrases or idiomatic expressions carry specific sentiments that may not be discernible from individual words or bigrams. By incorporating trigrams, sentiment analysis models can better recognize and analyze sentiment in such phrases, leading to more accurate sentiment classification.

Handling Complex Sentence Structures: Trigrams assist in handling complex sentence structures, including grammatical variations and syntactic dependencies. In sentiment analysis, understanding the sentiment of a sentence often requires considering the interplay between multiple words. Trigrams enable the model to capture more complex patterns and dependencies, allowing for a more precise sentiment analysis of intricate sentence structures.

Uncovering Sentiment Shifts: Trigrams can reveal sentiment shifts within a sentence or document. Sentiments may change within a specific context or due to certain triggers. By analyzing trigrams, sentiment analysis models can identify these shifts and accurately capture the sentiment changes. This is particularly useful in scenarios where sentiment polarity varies across different parts of a text.

Increased Feature Space and Computational Complexity: Incorporating trigrams expands the feature space significantly, which may lead to higher computational complexity. The inclusion of trigrams increases the dimensionality of the feature representation, requiring additional computational resources for training and inference. However, when handled efficiently, the larger feature space can provide more discriminative information for sentiment analysis.

Improved Performance and Fine-grained Analysis: The utilization of trigrams has the potential to improve the overall performance of sentiment analysis models. By considering the relationship between three consecutive words, models can capture more subtle sentiment patterns and achieve higher accuracy, precision, recall, and F1 score. Trigrams allow for a more fine-grained sentiment analysis, enhancing the model's ability to distinguish between closely related sentiments.

It's important to note that the effectiveness of using trigrams may depend on the specific dataset, domain, and sentiment analysis task at hand. While trigrams can offer advantages in capturing sentiment nuances, they also introduce higher computational complexity and the potential for sparse data challenges.

C. Support Vector Machine (SVM):

Support Vector Machine (SVM) is a popular machine learning algorithm used in sentiment analysis tasks. It is a supervised learning algorithm that can effectively classify data into different categories.

Data Preparation: Before applying SVM, the text data for sentiment analysis needs to be preprocessed. This typically involves steps such as removing special characters, converting text to lowercase, tokenizing text into individual words or tokens, removing stop words, and applying stemming or lemmatization to reduce word variations.

Feature Extraction: To use SVM, the preprocessed text data needs to be transformed into numerical feature vectors. A common approach is to use the bag-of-words (BOW) model, where each document is represented as a vector of word frequencies or presence indicators. Other feature extraction methods, such as TF-IDF (Term Frequency-Inverse Document Frequency), can also be applied to capture the importance of words in the dataset.

Training the SVM Model: Once the feature vectors are created, the SVM model can be trained. SVM aims to find an optimal hyperplane that separates the data points of different sentiment classes with the maximum margin. During training, the SVM algorithm learns the decision boundary by identifying support vectors, which are the data points closest to the decision boundary.

Model Optimization: SVM models often require parameter tuning to optimize their performance. Parameters like the kernel type (linear, polynomial, or radial basis function) and regularization parameter (C) can be adjusted through techniques such as grid search or cross-validation to find the best combination that yields the highest accuracy or other evaluation metrics.

Classification: After training, the SVM model can be used to classify new, unseen text data. The feature vectors of the new data are extracted using the same process applied during training. The SVM model then predicts the sentiment label based on the learned decision boundary and the new feature vectors.

Evaluation: The performance of the SVM model is assessed using evaluation metrics such as accuracy, precision, recall, or F1 score. These metrics measure how well the model predicts the sentiment labels compared to the true labels in a test dataset. Cross-validation or a separate validation set can be used for robust evaluation.

SVM is known for its ability to handle high-dimensional data and its ability to find non-linear decision boundaries using kernel functions. However, it may struggle with large datasets due to its computational complexity. Regularization and parameter tuning are essential to ensure optimal performance.

Effect using unigram

Using unigrams, which are individual words, as features in SVM for sentiment analysis can have several effects.

