

Detecting Algorithmic Bias in Turkish E-Commerce Reviews: A Systematic Comparison of Supervised, Lexicon-Based, and Unsupervised Polarity Analysis Methods

Betul Kan-Kilinc

Journal Title
XX(X):1–14
©The Author(s) 2016
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/

SAGE

Abstract

This study explores a critical feedback loop of algorithmic bias on e-commerce platforms, where the design of recommendation systems incentivizes user behaviors that ultimately distort the data these systems depend on. Specifically, algorithms that prioritize products with higher ratings encourage users to submit inauthentic 5-star reviews—even after negative experiences—in order to boost the visibility of their feedback. This tactic manipulates algorithmic sorting, creating a self-reinforcing cycle that artificially inflates product scores and misleads consumers. An exploratory analysis was conducted and developed a phrase-based polarity dataset to uncover sentiment-rating mismatches. We evaluated a range of natural language processing techniques, including supervised classifiers, unsupervised methods, and Turkish-specific sentiment corpora. These approaches were validated against a manually annotated subset of reviews. The findings offer a robust framework for detecting this pervasive form of platform manipulation and underscore the urgent need to design recommendation algorithms that are resilient to strategically motivated bias. Clustering enabled the identification of latent emotional patterns across reviews, particularly those expressing negative sentiment masked by high ratings.

Keywords

Text mining, NLP, clustering, polarity

Introduction

The ways of interacting with our surroundings have extended to various online platforms with the developments in technology. This results in vast amounts of unprocessed data on the internet. Today, many people rely on websites for travel, accommodation, and online shopping. Explorers, travelers, customers, and online shoppers share their vacation, entertainment, and shopping experiences through community-based platforms like Tripadvisor, Expedia, Yelp, Booking, Amazon, eBay, Trendyol, Hepsiburada, and MDPI, etc. Among these, Tripadvisor reached a major and historic Internet milestone: 1 billion reviews and opinions (Tripadvisor 2022). These websites enable a daily influx of user activity logs and reviews. Online reviews—whether positive or negative—are potential insights written by previous customers about products or services. This information, made available through companies, can reach a vast audience of individuals and institutions (Hennig-Thurau et al. 2008).

Online platforms offer users the opportunity to share their experiences and evaluate specific products. Considering the number of users on these platforms, along with the reviews and ratings they provide over time, an enormous amount of textual data emerges. For example, "Amazon" operates with a system focused on customer reviews. Globally, e-commerce systems are evolving, leveraging big data analytics to expand into international markets and create sustainable revenue models (Saura 2018). The research has

shown how online reviews influence consumer behavior. Chevalier and Mayzlin (2006) demonstrated that long and short-term book sales on "Amazon" and "bn" are influenced by star ratings. Similarly, Hofstede (2010) conducted a research on how culture affects consumer decision-making processes using data from 117,000 employees in 40 countries. Thus, these experiences accumulated on the web, termed "learning from others" directly reach millions of people and have an extraordinary impact regardless of geography.

Given the influence of user reviews, effective multilingual processing is essential for ensuring accessibility and accuracy in consumer insights, which is where advanced translation models come into play. The Helsinki-NLP OPUS-MT model is an open-source neural machine translation system covering over 1,200 corpora across 747 languages. The Helsinki-NLP OPUS-MT model focuses on English-to-Turkish translation, leveraging large-scale datasets to improve linguistic accuracy (Tiedemann and Thottingal 2020; Tiedemann 2020).

Eskisehir Technical University, Turkiye

Corresponding author:

Betul Kan-Kilinc, Eskisehir Technical University, Department of Statistics, Turkiye

Email: bkan@eskisehir.edu.tr

Building on these advancements in multilingual translation, recent developments in deep learning have further transformed natural language understanding—particularly in sentiment analysis tasks.

BERT (Bidirectional Encoder Representations from Transformers) is a groundbreaking NLP model introduced by Google in 2018 and uses the Transformer (Vaswani et al. 2017), (Devlin et al. 2018), which relies on self-attention mechanisms to weigh the importance of words in a sentence dynamically. Belaroussi et al. (2025) examines how BERT-based models (including DistilBERT, RoBERTa, and BERTweet) compare to LSTM-based models in classifying positive, negative, and neutral sentiment in Yelp reviews. The research highlights that transformer models outperform traditional RNN-based models, but more complex architectures do not always guarantee better results. The DistilBERT base uncased finetuned SST-2 model is a fine-tuned version of DistilBERT, optimized for sentiment analysis on the Stanford Sentiment Treebank dataset (Sanh et al. 2019). Reddy et al. (2025) presents a hybrid sentiment analysis framework combining rule-based models (VADER) and deep learning models (DistilBERT) to analyze corporate reputation on social media. The study finds that DistilBERT enhances sentiment classification accuracy, particularly in distinguishing positive and negative sentiment trends across major corporations explores fine-tuned DistilBERT models for sentiment analysis, comparing them to LSTM and VADER-based approaches. The study demonstrates that DistilBERT achieves 92.4% accuracy, significantly outperforming traditional methods in detecting polarity in restaurant reviews (Gao 2021).

BERTurk, is a community-driven, cased BERT model specifically trained for Turkish NLP tasks. It was developed by the MDZ Digital Library team (dbmdz) at the Bavarian State Library and is available on Hugging Face (Devlin et al. 2018). It was pretrained on diverse Turkish corpora, including Wikipedia, OPUS datasets, and the Turkish OSCAR corpus.

Hunspell is an open-source spell checker and morphological analyzer designed for languages with complex word structures, such as Turkish, Hungarian, Finnish and German. Hunspell supports Turkish through specific dictionaries, such as hunspell-tr, which is optimized for Turkish spell checking (Ooms 2025).

Eryiğit (2014) presented a Turkish natural language processing platform and provided tools for morphology, syntax and entity recognition.

SentiWordNet is a widely used lexical resource for sentiment analysis and opinion mining, built on top of the WordNet database (Baccianella et al. 2010). It assigns three sentiment scores—positivity, negativity, and objectivity—to each synset (a group of synonymous words) in WordNet. These scores range from 0.0 to 1.0 summing to 1.0 for each synset, allowing for nuanced sentiment representation. Since it is a sentiment lexicon built for the English language, directly tied to the English WordNet synsets, it is not considered to be used in this study (Ucan et al. 2016).

SentiTurkNet is a Turkish sentiment lexicon designed to support sentiment analysis tasks in Turkish-language texts. It was developed to address the lack of high-quality, language-specific polarity resources for Turkish, which is a

morphologically rich and agglutinative language—meaning words can carry complex sentiment cues through suffixes and compound forms (Dehkharghani et al. 2016).

An automated method to generate word-emotion lexicons by leveraging Google n-gram frequencies around target words, expanding a seed lexicon (NRC Word-Emotion Association Lexicon) is presented by Altinel-Girgin et al., (Perrie et al. 2013). The approach achieved improved performance over manual lexicons in affective text classification tasks()

Labille et al. (2017) generated a domain-specific sentiment lexicon using a combination of probabilistic and information theoretic weights. They showed that domain-specific lexicons are more accurate than generic lexicons in sentiment analysis.

Labille et al. (2016) proposes novel methods to automatically generate sentiment lexicons using probabilistic and information-theoretic techniques on large corpora, achieving 87.6% accuracy (single lexicon) and 88.75% (ensemble approach) in sentiment analysis tasks.

Akbaş and Taşkın (2024) et al., categorizes Turkish sentiment analysis studies into dictionary-based, machine learning-based, and hybrid methods, providing a systematic comparison of methodologies and datasets. It highlights challenges like morphological complexity in Turkish and the scarcity of labeled datasets.

Ye et al. (2009) in their research classified online reviews to travel destinations by supervised machine learning approaches. They indicated that the Support Vector Machine(SVM) and N-gram approaches outperformed the Naive Bayes approach when the training datasets have a large number of reviews.

Kaya et al. (2012) et al., compares Naive Bayes(SB), Maximum Entropy, SVM, and N-gram models for sentiment classification of Turkish political news columns, finding Maximum Entropy and N-gram models superior NB and SVM.

Recent advances in topic modeling have opened new avenues for sentiment-driven forecasting across high-impact domains. Notably, Zhu (2024) demonstrated the predictive power of BERTopic in capturing market sentiment fluctuations, offering scalable insight extraction from large-scale unstructured data streams.

