SCIENTIFIC CULTURE
www.sci-cult.com

SCIENTIFIC CULTURE, Vol. 12, No. 4, (2026), pp. 7942-7956
Open Access. Online & Print

DOI: 10.5281/zenodo.12426793

A GRAPH-BASED OPTIMIZATION OF MARATHI SENTIMENT DICTIONARY FOR ENHANCED LEXICAL MODELING

Kiran V.Sonkamble[1]*, Dr.Rajkumar Jagdale[2], Prof.Sachin N.Deshmukh[3]

[1]Department of computer science &IT Dr.Babasaheb Ambedkar Marathwada University Chhatrapati Sambhanjinagar ,India
[2]Department of Computer Science Vishwakarma University, Pune, rajkumar.jagdale@vupune.ac.in
[3]Department of computer science &IT Dr.Babasaheb Ambedkar Marathwada University Chhatrapati Sambhanjinagar ,India, sachin.csit@bamu.ac.in

Received: 29/12/2025
Accepted: 15/04/2026

Corresponding Author: Kiran V.Sonkamble (kiransonkamble@gmail.com)

ABSTRACT

Sentiment analysis is necessary for understanding consumer opinions on digital platforms, but creating efficient sentiment analysis systems for low-resource regional languages is difficult as there are few linguistic resources and no annotated corpora. In an effort to enhance sentiment classification in Marathi, this study proposes a method that integrates machine learning approaches with structured dictionary information. In order to overcome the limitations of present lexicon and machine learning techniques in controlling morphologically complex, low-resource languages, this research presents a graph-based sentiment analysis model for Marathi. In this research, the sentiment graph is built from an optimized subset of the Marathi Sentiment Dictionary, reduced from 63,897 to 38,468 lexical entries based on polarity strength, context discriminability, semantic density, and computational tractability. Sentiment graphs were constructed to capture semantic and syntactic relationships, and TF-IDF features were used to train multiple classifiers on 9,673 Marathi headlines. The Decision Tree model achieved 99.81% accuracy and 99.89% F1-score, outperforming other classifiers. The proposed approach enhances sentiment classification through graph-theoretic representation and optimized lexicon design, establishing a scalable and interpretable framework for regional language sentiment analysis.

KEYWORDS: Sentiment Analysis, Marathi Language, Dictionary, Knowledge Graph, Synonym Merging.

1. INTRODUCTION

The fast growth of digital networks and the extensive availability of tools that support typing in regional languages has helped the content growth on social media using local languages. The huge amount of content in local language has captured the attention of researchers for data mining. Social media sites, news sites, blogs, and communication channels are now filled with a huge and increasing amount of textual content in Marathi. Such texts tend to contain dense emotional expressions, judgments, and feelings, which are immensely valuable for public opinion mining, customer feedback analysis, and social behavior modeling applications.Sentiment analysis is a leading subarea of Natural Language Processing (NLP), dealing with the automatic detection, extraction, and categorization of sentiment or emotional content infused into text content. It allows systems to judge whether a particular piece of text carries a positive, negative, or neutral sentiment. Although sentiment analysis methods have come a long way for high-resource languages such as English, and to a certain extent Hindi as well, it is still a significant challenge to apply these methods effectively to Marathi. This is because of the distinctive linguistic features of Marathi, including its morphologically dense structure, agglutinative morphology, allowed syntactic word order, and context-bound semantics. These factors complicate lexical and syntactic analysis, thus rendering traditional sentiment classification techniques less effective [1].

Traditional approaches to sentiment analysis, namely lexicon-based and machine learning algorithms, are bound to linguistic resources and large annotated corpora. For a resource-poor language like Marathi, the quality of the sentiment lexicon that can be constructed may also vary. Therefore, most of the existing methods only have poor accuracy, weak generalization ability, and limited interpretability when processing Marathi text.

Recent studies have targeted these issues using knowledge graphs. A knowledge graph is a structured representation of entities and their relationships that captures factual, contextual, and conceptual links among entities. It can model opinion words, targets, and modifiers in sentiment analysis as nodes and edges. This structure makes context clearer and enhances reasoning and polarity disambiguation [2].

This paper proposes a graph-based approach specific to Marathi sentiment analysis. The proposed model constructs a Sentiment Graph from a portion of the Marathi Sentiment Dictionary. In this graph, words are represented as nodes, and their semantic or syntactic relations are drawn as edges. This structure maintains rich relational information, thereby helping in the flow of sentiment, clustering, and emotion categorization. Features extracted from the graph are then integrated with machine learning classifiers to form a hybrid model that balances linguistic detail with computational efficiency [3].

1.1 Understanding Lexicon-Based Sentiment Analysis

The lexicon-based method works like using an emotional dictionary to understand how people feel about something based on the words they use. Instead of teaching a computer through examples like machine learning does, this approach relies on pre-built word lists that already know whether each word expresses positive, negative, or neutral emotions. [4]

Think of it as having a special book where every word has an emotional score attached to it - words like "excellent," "amazing," and "love" might get positive points, while words like "terrible," "awful," and "hate" get negative points. When the computer reads a sentence, it simply looks up each word in this emotional dictionary, adds up all the positive and negative scores, and determines the overall feeling of the text. For example, if a restaurant review contains 5 positive words and 2 negative words, [5]

This method can work at different levels - it can analyze individual sentences to understand their emotional tone, or it can look at entire documents like reviews or articles to get an overall sentiment picture. The beauty of this approach is its simplicity there's no need to train the computer with thousands of examples because the emotional knowledge is already built into the word dictionary[6].These sentiment dictionaries can be created in three different ways. Manual construction involves human experts carefully reviewing words and assigning emotional scores based on their knowledge and experience. Dictionary-based approaches use existing emotional dictionaries and expand them by finding related words through synonyms and antonyms. Corpus-based methods analyze large collections of text to discover which words tend to appear together with known emotional words, helping identify new sentiment-bearing terms.

