A Deep Learning Approach for Text-Based Sarcasm Detection using Lexical Cues

Aimen Naveed, Faria Kanwal, Huma Tauseef, Komal Bashir
Lahore College for Women University, Lahore
*Correspondence: faria.kanwal@gmail.com

Citation | Naveed. A, Kanwal. F, Tauseef. H, Bashir. K, "A Deep Learning Approach for Text-Based Sarcasm Detection using Lexical Cues", IJIST, Vol. 8 Issue. 9 pp 3831-3858, October 2026
Received | Oct 10, 2026 **Revised |** Oct 23, 2026 **Accepted |** Oct 26, 2026 **Published |** Oct 29, 2026.

Sarcasm conveys a meaning opposite to its literal wording, making it a persistent source of error in sentiment analysis, especially on social media and across languages. This study examines whether explicit lexical cues improve recurrent sarcasm detectors. We propose Bi-LSTM-LC, which fuses the pooled Bi-LSTM representation with four lexical cues (sarcastic phrases mined from the training data, emojis, punctuation patterns, and interjections), and compare it with a Bi-LSTM baseline and four TF-IDF classifiers (Decision Tree, Nearest Centroid, Complement NB, and Multinomial NB) on five public datasets in English (news headlines and tweets), Arabic, Urdu, and Persian. All models are evaluated with a leakage-free protocol: deduplication, a single stratified 80/20 split, hyperparameter selection by 5-fold cross-validation on the training set only, and one evaluation on the held-out test set. Bi-LSTM-LC reached test macro-F1 scores of 79.8% (news headlines), 57.4% (English tweets), 67.4% (Arabic), 78.1% (Urdu) and 71.3% (Persian). It outperformed the Bi-LSTM baseline on four of the five datasets under identical fixed settings (−0.5 to +2.8 points), but not consistently once both models were tuned (−3.5 to +3.1 points), and TF-IDF Naive Bayes was the strongest model on three datasets. On the imbalanced Arabic dataset (16% sarcastic), Bi-LSTM-LC achieved the highest macro-F1 (67.4% vs. 64.3% for the tuned Bi-LSTM), with a balanced accuracy of 67.0% and sarcastic-class precision, recall, and F1 of 46.2%, 43.7%, and 44.9%, although its accuracy (83.0%) did not exceed the majority-class rate (84.1%). An architecture-matched ablation showed that the benefit of the cues is dataset-dependent: on Arabic, removing them lowered macro-F1 by 11.7 points and sarcastic-class recall from 43.7% to 11.4%; the mined phrases were the decisive cue for Urdu; and the cues gave no improvement on the English and Persian datasets. Lexical cues are therefore not a general-purpose improvement for Bi-LSTMs trained from scratch, but they are valuable for minority-class detection on imbalanced, low-resource data, and equally tuned baselines are essential when evaluating sarcasm detectors.

Keywords: Sarcasm Detection; Bi-LSTM; Lexical Cues; Multilingual Text Classification; Low-Resource Languages; Deep Learning

Introduction:

According to Gibbs [1], sarcastic remarks are more memorable than literal ones. Sarcasm usually arises from a combination of situational, semantic, and lexical cues [2]. Sarcasm captures the reader's attention while conveying an implicit or explicit negative message. Detecting sarcasm is an important and challenging task in natural language processing (NLP).

Sarcasm is commonly used to express humor, irritation, displeasure, or insult. Sentiment analysis classifies the emotional tone of a text as positive, negative, or neutral; sarcasm undermines this task because its surface sentiment often contradicts its intended sentiment, leading to misclassification [2]. Detecting sarcasm is therefore necessary for reliable sentiment analysis; recent work has shown that adding sarcasm information to a sentiment model improves its accuracy by roughly 2% to more than 10% [3]. In spoken communication, listeners recognize sarcasm through facial expressions and vocal tone [4]. These signals are absent in written text. Writers compensate by using lexical markers, such as exaggerated punctuation, interjections, emojis, and characteristic phrases, to signal sarcastic intent. These lexical cues are the textual counterpart of tone of voice, and they motivate the approach taken in this study.

Sarcasm frequently occurs in social media, online reviews, and other forms of informal digital communication, making automated detection important for applications such as sentiment analysis and opinion mining [5]. The informal and context-dependent nature of such text limits the effectiveness of purely rule-based methods. Machine-learning approaches have been used to exploit textual and sentiment-related cues for sarcasm detection [6]. Among deep learning models, Bi-LSTM is particularly useful because it processes text in both forward and backward directions, allowing contextual information from preceding and succeeding words to contribute to the representation [7].

