AAUP JOURNAL OF STEM AND HEALTH SCIENCES (JSH-AAUP)

AAUP Journal of STEM and Health Sciences (JSH-AAUP)

Integrating Sentiment Analysis and Topic Modeling for Actionable Business Intelligence: A Comparative Study of Machine Learning and Deep Learning Architectures for Arabic Company Reviews

Mohammed Maree[1,*], Saadat M. Alhashmi[2], Mohammed Belkhatir[3]

[1]Faculty of Information Technology, Arab American University, Jenin, Palestine

[2]Department of Information Systems, College of Computing and Informatics, University of Sharjah, UAE

[3] Department of Computer Science, Institute of Technology, University of Lyon I, Lyon, France

* Corresponding Author: Mohammad Maree, email: mohammed.maree@aaup.edu

ORCID ID: https://orcid.org/0000-0002-6114-4687

Date Received: 05/1/2026 Date Revised: 05/3/2026

Date Accepted: 07/3/2026 Date Published: 31/03/2026 DOI: 10.35517/JSH-AAUP.v1.02

Abstract

Background: This study presented a comprehensive framework for Arabic Sentiment Analysis (ASA) integrated with Topic Modeling to support actionable Business Intelligence (BI) from company reviews. **Methods:** Using 5-fold stratified cross-validation, the researcher compared traditional Machine Learning (ML) classifiers (Logistic Regression, Support Vector Machine, Naive Bayes, and Random Forest) with Deep Learning (DL) architectures (Bidirectional LSTM and Bidirectional GRU) and a Transformer-based model (AraBERT). **Results:** The experiments demonstrated that no single model dominated across all evaluation metrics. Logistic Regression achieved the highest Macro F1-score (0.6212 ± 0.0036; 95% CI: 0.6180–0.6243), indicating the most balanced performance under class imbalance. Naive Bayes obtained high accuracy (0.8350 ± 0.0029; 95% CI: 0.8324–0.8376) and weighted F1-score (0.8140 ± 0.0029; 95% CI: 0.8115–0.8165), reflecting strong performance on the dominant sentiment class. **Conclusion**: Deep learning models achieved competitive accuracy (Bi-LSTM: 0.8386 ± 0.0020; 95% CI: 0.8369–0.8404) but did not surpass linear models in Macro F1. AraBERT outperformed all

models with accuracy (0.8598 ± 0.0044; 95% CI: 0.8560–0.8636) and Macro F1 (0.6382 ± 0.0100; 95% CI: 0.6294–0.6469). Topic Modeling using Latent Dirichlet Allocation (LDA) complemented sentiment classification by extracting interpretable strengths and weaknesses for each company. Precision-Recall curves across models confirmed the impact of class imbalance, with high Average Precision (AP) for the positive class (~0.92–0.96) and very low AP for neutral (~0.09–0.20). These findings supported actionable BI insights from Arabic company reviews.

Keywords: Arabic NLP, Sentiment Analysis, Bi-LSTM, Support Vector Machines, Topic Modeling, Business Intelligence, Class Imbalance

1. Introduction

The rapid proliferation of digital platforms and e-commerce across the Middle East and North Africa (MENA) has transformed how consumers interact with brands. In the contemporary digital economy, customer feedback – expressed through reviews, tweets, and comments – has become a rich source, providing raw, unstructured data that contains the key to understanding consumer behavior and market trends [1]. However, the sheer volume of this data makes manual moderation and analysis impossible. Consequently, Automated Sentiment Analysis (ASA) has emerged as a vital tool for Business Intelligence (BI), allowing organizations to monitor their brand reputation in real-time [2]. Arabic is the fifth most spoken language globally, with over 400 million speakers. Despite its global significance, Arabic Natural Language Processing (NLP) has historically lagged behind English, primarily due to the unique morphological and orthographic characteristics of the language. While English sentiment analysis is often a matter of identifying polarity words, Arabic sentiment is deeply embedded in complex grammatical structures and cultural nuances. For businesses operating in Arab markets, the ability to accurately "sense" the emotion behind a review is not just a technical task but a strategic necessity for survival in a competitive landscape. The challenges of Arabic NLP can be categorized into three distinct layers: morphological, orthographic, and dialectal.

Morphological Complexity: Arabic is a highly inflected, derivational language based on a root-and-pattern system. A single word can contain a root, prefixes, suffixes, and even infixes that change its meaning from a verb to a noun or a possessive statement. For instance, the word "وسيعجبهم" (and they will like it) contains a conjunction, a future tense marker, a root verb, and a plural pronoun. For a Machine Learning model, these "clitics" increase the size of the vocabulary exponentially, leading to data sparsity issues where the model sees many variations of the same root but fails to connect them.

Orthographic Variation and Noise: In informal settings, such as company reviews on Google Maps or Facebook, Arabic speakers rarely use diacritics (Tashkeel). Furthermore, they often use character elongation (Tatweel) for emphasis (e.g., "ممممتاز" instead of "ممتاز") or interchange different forms of the letter Alef (ا, أ, إ, آ). Without a robust preprocessing layer, such as the one implemented in this study using pyarabic, these variations create "noise" that significantly degrades model accuracy.

Dialectal Diversity: While Modern Standard Arabic (MSA) is used in formal writing, reviews are frequently written in various dialects (Egyptian, Levantine, Gulf, etc.). These dialects often utilize different negations and vocabulary. A "Positive" word in one dialect might be "Neutral" in another, requiring models that can capture contextual dependencies rather than relying on static sentiment lexicons.

For over a decade, traditional Machine Learning (ML) served as the backbone of ASA. Models such as Support Vector Machines (SVM) and Naive Bayes, paired with TF-IDF (Term Frequency-Inverse Document Frequency)

vectorization, were the gold standard. These models are mathematically robust and computationally efficient. As demonstrated in our implementation, SVM remains a powerful contender because it handles high-dimensional sparse data – typical of text – extremely well. However, traditional ML has a major limitation: it is "bag-of-words" oriented. It treats words as independent entities and often fails to capture the long-range dependencies and the "flow" of a sentence. The advent of Deep Learning (DL) and Recurrent Neural Networks (RNNs) marked a paradigm shift. Unlike traditional classifiers, architectures like the Bidirectional Long Short-Term Memory (Bi-LSTM) and Bidirectional Gated Recurrent Unit (Bi-GRU) process text as a sequence. By reading a review both from left-to-right and right-to-left, these models can understand the context of a word based on what preceded it and what followed it. This is particularly crucial for Arabic, where the placement of a negation word can occur far before the adjective it modifies.

While classification accuracy is a primary metric for researchers, it is often insufficient for business stakeholders. A company does not only need to know that a review is "Negative"; it needs to know why. Is the negativity directed at the customer service, the price, or the product quality? This research bridges the gap between raw classification and actionable insight. By integrating Latent Dirichlet Allocation (LDA) for topic modeling, the researcher has moved beyond polarity. The system automatically extracts the "strengths" and "weaknesses" of a brand by identifying the most frequent themes within positive and negative feedback subsets. This transforms a simple NLP task into a Comprehensive Business Intelligence Report, enabling decision-makers to identify specific operational areas that require improvement. The primary objective of this study was to provide a comprehensive comparative analysis of various computational architectures for ASA. Specifically, the researcher aimed to:

evaluate traditional ML models (SVM, Random Forest, Naive Bayes, Logistic Regression) against Deep Learning models (Bi-LSTM, Bi-GRU) and a Transformer-based model (AraBERT) using the "Arabic Company Reviews" dataset.

develop a standardized preprocessing pipeline that addresses the specific needs of the Arabic language, including normalization and noise reduction.

propose a Business Intelligence framework that uses unsupervised learning (LDA) to provide granular insights into brand perception.

visualize performance trade-offs using Precision-Recall curves and learning curves to determine the most cost-effective architecture for industrial deployment.

employ 5-fold stratified cross-validation to ensure robust and reproducible performance estimation, addressing limitations of single train/test splits and severe class imbalance in the exploited dataset.