The sentiment library essentially becomes a comprehensive database where each word is tagged with its emotional polarity whether it typically expresses positive feelings, negative feelings, or remains neutral. This systematic labeling allows

computers to quickly process text and make emotional judgments without needing complex training processes that machine learning requires. The lexicon-based approach has actually made machine learning methods more effective because these emotional dictionaries provide a solid foundation of word knowledge that can enhance computer learning algorithms. When machine learning systems have access to these pre-built emotional vocabularies, they can learn patterns more efficiently and make more accurate predictions about human sentiment in text. [7]

1.2 Machine learning based Sentiment Analysis

Machine learning uses three key approaches to understand emotions in text through syntactic, semantic, and pattern analysis. Syntactic analysis examines how words are structured and arranged in sentences, while semantic analysis focuses on the actual meaning behind the words, and pattern mining identifies recurring emotional indicators throughout the text. In supervised learning, computers learn like students with a teacher they study thousands of text examples that humans have already labeled as positive, negative, or neutral, then use algorithms like Naive Bayes (which calculates the probability of sentiment based on word patterns), Support Vector Machine (which creates boundaries between different sentiment categories), Decision Trees (which make yes/no decisions about emotional content), and k-nearest Neighbor (which compares new text to similar examples it has seen before). The computer practices on a training dataset to learn these patterns,[8] then proves its understanding on a test dataset to measure how accurately it can identify emotions. Unsupervised learning works more like a detective solving a mystery without clues [8] the computer receives unlabeled text and must discover emotional patterns on its own by building vocabulary clusters and grouping words that frequently appear together based on similar sentiments. The main challenge here is creating these emotional word groups without any human guidance. Semi-supervised learning combines both approaches by using a small amount of labeled data as guidance while also learning from large amounts of unlabeled text, making it particularly valuable since unlabeled data is much more readily available and cost-effective than carefully labeled data, which requires significant time, expense, and human expertise to create, making this hybrid approach suitable for most real-world machine learning problems where labeled [9]data is scarce but unlabeled text is abundant.

1.3 Research Motivation and Hypothesis

The variation in Marathi texts demands many redundant and overlapping sentimental expressions, which reduces accuracy and efficiency of computer models. We conjecture that under low- resource settings there is also benefit in applying graph-based optimization to model sentiment lexicons as graphs and perform synonym merging while reducing redundancy, preserving semantic, and increasing SA.

2. RELATED WORK

Research on sentiment analysis has expanded across rule-based systems, lexicon-driven methods, supervised learning, optimisation techniques and, more recently, transformer models. Medvedeva et al. [10] showed that tonal vectorisation, which assigns sentiment values on a –3 to +3 scale, can capture emotional strength better than simple Bag-of-Words or TF-IDF features. Their logistic regression model reached 72.1 percent accuracy on a bank-comment dataset, though it still had trouble interpreting context-heavy or ambiguous language.Work on large language models has pushed sentiment analysis in a different direction. Zhan et al.[11] examined optimisation strategies for transformer models like GPT-3 and reported that fine- tuning on domain-specific hotel reviews raised accuracy to 85 percent. Their study also pointed out how earlier deep learning models such as RNNs, LSTMs and CNNs improved feature learning, but still relied on large labelled datasets and considerable computing power. Even with GPT-scale models, domain alignment and data quality remain essential.

Lexicon-based approaches come with their own difficulties Machová et al. [12],worked on reducing the subjectivity that comes with manual labeling by using PSO and BBPSO to adjust word polarity scores. Their refined Slovak lexicons improved macro F1 scores, and when combined with classifiers such as Naïve Bayes, J48, BFTree and OneR, the hybrid models reached up to 92.34 percent accuracy on Amazon and IMDB reviews. They validate that swarm-inspired optimization can be used to enhance operational properties of lexicon-based approaches, in particular for low resource languages.

Munir et al. [13] concentrated on the problem of high-dimensional and noisy social media text. Their hybrid RSTLBO based feature selection method achieved an 80.2% accuracy for airline Twitter data and could reduce the number of feature dimensions by nearly 99%. This demonstrates the importance of clean preprocessing and feature selection when working with informal or dirty text.Graph-based approaches offer a different perspective. Muthuvel et

al. [14] employ an optimisation framework using SVMs, grid-search tuning and cross-validation to classify sentiment in Twitter and IMDB datasets. Their larger body of work involving graph algorithms, including an application of Kruskal's method for urban infrastructure planning, and the combination of Dijkstra with A* and Q-learning for route optimization, hints that the graph structures can also support richer representations of text relationships in sentiment tasks.

Chen et al. [15] presented the MSIF framework that constructs domain-specific sentiment lexicons by fusing multiple emotional data sources and leveraging ADMM-based optimisation. Evaluations on five Amazon datasets demonstrated significant improvements compared with previous lexicon construction approaches.

Tang Duyu et al. [16] and Erramni et al. [17] point out the recent rise in the importance of opinion recognition in fields such as customer behaviour, public issues and finance. Although a large number of sentiment systems still use machine learning and NLP techniques [18], they are mostly bound to simple polarity categories for sentiment analysis, which is too coarse for the richness of human emotion displayed [19]. These problems are further exacerbated in languages like Marathi, where annotated datasets and language-specific tools are minimal.

Table 1 is a comparative analysis of representative sentiment analysis tasks, summarizing their design, datasets, pivotal techniques, performance, and weaknesses. It also points to an evolution from rule-based, lexicon-driven methods, to transformer approaches, graph solutions, and optimization-driven machine learning solutions. It further points to the fact that despite the efficiency in many tasks on English datasets, as well as domain-specific datasets, there are weaknesses in high computation complexity, lack of understanding, lexicon dependence, and inability to handle multilingual and low-resource languages efficiently. All these point to the need for efficient sentiment analysis systems, especially for morphologically complex languages like Marathi.

Table 1: Presents A Comparative Summary of the Major Sentiment Analysis Approaches.

Study	Approach / Method	Dataset / Language	Key Techniques	Reported Performance	Limitations
Medvedeva et al. [10]	Rule-based + tonal lexicon + logistic regression	20,000 bank customer comments (English)	Tonal vectorization (−3 to +3), supervised learning	72.1% accuracy (C = 0.1)	Weak on context-dependent sentiment; struggles with subtle expressions
Zhan et al. [11]	Transformer-based (GPT-3) with fine-tuning	Region-specific hotel reviews	Self-attention, transfer learning, domain tuning	85% accuracy with strong precision/recall	High computation; domain-specific tuning required
Machová et al. [12]	Lexicon optimization (PSO, BBPSO) + hybrid classifiers	Slovak datasets; Amazon and IMDB	Swarm optimization, Naïve Bayes, J48, BFTree, OneR	Up to 92.34% accuracy (OneR)	Relies on lexicon quality; limited contextual modelling
Munir et al. [13]	Hybrid feature selection (RST + TLBO)	Airline Twitter reviews	Text preprocessing + optimization-based feature reduction	80.2% accuracy; 98.9% feature reduction	Works best on structured tasks; still needs labelled data
Muthuvel et al. [14]	Graph-based sentiment classification + SVM (OSAF)	Twitter, IMDB (English)	Grid search, K-fold cross-validation, graph algorithms	Improved accuracy, recall, F-measure vs. baselines	Not designed for multilingual or low-resource settings
Chen et al. [15]	Domain-specific lexicon construction (MSIF)	Amazon review datasets	Multi-source emotional signals; ADMM optimization	Outperforms baseline lexicon methods	Limited evaluation outside English domains
Tang Duyu et al. [16]	General sentiment analysis overview	—	Feature extraction, NLP, ML	—	Highlights general challenges but no applied system
Erramni et al. [17]	Polarity-based ML/NLP methods	—	ML classification, text preprocessing	—	Limited emotional granularity; mainly binary/ternary sentiment
[18], [19]	Traditional ML + NLP	—	Feature-based classification	—	Oversimplified polarity; no deep emotional modelling