Multilingual sarcasm detection matters because a large and growing share of social media content is written in languages other than English, including Arabic, Urdu, and Persian, yet most sarcasm resources and models remain English-centric [5][8]. Models trained on English do not transfer directly to these languages: Arabic exhibits strong dialectal variation, Urdu and Persian are written in Perso-Arabic script with rich morphology and frequent code-mixing, and annotated corpora are small [9][10][11]. Even large language models still trail supervised pre-trained models on sarcasm benchmarks [12], and recent work guides models with explicit linguistic, contextual, and emotional cues of sarcasm [13]. Lexical features are attractive in this setting because they are inexpensive, interpretable, and can be compiled for any language without large pre-trained resources: punctuation, emojis, interjections, and recurrent sarcastic phrases are the written substitutes for the prosodic cues of spoken sarcasm [14][15].

Despite these advances, three gaps remain. First, most existing studies evaluate a single language, usually English, and a single domain. Second, sarcasm detection in Arabic, Urdu, and Persian remains comparatively under-resourced. Third, although lexicon-based features have occasionally been combined with deep models, most notably by [16] for English–Hindi text, explicit surface-level cues such as emojis, punctuation patterns, and interjections have rarely been integrated into deep models for Arabic, Urdu, or Persian. A further, methodological gap is that many reported results are obtained with single random splits, without class-specific measures for imbalanced data, or with preprocessing fitted on the complete dataset, which makes it difficult to compare studies. To address these gaps, this study makes the following contributions:

It proposes Bi-LSTM-LC, which integrates a four-dimensional vector of lexical cues (sarcastic phrases mined from the training data, emojis, punctuation patterns, and interjections) into a Bi-LSTM architecture through a separate input branch.

It evaluates Bi-LSTM-LC against four machine learning classifiers and a Bi-LSTM baseline on five datasets spanning two domains (news headlines and tweets) and four languages (English, Arabic, Urdu, and Persian).

It applies a leakage-free protocol in which each dataset is deduplicated and split into training (80%) and held-out test (20%) sets before any fitting, stratified 5-fold cross-validation is performed on the training set only, and the final models are evaluated once on the test set.

It reports balanced accuracy, macro-averaged and class-specific precision, recall, and F1, and the Matthews correlation coefficient (MCC), in addition to accuracy, so that performance on imbalanced data is not overstated.

It shows, through a controlled comparison, that the gains of Bi-LSTM-LC over a Bi-LSTM baseline under identical fixed training settings largely disappear when both models are tuned by cross-validation, and that TF-IDF Naive Bayes classifiers remain the strongest models on most datasets.

It quantifies the contribution of the lexical cues and of each cue type with an architecture-matched ablation study on all five datasets, showing that the cues are essential for minority-class detection on the imbalanced Arabic dataset and that the mined phrase list is the most useful cue for Urdu, but that they bring no gain on the English and Persian datasets.

Research Objectives:

The study pursues five objectives. Each objective is linked to a measurable criterion and to the part of the results in which it is evaluated.

O1 – Multilingual, leakage-free benchmark: Evaluate five datasets in four languages and two domains under one identical protocol. *Criterion:* zero text overlap between training and test sets after deduplication, and all fitted components (tokenizer, phrase list, scaler, TF-IDF) learned from training data only (Materials and Methods, Table 2).

O2 – Performance of Bi-LSTM-LC: Determine whether Bi-LSTM-LC improves on a simple Bi-LSTM and on four classical classifiers. *Criterion:* differences in macro-F1 and balanced accuracy on the held-out test set of each dataset, supported by the cross-validation mean and standard deviation and by a controlled comparison with identical training settings (Tables 4–7, 9, and 10).

O3 – Behaviour under class imbalance: Quantify how well the sarcastic minority class is detected on the imbalanced ArSarcasm corpus. *Criterion:* sarcastic-class recall and F1, balanced accuracy, and MCC compared with the majority-class baseline (Table 5, Figure 4).

O4 – Positioning against published work: Compare the results with published studies on the same datasets while accounting for differences in dataset version, split, and metric. *Criterion:* comparisons are labelled as direct only when the dataset version and metric match (Table 8).