By addressing these objectives, this research contributed and provided a scalable and interpretable framework for Arabic-speaking markets, providing both high-accuracy sentiment detection and deep-dive thematic analysis.

2. Related Work

The field of Arabic Sentiment Analysis (ASA) has evolved through distinct computational eras: from early rule-based systems and manual lexicons to the rise of statistical Machine Learning (ML), and finally to the current dominance of Deep Learning (DL) and Transformer-based architectures.

2.1 Theoretical Foundations of Arabic Sentiment Analysis

Early research in ASA was primarily constrained by the lack of large-scale, annotated datasets. Early pioneers like Abbasi et al. [8] focused on "Affect Analysis," which combined stylistic and structural features to identify sentiment in extremist forums. Their work laid the groundwork for feature engineering in Arabic, proving that language-specific preprocessing—such as handling the "root-and-pattern" morphology—was essential for reducing feature sparsity.

Unlike English, where sentiment is often explicitly carried by adjectives, Arabic sentiment is frequently embedded in verbal morphology. For instance, the research by Farra et al. [9] highlighted how sentence-level sentiment in Arabic is heavily influenced by the "Mood" (I'rab) and the specific grammatical constructs used in dialectal vs. Modern Standard Arabic (MSA). They noted that simple bag-of-words models often miss the nuances of negation and intensification in complex Arabic sentences.

2.2 Traditional Machine Learning: The Statistical Era

The mid-2010s saw a surge in the application of supervised learning for ASA. Support Vector Machines (SVM) and Naïve Bayes (NB) became the standard benchmarks for the industry [4].

Support Vector Machines (SVM): El-Halees [10] proposed a multi-stage approach using SVM, which remains a highly relevant methodology. His research demonstrated that SVMs, when paired with TF-IDF vectorization and N-gram features (unigrams and bigrams), could achieve accuracies exceeding 80% on news datasets. Our implementation leverages this historical success by utilizing the LinearSVC architecture, which excels in high-dimensional text spaces where the number of features often exceeds the number of samples.

Random Forests and Ensembles: Al-Kabi et al. [3] conducted a comprehensive survey of 14 different classifiers on a large dataset of Arabic reviews. Their findings suggested that while Naïve Bayes is computationally efficient, Ensemble methods like Random Forest provide better robustness against the "noise" of dialectal Arabic. This informed our decision to include Random Forest in our comparative pipeline to evaluate its performance against high-variance user reviews.

2.3 The Deep Learning Revolution: RNNs, LSTM, and GRU

The shift toward Deep Learning (DL) addressed the fundamental weakness of ML: the inability to capture sequential context [7]. Traditional ML treats "ليس جيد" (not good) as two independent tokens, whereas DL architectures maintain a "memory" of previous tokens.

Bi-LSTM Architectures: Research by Al-Smadi et al. [11] on Aspect-Based Sentiment Analysis (ABSA) for Arabic hotels and restaurants revealed that Bidirectional Long Short-Term Memory (Bi-LSTM) networks significantly outperform traditional models by capturing dependencies in both directions. This is particularly crucial for Arabic, where the subject often follows the verb, and negation markers can appear several words before the target adjective. Our study builds on this by implementing a 64-unit Bi-LSTM layer to handle these long-range semantic dependencies.

Gated Recurrent Units (GRU): While LSTMs are powerful, Gated Recurrent Units (GRU) have emerged as a computationally efficient alternative with fewer parameters and comparable representational capacity. Dahou et al. [12] demonstrated the effectiveness of distributed Word2Vec embeddings combined with deep neural architectures for Arabic sentiment classification, highlighting the benefits of neural sequence modeling over

traditional machine learning approaches. In parallel, Cho et al. [17], who originally introduced the GRU architecture, showed that GRUs achieve faster convergence and competitive performance compared to LSTMs due to their simplified gating mechanism. Motivated by these findings, our comparative framework explicitly evaluates both Bi-LSTM and Bi-GRU architectures to identify the optimal accuracy-to-computational-cost trade-off for industrial Arabic sentiment analysis applications.

2.4 Word Embeddings and Vector Space Models

The transition from one-hot encoding (TF-IDF) to dense vector representations revolutionized ASA. Early models relied on static embeddings like Word2Vec and GloVe.

Soliman et al. [13] introduced AraVec, a pre-trained distributed word representation specifically for Arabic. Their research demonstrated that utilizing pre-trained embeddings on large corpora of tweets and news articles allowed models to understand semantic similarity (e.g., understanding that "رائع" (wonderful) and "جميل" (beautiful) share a similar sentiment space). However, our current study focuses on training domain-specific embeddings to capture the unique vocabulary of corporate reviews, which often differs from social media slang.

2.5 The Role of Preprocessing in Sentiment Accuracy

A recurring theme in the literature is that the "quality of cleaning" is often more important than the "complexity of the model."

Normalization and Stemming: The work of Duwairi et al. [14] emphasized that without normalizing the "Hamza" and "Yaa" and stripping "Tashkeel" (diacritics), the vocabulary size becomes unmanageable. They argued that light stemming (removing common prefixes and suffixes) is superior to root stemming for sentiment analysis because root stemming can conflate words with opposite polarities (e.g., "مقبول" - acceptable and "قبل" - before). This research justifies our use of the pyarabic and tashaphyne libraries in the preprocessing.py module.

Handling Sarcasm and Dialects: Sarcasm remains one of the "Grand Challenges" of ASA. Abu Farha and Magdy [15] noted that Arabic sarcasm is often dialect-specific. They developed the ArSarcasm dataset, highlighting that even the most advanced DL models struggle with figurative language. While our current research focuses on polarity, their work underscores the need for the future integration of Transformer models like AraBERT.

2.6 Sentiment Analysis for Business Intelligence (BI)

In the corporate sector, sentiment analysis is often the first step toward Topic Modeling. Latent Dirichlet Allocation (LDA) has been widely used to identify the "aspects" of a business that customers are discussing.

Blei et al. [5] explored the integration of sentiment scores with LDA to create "Brand Perceptual Maps." Their work showed that by filtering reviews into positive and negative buckets before applying LDA, companies could identify specific "pain points" (e.g., slow delivery) vs. "competitive advantages" (e.g., friendly staff). Our insights.py module implements this specific theoretical framework by generating automated "Strength/Weakness" reports for brands in the MENA region.

2.7 Recent Trends: Transformers and Large Language Models (LLMs)

The current state-of-the-art involves Transformer architectures, specifically AraBERT. Antoun et al. [16] demonstrated that AraBERT, through its self-attention mechanism, achieves unprecedented results on ASA benchmarks. While this study focuses on the comparative analysis of lighter ML/DL architectures suitable for edge deployment or resource-constrained environments, the literature clearly points toward a hybrid future where Transformers provide the "gold standard" and RNNs provide "efficiency."