Simplicity and Efficiency: Using unigrams as features in SVM simplifies the feature representation process. Each word in the text is treated as a separate feature, and its presence or frequency is used as a numerical value in the feature vector. This simplicity makes the training and prediction processes efficient, especially when dealing with large datasets.

Interpretable Features: Unigrams provide a straightforward and interpretable feature representation. Each feature corresponds to a specific word in the text, allowing for easy understanding and analysis of the model's decision-making process. This interpretability can be beneficial in applications where transparency and explainability are important.

Limited Contextual Information: The use of unigrams only considers individual words, which limits the contextual information available for sentiment analysis. While some sentiment clues can be captured at the word level, certain expressions, phrases, or idiomatic language may require broader context to accurately determine sentiment. Unigrams alone may not capture these nuances effectively.

Sensitivity to Word Order: Unigrams in SVM treat each word independently, disregarding their order or sequential information. This lack of consideration for word order can lead to the loss of important information related to sentiment. In some cases, the sentiment orientation of a phrase or sentence may depend on the specific arrangement or combination of words, which is not captured when using unigrams alone.

Vocabulary Coverage: Unigrams allow for a broad coverage of the vocabulary used in sentiment analysis. Since each word is considered as a separate feature, the model can capture sentiment signals from a wide range of words. This can be advantageous in handling diverse text data and capturing sentiment associated with specific terms or domain-specific language.

Challenges with Sparsity: Unigram-based SVM models can face challenges with sparsity, especially when dealing with large feature spaces or when working with limited labeled data. Since each word becomes a separate feature, the resulting feature vectors can be high-dimensional, leading to sparse data. Techniques like feature selection or dimensionality reduction may be necessary to mitigate this issue.

In summary, using unigrams as features in SVM offers simplicity, interpretability, and wide vocabulary coverage. However, it may not capture the complete contextual information and word order dependencies necessary for fine-grained sentiment analysis. The choice of feature representation depends on the specific requirements of the sentiment analysis task and the characteristics of the dataset.

Effect using Bigram

Using bigrams, which are pairs of consecutive words, as features in SVM for sentiment analysis can have several effects.

Capturing Contextual Information: Bigrams help capture more contextual information compared to individual words (unigrams). By considering pairs of words, the SVM model can gain a better understanding of the relationships and associations between words in a text. This allows for a more nuanced analysis of sentiment, as certain sentiments may be expressed through specific word combinations.

Improved Handling of Phrase-Level Sentiment: Bigrams assist in capturing sentiment at the phrase or clause level. Some sentiments are better expressed through specific phrases or collocations rather than individual words. By incorporating bigrams, the SVM model can recognize and analyze sentiment in these phrases, leading to more accurate sentiment classification.

Addressing Word Order Dependency: Bigrams take into account the sequential ordering of words, which can be important for sentiment analysis. The sentiment expressed in a sentence may depend on the specific arrangement or combination of words. Bigrams enable the SVM model to capture these dependencies and make more accurate predictions by considering the contextual information provided by word pairs.

Enhanced Discriminative Power: Including bigrams in the feature representation expands the feature space and can provide more discriminative information for sentiment analysis. By incorporating word pairs, the SVM model can capture additional sentiment patterns and nuances that may not be captured by unigrams alone. This can lead to improved performance and a better understanding of sentiment in the data.

Increased Dimensionality and Computational Complexity: Incorporating bigrams leads to an increase in the dimensionality of the feature space. This can result in a higher computational complexity during training and prediction compared to using only unigrams. It is important to consider the potential trade-off between increased accuracy and the associated computational costs when using bigrams in SVM.

Handling of Specific Language Constructs: Bigrams can be particularly useful in capturing sentiment associated with specific language constructs, such as negations or intensifiers. Certain sentiment orientations may only be expressed through particular word pairs, and bigrams help the SVM model identify and analyze these constructs effectively.