2.1 Surveys on Indian Regional Languages

SentiWordnet was and a word lexical transfer technique [20] to determine the polarity of reviews by training the machine in one language and then using machine translation to determine the sentiment of reviews in another language. Hindi SentiWordnet has been improved to include rules for negation and discourse integration[21].Creating a dictionary based on a pre-annotated corpus of the most often used

Hindi words and determining the polarity of review and checking for negations This paper primarily emphasizes the sentiment classification of movie reviews and tweets. The initial step involves pre-processing the datasets by eliminating irrelevant terms and characters. Subsequently, multiple supervised learning models are trained and tested, and their accuracies, precision, and recalls are compared [22]. This study developed a tool for evaluating reviews of different categories of web movies and series using sentiment analysis. Using API information and Twitter access tokens, data was downloaded from the social media site. Sentiment analysis can be applied to each of the four genres of comedy, romance, horror, and crime to analyze the sentiment expressed in textual data related to these genres. For example, in the case of comedy, sentiment analysis can be used to identify whether the text data is expressing positive emotions such as happiness and fun, or negative emotions such as disappointment or frustration.

Similarly, sentiment analysis can be used to identify the sentiment expressed in textual data related to romance, horror, and crime genres. For instance, in the case of romance, sentiment analysis can be used to analyze the emotions expressed by characters in a romantic story or to identify the general sentiment of a romantic novel. In the case of horror, sentiment analysis can be used to identify the emotions expressed by characters when they encounter scary situations or to identify the overall sentiment of a horror movie. In the case of crime, sentiment analysis can be used to identify the emotions expressed by characters when they face danger or to identify the general sentiment of a crime story. Sentiment analysis can provide valuable insights into the emotions and attitudes expressed in textual data related to various genres, which can help businesses, researchers, and content creators better understand their audiences and improve their products or services accordingly [23].People are allowed to openly share their opinions and points of view, and because these views have developed through time, it is possible to study them to determine their polarity. Finding a predictive model to record and classify these user times for Indian movie reviews was our goal. [24]. An algorithm is proposed for Sentiment analysis of Movie reviews in Hindi. Pre-annotated corpus consisting of words/phrases and negation handling of review. Overall polarities if defined in the review then the system recognizes the overall polarity of the system and assigns that polarity to the review [25] With this approach, the reviews in Hindi would be separated into two groups: positive and negative. So, the user's understanding of the review's polarity would be made much simpler and simpler. Machine learning will be utilized as the classification technique [26]. the topic of identification of aspect groups was cast as a problem of multi-label classification, while the classification of emotion was modeled as a problem of multi-class classification [27]. They use their system to analyze the review content and give sentiment labels to each review component. Then, SentiWordNet is used to score each aspect of the text, with characteristics including adjectives, adverbs, verbs and n-grams being chosen It is possible to classify Hindi text based on ontology and perform multiclass classification along with sentiment analysis. Ontology refers to the hierarchical structure of knowledge representation that organizes concepts and their relationships within a domain. By using an ontology-based approach, the Hindi text is based on the concepts and relationships defined in the ontology. For example, let's say we have an ontology related to the food industry that contains concepts such as ingredients, recipes, cooking techniques, and types of food. This ontology classifies Hindi text related to food based on the concepts defined in the ontology. In addition to classification, they can also perform sentiment analysis on the text to determine the emotions and attitudes expressed in the text [28] Sentiment analysis (SA)is an NLP (natural language processing) activity that sources feelings from different texts and categories them into polarity (positive, negative or neutral) groups accordingly. Shopping online for every good is a classic example of For Indian languages, SA. Aditya Joshi and Balamurli developed cross-lingual sentiment analysis. Not every combination can be translated using machine translation. to extract customer sentiment from Kannada Web data by [29] The English feedback was translated into Kannada using machine translation, and the algorithm that focuses on pairing POSs and adjective analysis was also applied. The difference between positive and negative numbers is referred to as the disparity. presented a way for undertaking a fine-grained analysis of the reviewer's sentiment orientation and strength towards the various components of the film. It rates words using domain-specific and general opinion lexicons, and it realizes various word dependencies and assists in distributing the word score in the document using a dependency tree. By developing a unique feature-based algorithm for aspect-level sentiment analysis, After that, the whole document is graded using the total of everyone. .Several methods can be used to identify significant

aspects of online consumer reviews. One popular method is called aspect-based sentiment analysis. In aspect-based sentiment analysis, the text is first segmented into different aspects or features, and then the sentiment expressed towards each aspect is analyzed. Based on observations that these components receive the most comments in reviews and that customer opinions on these key characteristics have a significant impact on total product opinion, they were able to identify the critical aspects. In their technique, they formulate the aspect value distribution using Multivariate Gaussian Distribution. Each lexical phrase was linked to its appropriate component of speech. After tagging the speech using the Stanford Part-Of-Speech tagger and scanning the phrase for the vocabulary word, the lexical term is matched to the relevant part of speech in the sentence. These aspect-based split sentences were fed into the classifiers created for each aspect. A Naive Bayes classifier was employed for this. It influences the possibility that a word, or maybe a sentence, would be classified as positive or unfavorable. The classifier is trained and tested traditionally.

2.2 Research Gap in Sentiment Analysis

Such progress in many methods, several gaps still exist. Methods that rely on tonal vectorization and lexicon-based scoring generally struggle with highly contextualized expressions, such as sarcasm, idioms, and layered emotional meaning. While models using the transformer architecture train well on datasets matched to the domain and perform excellently, their accuracies usually drop whenever being applied to newer domains or low-resource languages.

All state-of-the-art systems are primarily trained on English datasets. Due to the limited availability of annotated corpora, coupled with a lack of linguistic tools, it is difficult to develop an accurate sentiment model for languages like Marathi. Also, creating high-quality datasets and fine-tuning large models are expensive; hence, their application to real-world settings with very limited resources remains limited.

Another gap is the limited combination of deep learning with bio-inspired optimization methods such as PSO. While these methods can enhance lexicon quality, they seldom appear together with modern contextual models or graph-based representations.

Finally, many systems still work with binary or simple three-class sentiment labels. This simplifies modelling but fails to reflect the full range of emotions found in natural language. A more unifiedframework that blends context awareness, deeper emotional representation, multilingual support and efficient computation is still missing.

Addressing these gaps would support more accurate, interpretable and practical sentiment systems for multilingual and emotion-rich real-world environments.

3. METHODOLOGY

This section presents a graph-based approach can be used to construct, optimize, and evaluate the Marathi Sentiment Dictionary. Linguistic resource design, data preprocessing, graph construction, synonym merging, and sentiment classification are integrated parts of the whole process

3.1 Comparative Analysis of Dictionary Versions

The Marathi Sentiment Dictionary serves as a formal lexical resource to support sentiment analysis in Marathi. Across several development stages, the dictionary has been refined through better annotations, removal of duplicate entries and steady expansion of its vocabulary. Each version was shaped to improve semantic consistency, run more efficiently and adapt to different domains.