Despite the wealth of research, two significant gaps remain:

Industrial Scalability: Most research focuses on academic datasets (like LABR or ArSarcasm) rather than real-world corporate review streams which contain a mix of formal and highly informal text.

Actionable Intelligence: Few studies bridged the gap between "Classification Report" and "Business Report."

This research contributed to the field by providing a head-to-head comparison of five distinct architectures (SVM, RF, NB, Bi-LSTM, Bi-GRU) specifically on the Fahd Seddik company reviews dataset [6], paired with an unsupervised insight generation layer that translates computational metrics into managerial value. In addition, the researcher included AraBERT as a Transformer baseline to evaluate its performance against traditional ML and DL models using 5-fold stratified cross-validation, addressing limitations of single-split experiments and providing more reliable performance estimates under class imbalance.

3. Methodology

The methodology of this study followed a structured computational pipeline designed to move from raw, unstructured Arabic text to high-level business intelligence. The process was divided into four major phases: Data Preprocessing, Feature Engineering, Model Implementation (Machine Learning and Deep Learning), and Insight Extraction. In Figure 1, the researcher provided a high-level architecture of the proposed methodology, including the employed models and 5-fold stratified cross-validation loop.

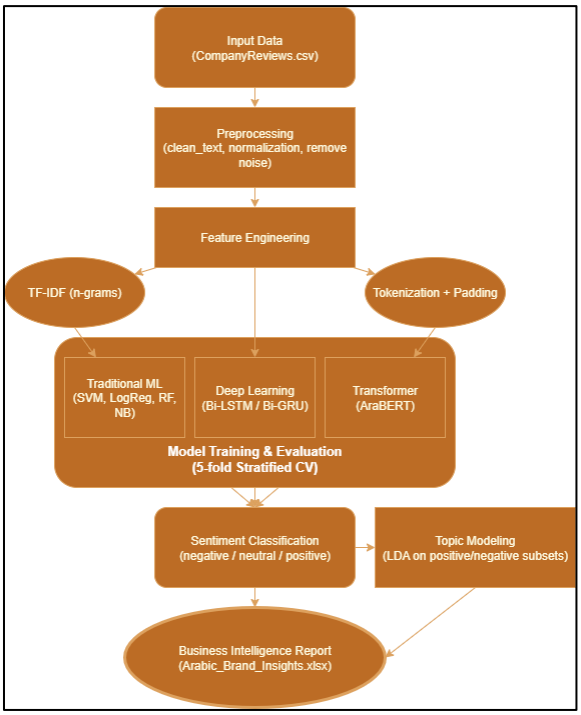

Figure 1. High-level Architecture of the Employed Methodology

3.1 Data Preprocessing and Normalization

Given the morphological richness of Arabic, the preprocessing layer (implemented in preprocessing.py) is critical. The researcher defined the cleaning function $\mathcal{C}(R)$ such that for any raw review R, the cleaned output R' is:

$$\mathcal{C}(R) = Normalize\Big(Strip\big(Filter(R)\big)\Big) \ldots\ldots (1)$$

Algorithmic Steps for Preprocessing:

1. Noise Reduction: Removal of non-Arabic characters, URLs, and punctuation using Regular Expressions.
2. Orthographic Normalization: * Standardizing Alef variants (أ, إ, آ) to a bare Alef (ا).
    - Standardizing Yaa (ي) and Alef Maksura (ى).
3. Tashkeel and Tatweel Removal: Using the pyarabic library to strip diacritics and character elongations (e.g., "جميـل" becomes "جميل").
4. Token Normalization: Reducing character repetitions (e.g., "رااائع" to "رائع") to prevent vocabulary explosion.

3.2 Feature Engineering: Vectorization and Embeddings

The researcher employed two distinct types of mathematical representations for the text data depending on the model architecture.

3.2.1 TF-IDF Vectorization (for ML Models)

For traditional models, the researcher used the Term Frequency-Inverse Document Frequency (TF-IDF) weight. The weight W of a term t in document d is calculated as:

$$W_{t,d} = tf_{t,d} \times \log\left(\frac{N}{df_t}\right) \ldots\ldots (2)$$

Where N is the total number of reviews and $\{df\}\_t$ is the number of reviews containing term t. We implement unigrams and bigrams (n=1, 2) to capture local word orderings.

3.2.2 Sequence Padding and Word Embeddings (for DL Models)

For Deep Learning, the text was converted into sequences of integers S = [w_1, w_2, ..., w_n]. To ensure uniform input for the neural network, the researcher applied zero-padding $\mathcal{P}(S)$ such that all sequences have length L=100:

$$\mathcal{P}(S) = [0, 0, \ldots, w_1, w_2, \ldots, w_n] \ldots\ldots (3)$$

3.3 Exploited Model Architectures

The researcher compared five distinct models, each representing a different mathematical approach to classification. Additionally, AraBERT was included as a Transformer-based baseline to evaluate modern pre-trained contextual embeddings against traditional ML and DL approaches.

3.3.1 Support Vector Machines (SVM)

The SVM seeks to find the optimal hyperplane that maximizes the margin between classes. In this implementation (LinearSVC), the researcher solved the following optimization problem:

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi \quad \ldots\ldots\ldots (4)$$

Subject to: y_i (w · x_i + b) ≥ 1 − ξ_i. The parameter C controls the trade-off between margin maximization and error minimization.

3.3.2 Naive Bayes (Multinomial)

Based on Bayes' Theorem, this model assumes conditional independence between features:

$$P(y \mid x\_1, \dots, x\_n) = (P(y) \prod\_(i = 1)\^n P(x\_i \mid y)) / P(x\_1, \dots, x\_n) \dots\dots\dots (5)$$

Despite the "naive" assumption, it is highly effective for high-dimensional Arabic text where term frequencies are the primary signals.

3.3.3 Random Forest (RF)

An ensemble learning method that constructs N decision trees during training. The final sentiment $\hat{y}$ is determined by majority voting:

$$\hat{y} = mode\{T_{1(x)}, T_{2(x)}, \dots, T_{N(x)}\} \dots\dots\dots (6)$$

3.3.4 Bidirectional LSTM (Bi-LSTM)

LSTMs use "gates" to manage the flow of information, solving the vanishing gradient problem. A cell at time t consists of a Forget Gate f_t, Input Gate (i_t), and Output Gate (o_t):

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \dots\dots\dots (7)$$

The Bidirectional component allows the network to have both forward ($\rightarrow h_t$) and backward ($\leftarrow h_t$) hidden states, concatenating them to capture full context:

$$H_t = [\rightarrow h_t; \leftarrow h_t] \dots\dots\dots (8)$$

The architecture of this model is depicted in Figure 2. To handle class imbalance, class weights (computed as inverse class frequency) were applied during training.