O5 – Contribution of the lexical cues: Quantify how much the lexical cues, and each of the four cue types, contribute within the Bi-LSTM-LC architecture. *Criterion:* the change in cross-validation and test macro-F1 when the cue branch or a single cue is removed, with all other components and settings unchanged, together with fold-wise differences and an exact McNemar test (Table 11).

Novelty Statement:

Bi-LSTM-LC differs from existing approaches in five respects. (i) Data-driven, language-specific cue lists. Instead of fixed English lexicons such as SentiWordNet [15][16], the sarcastic-phrase list of each language is mined from the training data with an explicit, reproducible rule (Eq. 1), so the same procedure can be applied to Arabic, Urdu, and Persian without external lexical resources. (ii) Explicit fusion branch. The four cue counts are normalized and projected by a separate dense branch and fused with the pooled Bi-LSTM representation (Eq. 6), rather than being used in a rule-based or feature-engineered classifier [15][17] or replaced by attention mechanisms alone [7][18] (iii) One architecture across languages and domains. The same cue definitions, architecture, and hyperparameters (apart

from the learning rate and number of epochs, which are selected by cross-validation on each training set) are applied to five datasets, whereas most prior models are designed and tuned for a single language and dataset (Table 1). (iv) Leakage-free and imbalance-aware evaluation. All fitted components are learned inside the training data, cross-validation is confined to the training set, and class-specific measures and balanced accuracy are reported for every model. (v) Architecture-matched ablation. The contribution of the cue branch and of each cue type is measured within the same architecture and training configuration, which separates the effect of the cues from that of the additional layers. The contribution of the study therefore lies in the cue-fusion design, in its controlled multilingual evaluation, and in two findings: under fair tuning, the cues do not provide a general improvement over a Bi-LSTM trained from scratch, but the ablation shows that they are decisive for detecting the sarcastic minority class on imbalanced Arabic data and that the mined phrase list carries real signal for Urdu.

Related Work:

Deep learning approaches have consistently outperformed classical machine learning classifiers for sarcasm detection because they learn semantic and contextual features directly from text [19]. Researchers have further improved these models by adding attention mechanisms, pooling layers, and hybrid architectures, as reviewed below.

For English, the most common strategy has been to combine LSTM-based models with attention or pooling layers. [7] used a Bi-LSTM with multi-head attention (MHA-BiLSTM) to detect sarcastic tweets and showed that it outperformed a feature-rich SVM. [20] combined global max pooling with an LSTM (GMP-LSTM) and reached 92.54% accuracy on the News Headlines dataset, outperforming Bernoulli Naive Bayes and support vector classifiers. [21] applied an attention-based hybrid neural network with 1-D max pooling and n-gram features to the same dataset, reporting 89.7% accuracy. [22] proposed RCNN-RoBERTa, a transformer-based method combining a recurrent CNN with RoBERTa, which performed well on four benchmark datasets. [23] reported accuracies of 92–94% with BERT and CNN+Bi-LSTM models using several word embeddings, and [24] reported 95.27% accuracy with a deep CNN-LSTM model using TF-IDF features. [18] combined a soft-attention Bi-LSTM with a convolutional network and GloVe embeddings.

A second line of work has focused on strengthening the Bi-LSTM itself. [25] applied ELMo, a deep contextualized word representation derived from a Bi-LSTM language model, to sarcasm and irony detection and achieved state-of-the-art results on six of seven datasets. [26] presented a three-layer stacked Bi-LSTM for text-based sarcasm detection and reported 95.30% accuracy. [27] combined a multi-layer Bi-LSTM with GloVe embeddings, which capture semantic relationships between words; this reduced overfitting and led to 95.41% accuracy. [28] combined an LSTM with an autoencoder and obtained an 84% F1-score on a benchmark dataset. Across these English-language studies, reported accuracies range from about 90% to 95.4%, and nearly all improvements come from architectural additions rather than explicit linguistic features. [16] are among the few to extend such models beyond English: they combined a soft-attention Bi-LSTM with a feature-rich CNN for English–Hindi text, using GloVe embeddings for English and Hindi-SentiWordNet features for Hindi, and achieved 92% accuracy and an 89% F1-score.