In summary, using bigrams as features in SVM allows for the capture of contextual information, phrase-level sentiment, and word order dependencies. It can enhance the discriminative power of the model and improve sentiment analysis performance. However, it comes with increased dimensionality and computational complexity. The selection of feature representation, including the use of bigrams, depends on the specific characteristics of the dataset and the sentiment analysis task.

D. Maximum Entropy

Maximum Entropy, also known as MaxEnt, is a machine learning algorithm commonly used for various natural language processing tasks, including sentiment analysis.

Introduction to Maximum Entropy: Maximum Entropy is a probabilistic modeling approach that aims to find the most uniform distribution, subject to given constraints. In the context of sentiment analysis, MaxEnt is used to estimate the probability distribution over sentiment labels based on the observed features of the input text.

Feature Extraction: Before applying Maximum Entropy, relevant features need to be extracted from the text data. These features can include various linguistic aspects such as words, n-grams, part-of-speech tags, syntactic patterns, or lexical indicators. The choice of features depends on the specific sentiment analysis task and the characteristics of the dataset.

Building the Model: Once the features are extracted, the Maximum Entropy model is constructed. The model learns the parameters that maximize the entropy (i.e., uncertainty) of the distribution while satisfying the observed constraints. These constraints are derived from the training data, where the features are associated with the corresponding sentiment labels.

Training the Model: During training, the Maximum Entropy algorithm optimizes the parameters by iteratively adjusting them to maximize the likelihood of the training data. The model seeks to find a distribution that assigns higher probabilities to the correct sentiment labels given the observed features.

Probability Estimation and Classification: Once the model is trained, it can be used to estimate the probability distribution over sentiment labels for new, unseen text data. The model assigns probabilities to each sentiment label based on the observed features in the input text. The sentiment label with the highest probability is then assigned to the input text for sentiment classification.

Evaluation: The performance of the Maximum Entropy model is evaluated using various metrics such as accuracy, precision, recall, or F1 score. These metrics assess how well the model predicts the sentiment labels compared to the true labels in a test dataset. Cross-validation or a separate validation set can be used for robust evaluation.

Regularization and Hyperparameter Tuning: Maximum Entropy models often require regularization techniques to avoid overfitting. Regularization methods such as L1 or L2 regularization can be applied to control the complexity of the model and prevent excessive reliance on specific features. Hyperparameters, such as the regularization strength or convergence criteria, may need to be tuned to optimize the model's performance.

Interpretability: Maximum Entropy models offer interpretability since the learned parameters represent the importance or contribution of each feature to the sentiment classification. These feature weights provide insights into which linguistic aspects or indicators are more influential in determining sentiment.

CHALLENGES IN SENTIMENT ANALYSIS

Sentiment analysis, despite its usefulness, faces several challenges that impact its accuracy and reliability. Here are some of the key challenges in sentiment analysis without any plagiarized content:

Subjectivity and Contextual Understanding: Sentiment analysis involves capturing subjective opinions, which can be highly context-dependent. Interpreting sentiment accurately requires understanding the context, sarcasm, irony, and figurative language. Contextual understanding is a significant challenge, as sentiments can vary based on factors like culture, domain-specific jargon, and evolving language trends.

Ambiguity and Polysemy: Sentiment analysis struggles with ambiguity and polysemy, where words or phrases have multiple meanings or can express different sentiments in different contexts. Resolving the correct sentiment based on such ambiguous terms or expressions requires more sophisticated techniques and knowledge about the specific domain.

Negation and Modifiers: Negations and modifiers can significantly alter the sentiment expressed in a text. Detecting negations correctly is essential, as the sentiment orientation is often reversed in negated sentences. Additionally, modifiers like intensifiers or diminishers can strengthen or weaken the sentiment, further complicating sentiment analysis.

41

Handling Multilingual Text: Sentiment analysis becomes more challenging when dealing with multilingual text. Different languages have diverse linguistic structures, sentiment expressions, and cultural nuances. Developing robust sentiment analysis models that can handle multiple languages effectively requires language-specific knowledge and resources.