Drawing upon these foundational resources and methodologies, this study applies sentiment analysis to a real-world challenge in Turkish e-commerce: the mismatch between user-generated review text and numerical ratings. By leveraging both lexicon-based and machine learning approaches, the study aims to uncover biases and improve product evaluation systems. It addresses a critical challenge in e-commerce analytics: the discrepancy between user-generated text and numerical ratings. Specifically product reviews are investigated, where users sometimes assign high star ratings despite writing negative comments, creating a significant bias in product evaluation and recommendation systems. To detect sentiment-rating mismatches, this study conducts a comprehensive comparative analysis of NLP techniques. It evaluates multiple supervised classifiers (Logistic Regression(LR), support vector machines(SVM),

Random Forest(RF), XGBoost), unsupervised methods (K-means, Hierarchical Clustering, Latent Dirichlet Allocation-LDA), and lexicon-based approaches (BERTurk, Senti-TurkNet, TRSAv1, HUMIR) on a manually annotated subset of reviews and ratings, which serves as a gold standard for polarity prediction. Subsequently, sentiment polarity is predicted for a larger dataset comprising 260,308 reviews from a major Turkish e-commerce platform.

The paper is organized as follows: Section 2 describes the data acquisition process, including sources and preprocessing steps. Section 3 outlines the methodology, detailing the analytical techniques and tools employed. Section 4 presents the results and discusses key findings, while Section 5 concludes with implications and future research directions. The subsequent sections provide a comprehensive exploration of each stage, ensuring a clear understanding of the study's workflow and contributions.

Data acquisition

All data used in this study were collected from publicly accessible pages of a leading Turkish e-commerce platform. The data collection process complied fully with the platform's terms of service, and no login was involved. To protect user privacy, all personally identifiable information (PII) was excluded or anonymized during preprocessing. Usernames and seller identifiers were retained only in aggregate or pseudonymized form to support reproducibility without compromising privacy and not shared. Reviewer data were accessed, and all analyses were conducted on publicly visible content available to any site visitor.

In this study, the e-commerce study examines algorithmic biases in product reviews, relying on massive textual data generated by consumers. It is assumed that sufficient reviews exist to investigate algorithmic biases, considering the evaluations accompanying these sales. The data were gathered from a leading Turkish e-commerce platform. This platform was selected due to the significant increase in sales observed during promotional periods, where companies offering discounts often experience up to a fourfold rise in sales compared to regular periods, resulting in millions of product transactions (Statista 2025). E-commerce site facilitates online product sales across various categories such as home, furniture, cosmetics, clothing, sports, accessories, and shoes. The research focuses on developing a method to separate erroneous information caused by the algorithm that ranks products at the top from user-driven bias. For this purpose, the first step involves examining the "women/men" category on the platform, specifically the apparels, and exporting the required information for analysis through web scraping.

Compared to related works, no definitive method exists in the literature for determining the number of reviews to examine. This study examines 260,308 user comments collected from an online shopping platform to investigate the bias in high ratings associated with negative phrases (Kan et al. 2025). Web scraping techniques were employed to acquire this dataset and was programmatically collected via the platform's API, a well-known shopping platform in Türkiye, and stored for customized analysis. The API returns a JSON object containing review data, which was

parsed using the jsonlite package to extract key fields such as ratings (1–5 stars), reviews, reviewer details (e.g., elite status) (Ooms 2022). These structured data points then facilitated subsequent analytical operations.

It is important to note that while rvest is commonly used for scraping static web content, it cannot fully handle dynamic JavaScript-rendered pages on its own (Wickham 2024). Hence, chromote allows for direct interaction with the browser environment, enabling the function to execute JavaScript and mimic user actions, such as scrolling, to load dynamic content (Aden-Buie and Schloerke 2025). This combination of tools ensures that both static and dynamic web elements are effectively captured during the scraping process.

Data manipulation and processing steps were involved in organizing the scraped data using tidyverse and its dependencies (Wickham et al. 2019; Wickham 2019, 2022). Functions such the purrr package in tidyverse are used to efficiently iterate through the extracted comments and convert them into data frames, ensuring that the data is well-structured (Wickham and Henry 2023). Additionally, duplicate entries are removed while maintaining a clean and unique dataset for analysis.

As the primary aim of this study is to identify negative user comments that are paired with high rating scores, only publicly available fields—user comments, ratings, and document IDs—are retained for further analysis. Additionally, the research process will cover product reviews and other information for products sold between June 11, 2025, and April 16, 2023.

Creating shoppingwords-based lexicon

A corpus is a large collection of written or spoken texts used for linguistic analysis. These texts can come from diverse sources, such as books, newspapers, conversations, or social media. Corpora are often used to study language usage, patterns, and frequencies. They provide real-world examples of how language is used in context. While it is a crucial preprocessing step, a full corpus typically includes contextual data rather than standalone tokens.

However, a lexicon is essentially a dictionary or a collection of words along with their meanings, definitions, and sometimes additional information like pronunciation, usage, or grammatical details. Lexicons are often used to understand word meanings and structure, and may form the basis for natural language processing (NLP) tasks like language generation or sentiment analysis. For instance: Merriam-Webster Dictionary or a specific lexicon for computational linguistics like WordNet are differ from corpora.

In this study, we introduce a domain-specific corpus derived from Turkish e-commerce of reviews, made publicly available via the shoppingwords R package (Kan et al. 2025). Below is a summary of the contents provided on CRAN:

- **Text Data:** Sentences or paragraphs, including the original comments from users.
- **Metadata:** Information about the rating scores and id.
- **Storage Format:**

- CSV (.csv): Useful for tokenized data or structured metadata (rows for sentences/words with attribute columns).

All analyses in this study were conducted using the datasets provided by the shoppingwords R package (Kan et al. 2025).

Data collection begins by specifying a product URL—such as a link to a women’s bodysuit—and then constructing two key API endpoints: one for retrieving review content (including ratings, comments, and metadata) and another for obtaining the number of likes (i.e., helpful votes) associated with each review. After retrieving and merging data from both endpoints, the script outputs a structured CSV file. This file contains detailed columns, including review ID, rating, comment, author, date, reviewer status (elite or influencer), as well as product-specific characteristics like size, height, weight, and the number of likes each review has received—reflecting its perceived helpfulness. However due to package size limitation only, 260,308 of reviews, ratings and id columns are publicly available on CRAN(Kan et al. 2025).

After removing duplicate product URLs by performing a full comparison of the imported data reducing multiple occurrences to a single entry—a refined sample of 1,481 documents was obtained. This sample size was informed by a pre-study conducted on the broader dataset, which revealed that only approximately 0.5% of reviews with high star ratings contained negative sentiments. Given the rarity of such cases, the selected sample was deemed sufficient to support a focused analysis of sentiment mismatches while keeping manual annotation efforts manageable. This dataset was used to examine discrepancies between high ratings (e.g., 5 stars) and the presence of negative phrases in user reviews, where each review was manually annotated with polarity labels (negative/positive). Regarding annotation, all polarity labels were manually assigned by a single researcher to prevent subjective interpretation of multiple reviewers. While existing Turkish sentiment lexicons and tools (e.g., BERTurk, Zemberek) provide pre-defined polarity levels, we manually annotated sentences to enable direct comparison across multiple lexical resources with differing polarity criteria (Schweter 2020; Ahmet 2023; Yıldırım 2024). This approach was critical because: (1) Automated tools often disagree due to variations in lexicon design (e.g., word coverage, polarity thresholds), and (2) Python cannot natively harmonize these discrepancies without manual validation. By combining manual labels with programmatic integration of external lexicons (e.g., Turkish sentiment wordlists), we rigorously evaluated bias detection under conflicting polarity standards. The BERTurk model itself is distributed as a Python-compatible Hugging Face transformer model, however many Turkish polarity resources—such as SentiTurkNet (Dehkharghani et al. 2016) or TRSAv1 (Aydoğan and Kocaman 2022)—are published as standalone datasets in formats like XML or CSV. The installation of python packages was difficult and complicated or not possible, as at that time, the researchers were unable to install some of them and had to install a virtual machine instead and the reticulate R package as a workaround(Ushey et al. 2025).

After preprocessing, the documents remain, irrelevant comments not in Turkish and also irregular spaced were removed. A list of a shoppingwords based stopwords were created after user comments, that is present on CRAN, were gathered (Kan et al. 2025). The total number of user comments extracted is referred to n documents, each representing a corpus. For further analysis, the text is divided into tokens and word-groups. In m consecutive words, this token is called a bigram, trigram or n-gram. A group of n-gram words are created from user comments and is accessible on CRAN (Kan et al. 2025). Word-groups were chosen based on exploratory analysis of the training data, such as observing high ratings with negative comments. The selected word-groups clearly indicate the categories under consideration. Our method is heavily based on the selected word-groups. This approach aligns with the rule-based classification. Data manipulation and processing steps were involved in organizing the scraped data using tidyverse and its dependencies.