More recent studies (2023–2026) have moved towards transformers and large language models. [12] benchmarked large language models on six sarcasm datasets and found that they still underperform supervised pre-trained language models, with few-shot prompting outperforming chain-of-thought prompting. [13] decomposed sarcasm detection into linguistic, contextual, and emotional dimensions and trained a reasoning model with dimension-specific rewards, outperforming state-of-the-art baselines in most settings on five datasets. Two recent reviews [8][19] conclude that deep learning, and increasingly multimodal,

models now dominate sarcasm detection, while multilingual sarcasm detection remains a recognised challenge.

[29] contributed to Arabic sarcasm detection by constructing the ArSarcasT corpus and proposing a static and context-based hybrid deep network (SC-HDNN). [30] addressed sentiment analysis and sarcasm detection in Arabic tweets, including unseen domains. [9] highlighted the challenges of detecting sarcasm in Arabic text given the scarcity of Arabic datasets. They evaluated pre-trained models on ArSarcasm-v2 and the Saudi Dialect Irony dataset. AraBERT-v2 achieved 78.73% accuracy, whereas mBERT achieved 71.18% on the Saudi Dialect Irony dataset. [31] addressed both the detection and the quantification of sarcasm in Arabic tweets. [11]introduced the ArSa-Tweets corpus and found that AraBERT-v02 outperformed LSTM, CNN-LSTM-GRU, and BERT models, while [3] showed that adding sarcasm features to Bi-LSTM sentiment models improves results on Arabic (ArSarcasm-v2), English, and code-mixed data. These studies show that Arabic sarcasm detection lags behind English, with the best reported accuracy on the Saudi Dialect Irony dataset reaching 78.73% [9] largely due to limited annotated data, class imbalance, and dialectal variation.

Research on sarcasm detection in Urdu and Persian is even more limited. [32]proposed a context-driven approach for both sentiment and sarcasm classification in Urdu text and highlighted the importance of using contextual information and cognitive relationships in text classification tasks. [10]proposed a hybrid architecture combining mBERT embeddings with a Bi-LSTM and multi-head attention (MHA) on a new dataset of 12,910 Urdu tweets, which achieved 79.51% accuracy and an 80.04% F1-score and outperformed models based on fastText embeddings [33]released the Urdu Sarcastic Tweets dataset used in this study and evaluated classical classifiers with textual features, and later proposed UrduSarcasmNet, a transformer with cascaded group multi-head attention for the same corpus [34]Islam and Azhar [35]fine-tuned mBERT and UrduBERT on English news headlines and their Urdu translations and found mBERT to perform best.

Persian is similarly under-resourced. [36]fine-tuned transformer-based language models for Persian humor detection, and [37]combined deep learning and machine learning techniques to detect sarcasm in Persian tweets, highlighting the value of a Persian proverb-based dictionary. [38]applied transfer learning from an emoji prediction task, using multi-task and single-task architectures; their best model achieved 83% accuracy. For the related task of sentiment classification, [39] fine-tuned ParsBERT on 28,710 Instagram comments from several domains, labelled as positive or negative, and reported 68% accuracy across three cross-domain settings. Although that work addresses sentiment rather than sarcasm, it illustrates the difficulty of Persian text classification.

Beyond architecture, lexical cues provide direct textual evidence of sarcasm. [14]identified hyperbole and ironic tone of voice as two key cues for verbal irony. [6] further investigated hyperbole together with sentiment information for sarcasm detection in tweets. [17]proposed figurative-language features for detecting humor and irony in tweets. In written text, where tone of voice is unavailable, writers signal the same intent through lexical markers such as intensifiers, interjections, repeated punctuation, and emojis. [15] exploited such markers, including interjection words, emoticons, and hashtags, together with SentiWordNet and WordNet scores, to detect sarcasm in online comments. However, these cues have mostly been used in rule-based or feature-engineered systems for English and have seldom been integrated into deep learning models for other languages.

Table 1 summarizes representative deep learning studies, including their publication year, dataset size, reported performance, and main limitations, and contrasts them with the proposed approach.