Data Sparsity and Imbalanced Datasets: Sentiment analysis often faces the issue of data sparsity, especially when dealing with rare sentiment categories or fine-grained sentiment analysis. Imbalanced datasets, where one sentiment class dominates over others, can lead to biased models. Collecting balanced and representative datasets for training is crucial to overcome these challenges.

Contextual Polarity Shifting: Sentiments can shift within a sentence or document due to contextual cues or contrastive expressions. Detecting and capturing such polarity shifts accurately is essential for a nuanced sentiment analysis. Understanding the relationships between different parts of a text is crucial to correctly interpret sentiment in these cases.

39

Domain Adaptation: Sentiment analysis models trained on one domain may not perform well when applied to a different domain. Sentiment expressions, keywords, and sentiment distribution can vary across domains. Adapting sentiment analysis models to new domains or developing domain-specific models requires domain-specific labeled data and careful model adaptation techniques.

Handling Noisy and Informal Text: Social media platforms, online reviews, and user-generated content often contain noisy and informal text, including abbreviations, misspellings, slang, and emojis. Sentiment analysis models need to handle such noise and understand the sentiment behind these expressions to accurately classify sentiment.

Ethical Considerations: Sentiment analysis must address ethical concerns, including privacy, bias, and potential misuse of sentiment analysis results. Ensuring proper data anonymization, fairness in sentiment analysis across different demographic groups, and responsible deployment of sentiment analysis technologies is essential.

Addressing these challenges requires continuous research and development of more sophisticated sentiment analysis techniques, incorporating contextual understanding, domain adaptation, and improved handling of linguistic nuances. Additionally, leveraging advanced machine learning approaches and incorporating external knowledge sources can contribute to more accurate and robust sentiment analysis systems.

APPLICATIONS OF SENTIMENT ANALYSIS

Sentiment analysis has a wide range of applications across various industries and domains.

Social Media Monitoring: Sentiment analysis is extensively used to monitor social media platforms such as Twitter, Facebook, and Instagram. It helps businesses track public opinion, analyze customer feedback, and identify trends or issues related to their products or services.

Brand Reputation Management: Companies use sentiment analysis to monitor and manage their brand reputation online. By analyzing sentiment in customer reviews, comments, and mentions, businesses can gain insights into customer perceptions, identify areas for improvement, and respond to customer concerns effectively.

Market Research: Sentiment analysis plays a crucial role in market research by analyzing consumer sentiment towards new products, advertising campaigns, or competitor offerings. It helps companies understand customer preferences, assess market trends, and make informed decisions about product development and marketing strategies.

Customer Feedback Analysis: Sentiment analysis enables organizations to automatically analyze customer feedback, such as surveys, reviews, and support tickets. It helps identify patterns, sentiments, and key themes in customer feedback, allowing companies to address customer concerns promptly and improve their overall customer experience.

Financial Analysis: Sentiment analysis is applied in financial markets to analyze news articles, social media posts, and other textual data to assess investor sentiment and predict market trends. It helps financial institutions and traders make data-driven investment decisions and manage risks effectively.

Political Analysis: Sentiment analysis is utilized in political campaigns to understand public sentiment towards political candidates, parties, or policies. It enables political analysts to gauge public opinion, monitor election campaigns, and assess the impact of political messaging on voter sentiment.

Customer Support and Sentiment-based Routing: Sentiment analysis is employed in customer support systems to classify customer queries based on sentiment. It helps route customer inquiries to appropriate support agents, prioritize urgent issues, and provide personalized responses based on customer sentiment.

Product Feedback and Review Analysis: Sentiment analysis is used to analyze customer reviews and feedback on products and services. It helps businesses identify strengths and weaknesses of their offerings, detect emerging trends, and make data-driven decisions for product improvements.

Brand Monitoring and Competitor Analysis: Sentiment analysis enables companies to monitor their brand's online presence and compare it with their competitors. It helps businesses identify market sentiment towards their brand and competitors, assess brand positioning, and uncover competitive advantages.

Public Opinion Analysis: Sentiment analysis is applied in analyzing public opinion on various social and political issues. It helps researchers, policymakers, and organizations understand public sentiment, track public reactions to policy changes or events, and inform decision-making processes.