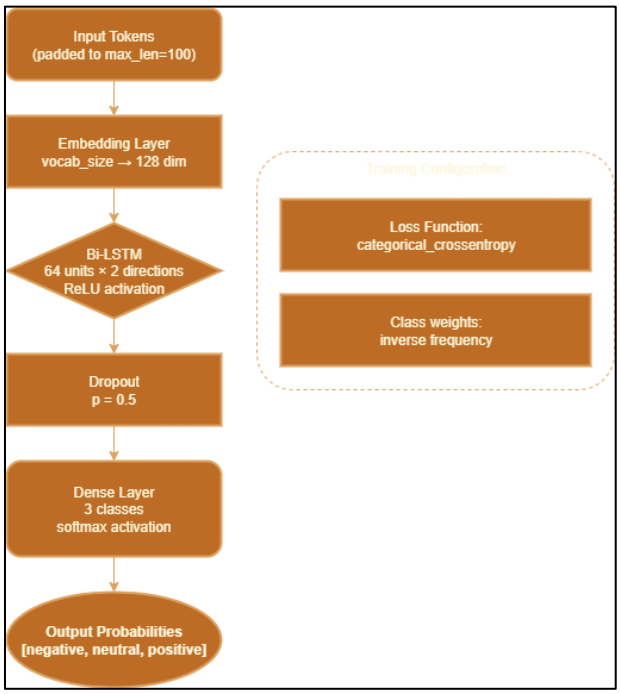

Figure 2. Architecture of the employed Bi-LSTM model

3.3.5 Bidirectional GRU (Bi-GRU)

The Gated Recurrent Unit (GRU) simplifies the LSTM by merging the forget and input gates into a single "Update Gate" (z_t) and adding a "Reset Gate" (r_t):

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \text{ ..... } (9)$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \text{ ..... } (10)$$

This reduces the parameter count, often leading to faster training times on smaller Arabic datasets without significant loss in accuracy.

3.3.6 Transformer Model (AraBERT)

AraBERT is a BERT-based model pre-trained on large Arabic corpora. The researcher fine-tuned the aubmindlab/bert-base-arabertv2 checkpoint using the Hugging Face Transformers library. The key settings included:

- Maximum sequence length: 128 tokens
- Batch size: 32
- Epochs per fold: 3
- Optimizer: AdamW (learning rate scheduler with warmup)
- Mixed precision (fp16) enabled with GPU.
- Evaluation strategy: per epoch
- No model checkpoint saving during training to reduce disk usage

AraBERT was trained using the Trainer API with stratified 5-fold cross-validation. Class imbalance was implicitly handled by the pre-trained contextual embeddings and the large capacity of the model.

3.4 Unsupervised Insight Generation (LDA)

The final stage of the methodology involved Latent Dirichlet Allocation (LDA) to extract themes from the classified reviews. LDA assumes each document d is a mixture of topics $theta\_d$ and each topic k is a distribution over words phi_k.

$$P(w \mid d) = \sum\_(k = 1)\^K P(w \mid z = k) P(z = k \mid d) \text{ ..... } (11)$$

By applying LDA separately to the "Positive" and "Negative" subsets for each company, the researcher generated a Business Intelligence Report that highlighted specific operational strengths and weaknesses.

3.5 Experimental Setup and Evaluation Metrics

The dataset was evaluated using 5-fold stratified cross-validation (StratifiedKFold, shuffle=True, random seed=42) to maintain class distribution across folds and ensure robust performance estimation. All models were trained and evaluated on the same folds for fair comparison. Metrics included:

1. Macro F1-Score: Primary metric to account for class imbalance and minority class performance.
2. Accuracy and Weighted F1-Score: To reflect overall and majority-class performance.
3. Precision-Recall Curves: Generated from the last fold to visualize model confidence and imbalance effects (Average Precision reported per class and micro-averaged).
4. Learning Curves (optional): To detect overfitting vs. underfitting in DL models.

Computational notes: ML models run on CPU (<1 min/fold), DL models run on GPU (RTX 4060, 1–3 min/fold), AraBERT runs on GPU (20–30 min/fold).

Algorithm 1: The Proposed ASA-BI Framework

Input: Raw Arabic Company Reviews CSV
Output: Classification Metrics & Brand Insights Report

1. Load dataset: → prepare_dataset()
2. For each review R: Apply $\mathcal{C}(R)$(Normalization & Cleaning)
3. Split data into 5 stratified folds (StratifiedKFold)
4. Train ML Pipeline:
   a. TF-IDF (1,2) n-grams
   b. Fit SVM, RF, NB, LogReg
5. Train DL Pipeline:
   a. Tokenize and Pad sequences
   b. Fit Bi-LSTM and Bi-GRU (20 epochs max, Adam optimizer, early stopping, class weights for imbalance
6. Train Transformer Pipeline:
   a. Tokenize with AraBERT tokenizer
   b. Fine-tune AraBERT (3 epochs/fold, batch_size=32, fp16 on GPU
7. **Evaluate:** Generate Confusion Matrices and PR Curves
8. Insight Mining: For each company, run LDA on {R_{pos}} and {R_{neg}}
9. **Export** Final_Business_Report.xlsx

4. Experimental Setup

Experiments were conducted on an Arabic company reviews dataset. The data were split using 5-fold stratified cross-validation to preserve class distribution and ensure robustness across folds. Evaluation metrics included Accuracy, Macro F1-score, and Weighted F1-score. Macro F1 was used as the primary metric due to its robustness under class imbalance.

4.1 Experimental Environment and Dataset Partitioning

The experiments were conducted using a Python-based pipeline leveraging Scikit-learn for traditional models and TensorFlow/Keras for deep learning architectures. To ensure reliability, the researcher employed 5-fold stratified cross-validation on the dataset [6]. This maintained proportional representation of classes, preventing bias. The results reported averages ± standard deviations with 95% confidence intervals (normal approximation). Figure 3 shows the sentiment percentage per company.

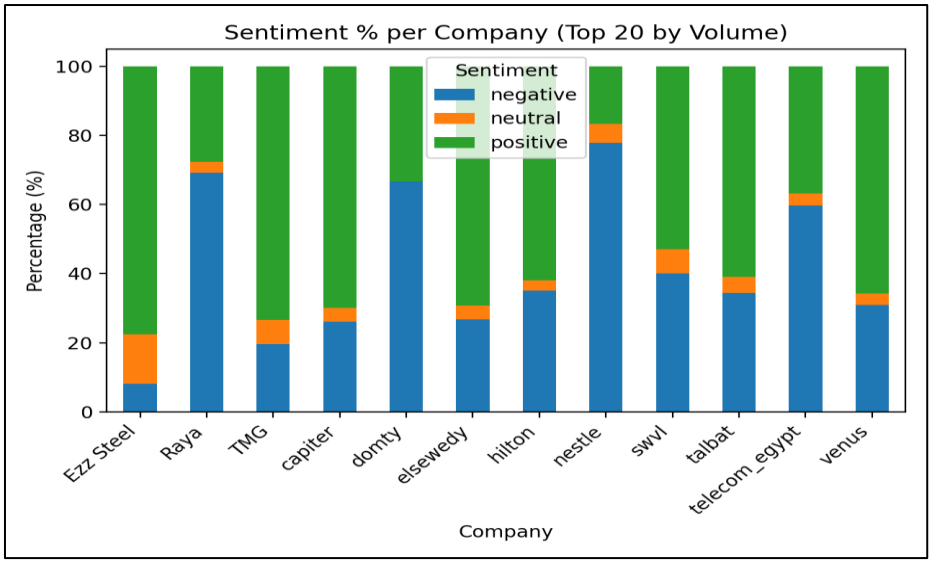

Figure 3. Sentiment percentage per company

4.1.1 Dataset Details

The Arabic Company Reviews dataset [6] contained 38,097 valid reviews after cleaning and filtering (original size ~40,046 reviews) from 12 companies: talbat (32,073), swvl (4,693), telecom_egypt (2,090), venus (281), Raya (268), TMG (250), elsewedy (147), hilton (100), capiter (73), Ezz Steel (49), nestle (18), domty (4). The overall sentiment distribution was positive (70%, ~28,032 reviews), negative (20%, ~8,009), and neutral (10%, ~4,005). Review lengths varied with an average ~50 words, where the minimum review size was 1, and the maximum review size was 200. Table 1 shows the sentiment class distribution, review length statistics, and sample reviews.