Application

This work focuses exclusively on capturing negative comments associated with high product ratings from online shopping platforms, as they particularly create a bias for rating scores. We introduce shoppingwords, an R package, featuring Turkish test dataset, which contains 1,481 manually annotated reviews including true polarity. To systematically address this bias and improve sentiment classification, first a supervised learning framework capable of detecting nuanced patterns in user reviews were implemented. This approach enables us to move beyond rule-based heuristics and leverage data-driven models for more robust sentiment analysis.

Model performance was assessed using accuracy, precision, recall, F1-score, specificity, Cohen’s kappa, and the Youden index. These are formally defined as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision}_{\text{neg}} = \frac{TN}{TN + FN}$$

$$\text{Recall}_{\text{neg}} = \frac{TN}{TN + FP}$$

$$F1_{\text{neg}} = 2 \cdot \frac{\text{Precision}_{\text{neg}} \cdot \text{Recall}_{\text{neg}}}{\text{Precision}_{\text{neg}} + \text{Recall}_{\text{neg}}}$$

$$\text{Youden's Index} = \text{Sensitivity} + \text{Specificity} - 1$$

$$\text{Cohen's Kappa} = \frac{p_o - p_e}{1 - p_e} \text{ where}$$

$$p_o = (TP + TN) / \text{total}$$

$$p_e = ((TP + FP) * (TP + FN) + (FN + TN) * (FP + TN)) / \text{total}^2$$

The best-performing models were compared across test sets to determine the optimal strategy for emotion classification.

The structure of each algorithm was given in the tables.

The algorithm 1 represents a supervised classification framework employed using multiple machine learning

models trained on manually labeled user reviews to evaluate sentiment in the curated dataset. Following normalization and token filtering, each review was converted into TF-IDF vectors to represent word importance across the corpus. LR, SVM, RF, and XGBoost were implemented as supervised learners. Each model was trained on the TF-IDF features, and its performance was assessed using 5-fold cross-validation. For the final test set, sentiment was assigned according to a calibrated decision threshold. All processing, modeling, and validation steps were conducted using the tidymodels framework, which ensures reproducibility and consistency across workflows.

Algorithm 1 TF-IDF Feature-Based Sentiment Classification Using Multiple Models

Input:

Corpus $D = \{d_1, \dots, d_n\}$ with labels $y_i \in \{\text{negative}, \text{positive}\}$

Stopword list S , rating scores r_i

Models: Logistic Regression, SVM, Random Forest, XGBoost

Procedure:

1. **Text Preprocessing:**

- Normalize text: lowercase, strip emojis/punctuation
- Tokenize and remove stopwords S
- Reconstruct cleaned text per review

2. **Train/Test Split:**

- Divide D into training and testing sets using stratified sampling

3. **TF-IDF Feature Construction:**

- Apply tokenization and filtering (max 300 tokens)
- Transform tokens into TF-IDF matrix

4. **Model Fitting:**

For each model $m \in \{\text{LR}, \text{SVM}, \text{RF}, \text{XGB}\}$:

- Train model m on training set
- Perform 5-fold cross-validation
- Compute metrics

5. **Threshold-Based Prediction:**

- For final test set prediction:
- Assign label $\hat{y}_i = \text{negative}$ if $p_i > 0.35$; else positive
- Evaluate confusion matrix and compute metrics per model

Output:

Per-model evaluation metrics \mathcal{M}_m on test set and cross-validation summary

The second approach uses unsupervised learning methods offering a powerful approach for identifying negative user comments in datasets without requiring manually labeled examples. Techniques such as clustering, topic modeling, and anomaly detection allow algorithms to uncover hidden patterns in text data. For example, unsupervised methods such as k-means with PCA, hierarchical clustering with PCA, and topic modeling can group comments by semantic patterns, effectively surfacing clusters characterized by negative sentiment. The algorithm 2 details the PCA-enhanced clustering methodology, demonstrating how dimensionality reduction improves the separation of sentiment-driven patterns.

Algorithm 2 formalizes the hierarchical clustering procedure applied to PCA-transformed features, outlining

the key steps for dimensionality reduction and subsequent cluster analysis.

Algorithm 2 Cross-Validated Hierarchical Clustering with PCA

Input:

Corpus $D = \{d_1, \dots, d_n\}$ with emotion labels $y_i \in \{\text{negative}, \text{positive}\}$

Predefined phrase list $P = \{p_1, \dots, p_m\}$

k -fold partition $\{F_1, \dots, F_k\}$ of D

Procedure:

For each fold F_i :

1. **TF-IDF Feature Extraction:**

- Tokenize each document d_i into terms t_j
- Compute tf-idf: $\text{tf-idf}_{ij} = \frac{f_{ij}}{\sum_j f_{ij}} \cdot \log\left(\frac{N}{n_j}\right)$

2. **Dimensionality Reduction via PCA:**

Project TF-IDF matrix X to top $q = 50$ components: $X_{\text{PCA}} = X \cdot W_q$

3. **Clustering via Ward's Method:**

Apply hierarchical clustering with linkage: $C = \arg \min_{C_1, C_2} \sum_{i=1}^2 \sum_{x \in C_i} \|x - \mu_i\|^2$

4. **Cluster Labeling:**

Define mapping $\ell(C_j)$ based on majority emotion label in each cluster: $\ell(C_j) = \text{negative}$ if $\frac{n_{\text{neg}}}{n_{\text{total}}} > \tau$, else positive

5. **Test Sample Assignment:**

Assign each test sample x_i^{test} to nearest cluster: $\hat{C}_i = \arg \min_j \|x_i^{\text{test}} - \mu_j\|^2$

6. **Performance Evaluation:**

Compare predicted \hat{y}_i to ground truth y_i , compute metrics

Output:

Per-fold predictions \hat{y}_i , with aggregated mean \bar{M} and standard deviation σ_M across folds

Similarly, Latent Dirichlet Allocation (LDA) enables the extraction of prevalent negative topics from large corpora. Sentiment-based embeddings or lexical anomaly scores can further enhance detection by flagging emotionally charged language or outliers from typical conversational norms. These methods are particularly valuable for scalable monitoring of feedback in online platforms, where user sentiment is diverse, dynamic, and not always explicitly labeled. Algorithm 3 represents the structure of the analysis.

Algorithm 3 Cross-Validated Topic Modeling with LDA

Input:

Corpus $D = \{d_1, d_2, \dots, d_n\}$ where each document d_i has label $y_i \in \{\text{negative}, \text{positive}\}$

A k -fold partition $\{F_1, F_2, \dots, F_k\}$ of D

Procedure:

For each fold F_i :

1. **Text Preprocessing:**

- Tokenize document d_i into word sequence $w_i = \{w_{i1}, w_{i2}, \dots, w_{im}\}$
- Filter top T tokens: $w'_i \subseteq w_i, |w'_i| \leq T = 500$
- Compute Term Frequency: $\text{tf}(t, d_i) = \frac{f_{t, d_i}}{\sum_{t'} f_{t', d_i}}$

2. **Train LDA Model with $K = 2$ Topics:**

- Fit LDA using Gibbs Sampling: $\max_{\phi, \theta} p(W|\phi, \theta, z)$ where $z_i \sim \text{Multinomial}(\theta_{d_i})$

3. **Assign Topic Labels:** $\hat{z}_i = \arg \max_k p(z_i = k|d_i)$ for $k \in \{1, 2\}$

4. **Map Topics to Emotion Labels:** Based on majority emotion label:

$$\hat{y}_i = \begin{cases} \text{negative} & \text{if } \frac{n_{\text{neg},k}}{n_k} > 0.15 \\ \text{positive} & \text{otherwise} \end{cases}$$

5. **Performance Evaluation:** Compare predicted \hat{y}_i against true y_i

Output:

Per-model evaluation metrics \mathcal{M}_m on test set and cross-validation summary

Common Turkish lexicon and corpora

This study also compares four complementary approaches to Turkish sentiment analysis. It first leverages a multilingual transformer-based model (BERTurk) to perform binary sentiment classification—specifically identifying negative vs. positive emotional content in Turkish user-generated comments (Schweter 2020; Yildirim 2024). The implementation leveraged Hugging Face’s Transformers pipeline through Python’s reticulate interface (Ushey et al. 2025; Hugging Face 2024). Text preprocessing involved lowercase conversion, Unicode symbol removal, and punctuation stripping, followed by Turkish stopword filtering using a predefined lexicon. Algorithm 4 represents the structure of the BERTurk model analysis.