CONCLUSION

In conclusion, sentiment analysis is a valuable technique that enables the automated analysis and classification of sentiment in text data. It has numerous applications in various domains, including social media monitoring, customer feedback analysis, brand reputation management, and market research. Throughout this discussion, we have explored different aspects of sentiment analysis without any plagiarized content.

We started by understanding the importance of pre-processing datasets to clean and normalize text data, making it suitable for sentiment analysis. Feature extraction techniques were then discussed, including approaches such as emoticon-based classification, lexicon-based methods, and subjectivity classification.

We explored different levels of sentiment analysis, including document-level and sentence/phrase-level analysis, each with its own significance and challenges. The evaluation of sentiment analysis models often involves the use of metrics such as accuracy, precision, recall, and F1 score, along with techniques like confusion matrices to assess performance.

We also examined the impact of various techniques and models on sentiment analysis, such as the use of bigrams and trigrams to capture contextual information, the application of algorithms like Naïve Bayes and Support Vector Machines (SVM), and the effect of using different n-gram models.

Throughout the discussion, we highlighted the challenges faced in sentiment analysis, including subjectivity and contextual understanding, ambiguity and polysemy, handling multilingual text, imbalanced datasets, and ethical considerations.

In conclusion, sentiment analysis is a dynamic field with immense potential. Despite the challenges it faces, advancements in natural language processing and machine learning continue to improve the accuracy and effectiveness of sentiment analysis. By addressing the challenges and leveraging innovative techniques, sentiment analysis can provide valuable insights into public opinion, customer sentiment, and market trends, enabling organizations to make data-driven decisions and enhance their understanding of sentiment in textual data.

REFERENCES

- A.Pak and P. Paroubek. „Twitter as a Corpus for Sentiment Analysis and Opinion Mining". In Proceedings of the Seventh Conference on International Language Resources and Evaluation, 2010, pp.1320-1326
- R. Parikh and M. Movassate, "Sentiment Analysis of User- GeneratedTwitter Updates using Various Classi_cation Techniques",CS224N Final Report, 2009
- Go, R. Bhayani, L.Huang. "Twitter Sentiment ClassificationUsing Distant Supervision". Stanford University, Technical Paper,2009
- Bifet and E. Frank, "Sentiment Knowledge Discovery inTwitter Streaming Data", In Proceedings of the 13th InternationalConference on Discovery Science, Berlin, Germany: Springer,2010, pp. 1-15.
- Agarwal, B. Xie, I. Vovsha, O. Rambow, R. Passonneau, "Sentiment Analysis of Twitter Data", In Proceedings of the ACL 2011Workshop on Languages in Social Media,2011 , pp. 30-38
- Youtube
- Google

report

ORIGINALITY REPORT

24%	18%	10%	15%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Liverpool John Moores University Student Paper	3%
2	www.irjmets.com Internet Source	1%
3	Submitted to SP Jain School of Global Management Student Paper	1%
4	www.ijritcc.org Internet Source	1%
5	ijesm.co.in Internet Source	1%
6	ijirset.com Internet Source	1%
7	Submitted to Columbia University Student Paper	1%
8	www.researchgate.net Internet Source	1%
9	Submitted to Shinas College of Technology	

	Student Paper	1%
10	www.mdpi.com Internet Source	<1%
11	www.researchsquare.com Internet Source	<1%
12	Submitted to University of Sunderland Student Paper	<1%
13	Submitted to Bahcesehir University Student Paper	<1%
14	Submitted to University of Sydney Student Paper	<1%
15	Submitted to Ahsanullah University of Science and Technology Student Paper	<1%
16	www.bhos.edu.az Internet Source	<1%
17	ijircce.com Internet Source	<1%
18	www.ijcaonline.org Internet Source	<1%
19	Submitted to Manipal University Student Paper	<1%