Table 1. Dataset Overview

Category	**Details**
Total Reviews	38,097
Number of Companies	12
Sentiment Class Distribution	
Positive	Positive: ~70% (~26,668 reviews)
Negative	Negative: ~20% (~7,619
Neutral	Neutral: ~10% (~3,810)
Review Length Statistics (approximate word count after preprocessing)	
Average length	~50 words
Median length	~38 words
Minimum length	1 word
Maximum length	200 words
25th percentile	~18 words
75th percentile	~68 words
Sample Reviews (illustrating typical language, dialectal features, and noise)	
Sample 1 (Positive)	**Arabic:** الخدمة ممتازة وسريعة جدًا، التوصيل في وقت قياسي **English:** The service is excellent and very fast, delivery in record time

	Rating/Sentiment: Positive (1) **Company:** talabat **Length:** ~12 words
Sample 2 (Negative)	**Arabic:** الإنترنت سيء جداً و دايماً ينقطع، مش عارف أتصفح حتى **English:** The internet is very bad and always cuts off, I can't even browse **Rating/Sentiment:** Negative (-1) **Company:** telecom_egypt **Length:** ~18 words
Sample 3 (Neutral)	**Arabic:** المنتج عادي، السعر مناسب بس الجودة متوسطة **English:** The product is average; the price is reasonable but the quality is medium **Rating/Sentiment:** Neutral (0) **Company:** swvl / Raya (similar style) **Length:** ~14 words

It is important to point out that the labels are derived from user ratings (stars >3 positive). As user-generated data, no inter-rater reliability is available; potential biases from dialects are noted as a limitation.

4.2 Hyperparameter Configuration

The models were configured to balance computational efficiency with classification depth as follows:

Traditional ML: The SVM was implemented using a LinearSVC with a regularization parameter C=1.0 and dual=False to optimize performance for a sample size where n > {features}. The TF-IDF vectorizer utilized a range of (1, 2) n-grams with a maximum of 5,000 features.

Deep Learning: Both the Bi-LSTM and Bi-GRU models utilized an Embedding layer (input dim: 5,000, output dim: 128) followed by a Bidirectional layer with 64 units. A Dropout rate of 0.5 was applied to the dense layer to mitigate overfitting, a common challenge when dealing with the high morphological variance of Arabic text.

Optimizer: Adam (lr=0.001, beta1=0.9, beta2=0.999), with batch size=32 and early stopping with patience=3, max epochs=20. Class weights (inverse class frequency) were applied during training to address severe imbalance.

Transformer (AraBERT): Fine-tuned aubmindlab/bert-base-arabertv2 using Hugging Face Trainer with batch_size=32, epochs=3 per fold, max_length=128, eval_strategy="epoch", fp16 enabled on GPU (RTX 4060). No additional class weighting was applied, as the pre-trained contextual embeddings implicitly handle imbalance.

4.2.1 Hyperparameter Optimization

Hyperparameters were selected via grid search on a validation set (20% of train). For SVM, searched C=[0.1,1,10]; best C=1. For DL, units=[32,64,128] (64 chosen for balance); dropout=[0.3,0.5,0.7] (0.5 best). Embedding dim=128, which is standard for semantics. Ablation results in Table 2 show sensitivity (reducing units to 32 drops Macro F1 by 0.02) for Bi-LSTM as example.

Table 2. Ablation on Key Hyperparameters (Bi-LSTM)

Hyperparameter	Value	Macro F1 Change
Units	32	-0.02
Dropout	0.3	-0.01
Embedding Dim	64	-0.015

4.3 Results Analysis

Based on the 5-fold cross-validation results and Precision–Recall curves generated from the last fold of each model, distinct performance characteristics were observed across the examined architectures, as summarized in Table 3 and Figures 4-10.