The next sections describe the structural updates and optimization steps used in each version of the dictionary.

3.1.1 Lexicon Set Development Process

3.1.1.1 Dictionary 2.0 and 2.0 (Optimized)

The initial release of the Marathi Sentiment Dictionary 2.0, had 29,900 set of lexicon words entries that comprised the core resource for sentiment categorization tasks. Comprehensive lexical analysis was the priority in this release but with a number of redundancies [30]. To enhance its effectiveness and usability in real-time systems, Dictionary 2.0 (Optimized) has been refined significantly. Low-impact and redundant entries were eliminated, reducing the word count to 15,486, without compromising crucial sentiment information.

3.1.2 Graph-based synonym-merging

Each unique word form as a node in a graph. Whenever two words are known to be synonymous, morphological variants, or strong near-synonyms, place an edge between their nodes. After inserting all edges from your source files (synonym pairs, morphological rules, variant lists), the graph breaks into connected components. Each connected component is a set of words that are directly or indirectly related. We then collapse each component to a single representative entry for the lexicon. The

edges in the graph are created by bringing together different sources that show how words are related to one another. These links can come from synonym pairs found in existing lexicons or thesauri, as well as from morphological variants generated through lemmatizes or affix rules. Spelling differences and transliteration forms are also included so that words with the same meaning but different surface forms are connected. In addition, any manually curated synonym groups or expert-reviewed lists contribute to the network. When needed, high-confidence relationships from distributional similarity or embedding-based nearest-neighbor checks can also be added. Taken together, these sources help build a annoying and reliable set of connections, supporting accurate synonym merging.

The synonym-merging process begins by collecting all lexicon entries and any lists of external synonyms or variants, then normalizing the word forms so that comparisons are consistent-such as by lowercasing and normalizing. We then build an undirected graph in which each unique word forms a node, and links between words that the source material identifies as related are added; the members of multi-word groups are linked according to source confidence. When available, those links can carry confidence weights, though plain merging can proceed without them. Once the full graph is constructed, a connected-components routine finds clusters of words that are directly or indirectly related; each cluster becomes a candidate merge unit. For every cluster we pick a single representative term—typically the most frequent corpus form, a canonical lemma, or a human- validated entry (or the highest-weighted node when weights exist)—and map all other members to that representative. The merged entries, their provenance and any aggregated polarity scores are recorded in a mapping table, and finally we recompute polarity if needed and run a short validation pass (automated checks plus spot manual review) to make sure no important distinctions were lost.

3.1.1.3 Synonym Merging Using a Graph-Based Approach

Structural consistency and lexical recurrence control are crucial in enhancing performance and comprehension while using sentiment lexicons for computational tasks, especially in low-resource languages such as Marathi. The major problem of creating a positive and negative lexicon is that there are many words having the same meaning, hence duplication, additional noise, and spurious computational complexity. To do this and utilize it in order to come up with a graph-based synonym conflation approach that allows to bring together words of close meaning in a way that can be scaled and accomplished in a systematic manner consists of three main fields i.e a list of polarity measures (usually numerical scores that indicate positive, negative, or neutral sentiment), a list of synonyms that were gathered manually or semi-automatically based on linguistic data or expert tagging, and a part-of-speech (POS) tag that illustrates the word's grammatical function(noun, verb, adjective, etc.). Every word in graph theory is a vertex, and words that appear in the same list of synonyms, vertices are connected to one another by an edge.

This creates unoriented relations which illustrate semantic equivalence or contextual similarity. Therefore, the graph becomes a set of interconnected components, each being a cohesive cluster of synonyms that relate to one another. To synonym groups through the submission of normal graph traversal. In instruction to retain valuable data regarding sentiment, such as polarity scores and POS tags, without unnecessary redundancy, each connected component in the system is assigned a representative word. This representative word chosen by a person is often the first use that appeared in the original data set or is chosen in some fashion, including by frequency of use or relevance of its presence on a graph.

Then, the sentiment data of the representative term is stored, and the rest of the words within the cluster are considered synonyms of this one item. The final result of this approach will be a normalized emotion dictionary in which every set of synonyms will be condensed into a single lexical representation that ensures metadata. Not only does it make the sentiment resource more consistent and denser, but it also makes applications of sentiment classification in the future more accurate and effective because semantic redundancy is avoided and the polarity of sentiments is the same across similar expressions. The synonym integration process in this paper is graph-based, which leverages features and uses to create a robust framework for enhancing Marathi sentiment lexicons and making them cope with high-performance sentiment analysis in academic as well as real-world NLP applications.

3.2 Dictionary 3.0 and 3.1

Later on, Dictionary 3.0 further broadened the coverage with other domain-specific and informal words, bringing the total to 32,891 sets of lexicon words entries. This addition was further directed

towards better generalization over diverse text domains such as informal and news articles.

The latest base version before optimization, Dictionary 3.1, had 32,997 sets of lexicon words entries, extending the architecture and semantics of the previous version while adding more detailed polarity scoring and syntactic coherence.

In order to combine resources and optimize lexical diversity, Dictionary 2.0 and Dictionary 3.1 were combined, creating a large corpus of 63,897 sets of lexicon words entries. This combination did, however, introduced extensive redundancy and conflicting polarity labels.

The lexicon was filtered by removing redundant, low-impact, and conflicting entries while conserving high-polarity, semantically important terms to optimize sentiment classification performance. With a final optimization phase, this combined lexicon was filtered down to 38,468 sets of lexicon words entries, eliminating duplicate or poorly informative words, while keeping semantically rich and sentiment-carrying entries. This compression provided optimal performance in machine learning tasks by minimizing noise and enhancing polarity consistency.

The three-member expert committee in the Marathi language validated the Marathi Dictionary. The committee consisted of two university faculty members from Marathi Department and one post-doctoral fellow in Marathi and one expert from Computer Science and Engineering as the chairperson.

The process of validation (flesiss Kappa), To measure the level of agreement among the evaluators, (Wikipedia n.d) we used Fleiss' Kappa. This statistical measure was calculated using its standard formula to determine how consistently the experts evaluated the POS tags, sentiment scores, and synsets.

$$K = \frac{\bar{P} - \bar{Pe}}{1 - \bar{Pe}}$$

.....Equation 1

where the factor $1 - Pe$represents the degree the of agreement that can be achieved beyond chance. The degree of agreement that was actually attained above chance is given by $P - Pe$ and k= 1 if evaluator is completely in accord. If no agreement exits amongst the evaluator, then k=0. Here is a smooth and natural integration of your added point:

The combined MSWN 2.0 and 3.1 lexicon achieved a kappa value of 0.8474, which also falls within the range of almost perfect agreement. This strengthens confidence in the annotation process and confirms that the optimized lexicon is supported by a clear validation step. We used Fleiss' Kappa to measure how well multiple reviewers agreed on the sentiment labels, and the proposed lexicon-based model for Marathi sentiment analysis reached an agreement score of K = 0.8358, which also falls in the almost perfect range. This result indicates that the validated lexicon provides an effective foundation for subsequent sentiment analysis studies and that the annotations represent consistent reviewer opinions.