Algorithm 4 BERTurk Model for Turkish Sentiment Classification

Input:

Corpus $D = \{d_1, \dots, d_n\}$ with labels $y_i \in \{\text{negative}, \text{positive}\}$

Pre-trained transformer model: (Yildirim, 2024)

5-fold cross-validation splits $\{F_1, \dots, F_5\}$

Procedure:

1. **Text Cleaning:**

- Convert to lowercase, strip emojis and punctuation
- Tokenize and remove stopwords
- Reconstruct each d_i as preprocessed sentence

2. **Model Loading and Inference:**

- Load transformer pipeline with tokenizer and model
- Predict sentiment \hat{y}_i for each d_i via mini-batched inference

3. **Evaluation Loop:**

For each fold F_i : - Perform inference on fold’s held-out validation set

- Compare predictions \hat{y}_i with labels y_i
- Compute metrics

4. **Final:**

- Predict on held-out test set

Output:

Cross-validation metrics \mathcal{M}_{cv} and final test set performance \mathcal{M}_{test}

The second Turkish dataset, known as SentiTurk, processes XML-annotated words with positive (PSCORE) and negative (NSCORE) polarity scores, where document-level sentiment is determined by majority voting: a review is classified as positive when $\sum \text{PSCORE} > \sum \text{NSCORE}$, negative when $\sum \text{NSCORE} > \sum \text{PSCORE}$, and neutral

otherwise (Dehkharghani et al. 2016). Algorithm 5 represents the structure of the analysis calculated exclusively on classifiable reviews (excluding neutral predictions).

Algorithm 5 Lexicon-Based Sentiment Analysis using SentiTurk

Input:

Corpus $D = \{d_1, \dots, d_n\}$ with emotion labels $y_i \in \{\text{negative}, \text{positive}\}$

SentiTurk lexicon $L = \{(w_j, p_j, n_j)\}$ with polarity scores

5-fold cross-validation splits $\{F_1, \dots, F_5\}$

Procedure:

For each fold F_i :

1. **Lexicon Preparation:**

- Extract words and polarity scores from XML
- Assign each word to dominant sentiment:
- If PSCORE > NSCORE: positive
- If NSCORE > PSCORE: negative

2. **Token Matching and Prediction:**

- Tokenize review d_i into words
- Count matched words: $n_{\text{pos}}, n_{\text{neg}}$
- Predict sentiment:
- If $n_{\text{pos}} > n_{\text{neg}}$: $\hat{y}_i = \text{positive}$
- If $n_{\text{neg}} > n_{\text{pos}}$: $\hat{y}_i = \text{negative}$
- Else: neutral

3. **Filter Valid Predictions:**

- Keep documents where $\hat{y}_i \neq \text{neutral}$

4. **Evaluation:**

- Compare \hat{y}_i to true label y_i
- Compute metrics

Output:

Cross-validation and held-out test set evaluation metrics

For each review in the test dataset, the system tokenizes the text into individual words, matches them against entries in the SentiTurk sentiment lexicon, and assigns a sentiment label—either positive or negative—based on which polarity has more matches, defaulting to neutral when no sentiment-based words are found. The testdata consists of true emotions, it then calculates the metrics (accuracy, F1, etc.). The implementation followed a stratified 5-fold cross-validation protocol, with performance metrics.

Algorithm 6 represents the structure of the analysis using TRSAV1 dataset. It, serving both as a standalone corpora through term-frequency analysis and as a benchmark for model comparisons to support NLP research in Turkish, especially for sentiment classification tasks, provides a balanced corpus of 150,000 reviews. It was processed through a rigorous text normalization pipeline, converting all terms to lowercase and removing non-alphabetic characters before tokenization into individual word-sentiment pairs. The lexicon’s pre-annotated polarity scores were mapped to each token, creating a standardized sentiment dictionary. For document-level classification, reviews were analyzed using a majority voting system where sentiment was determined by aggregating polarities of constituent words - classified as positive when containing more TRSAV1-annotated positive terms than negative, with neutral assignments for ties or no matches. Evaluation followed a stratified 5-fold cross-validation, with confidence thresholds ($\zeta 0.01$) applied to filter low-certainty predictions.

Algorithm 6 Turkish Sentiment Evaluation with TRSAVI dataset**Input:**

Corpus $D = \{d_1, \dots, d_n\}$ with labels $y_i \in \{\text{negative}, \text{positive}\}$

Emotion lexicon $L = \{(w_j, s_j)\}$ in Turkish

Stratified 5-fold splits $\{F_1, \dots, F_5\}$ of D

Procedure:

For each fold F_i :

1. **Clean and Tokenize:**

- Remove stopwords, punctuation, emoji from each review d_i

- Tokenize into words $\{w_{i1}, \dots, w_{im}\}$

2. **Lexicon Matching:**

- Count $n_{\text{pos}}, n_{\text{neg}}$ words matching L per document

3. **Document Prediction:**

- Assign \hat{y}_i based on count:

- If $n_{\text{pos}} > n_{\text{neg}}$: positive

- If $n_{\text{neg}} > n_{\text{pos}}$: negative

- Else: neutral

- Calculate confidence: $c_i = \frac{\max(n_{\text{pos}}, n_{\text{neg}})}{n_{\text{pos}} + n_{\text{neg}} + \varepsilon}$

4. **Evaluate Predictions:**

- Keep documents with $\hat{y}_i \in \{\text{positive}, \text{negative}\}$ and $c_i > 0.01$

- Compute metrics

Output:

Fold summary metrics and test performance

Algorithm 7 Lexicon-Based Emotion Classification with HUMIR Dataset**Input:**

Corpus $D = \{d_1, \dots, d_n\}$ with labeled emotion $y_i \in \{\text{negative}, \text{positive}\}$

A manually constructed lexicon $L = \{(w_j, s_j)\}$ with word-sentiment pairs

5-fold cross-validation splits $\{F_1, \dots, F_5\}$ of D

Procedure:

For each fold F_i :

1. **Tokenize and Match Lexicon:**

Split each review d_i into words $\{w_{i1}, \dots, w_{im}\}$

Count $n_{\text{pos}}, n_{\text{neg}}$ matching words from lexicon

2. **Predict Polarity per Document:**

Assign label \hat{y}_i by comparing counts:

- If $n_{\text{pos}} > n_{\text{neg}}$: assign positive

- If $n_{\text{neg}} > n_{\text{pos}}$: assign negative - Else: assign neutral

Compute confidence $c_i = \frac{\max(n_{\text{pos}}, n_{\text{neg}})}{n_{\text{pos}} + n_{\text{neg}} + \varepsilon}$

3. **Filter Valid Predictions:**

Keep documents with $\hat{y}_i \in \{\text{positive}, \text{negative}\}$ and $c_i > 0.01$

4. **Evaluate Prediction Quality:**

- Compare \hat{y}_i to true label y_i

- Calculate metrics

Output:

Final performance on held-out test set and fold summary metrics

Next, the HUMIR sentiment dataset was processed from its raw CSV format through a structured cleaning pipeline that extracted and standardized polarity annotations (Ucan et al. 2016). The original dataset underwent text normalization, including case unification and non-alphabetic character removal, before being tokenized into individual word-sentiment pairs. Each term was mapped to its annotated polarity (Positive/Negative) while preserving morphological richness through careful Turkish-language aware processing. For evaluation, it was applied through a majority voting system where document-level sentiment was determined by aggregating polarities of constituent words. Reviews were classified as positive when containing more HUMIR-annotated positive terms than negative, with neutral assignments for ties or no matches. This classification was implemented through a cross-validated framework using five stratified folds, with confidence thresholds applied to filter low-certainty predictions (confidence ≥ 0.01). The evaluation metrics accounted for coverage limitations by tracking the proportion of classifiable reviews while maintaining strict alignment between predicted labels and ground truth emotions in the test set. Algorithm 7 represents the structure of the emotion classification using HUMIR dataset.

Results

Model performance with supervised classification methods

The analysis replicated the previous emotion classification prediction while incorporating review ratings as an additional predictive feature after the user comments are cleaned from stopwords, unnecessary punctuations

and emojis. Also the comments trimmed, squished and whitespace was collapsed. The same four models (LR, SVM, RF, and XGBoost) were evaluated using identical preprocessing steps and evaluation metrics, with the key modification being the inclusion of rating information in the base recipe. The data were split into training (80%) and test (20%) sets with stratified sampling to maintain class balance. Four supervised classification models were evaluated each tuned via 5-fold cross-validation.