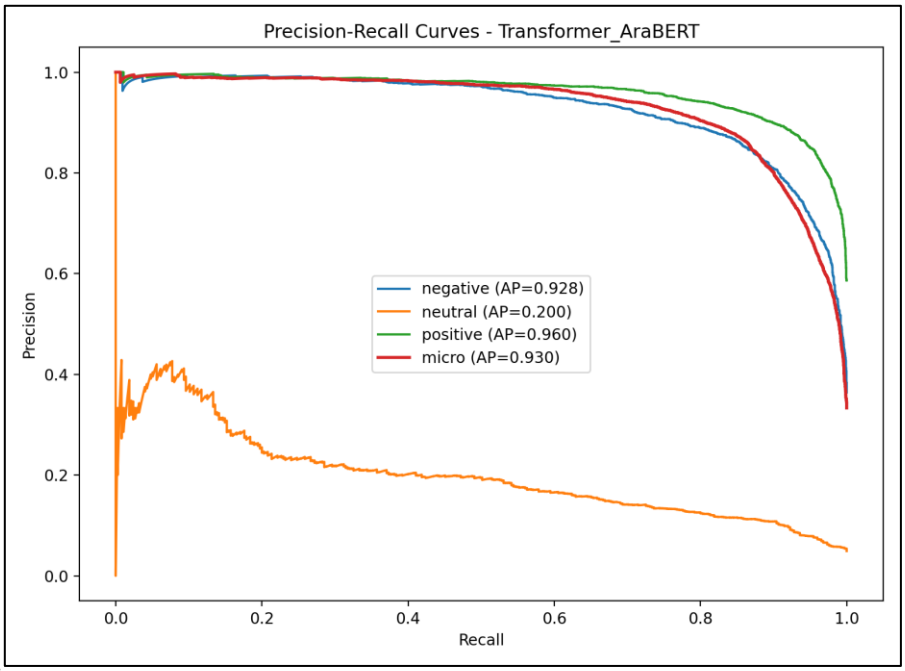

Figure 4. Precision/Recall of the AraBERT Model

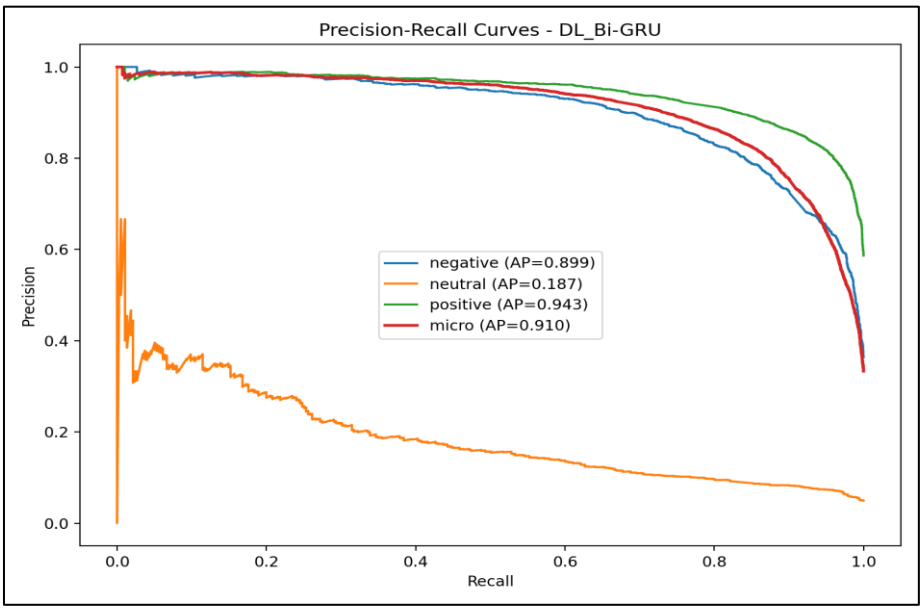

Figure 5. Precision/Recall of the Bi-GRU Model

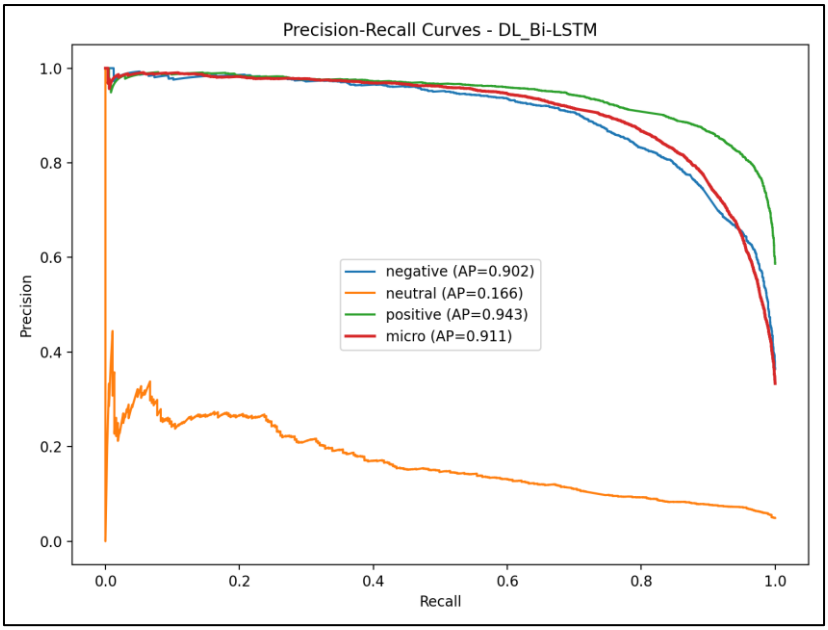

Figure 6. Precision/Recall of the Bi-LSTM Model

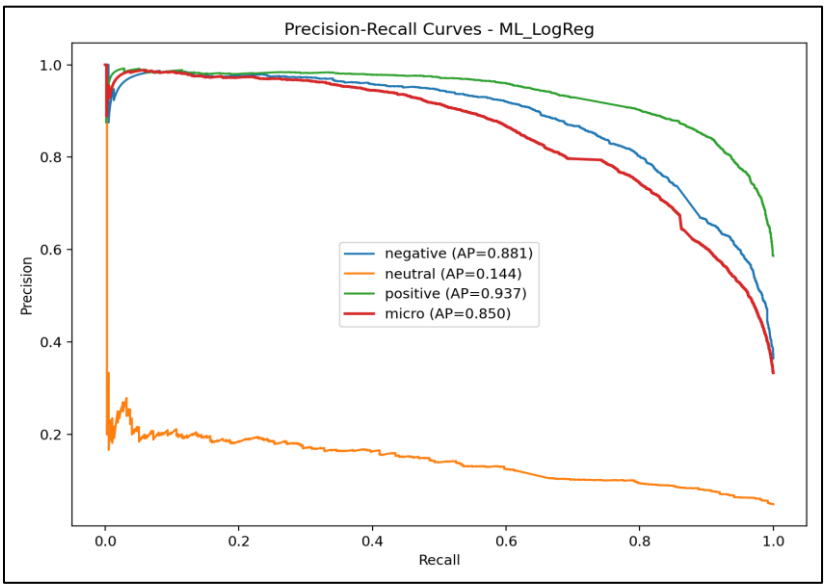

Figure 7. Precision/Recall of the Logistic Regression Model

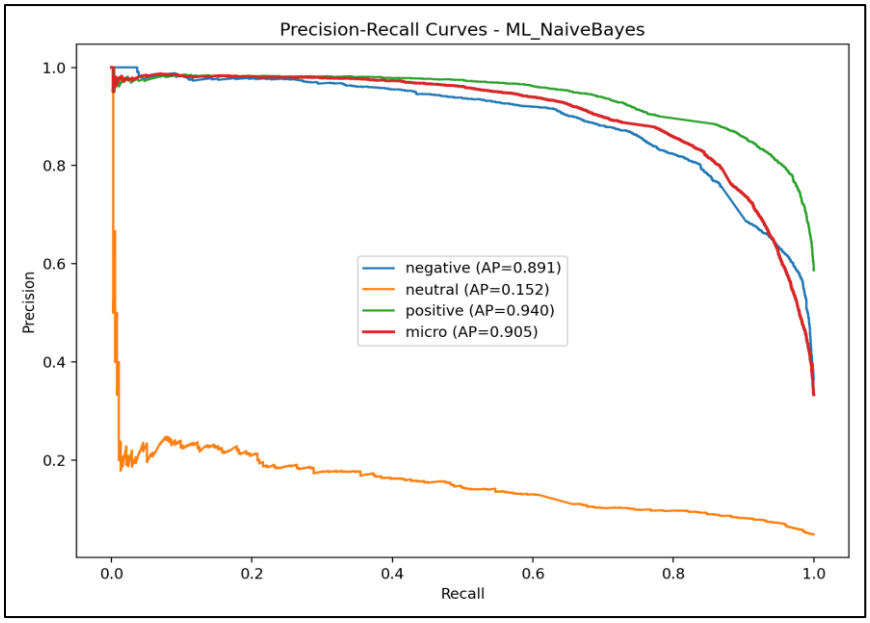

Figure 8. Precision/Recall of the Naive Bays Model

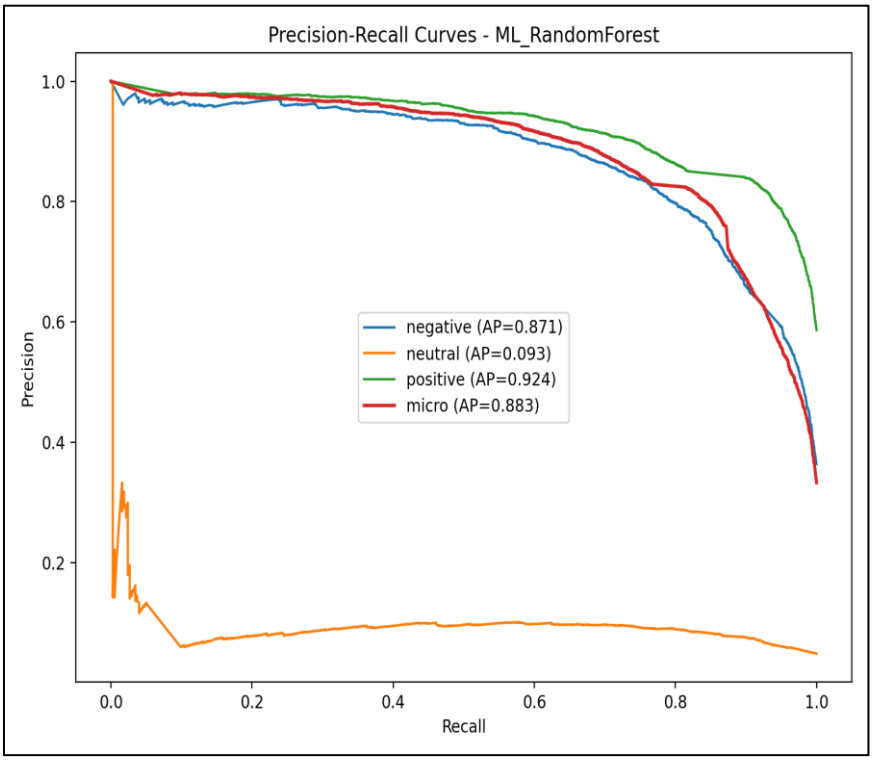

Figure 9. Precision/Recall of the Random Forest Model

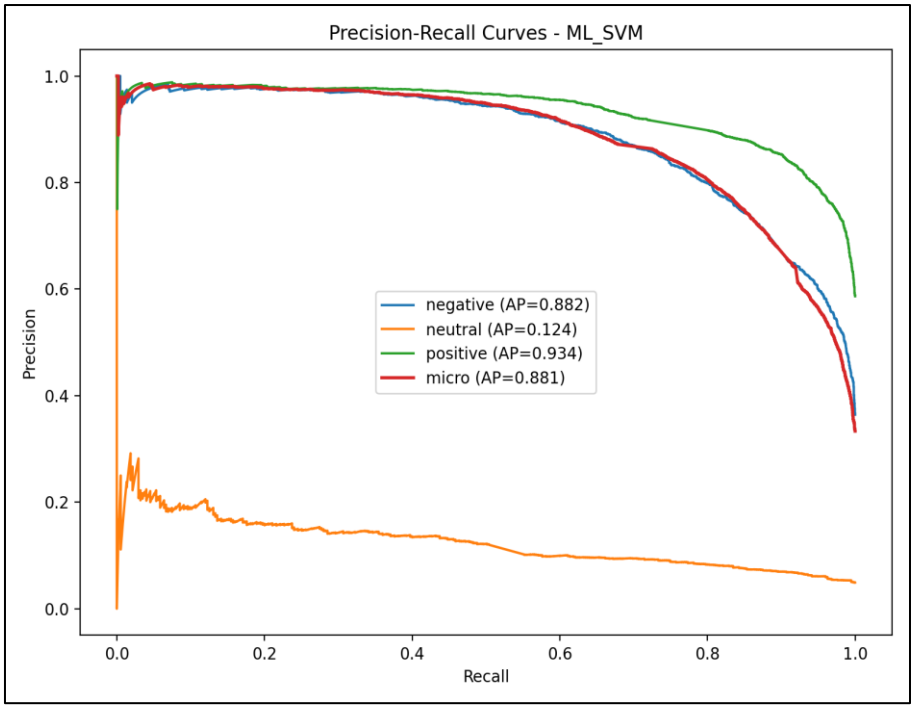

Figure 10. Precision/Recall of the SVM Model

Table 3. Comparative 5-Fold CV Performance (Mean ± Std; 95% CI)

Model	Accuracy	Macro F1	Weighted F1
SVM	0.8036 ± 0.0049 (0.7993–0.8079)	0.6205 ± 0.0058 (0.6154–0.6256)	0.8103 ± 0.0042 (0.8066–0.8140)
RandomForest	0.8158 ± 0.0032 (0.8130–0.8186)	0.5689 ± 0.0055 (0.5641–0.5737)	0.7990 ± 0.0028 (0.7965–0.8015)
NaiveBayes	0.8350 ± 0.0029 (0.8324–0.8376)	0.5673 ± 0.0022 (0.5654–0.5692)	0.8140 ± 0.0029 (0.8115–0.8165)
LogReg	0.7683 ± 0.0036 (0.7651–0.7714)	**0.6212 ± 0.0036 (0.6180–0.6243)**	0.7965 ± 0.0031 (0.7937–0.7993)
Bi-LSTM	0.8386 ± 0.0020 (0.8369–0.8404)	0.5681 ± 0.0020 (0.5663–0.5698)	0.8171 ± 0.0025 (0.8149–0.8194)
Bi-GRU	0.8365 ± 0.0011 (0.8355–0.8375)	0.5741 ± 0.0059 (0.5689–0.5793)	0.8162 ± 0.0022 (0.8143–0.8182)
AraBERT	**0.8598 ± 0.0044 (0.8560–0.8636)**	**0.6382 ± 0.0100 (0.6294–0.6469)**	**0.8479 ± 0.0043 (0.8441–0.8516)**

As shown in Table 3, AraBERT achieved the highest overall Accuracy and Macro F1-score, demonstrating the advantage of pre-trained contextual embeddings. Naive Bayes and Bi-LSTM/Bi-GRU models showed high accuracy and weighted F1, reflecting strong performance on the dominant positive class. In contrast, Logistic Regression obtained the highest Macro F1 among non-Transformer models, indicating a more balanced treatment of minority and majority sentiment classes. This behavior highlights the trade-off between overall correctness and class-balanced performance in