As shown in **Table 3.1,** Dictionary 3.1 (Optimized), the most refined and application-ready version, comprises 22,982 sets of lexicon word entries, a reduction from the original Dictionary 3.1. This version is the preferred choice for subsequent sentiment classification tasks because it strikes a compromise between lexical depth and computational usability in reality.

Table 2 summarizes the version-wise distribution of word counts along with positive and negative sentiment entries across different stages of the Marathi Sentiment Dictionary.

Version	Word Count	Positive	Negative
Dictionary 2.0	29,900	4228	5038
Dictionary 2.0 (Optimized)	15,486	2190	2609
Dictionary 3.0	32,891	5906	6407
Dictionary 3.1	33,997	9025	16255
Dictionary 3.1 (Optimized)	22,982	6286	11322
Dictionary 2.0 + 3.1	63,897	14991	21641
Dictionary 2.0 + 3.1 (Optimized)	38,468	9025	13028

The dictionaries reveal important differences across versions in word count as well as positive and negative entries. Dictionary 2.0 had 29,900 words, with 4,228 positive and 5,038 negative ones, whereas the optimized version the word count to 15,486, with 2,190 positive and 2,609 negative ones. Dictionary 3.0 consisted of 32,891 words, comprising 5,906 positive and 6,407 negative ones. Version 3.1 moved up slightly to 32,997 words, with an affected growth in positive entries to 9,025 and negative entries to 16,255. Optimized version 3.1 reduced the word count to 22,982, with 6,286 positive and 11,322 negative entries. Merging Dictionary 2.0 and 3.1 gave 63,897 words for a total of 14,991 positive and 21,641 negative entries, whereas optimized merge had a total of 38,468 words with 9,025 positive and 13,028

negative entries. These differences show the effect of optimization in reducing word count and still keeping a substantial number of sentiment entries.

3.2.1 Optimization Implications

There are some pragmatic implications of the several development and optimization of dictionary:

- Improved performance: Getting rid of bad sentiment words and redundancy can help classifiers get better in accuracy and recall.
- Time and memory: reduces the the amount of time and memory performed on data, which is important for embedded systems and real-time NLP.
- Enhanced semantic focus: Optimized versions emphasize strongly-polararised and contextually-weighted words resulting in the increased detection of important sentiment cues.
- Larger domain coverage: Versions 3.0 and 3.1 included domain-specific specific terms, making the lexicons more useful for social media analysis, news and review data.

Together, these updates support the creation of scalable and interpretable sentiment analysis systems, especially for languages like Marathi that are rich in morphology but limited in resources.

3.3 Data and Preprocessing

As described in **Table 3.2,** dataset of 9673 Marathi news headlines from an open Kaggle corpus. This corpus widths a wide variety of actual incidents and points of view, which makes it an ideal candidate for sentiment classification in Marathi a low resource morphologically rich language [16]. In instruction to maintain class balance for the sentiment labels, we split the dataset into 80% training and 20% test sets by lamination.

Table 3. The detailed composition of the training and testing datasets is presented.

Dataset	Number of Samples	No. of Positive Samples	No. of Negative Samples	Percentage
Training Set	7,738	6084	1654	80%
Testing Set	1,935	1520	414	20%
Total	**9,673**	**7605**	**2068**	**100%**

The lexicon contains the following metadata:

- Word or phrase (in Marathi)
- Part-of-speech (POS) tag
- Sentiment polarity values (positive and/or negative scores)
- List of synonyms, usually comma-separated

For pre-processing used a typical pipeline to clean and normalize the text raw data. Null values were dropped initially to maintain the dataset accuracy. Punctuation and diacritics were afterwards either

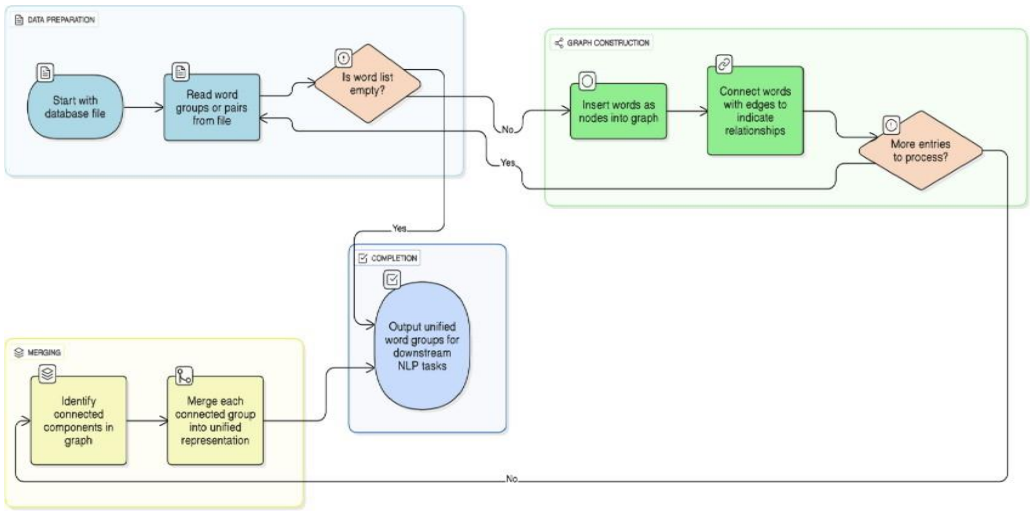

Fig 1. The overall workflow of the proposed graph-based word normalization and synonym merging process

is illustrated standardized or removed to provide a consistent text. After that, the headlines were processed into tokens using rules appropriate for the language. This was followed by lemmatization of words, so that inflected forms could be clustered together.For the same reason, we removed these very frequent stop words as they have little sentiment information and may degrade potential classification performance. They are removed so that the model can concentrate on words which actually emote. The POS tags were finally verified against the tokenized words in the right sequence. These actions also enhanced the text representation quality and ensured a better modelling of the graph, since it maintained token boundaries and sentiment tags consistent.