In Table 1, reporting metrics as mean \pm standard deviation, RF emerged as the most consistent and high-performing classifier across the board, achieving strong scores in accuracy, recall, and F1, while also maintaining impressive specificity. Its predictive balance between sensitivity and specificity suggests it handles both positive and negative deftly, making it a robust choice for classification tasks. LR was close behind, presenting reliable scores with relatively low variability, which reinforces its reputation for stability and interpretability. XGBoost offered competitive metrics as well, trading a slight dip in accuracy for excellent overall balance—particularly appealing when one values model flexibility. Meanwhile, SVM lagged, delivering lower average metrics and greater variability.

The held-out test results in Table 2 presents the performance of four supervised classification models on held-out test data. Each model is evaluated using a suite of metrics that reflect different aspects of predictive quality. Among the evaluated models, RF, followed closely by LR, emerge as the most reliable performers, demonstrating consistently high accuracy, strong agreement with true labels, and a well-balanced trade-off between precision and recall. XGBoost also delivers competitive results,

Table 1. Cross-validation results for supervised classification models

Metric	Logistic	RF
Accuracy	0.930 ± 0.010	0.940 ± 0.024
F1 Score	0.813 ± 0.029	0.835 ± 0.069
Recall	0.764 ± 0.055	0.764 ± 0.082
Precision	0.873 ± 0.050	0.923 ± 0.062
Specificity	0.971 ± 0.013	0.984 ± 0.012
Kappa	0.770 ± 0.035	0.799 ± 0.084
Youden Index	0.735 ± 0.049	0.748 ± 0.091

Metric	SVM	XGB
Accuracy	0.901 ± 0.016	0.922 ± 0.015
F1 Score	0.752 ± 0.041	0.793 ± 0.038
Recall	0.751 ± 0.075	0.743 ± 0.052
Precision	0.760 ± 0.065	0.854 ± 0.057
Specificity	0.939 ± 0.021	0.967 ± 0.016
Kappa	0.691 ± 0.050	0.745 ± 0.046
Youden Index	0.690 ± 0.067	0.710 ± 0.051

particularly in precision and specificity, making it effective at minimizing false positives. While SVM shows slightly lower overall performance, it still maintains respectable recall and specificity, indicating its ability to detect sentiment with reasonable consistency. Taken together, these results suggest that traditional machine learning models — especially ensemble-based approaches — remain highly effective for sentiment classification tasks in Turkish-language review corpora.

Table 2. Performance metrics on held-out test data for supervised classification models

Metric	Logistic	RF	SVM	XGB
Accuracy	0.923	0.936	0.889	0.929
F1 Score	0.807	0.846	0.732	0.826
Recall	0.800	0.867	0.750	0.833
Precision	0.814	0.825	0.714	0.820
Specificity	0.954	0.954	0.924	0.954
Kappa	0.758	0.805	0.662	0.782
Youden Index	0.754	0.820	0.674	0.787

Overall, while RF and XGBoost are top performers in terms of accuracy and recall, RF stands out for its higher and robust classification performance, especially in handling negative classes effectively.

Model performance with common Turkish corpora

This section presents the comparative performance and reliability of four sentiment datasets—retrieved from BERTurk, SentiTurk, TRSAV1 and HUMIR (Hacettepe University Multimedia Information Retrieval Laboratory) in detecting negative comments collected from online sources (Yıldırım 2024; Demirtaş and Peçenak 2020; Aydoğan and Kocaman 2022; Ucan et al. 2016). The results obtained using the Turkish sentiment classifiers via 5-fold cross-validation are summarized in Table 3. Each row in this table represents the aggregated confusion matrix metrics for one

corpora type, calculated over all 5-fold data; results from the held-out test set are separately reported in Table 4. During each fold, the pre-trained sentiment classifier was applied to the corresponding validation subset. In Table 3, SentiTurk stands out with the highest recall and overall accuracy, indicating its strong ability to detect negative emotions in user reviews, even if its precision is modest. BERTurk demonstrates a well-balanced performance, with high precision and F1 score, suggesting it is effective at identifying negative sentiment when present, though its recall is slightly lower than SentiTurk, indicating it may miss some subtle cases. TRSAV1 and HUMIR, despite their high precision and specificity, struggle with recall and overall agreement, suggesting they are overly conservative and fail to capture the broader spectrum of negative sentiment.

Table 3. Cross validation results for Turkish Lexicon model predictions

Metric	BERTurk	SentiTurk
Accuracy	0.780 ± 0.050	0.805 ± 0.041
F1 Score	0.809 ± 0.037	0.620 ± 0.069
Recall	0.692 ± 0.057	0.814 ± 0.084
Precision	0.979 ± 0.021	0.502 ± 0.060
Specificity	0.971 ± 0.030	0.803 ± 0.036
Kappa	0.569 ± 0.095	0.499 ± 0.095
Youden Index	0.663 ± 0.059	0.617 ± 0.111

Metric	TRSAV1	HUMIR
Accuracy	0.294 ± 0.028	0.495 ± 0.041
F1 Score	0.383 ± 0.019	0.449 ± 0.033
Recall	0.241 ± 0.013	0.306 ± 0.030
Precision	0.939 ± 0.034	0.847 ± 0.030
Specificity	0.827 ± 0.127	0.887 ± 0.011
Kappa	0.019 ± 0.031	0.145 ± 0.039
Youden Index	0.068 ± 0.138	0.193 ± 0.038

Overall, while lexicon-based models like TRSAV1 and HUMIR offer clean predictions, they lack the sensitivity needed for nuanced sentiment detection, whereas SentiTurk and BERTurk offer complementary strengths in coverage and confidence.

The Table 4 presents performance metrics for four Turkish corpora-based models evaluated on held-out test data. The BERTurk model, despite its high precision (0.967), struggles to consistently identify negative sentiment — its recall for negative reviews is only 0.565, meaning it misses a significant portion of them. Its low F1 score (0.360) and modest kappa (0.327) suggest that while it's cautious and rarely mislabels a review as negative, it fails to catch many that truly are.

In contrast, the SentiTurk model performs much more reliably. With a recall of 0.826, it successfully identifies the majority of negative reviews. Its balanced F1 score (0.655) and higher Youden Index (0.649) indicate strong overall discrimination between positive and negative sentiment. This model is better suited for uncovering hidden dissatisfaction, especially in cases of rating inflation where textual cues contradict the star rating.

TRSAV1 and HUMIR indeed lean heavily toward precision, with values of 0.958 and 0.949 respectively,

Table 4. Performance metrics on held-out test data for Turkish corpora

Metric	BERTurk	Sentiment	TRSAV1	HUMIR
Accuracy	0.646	0.824	0.318	0.592
F1 Score	0.360	0.655	0.413	0.544
Recall	0.565	0.826	0.263	0.381
Precision	0.967	0.543	0.958	0.949
Specificity	0.525	0.823	0.882	0.964
Kappa	0.327	0.544	0.033	0.281
Youden Index	0.532	0.649	0.145	0.345

indicating that they are highly cautious when labeling reviews as negative. However, this caution comes at a significant cost to recall: TRSAV1 captures only 26.3% of actual negative reviews, making it the least sensitive model in the group, while HUMIR performs slightly better at 38.1% but still misses a substantial portion of negative sentiment. Both models show high specificity, meaning they rarely misclassify positive reviews, yet their low recall and moderate F1 scores limit their effectiveness in tasks where identifying dissatisfaction is critical.

In contrast, the Sentiment model offers the most balanced performance, with high recall and F1 score, making it well-suited for detecting negative emotions even when they're subtly expressed. BERTurk, while extremely precise, sacrifices recall and overall balance, suggesting it only flags negativity when highly confident but overlooks many true cases. Overall, Sentiment excels in sensitivity and general reliability, while TRSAV1, HUMIR, and BERTurk prioritize precision at the expense of coverage.

Comparison of unsupervised methods

This section evaluates three clustering approaches (k-means with pca, hierarchical clustering with pca, and topic modeling) for sentiment analysis using stratified cross-validation, where TF-IDF/PCA-processed text features are grouped into sentiment clusters and rigorously compared against true emotions through comprehensive performance metrics. For each fold, the user comments are processed through a TF-IDF transformation followed by PCA dimensionality reduction to extract meaningful features. These features are clustered using k-means into two groups, with each cluster automatically labeled as "positive" or "negative" based on the predominant emotion in the training data. The model's performance is evaluated on test data by comparing predicted clusters against actual emotions, calculating various metrics including accuracy, precision, recall, F1-score, sensitivity, specificity, Youden's index, and Cohen's kappa. The cross-validation results given in Table 5 are aggregated to provide mean performance metrics with standard deviations, offering a robust assessment of the clustering approach's effectiveness in distinguishing sentiment categories.