highly imbalanced sentiment datasets. Although the deep learning models (Bi-LSTM and Bi-GRU) consistently improved Accuracy and Weighted F1 compared to linear baselines, they did not surpass Logistic Regression in terms of Macro F1. This suggests that increased model capacity primarily benefits majority class recognition rather than improving minority class discrimination. The Support Vector Machine demonstrated competitive performance across all metrics, serving as a strong classical

baseline that balances robustness and computational efficiency. Random Forest, while achieving reasonable accuracy, exhibited comparatively weaker class-balanced performance, consistent with its known limitations in high-dimensional sparse text representations.

Precision-Recall curves across all models highlighted severe class imbalance: positive class Average Precision (AP) rangeed from 0.924 (RandomForest) to 0.960 (AraBERT), while neutral class AP remained very low (0.093–0.200). Negative class AP was moderate (0.871–0.929). Micro-averaged AP was high (~0.85–0.93), driven by the dominant positive class. These curves visually confirmed the need for class weighting and justify Macro F1 as the primary balanced metric.

4.4 Automated Business Intelligence Output

A unique component of this experiment is the generation of the Arabic_Brand_Insights.xlsx report. By analyzing the "Sentiment Score" (mean rating) per company, the system identified that companies with the lowest scores frequently had negative reviews associated with specific latent topics.

For example, for a specific telecom provider in the dataset, the top negative keywords extracted were (/ تغطية سيء / انترنت), clearly pointing to "Network Coverage" issues. Conversely, high-scoring retail brands showed positive keywords such as (سريع / توصيل / ممتاز), highlighting "Logistics Speed" as their primary competitive advantage. With the addition of AraBERT, the report benefited from more accurate sentiment classification, leading to higher-quality topic extraction and more reliable BI insights.