3.4 Graph-Based Word Normalization

Figure 3.1 illustrates a graph-based word merging

process typical procedure in NLP works like lexical normalization and synonym summarization. The database file contains groups or pairs of words that convey the same meaning. Having read the data, it goes into a structure that in turn sets up a network. In this graph, words are the nodes, and their relationships, such as synonymy or morphological variants, are represented as edges. Then, the algorithm checks if the list of words is empty. If not, the algorithm inserts the words from the current entry as nodes into the graph and connects them with edges to indicate that they are connected in meaning or function. The algorithm then identifies groups of associated words that are connected directly or indirectly after all the significant words have been inserted into the graph. Each of these elements is a set of words that can be regarded as a single unified group. This merging process allows the system to reduce multiple forms or variations of words into one representation. This process is particularly beneficial for activities that follow, such as translation and sentiment analysis. The process continues until all words in the dataset file have been processed.

3.5 Synonym Merging Workflow

Input: Dataset containing word groups or synonym pairs Output: List of merged word clusters

1. Create an empty graph G
2. Read the dataset line by line
3. For each entry in the dataset:

a. Extract the words in the current group

b. For each word:If the word is not already in G, add it

as a node

c. For every pair of words in the group:

- Add an edge between them to show they are related

4. After all entries are processed:

a. Find all connected components in graph G

b. Each connected component becomes a merged synonym cluster

5. Return the list of all clusters

The algorithm begins with an empty graph. Each line from the dataset is treated as a set of related words, such as synonyms, spelling variants or morphologically linked forms. Every word becomes a node in the graph. When two words belong to the same group, the system connects them with an edge. Once all entries are processed, the graph contains several connected regions. Each connected region represents a cluster of words that share meaning, either directly or through chains of associations. The final step collects these regions and treats each one as a single merged lexical unit, which reduces redundancy and helps later stages of sentiment classification.

3.6 Sentiment Classification using Machine Learning Models

The optimized lexicon and graph-informed features were evaluated using seven machine learning classifiers to measure performance. Feature vectors were generated using TF-IDF, applied after graph-based normalization. During the classification process, several machine learning models were employed in order to evaluate how effective the sentiment-rich features of the optimized Marathi Sentiment Dictionary 3.1 and graph-based processing were, utilizing seven classifiers: logistic regression, stochastic gradient descent (SGD), support vector machine (SVM), nearest neighbour, decision tree, naïve Bayes, and an ensemble classifier.

Logistic regression was an appropriate choice because it could represent linear relationships in sentiment-weighted TF-IDF feature space. SGD could process large datasets and perform normalization effectively. It performed particularly well with sparse, high-dimensional input. SVM performed well in classification by identifying the optimal hyper planes to differentiate between positive and negative emotions. The system performed well in the new feature space. The Nearest Neighbour classifier ranked documents based on their proximity to labelled instances. This approach relied heavily on the performance of the distance estimates in the sentiment-laden space. Decision Tree performed best accuracy by exploiting the graph-informed features to construct rule-based trees that were capable of capturing sophisticated patterns of emotion. Although Naïve Bayes made the assumption that features were independent, it performed well because the dataset contained a lot of words that conveyed emotions. Finally, the ensemble classifier aggregated the strengths of a large number of base models to make the predictions more precise and robust in general. All of these models demonstrated that aggregating language preprocessing, sentiment graph construction, and optimised feature engineering significantly enhanced sentiment classification performance on Marathi. Using seven models different algorithms handle the feature space and which one performs best for Marathi sentiment data. They are all fit for the data because text classification typically benefits from a variety of models some handle sparse high-dimensional data better (SGD, SVM), some capture complex patterns (Decision Tree), some provide probabilistic reasoning (Naïve Bayes), and ensembles

combine these strengths. The comparison ensures the chosen method is both effective and reliable across different algorithmic perspectives.

4. EXPERIMENTS AND RESULTS

The experimental work, the dataset used in this study includes 9,673 Marathi news headlines taken from an open Kaggle corpus. The data was split in an 80:20 ratio, with 7,738 headlines used for training and 1,935 reserved for testing. Every headline was manually tagged as positive or negative. In the training set, 6,084 samples were labelled positive and 1,654 were labelled negative. There are 1,520 positive and 414 negative samples in the test set. The full dataset consists of 7,605 positive and 2,068 negative headlines. Proposed a method for analysis of opinion of the Marathi language using graph-based approach and machine learning classifiers. The intuition of the approach is to construct a sentiment graph using pruned subset of Marathi Sentiment Dictionary 3.1. The original linguistic dictionary had 63,897 lexicon sets, however in order to keep it useful and manage-able. It was reduced to 38,468 sets. This reduce was based on factors like polarity strength, which shows how strongly a word expresses positive or negative sentiment, and contextual distinctiveness, which indicates how well a word helps distinguish sentiment in different situations. This trimming kept the lexicon rich enough to be useful and efficient for systems with limited resources.

To normalize and remove noise the data, the raw dataset was processed using a standard preprocessing pipeline consisting of several key steps. Initially, null entries were removed to ensure data integrity. The text was then normalized by eliminating punctuation and standardizing accents. Tokenization was applied to segment news stories into individual tokens based on language-specific rules. Subsequently, lemmatization was performed to reduce words to their base or dictionary forms, thereby consolidating different inflected variants. Each node in the graph was assigned a sentiment polarity score using MSWN 3.1 for positive, negative, and objective classes. For ambiguous nodes, such as having more than one synset, averaging or choosing the most frequent sense was carried out for the preliminary assignment of scores to

Unigram features were extracted using TF-IDF, which captured the sentiment information inherent in the graph structure to prepare the data for classification, inputting these into a set of machine learning classifiers.

4.1 Graph Construction Strategies

The selection of nodes and edges in a graph created from Marathi text for sentiment analysis has a big influence on the kind of data the network can capture and how sentiment may be deduced or spread.

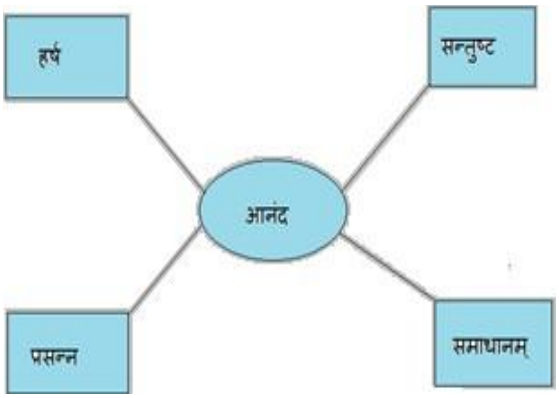

Fig. 2 Graph-based co-occurrence structure showing edges between word nodes

In a co-occurrence graph, nodes are individual words, and edges show how often or strongly they occur together in a certain context window, such as a phrase or a ±5-word span. Preprocessing is quite important when making a graph like this for Marathi. This includes tokenization, lemmatization, and often stop-word removal to reduce noise. Lemmatization is especially critical in Marathi due to its rich inflectional morphology for instance, different gendered and number forms like As illustrated in Figure 4.1, related word forms such as आनंदः (Ānandaḥ - Joy), सन्तुष्ट (Santuṣṭa - Contentment), and समाधानम् (Samādhānama –

Satisfactory) should ideally be mapped to a base lemma likeआनंद to ensure that semantically equivalent words are treated as a single unit.Edges between nodes are formed when two words appear within the defined context window. These edges can be weighted in various ways: the simplest being frequency-based weighting, counting how often two words co-occur. A more insightful approach involves Pointwise Mutual Information (PMI), which captures how much more likely two words are to appear together than by chance.

$$PMI(w_1,w_2)=log_2\left(\frac{P(w_1,w_2)}{P(w_1).P(w_2)}\right) \quad \text{…..}$$

Equation 2

Where PMI- Pointwise Mutual Information

w_1 = the first word you are interested in

w_2 = the second word you are interested in

P (w_1) is the probability of the word $w_1$appearing in your corpus.