The metrics of hierarchical clustering approach show model performance variability during development, demonstrating moderate consistency (accuracy 0.889) but lower recall (0.706) compared to k-means (accuracy 0.910), which achieved better balance between sensitivity (0.814) and specificity (0.934). LDA performed substantially worse

Table 5. Cross validation results for unsupervised methods

Metric	Hierarchical	k-means
Accuracy	0.889 \pm 0.03	0.910 \pm 0.02
F1 Score	0.713 \pm 0.09	0.783 \pm 0.03
Recall	0.706 \pm 0.20	0.814 \pm 0.07
Precision	0.785 \pm 0.15	0.765 \pm 0.09
Specificity	0.935 \pm 0.07	0.934 \pm 0.03
Kappa	0.648 \pm 0.10	0.727 \pm 0.04
Youden Index	0.642 \pm 0.15	0.747 \pm 0.05

Metric	LDA
Accuracy	0.393 \pm 0.04
F1 Score	0.200 \pm 0.05
Recall	0.380 \pm 0.10
Precision	0.136 \pm 0.04
Specificity	0.396 \pm 0.02
Kappa	-0.134 \pm 0.07
Youden Index	-0.224 \pm 0.12

across all metrics (accuracy 0.393 \pm 0.038), with negative recall values indicating systematic misclassification. The Youden index reveals k-means (0.747) outperforms hierarchical clustering (0.642) in overall discriminative power, while both significantly exceed LDA's near-zero performance.

Final evaluation on unseen data given in Table 6 confirms k-means as the strongest performer (accuracy 0.946, F1 0.849), maintaining robust sensitivity-specificity balance (both 0.865). Hierarchical clustering shows slightly degraded but still reasonable real-world performance (accuracy 0.929, F1 0.800), while LDA's test metrics (accuracy 0.448) expose fundamental inadequacies despite its slightly improved recall (0.617) versus cross-validation. The Youden index gap widens substantially between k-means (0.829) and other methods, demonstrating its superior generalizability to new data.

Table 6. Performance metrics on held-out test set evaluation for unsupervised model predictions

Metric	Hierarchical	k-means	LDA
Accuracy	0.929	0.946	0.448
F1 Score	0.800	0.849	0.311
Recall	0.808	0.865	0.617
Precision	0.792	0.833	0.208
Specificity	0.955	0.865	0.405
Kappa	0.757	0.816	0.013
Youden Index	0.763	0.829	0.022

The visualization in Figure 1 reveals two distinct clusters of hierarchical clustering with pca emerging from the linguistic features of reviews with 5 stars from the held-out test set, projected into two-dimensional space using PCA. The clusters naturally separate reviews by emotional tone, with one grouping containing predominantly negative feedback despite high star ratings. The reviews shown are not manually selected but represent all held-out examples that were correctly predicted and contained flagged linguistic phrases associated with visibility bias. Their English translations reveal consistent instances of "rating inflation,"

where customers assign 5-star ratings while describing serious product flaws — a tactic often used to boost review visibility.

despite critical feedback to boost visibility. The spatial separation of clusters corresponds to measurable sentiment differences, validating that this approach captures emotional signals. Faint background points represent all reviews in that test set, while bold, colored points highlight those containing specific phrases.

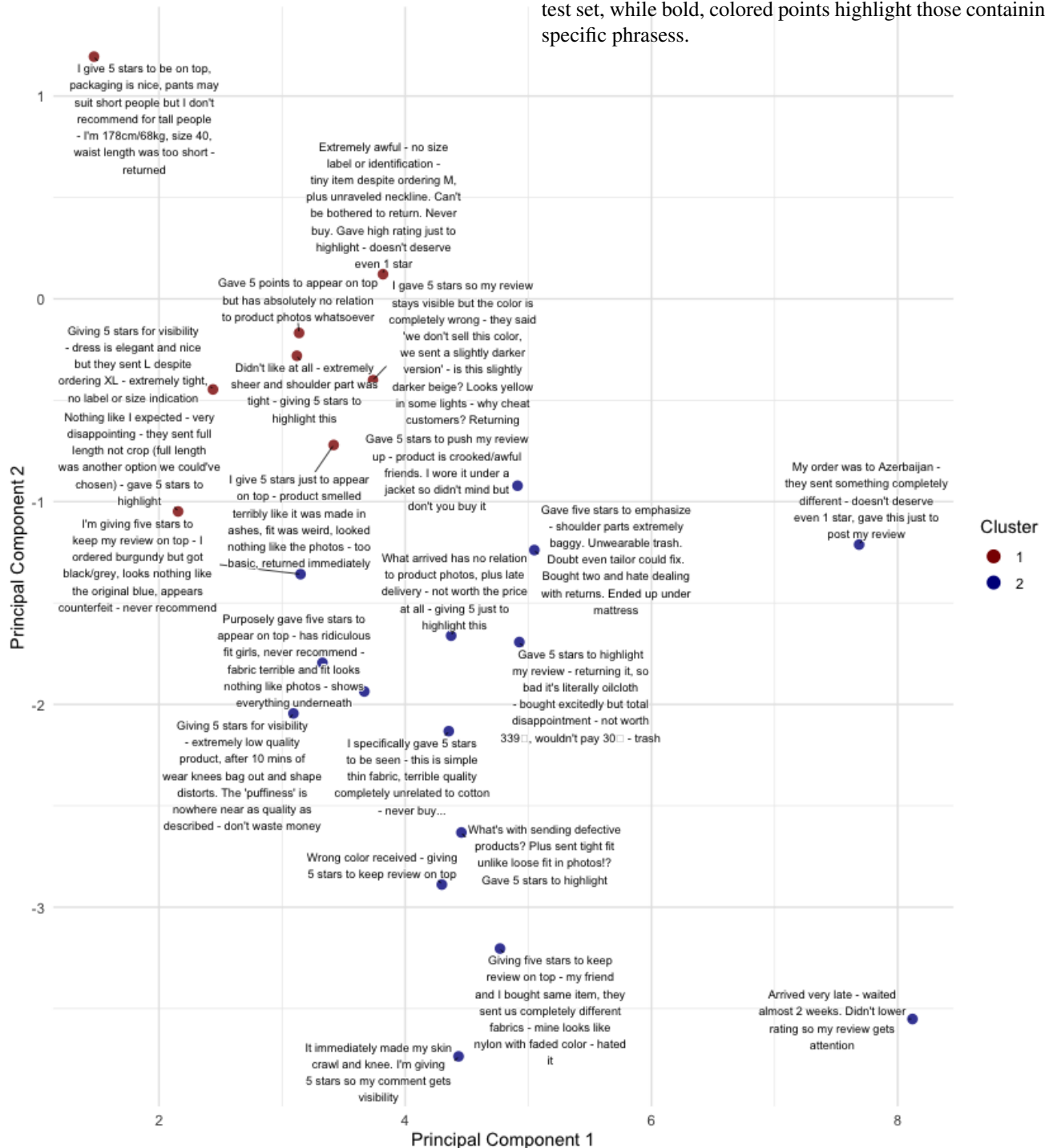


Figure 1. Hierarchical clustering with PCA on held-out data set

The Figure 2 visualization demonstrates how k-means clustering with PCA organizes customer reviews from the heldout test set into meaningful groups based on linguistic patterns, while revealing the striking disconnect between star ratings and actual sentiment. The plot shows two distinct clusters emerging from the PCA-reduced feature space, with representative English translations highlighting how negative reviews frequently appear in 5-star ratings—clear evidence of systematic rating inflation where users give high scores