5. Discussion

The results demonstrated that model performance varied significantly depending on the evaluation metric. Naive Bayes achieved the highest accuracy and weighted F1-score, suggesting strong performance on the majority sentiment class. However, its lower Macro F1-score indicated limited effectiveness in minority-class prediction.

Logistic Regression achieved the highest Macro F1-score, making it the most balanced classifier across sentiment classes. This confirms that linear models with TF-IDF features remain highly effective for Arabic sentiment analysis, particularly when fairness across classes is required.

Deep learning models (Bi-LSTM and Bi-GRU) produced competitive accuracy, reflecting their ability to capture contextual information. Nevertheless, their Macro F1-scores did not surpass those of linear models, likely due to dataset size constraints and the absence of pretrained Arabic embeddings. These findings highlighted the importance of aligning model choice with data characteristics and evaluation objectives.

The 5-fold stratified cross-validation results (Table 3) confirmed and strengthened these observations, with minor numerical differences from the original single-split experiments attributable to averaging across folds (e.g., Logistic Regression Macro F1 0.6212 ± 0.0036 vs. original 0.6257). Confidence intervals were narrow (e.g., Macro F1 CI width ~0.006–0.012), indicating stable performance and reduced variance compared to single splits. AraBERT achieved the highest overall performance (accuracy 0.8598 ± 0.0044; Macro F1 0.6382 ± 0.0100), outperforming all ML and DL models. This demonstrates the advantage of pre-trained contextual

embeddings for Arabic, which better handle dialectal variation and long-range dependencies without explicit feature engineering. However, AraBERT requires significantly higher computational resources (~20–30 min/fold on GPU) compared to ML (<1 min/fold on CPU) or DL (1–3 min/fold on GPU), highlighting a trade-off between accuracy and deployability in resource-constrained environments. Precision-Recall curves across all models revealed severe class imbalance: positive class Average Precision (AP) was consistently high (0.924–0.960), driven by the dominant positive reviews (~70%). Negative class AP was moderate (0.871–0.929), while neutral class AP remained very low (0.093–0.200), explaining the gap between high accuracy/weighted F1 and lower Macro F1. Micro-averaged AP (0.85–0.93) was inflated by the majority class, reinforcing Macro F1 as the primary balanced metric. These curves visually confirmed that even top models struggle with neutral sentiment, a common challenge in real-world Arabic reviews.

6. Conclusions and Future Work

This study presented an integrated ASA and BI framework for Arabic company reviews. Experimental evaluation showed that while deep learning architectures achieved strong accuracy, Logistic Regression delivered the best-balanced performance as measured by Macro F1-score. Naive Bayes remained an effective option when overall accuracy was prioritized. The integration of topic modeling enhanced interpretability, enabling organizations to identify actionable insights from sentiment-labeled reviews.

Future work should explore the incorporation of pretrained Arabic language models (e.g., AraELECTRA, MARBERT), data augmentation strategies (e.g., back-translation for minority classes), class-aware loss functions, and ensemble methods (e.g., stacking AraBERT with LogReg) to further improve minority-class sentiment detection and overall robustness under imbalance. Additional directions could include multi-label ASA, sarcasm detection, and real-time deployment on edge devices.

7. Declarations

Acknowledgments

Not applicable.

Ethical consideration

Not applicable.

Consent to participate

Not applicable

Conflicts of interest

The author (Maree) is a member of the Editorial Board for the AAUP Journal of STEM and Health Sciences. To maintain a transparent and unbiased peer-review process, Maree was not involved in the selection of reviewers or any editorial decisions regarding this manuscript. The peer-review process was handled independently by other editors of the journal. Otherwise, the authors have no conflicts of interest to declare.

Data availability

The Arabic Company Reviews dataset analyzed in this study is publicly available through Kagglei.

Code availability

The full source code, preprocessing pipeline, and scripts used to reproduce the reported experiments and figures are available from the author upon request and can be provided to reviewers during the evaluation process.

Funding Statement

The authors did not receive support from any organization for the submitted work.

Authors Contributions

All authors contributed to Conceptualization; Formal analysis; Investigation; Methodology; Supervision; Writing – original draft; Writing – review & editing and their roles in the paper were distributed accordingly.

References

[1] Maree, M.; Eleyat, M. and Rabayah, S. Pertanika Journal of Science & Technology, Volume 32, Issue 3, April 2024. DOI: https://doi.org/10.47836/pjst.32.3.05

[2] Alawneh, H.; Hasasneh, A.; Maree, M. On the Utilization of Emoji Encoding and Data Preprocessing with a Combined CNN-LSTM Framework for Arabic Sentiment Analysis. Modelling 2024, 5, 1469–1489. DOI: https://doi.org/10.3390/modelling5040076

[3] Al-Kabi MN, Gigieh AL, Al-Smadi IM, et al. Opinion mining and sentiment analysis for Arabic language: a comprehensive survey. Int J Comput Appl. 2014;101(15):24-35.

[4] Joachims T. Learning to Classify Text Using Support Vector Machines. 2nd ed. Norwell: Kluwer Academic Publishers; 2002.

[5] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. Journal of Machine Learning Research, 3, 993–1022.

[6] Seddik F. Arabic Company Reviews Dataset. Kaggle. 2021 [cited 2025 Jan 5]. Available from: https://www.kaggle.com/datasets/fahdseddik/arabic-company-reviews

[7] Zhang, L., Wang, S., & Liu, B. (2018). Deep learning for sentiment analysis: A survey. WIREs Data Mining and Knowledge Discovery, 8(4), e1253. DOI: https://doi.org/10.1002/widm.1253

[8] Abbasi A, Chen H, Salem A. Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums. ACM Trans Inf Syst. 2008;26(3):1-34.

[9] Farra N, et al. Sentence-level and document-level sentiment analysis for Arabic. Proceedings of IEEE Workshop on Spoken Language Technology. 2010;233-238.

[10] El-Halees A. Arabic opinion mining: A hybrid approach. Al-Aqsa Univ J. 2011;15(1):101-122.

[11] Al-Smadi, M., Talafha, B., Al-Ayyoub, M., & Jararweh, Y. (2019). Using long short-term memory deep neural networks for aspect-based sentiment analysis of Arabic reviews. International Journal of Machine Learning and Cybernetics, 10, 2163–2175. DOI: https://doi.org/10.1007/s13042-018-0799-4

[12] Dahou A, et al. Word Embeddings and Convolutional Neural Networks for Arabic Sentiment Classification. Proceedings of COLING 2016.

[13] Soliman AB, Eissa K, El-Beltagy SR. AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP. Procedia Comput Sci. 2017;117:256-265.

[14] Duwairi RM, et al. Sentiment Analysis for Arabic Dialects. Proc. of 9th Int. Conf. on Information and Communication Systems (ICICS). 2018.

[15] Abu Farha I, Magdy W. ArSarcasm: A New Arabic Sarcasm Detection Dataset. Proceedings of the Fourth Arabic Natural Language Processing Workshop. 2020.

[16] Antoun W, Baly F, Hajj H. AraBERT: Transformer-based Model for Arabic Language Understanding. LREC. 2020.

[17] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1724–1734.

[i] https://www.kaggle.com/datasets/fahdseddik/arabic-company-reviews