P (w_2) is the probability of the word $w_2$appearing in your corpus

For sentiment analysis, initial sentiment scores are

assigned to each node using the Marathi sentiment Dictionary 3.1. Each word is looked up in the lexicon. If found and associated with one or more synsets (word senses), sentiment attributes (positive, negative, and possibly objective scores) are extracted. Handling polysemy (multiple meanings) may require strategies like averaging all synset scores or using only the most frequent sense. Words not found in Dictionary 3.1are typically assigned neutral (zero) sentiment values initially. These scores can then serve as a foundation for further tasks such as sentiment propagation across the graph or contextual sentiment analysis.

4.2 Analysis and Interpretation

A consistent sentiment analysis feature set was used to evaluate seven classifiers. Accuracy and F1-Score, two well-known and accurate measures for text categorization tasks, were used to evaluate their performance.

Table 4: Classifier Performance with TF-IDF.

Sr. No.	Classifier	Unigram + TF-IDF optimization (MSD2.0+MSD3.1 WKWS-8510)	
		Accuracy %	F-Score %
1	Logistic Regression	90.38	94
2	Stochastic Gradient Decent	95.85	98.21
3	SVM	92.83	96.18
4	Nearest Neighbour	92.46	95.93
5	Decision Tree	99.81	99.89
6	Naïve Byes	90.29	94.89
7	Ensemble Classifier	95	97.36

To evaluate the effectiveness of this approach, several classifiers were trained and tested, including Logistic Regression, Stochastic Gradient Descent (SGD), Support Vector Machine (SVM), Nearest Neighbour, Decision Tree, Naïve Bayes, and an Ensemble Classifier. As reported in **Table 4.2** all classifiers were tested with the same features taken from the sentiment graph and the optimized MSWN. Their performance differed quite a bit. The Decision Tree model stood out, reaching 99.81% accuracy and an F-score of 99.89. Its gain possibly comes from its ability to capture non-linear patterns and sudden shifts in sentiment. The Stochastic Gradient Descent model also performed well, reaching 95.85 percent accuracy and a 98.21 F-score, which reflects solid generalization. Ensemble models stayed reliably strong, while Naïve Bayes and Logistic Regression delivered stable but comparatively lower results.

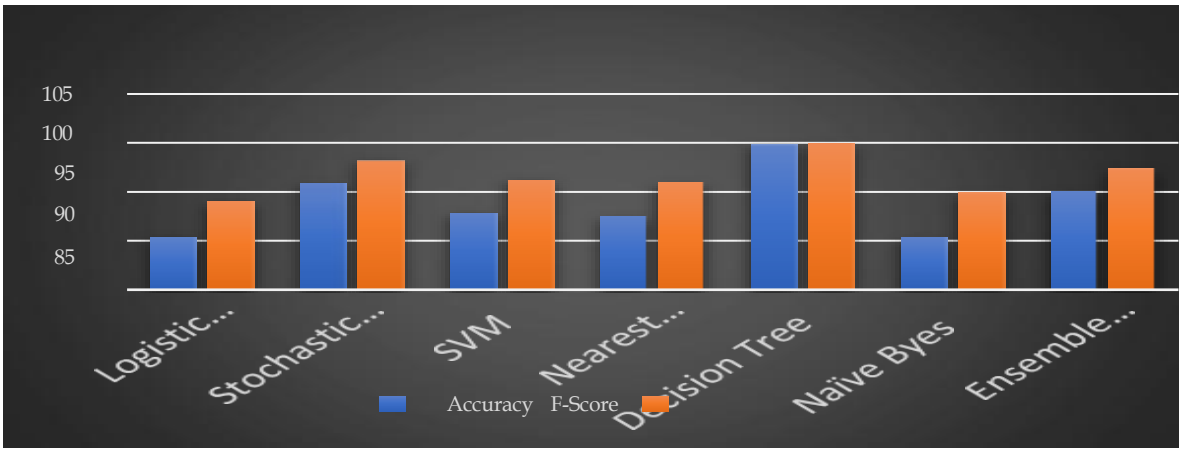

Fig 3 presents the classification results obtained using TF-IDF features derived from the optimized sentiment graph.

Figure 4.2 presents the overall framework used in this to improve sentiment analysis for morphologically complex languages such as Marathi.This paper determines that the sentiment analysis of morphologically complex languages, such as Marathi, can be enhanced by combining lexical optimization, graph-based modelling, and machine learning. Although the revised MSWN subset keeps the feature space inexpensive and focused, the sentiment graphs contribute to consistent normalization and smoother sentiment flow between related phrases. Everything considered, the approach handles linguistic variety, is scalable, and provides a useful basis for sentiment analysis in low-resource languages.

5. CONCLUSIONS

This work the challenges of sentiment analysis in

low-resource and morphologically complex languages and presents a comprehensive linguistically informed graph-based approach for Marathi. The method combines lexical refinement, graph building, synonym merging, and syntactic dependency information to form a sentiment-aware network that can capture deeper semantic relationships in Marathi text.

One of the major contributions is the development of the structured sentiment graphs, which are created from the optimized Marathi Sentiment Dictionary 3.1. In such a graph, words appear as nodes, and their links show synonymy, contextual links, and grammatical relationships. Merging synonyms within this graph removes duplicates, while it brings together words that share the same meaning, keeping the lexicon both compact and semantically reliable.

The features extracted from the sentiment graph are then transformed into a TF-IDF weighted vector space and used with several traditional machine learning models. The best performance was given by a Decision Tree model, reaching an accuracy of 99.81%, with a 99.89 F1 score, proving the capability of successfully handling nonlinear and rule-driven sentiment patterns. The performance of the SGD model turned out to be very strong, especially on high-dimensional and sparse textual data. Finally, ensemble methods added even more stability and maintained high accuracy across different evaluations.

Future work may involve testing deep learning architectures; extensions of the approach to cross-lingual sentiment transfer; and domain-specific sentiment lexicon development to enhance the adaptability and reach of this framework.