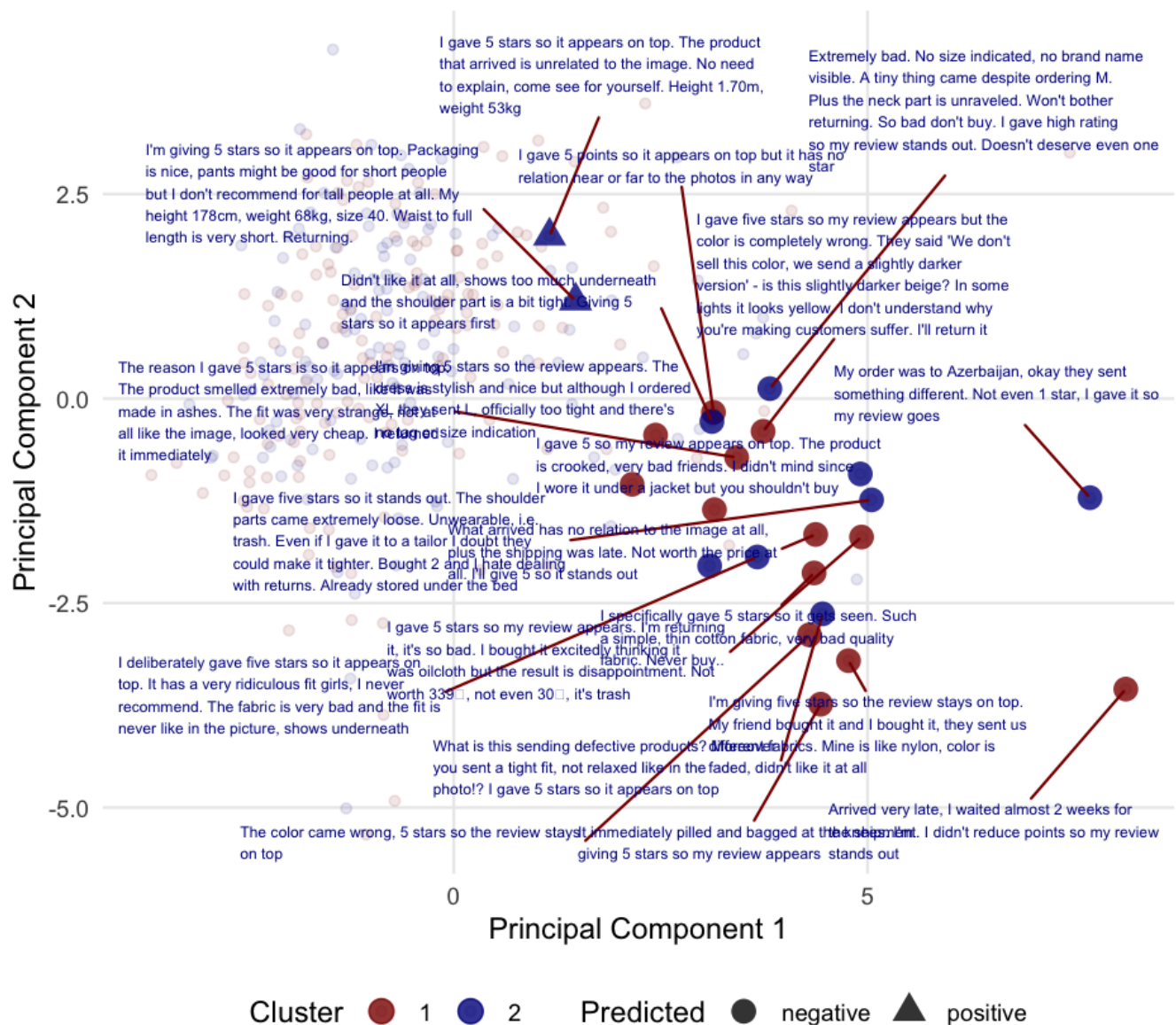


Figure 2. K-means clustering with pca on held-out data set

Generalization to Unlabeled E-Commerce Reviews

While our supervised models achieved strong performance on labeled data (recall = 0.867), applying them to 260,308 unlabeled reviews revealed critical real-world insights: 2.4% of 5-star reviews were predicted as negative—exposing systematic rating-text mismatches where users awarded top ratings despite negative language (e.g., '5 stars to get attention) as shown in Figure 3. This discrepancy, quantifiable only through large-scale deployment, highlights both the model's ability to detect deceptive patterns and the limitations of relying solely on star ratings in e-commerce platforms.

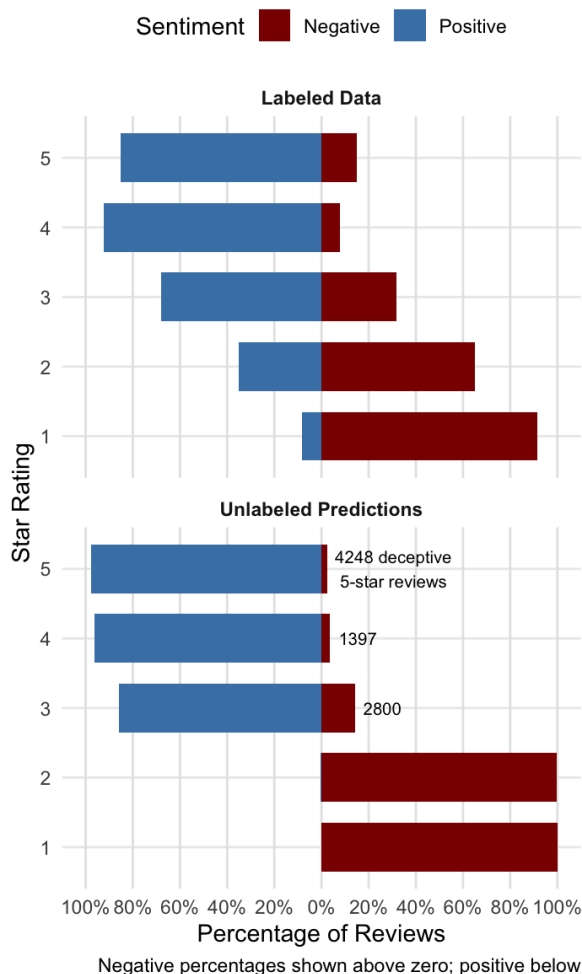


Figure 3. Rating-Sentiment Discrepancies: Labeled vs. Predicted

Conclusion

This paper presents a comprehensive framework for detecting sentiment bias in Turkish e-commerce reviews, where high star ratings sometimes conceal negative user experiences. Through systematic evaluation of supervised models, corpora-based techniques, and unsupervised clustering applied to a novel annotated dataset, we demonstrate that hybrid strategies combining textual features with rating metadata yield the most accurate sentiment interpretations.

Our research offers three core contributions to Turkish natural language processing. First, we present a comprehensive sentiment analysis pipeline tailored to Turkish-language e-commerce reviews—encompassing data acquisition, modeling, and practical implementation. Second, we introduce shoppingwords, an R package featuring four Turkish datasets, one of which contains 1,481 manually annotated reviews. Third, we demonstrate that RF is the optimal classifier among tested methods—achieving the best predictive performance across multiple evaluation strategies. Among all evaluated approaches, RF emerges as the optimal choice for emotion prediction across 260,308 user comments, consistently achieving high accuracy and recall in held-out tests. While XGBoost shows comparable performance, RF demonstrates greater scalability (extremely quicker) and robustness,

particularly in handling the negative expressions of emotional text. The performance kmeans clustering with pca is remarkably close to RF, which is a testament to how well the data clusters. However, RF still edges it out in key metrics (Recall, F1).

The clustering results offer particularly compelling evidence of our framework’s discovery potential. Visualization of PCA-projected clusters clearly separates negative expressions from positive language, even in high-rated reviews. While supervised methods ultimately achieve superior classification metrics, this unsupervised capability provides a crucial first analytical step for real-world scenarios requiring rapid, label-free assessment.

This work advances sentiment analysis methodology while challenging default assumptions about rating-sentiment alignment. By integrating data and practical tooling into a unified framework, robust solutions for detecting and mitigating sentiment bias in global e-commerce environments can create less inaccurate results. Future directions include developing semi-supervised extensions to better leverage both annotated and unannotated data, and adapting the framework for other linguistically underrepresented markets.

Acknowledgements

This work was financially supported by the Tubitak-BİDEB-2219 International Postdoctoral Research Scholarship Programme, and additional support, including server resources, was provided by the Department of Statistical Science at Duke University in the U.S., which greatly facilitated this research. I would also like to express my sincere gratitude to Prof. Dr. Mine Çetinkaya-Rundel for her valuable comments, insightful suggestions, and generous guidance throughout the development of this paper.