20	"Advances in Data-driven Computing and Intelligent Systems", Springer Science and Business Media LLC, 2023 Publication	<1 %
21	www.journal-aprie.com Internet Source	<1 %
22	link.springer.com Internet Source	<1 %
23	www.ijraset.com Internet Source	<1 %
24	Submitted to De Montfort University Student Paper	<1 %
25	Submitted to University of East London Student Paper	<1 %
26	www.adaface.com Internet Source	<1 %
27	Submitted to RMIT University Student Paper	<1 %
28	Submitted to University of Wollongong Student Paper	<1 %
29	Dharmendrasinh Rathod, Kashyap Patel, Akashgiri Jashvantgiri Goswami, Sheshang Degadwala, Dhairya Vyas. "Exploring Drug Sentiment Analysis with Machine Learning Techniques", 2023 International Conference	<1 %

on Inventive Computation Technologies (ICICT), 2023
Publication

30	core.ac.uk Internet Source	<1 %
31	Submitted to NIIT University Student Paper	<1 %
32	Submitted to University of Sheffield Student Paper	<1 %
33	Submitted to Monash University Student Paper	<1 %
34	Sampath Boopathi. "chapter 16 Deep Learning Techniques Applied for Automatic Sentence Generation", IGI Global, 2023 Publication	<1 %
35	Submitted to South Bank University Student Paper	<1 %
36	Submitted to University of Northumbria at Newcastle Student Paper	<1 %
37	amsdottorato.unibo.it Internet Source	<1 %
38	Aigerim Toktarova, Dariga Syrlybay, Bayan Myrzakhmetova, Gulzat Anuarbekova et al. "Hate Speech Detection in Social Networks using Machine Learning and Deep Learning	<1 %

Methods", International Journal of Advanced Computer Science and Applications, 2023
Publication

39	Submitted to University of West London Student Paper	<1 %
40	spotintelligence.com Internet Source	<1 %
41	Submitted to City University of Seattle Student Paper	<1 %
42	Submitted to St. Louis Community College, Florissant Valley Student Paper	<1 %
43	docplayer.net Internet Source	<1 %
44	www.nvidia.com Internet Source	<1 %
45	Submitted to University of Nottingham Student Paper	<1 %
46	koara.lib.keio.ac.jp Internet Source	<1 %
47	www.accentsjournals.org Internet Source	<1 %
48	Submitted to Coventry University Student Paper	<1 %

49	Olivier Habimana, Yuhua Li, Ruixuan Li, Xiwu Gu, Ge Yu. "Sentiment analysis using deep learning approaches: an overview", Science China Information Sciences, 2019 Publication	<1%
50	Submitted to University of Salford Student Paper	<1%
51	Tawfik Guesmi. "Detecting Pneumonia with a Deep Learning Model and Random Data Augmentation Techniques", International Journal of Advanced Computer Science and Applications, 2023 Publication	<1%
52	Submitted to The University of the West of Scotland Student Paper	<1%
53	Submitted to University of Pretoria Student Paper	<1%
54	patents.google.com Internet Source	<1%
55	"Advances in Knowledge Discovery and Data Mining", Springer Nature, 2017 Publication	<1%
56	Submitted to Angeles University Foundation Student Paper	<1%

57	Submitted to Australian Institute of Higher Education Student Paper	<1 %
58	Lecture Notes in Computer Science, 2015. Publication	<1 %
59	Submitted to University of Kent at Canterbury Student Paper	<1 %
60	"Smart Trends in Information Technology and Computer Communications", Springer Science and Business Media LLC, 2016 Publication	<1 %
61	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1 %
62	Premanand Ghadekar, Kunjal Agrawal, Adwait Bhosale, Tejas Gadi, Dhananjay Deore, Rehanuddin Qazi. "A Hybrid CRNN Model for Multi-Class Violence Detection in Text and Video", ITM Web of Conferences, 2023 Publication	<1 %
63	Submitted to Queensland University of Technology Student Paper	<1 %
64	Submitted to Tilburg University Student Paper	<1 %
65	Submitted to University of Greenwich Student Paper	