Table 1. Summary of recent deep learning studies on sarcasm detection and comparison with the proposed approach

Ref. (year)	Dataset (size)	Language	Features / model	Reported performance	Main limitation
[40] (2022)	News Headlines; Reddit	English	fastText, Word2Vec, GloVe / weighted ensemble of LSTM, GRU and CNN	≈99% acc. (headlines); ≈82% (Reddit)	English only; heavy multi-model ensemble
[21](2023)	News Headlines v2 (28,619)	English	Word2Vec / attention-based hybrid NN	89.7% acc.	Single domain and language
[20] (2023)	News Headlines v1 (≈26,700) + Sarcasm Corpus V2 (9,386)	English	Global max pooling / GMP-LSTM	92.5% acc.	70:30 split without CV; accuracy-centred
[41] (2021)	Tweets (39,267) + News Headlines (26,709)	English	TF-IDF, BoW, GloVe / 1-D CNN-LSTM	91.6% acc. (multi-domain); 73% on unseen domains	Single 3:1 split; weak cross-domain generalisation
[42] (2022)	ArSarcasm (10,547)	Arabic	BoW, TF-IDF, BERT / AraBERT	88% acc. (binary)	Accuracy only on a 16%-sarcastic corpus
[43] (2021)	ArSarcasm-v2 (12,548 train / 3,000 test)	Arabic	MARBERT / multi-task (sarcasm + sentiment)	76.8% acc.; sarcastic-class F1 0.60	One official split; no variance reported
[3] (2024)	ArSarcasm-v2, iSarcasmEval, IMDB, SentiMixArEn	Arabic, English, code-mixed	Word2Vec, fastText, GloVe, BERT / (Bi-)LSTM with sarcasm features	+2% to >10% acc. when sarcasm information is added	Sarcasm used to support sentiment analysis rather than evaluated as a detector
[10] (2024)	Urdu tweets (12,910)	Urdu	mBERT / Bi-LSTM + multi-head attention	79.5% acc.; 80.0% F1	Single language; costly transformer embeddings
[35] (2024)	English headlines and their Urdu translation	English, Urdu	mBERT, UrduBERT fine-tuning	mBERT best (precision, recall, F1)	Urdu side machine-translated, not native Urdu text
[38] (2020)	MirasIrony (4,339 tweets)	Persian	fastText / Bi-LSTM + attention, emoji-prediction transfer	83.1% acc.	Small corpus; F1 not reported
This study	Five datasets (Table 2)	English, Arabic, Urdu, Persian	Mined phrase list, emojis, punctuation, interjections / Bi-LSTM-LC (two-branch fusion)	Tables 4–7 and 11	Cue gains dataset-dependent; no pre-trained embeddings (see Limitations)

Research gaps and how this study addresses them:

Table 1 reveals four gaps.

G1 – single language and domain: most studies address one language and one dataset, most often English news headlines; this study evaluates one architecture on five datasets in four languages and two domains.

G2 – limited use of explicit cues in deep models for low-resource languages: lexical cues have mainly been used in rule-based English systems [15][17]. Bi-LSTM-LC mines cue lists from the training data of each language, fuses them with the sequence representation, and quantifies their contribution with an ablation study.

G3 – inconsistent evaluation: reported results rely on different dataset versions, single splits, and accuracy alone, and the ArSarcasm and Urdu studies. [33][42] report accuracy without class-specific measures; this study uses a leakage-free protocol with cross-validation on the training set and reports balanced accuracy and class-specific F1.

G4 – class imbalance: Arabic studies report much lower sarcastic-class F1 than accuracy. [42][43] this study reports minority-class performance explicitly and analyses its failure modes (Results and Discussion).

Materials and Methods:

Bi-LSTM-LC combines a deeper Bi-LSTM sequence-processing branch with an explicit lexical-cue branch to capture complementary signals associated with sarcasm. The approach was applied to datasets from multiple domains and languages. The pipeline begins with data acquisition and ends with performance evaluation (Figure 1). After cleaning, every dataset is split once into a training set (80%) and a held-out test set (20%); all components that are learned from data are fitted on the training set only, model development uses stratified 5-fold cross-validation inside the training set, and each final model is evaluated once on the test set.