References

- Tripadvisor (2022) *Travelers Push Tripadvisor Past 1-Billion Reviews and Opinions*. Retrieved from <https://tripadvisor.mediaroom.com/2022-02-01-Travelers-Push-Tripadvisor-Past-1-Billion-Reviews-Opinions>
- Hennig-Thurau T, Wiertz C and Feldhaus F (2008) *The Impact of Online Reviews on Film Marketing*. Journal of Marketing, 72, 74–89.
- Saura JR (2018) *Big Data for Big Impact: E-Commerce Analytics*. Journal of Business Research, 88, 169–178.
- Chevalier JA and Mayzlin D (2006) *The Effect of Word of Mouth on Sales: Online Book Reviews*. Journal of Marketing Research, 43(3), 345–354.
- Hofstede G (2010) *Cultures and Organizations: Software of the Mind*. McGraw-Hill.
- Zhu E (2024) *BERTopic-Driven Stock Market Predictions: Unraveling Sentiment Insights*. arXiv. Available at: <https://arxiv.org/html/2404.02053v2>
- Statista (2025) *E-commerce in Turkey – Statistics & Facts*. Available at: <https://www.statista.com/topics/9411/e-commerce-in-turkey>
- Tiedemann J and Thottingal S (2020) *OPUS-MT: Building Open Translation Services for the World*. In Proceedings of the 22nd Annual Conference of the European Association for Machine

- Translation, Lisboa, Portugal, 479–480. Available at: <https://aclanthology.org/2020.eamt-1.61>
- Tiedemann J (2020) *The Tatoeba Translation Challenge: Realistic Data Sets for Low-Resource and Multilingual MT*. In Proceedings of the 5th Conference on Machine Translation, 1174–1182. Available at: <https://aclanthology.org/2020.wmt-1.139>
- Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł and Polosukhin I (2017) *Attention Is All You Need*. arXiv:1706.03762. Available at: <https://arxiv.org/abs/1706.03762>
- Devlin J, Chang M-W, Lee K and Toutanova K (2018) *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. Available at: <https://arxiv.org/abs/1810.04805>
- Belaroussi R, Noufe SC, Dupin F and Vandanjon P-O (2025) *Polarity of Yelp Reviews: A BERT-LSTM Comparative Study*. Big Data Cogn. Comput., 9(5), 140. Available at: <https://www.mdpi.com/2504-2289/9/5/140>
- Sanh V, Wolf T and Debut L (2019) *DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper, and Lighter*. Hugging Face. Available at: <https://arxiv.org/abs/1910.01108>
- Reddy YA, Agarwal S, Parashar V and Arora A (2025) *Real-Time Sentiment Insights from X Using VADER, DistilBERT, and Web-Scraped Data*. arXiv:2504.15448. Available at: <https://arxiv.org/abs/2504.15448>
- Gao H (2021) *Improved Sentiment Analysis Using a Customized DistilBERT NLP Configuration*. Advances in Engineering: An International Journal (ADEIJ), 3(2). Available at: https://www.academia.edu/56705388/Improved_Sentiment_Analysis_using_a_Customized_Distilbert_NLP_Configuration
- Ooms J (2025) *hunspell: High-Performance Stemmer, Tokenizer, and Spell Checker*. R package version 3.0.7. Available at: <https://docs.ropensci.org/hunspell/>
- Ye Q, Zhang Z and Law R (2009) *Sentiment Classification of Online Reviews to Travel Destinations by Supervised Machine Learning Approaches*. Expert Systems with Applications, 36(3 PART 2), 6527–6535. Available at: <https://doi.org/10.1016/j.eswa.2008.09.011>
- Labille K, Alfarhood S, Gauch S and Özgür A (2016) *Estimating Sentiment via Probability and Information Theory*. In Proceedings of the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, 121–129. Available at: <https://doi.org/10.5220/0006072101210129>
- Eryigit G (2014) *ITU Turkish NLP Web Service*. In Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics, 1–4. Gothenburg, Sweden: Association for Computational Linguistics. Available at: <https://doi.org/10.3115/v1/E14-2001>
- Kaya M, Fidan G and Toroslu İH (2012) *Sentiment Analysis of Turkish Political News*. In Proceedings of the IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT), 174–180. Macau: IEEE. Available at: <https://doi.org/10.1109/WI-IAT.2012.179>
- Perrie J, Islam A, Milios E and Keselj V (2013) *Using Google n-Grams to Expand Word-Emotion Association Lexicon*. In Gelbukh A (editor) *Computational Linguistics and Intelligent Text Processing*, 137–148. Springer Berlin Heidelberg. Available at: https://doi.org/10.1007/978-3-642-37256-8_12
- Baccianella S, Esuli A and Sebastiani F (2010) *SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining*. In Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC'10), Valletta, Malta. European Language Resources Association (ELRA). Available at: http://www.lrec-conf.org/proceedings/lrec2010/pdf/769_Paper.pdf
- Ucan A, Naderalvojud B, Akcinar-Sezer E and Sever H (2016) *SentiWordNet for New Language: Automatic Translation Approach*. In 2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), 308–315. Available at: <https://doi.org/10.1109/SITIS.2016.57>
- Dehkharghani R, Saygin Y, Yanikoglu B and Oflazer K (2016) *SentiTurkNet: A Turkish Polarity Lexicon for Sentiment Analysis*. Language Resources and Evaluation, 50(3), 667–685. Available at: <https://doi.org/10.1007/s10579-015-9315-6>
- Demirtaş E and Peçenak Z (2020) *SentiTurkNet: A Turkish Polarity Lexicon for Sentiment Analysis*. In Proceedings of the 28th Signal Processing and Communications Applications Conference (SIU), IEEE.
- Labille K, Gauch S and Alfarhood S (2017) *Creating Domain-Specific Sentiment Lexicons via Text Mining*. In Proceedings of the Workshop on Issues of Sentiment Discovery and Opinion Mining (WISDOM'17), Halifax, Canada, 1–8. ACM. Available at: <https://doi.org/10.1145/nnnnnnn.nnnnnnn>
- Kan-Kilinc B, Çetinkaya-Rundel M and Rundel C (2025) *shoppingwords: Text Processing Tools for Turkish E-Commerce Data*. R package version 0.1.0. Available at: <https://cran.rstudio.com/web/packages/shoppingwords>
- Ooms J (2022) *jsonlite: A Simple and Robust JSON Parser and Generator for R*. R package version 1.8.3. Available at: <https://CRAN.R-project.org/package=jsonlite>
- Wickham H (2024) *rvest: Easily Harvest (Scrape) Web Pages*. R package version 1.0.4. Available at: <https://CRAN.R-project.org/package=rvest>
- Aden-Buie G and Schloerke B (2025) *chromote: Headless Chrome Interface*. R package version 0.2.0. Available at: <https://CRAN.R-project.org/package=chromote>
- Wickham H, Averick M, Bryan J, Chang W, D'Agostino McGowan L, François R, Golemund G, Hayes A, Henry L, Hester J, Kuhn M, Lin Pedersen T, Miller E, Milton Bache S, Müller K, Ooms J, Robinson D, Seidel DP, Spinu V, Takahashi K, Vaughan D, Wilke C, Woo K and Yutani H (2019) *Welcome to the Tidyverse*. Journal of Open Source Software, 4(43), 1686. Available at: <https://doi.org/10.21105/joss.01686>
- Wickham H (2019) *Advanced R*. Chapman & Hall/CRC.
- Wickham H and Henry L (2023) *purrr: Functional Programming Tools*. R package version 1.0.1. Available at: <https://CRAN.R-project.org/package=purrr>
- Wickham H (2022) *httr: Tools for Working with URLs and HTTP*. R package version 1.4.4. Available at: <https://CRAN.R-project.org/package=httr>

- Ahmet A (2023) *Zemberek NLP: Turkish Natural Language Processing Library*. GitHub repository. Python version. Available at: <https://github.com/ahmetaa/zemberek-nlp>
- Hugging Face (2024) *Tokenizers - Hugging Face Documentation*. Available at: <https://huggingface.co/docs/tokenizers/index>
- Yıldırım S (2024) *Fine-tuning Transformer-based Encoder for Turkish Language Understanding Tasks*. arXiv preprint arXiv:2401.17396. Available at: <https://arxiv.org/abs/2401.17396>
- Schweter S (2020) *BERTurk - BERT Models for Turkish*. GitHub repository. Available at: <https://github.com/stefan-it/turkish-bert>
- Ushey K, Allaire J and Tang Y (2025) *reticulate: Interface to Python*. R package version 1.42.0. Available at: <https://rstudio.github.io/reticulate/>
- Aydoğan M and Kocaman V (2022) *TRSAv1: A New Benchmark Dataset for Classifying User Reviews on Turkish E-commerce Websites*. Journal of Information Science. Available at: https://www.researchgate.net/publication/358582069_TRSAv1_A_new_benchmark_dataset_for_classifying_user_reviews_on_Turkish_e-commerce_websites
- Akbaş MN and Taşkın T (2024) *Turkish Sentiment Analysis: A Comprehensive Review*. Sigma Journal of Engineering and Natural Sciences, 42(4), 1292–1314. Available at: <https://doi.org/10.14744/sigma.2024.00033>