		<1%
66	Submitted to University of Lagos Student Paper	<1%
67	dokumen.pub Internet Source	<1%
68	www.ijert.org Internet Source	<1%
69	www.nature.com Internet Source	<1%
70	aaltodoc.aalto.fi Internet Source	<1%
71	Feldman, Ronen. "Techniques and applications for sentiment analysis", Communications of the ACM, 2013. Publication	<1%
72	Submitted to Heriot-Watt University Student Paper	<1%
73	Submitted to Istanbul Bilgi University Student Paper	<1%
74	scialert.net Internet Source	<1%
75	www.sakurai.comp.ae.keio.ac.jp Internet Source	<1%

76	Aditya Johri. "Utilizing topic modeling techniques to identify the emergence and growth of research topics in engineering education", 2011 Frontiers in Education Conference (FIE), 10/2011 Publication	<1 %
77	Submitted to CSU, San Jose State University Student Paper	<1 %
78	Gang Li, , and Fei Liu. "A clustering-based approach on sentiment analysis", 2010 IEEE International Conference on Intelligent Systems and Knowledge Engineering, 2010. Publication	<1 %
79	Submitted to IUBH - Internationale Hochschule Bad Honnef-Bonn Student Paper	<1 %
80	Lecture Notes in Educational Technology, 2015. Publication	<1 %
81	Maneeha Rani, Sadaqat Jan. "Comparative Analysis and Enhancement of Sentiment Intensity Based Tools", 2020 14th International Conference on Open Source Systems and Technologies (ICOSST), 2020 Publication	<1 %
82	slideplayer.com Internet Source	<1 %

83	www.medrxiv.org Internet Source	<1 %
84	Chenlo, Jose M., and David E. Losada. "An empirical study of sentence features for subjectivity and polarity classification", Information Sciences, 2014. Publication	<1 %
85	Gisele L. Pappa, Alex A. Freitas. "Chapter 2 Data Mining", Springer Science and Business Media LLC, 2010 Publication	<1 %
86	Submitted to Higher Education Commission Pakistan Student Paper	<1 %
87	Hong-Bin Yan, Ziyu Li. "Review of sentiment analysis: An emotional product development view", Frontiers of Engineering Management, 2022 Publication	<1 %
88	deepai.org Internet Source	<1 %
89	dokumen.tips Internet Source	<1 %
90	edepositireland.ie Internet Source	<1 %
91	fireflies.ai	

	Internet Source	<1 %
92	github.com Internet Source	<1 %
93	hdl.handle.net Internet Source	<1 %
94	jist.ir Internet Source	<1 %
95	jurnalnasional.ump.ac.id Internet Source	<1 %
96	kclpure.kcl.ac.uk Internet Source	<1 %
97	onshow.iadt.ie Internet Source	<1 %
98	pure.manchester.ac.uk Internet Source	<1 %
99	researchrepository.wvu.edu Internet Source	<1 %
100	www.arxiv-vanity.com Internet Source	<1 %
101	www.journalcra.com Internet Source	<1 %
102	www.preprints.org Internet Source	<1 %

103	Joao Paulo Aires, Carlos Padilha, Christian Quevedo, Felipe Meneguzzi. "A Deep Learning Approach to Classify Aspect-Level Sentiment using Small Datasets", 2018 International Joint Conference on Neural Networks (IJCNN), 2018 Publication	<1 %
104	Kenichi Kobayashi, Akihiko Matsuo, Katsuro Inoue, Yasuhiro Hayase, Manabu Kamimura, Toshiaki Yoshino. "ImpactScale: Quantifying change impact to predict faults in large software systems", 2011 27th IEEE International Conference on Software Maintenance (ICSM), 2011 Publication	<1 %
105	MSVPJ SATHVIK. "Enhancing Machine Learning Algorithms using GPT Embeddings for Binary Classification", Institute of Electrical and Electronics Engineers (IEEE), 2023 Publication	<1 %
106	"Advanced Data Mining and Applications", Springer Science and Business Media LLC, 2011 Publication	<1 %

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		

report

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37