Acknowledgment

The authors express their gratitude towards BARTI, Pune, Maharashtra, India, for the fellowship received. The Data analytics Research Laboratory at Dr. Babasaheb Ambedkar Marathwada University in Maharashtra, India, provided valuable support and lab facilities for this research.
We acknowledge the contributors of the publicly available Kaggle datasets used in this work for utilized their resources accessible to the research .
Competing Interests: We declare that we have no competing interests that could have influenced the research, analysis, or conclusions presented in this work. Our sole intention is to contribute to the academic and scientific community by providing unbiased and transparent findings.
Funding Information: This research was carried out without financial assistance. The authors declare that no funding was received from any governmental, private, or non-profit organizations.

Author contribution

Kiran Sonkamble: Conceptualized the study and designed the methodology developed the research framework and handled data collection and analysis.
Saroj S.Date: This contribution was instrumental in analyzing, reviewing, conceptualizing the study, and designing the methodology.
Prof.S.N.Deshmukh: This contribution played a key role in supervising the entire research, analyzing and reviewing findings, conceptualizing the study, and designing the methodology and results. Data Availability Statement: The dataset is publicly available and can be accessed at Kaggle. All preprocessing and modifications made during the research are documented and available upon request.

Data Availability Statement

Marathi news headlines from an open Kaggle corpus

Research Involving Human and/or Animals:

Not applicable Informed
Consent: Not applicable

REFERENCES

1. Jain, Y. S. Rathore, and S. Agarwal. "Sentimental Analysis Based on Machine Learning Technique." *Proceedings of the 2024 3rd International Conference for Innovation in Technology (INOCON)*, IEEE, 2024, pp. 1–7.
2. Pinto, Francisco. "Structured Methods of Representation of the Knowledge." 2019.
3. Wu, Yongliang, et al. "Knowledge Graph-Based Hierarchical Text Semantic Representation." 12 Jan. 2024.

4. Sharma, A., and U. Ghose. "Lexicon: A Linguistic Approach for Sentiment Classification." *Proceedings of the 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, IEEE, 2021, pp. 887–893.
5. Singh, V., et al. "Sentiment Analysis Using Lexicon-Based Approach." *Proceedings of the 2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, IEEE, 2018, pp. 13–18.
6. Raees, Muhammad, and Samina Fazilat. "Lexicon-Based Sentiment Analysis on Text Polarities with Evaluation of Classification Models." *arXiv*, 2024.
7. Taj, Soonh, Baby Bakhtawer Shaikh, and Areej Fatemah Meghji. "Sentiment Analysis of News Articles: A Lexicon-Based Approach." *Proceedings of the 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, IEEE, 2019.
8. Sumana, and P. Kanchan. "Hindi and Kannada Twitter Sentiment Analysis Using Machine Learning Algorithm." *Proceedings of the 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, IEEE, 2022, pp. 370–377.
9. Povoda, L., et al. "Sentiment Analysis Based on Support Vector Machine and Big Data." *Proceedings of the 2016 International Conference on Telecommunications and Signal Processing (TSP)*, IEEE, 2016, pp. 543–545.
10. Lutfullaeva, M., et al. "Optimization of Sentiment Analysis Methods for Classifying Text Comments of Bank Customers." *IFAC-PapersOnLine*, vol. 51, no. 32, 2018, pp. 55–60.
11. Zhan, Tong, et al. "Optimization Techniques for Sentiment Analysis Based on LLM (GPT-3)." *Applied and Computational Engineering*, vol. 67, 2024, pp. 41–47.
12. Machová, K., et al. "Lexicon-Based Sentiment Analysis Using Particle Swarm Optimization." *Electronics*, vol. 9, no. 8, 2020, p. 1317.
13. Ahmad, Munir, et al. "Sentiment Analysis of Tweets Using SVM." *International Journal of Computer Applications*, vol. 177, 2017.
14. Muthuvel, P., et al. "Optimizing Road Networks: A Graph-Based Analysis with Path-Finding and Learning Algorithms." *International Journal of Intelligent Transportation Systems Research*, 2024.
15. Chen, Zuo, et al. "Domain Sentiment Dictionary Construction and Optimization Based on Multi-Source Information Fusion." *Intelligent Data Analysis*, vol. 24, 2020, pp. 229–251.
16. Tang, Duyu, Bing Qin, and Ting Liu. "Deep Learning for Sentiment Analysis: Successful Approaches and Future Challenges." *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 5, no. 6, 2015, pp. 292–303.
17. Errami, Mouaad, et al. "Sentiment Analysis on Moroccan Dialect Based on ML and Social Media Content Detection." *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 3, 2023.
18. Gaind, Bharat, et al. "Emotion Detection and Analysis on Social Media." *arXiv*, 2019.
19. Chalupa, Stepan, et al. "Improving Service Quality Using Text Mining and Sentiment Analysis of Online Reviews." *Quality – Access to Success*, vol. 22, no. 182, 2021.
20. Jain, T., et al. "Sentiment Analysis of Movie Reviews in Hindi Language Using Machine Learning." *Proceedings of the 2022 IEEE International Conference on Data Science and Information System (ICDSIS)*, IEEE, 2022, pp. 1–5.
21. Sharma, A., and U. Ghose. "Toward Machine Learning Based Binary Sentiment Classification of Movie Reviews for Resource Restraint Language (RRL) Hindi." *IEEE Access*, vol. 11, 2023, pp. 58546–58564.
22. Akhtar, Md Shad, et al. "Aspect-Based Sentiment Analysis: Category Detection and Sentiment Classification for Hindi." Springer, 2018.
23. Chaware, S., and S. Rao. "Ontology Supported Inference System for Hindi and Marathi." *Proceedings of the 2012 IEEE International Conference on Technology Enhanced Education (ICTEE)*, IEEE, 2012, pp. 1–6.
24. Jain, A., et al. "KNetwork: Advancing Cross-Lingual Sentiment Analysis for Enhanced Decision-Making in Linguistically Diverse Environments." *Knowledge and Information Systems*, 2024.
25. Sunil, M. E., and S. Vinay. "Kannada Sentiment Analysis Using Vectorization and Machine Learning." Springer, 2022, pp. 677–689.
26. Thet, Tun Thura, et al. "Aspect-Based Sentiment Analysis of Movie Reviews on Discussion Boards." *Journal of Information Science*, vol. 36, 2010, p. 823.
27. Thet, T. T., et al. "Sentiment Classification of Movie Reviews Using Multiple Perspectives." *Lecture Notes in Computer Science*, vol. 5362, Springer, 2008.

28. Singh, Vivek, et al. "Sentiment Analysis of Movie Reviews: A Feature-Based Heuristic for Aspect-Level Sentiment Classification." 2013, pp. 712–717.
29. Yu, Jianxing, et al. "Aspect Ranking: Identifying Important Product Aspects from Online Consumer Reviews." 2011, pp. 1496–1505.
30. Passos, N. A. R. A., et al. "NetworkX-Temporal: Building, Manipulating, and Analysing Dynamic Graph Structures." 2024.