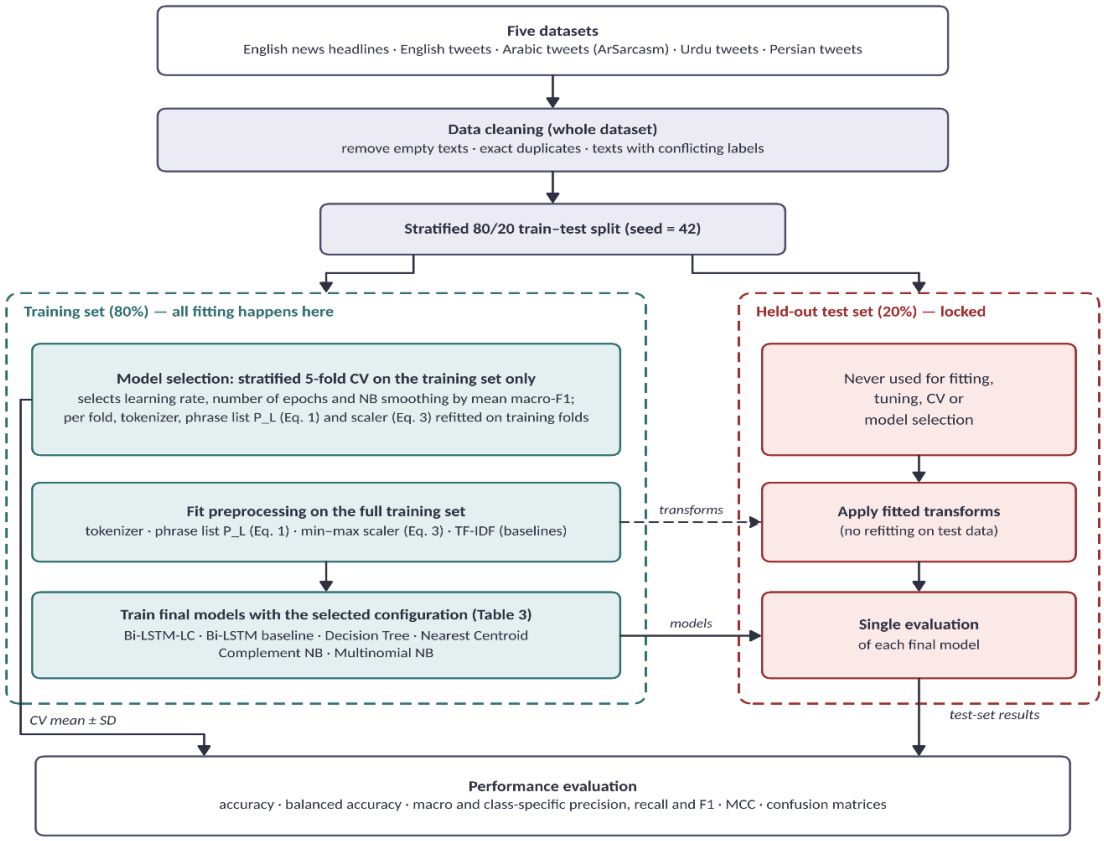


Figure 1. Overview of the leakage-free evaluation pipeline of the proposed Bi-LSTM-LC approach

Datasets:

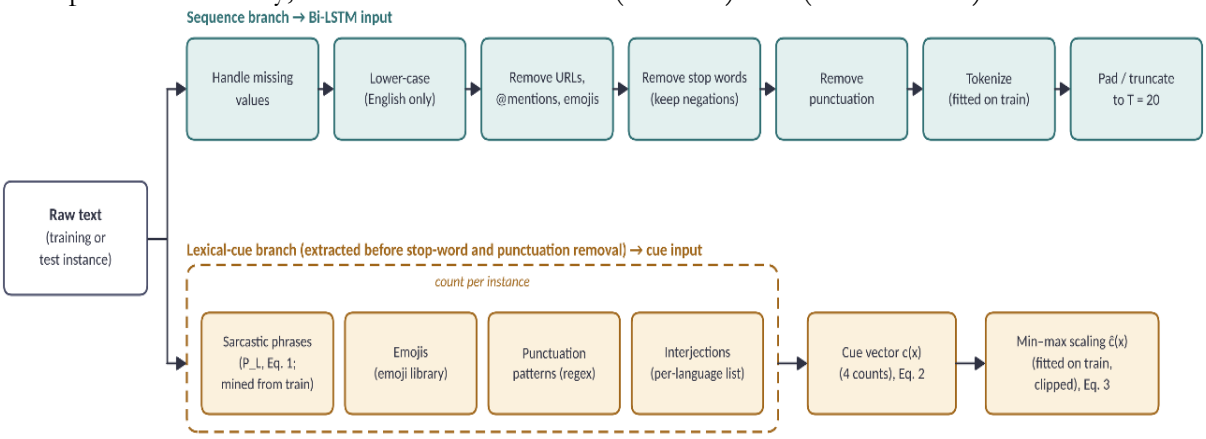
Five publicly available datasets were used: two English datasets (news headlines and tweets) and three tweet datasets in Arabic, Urdu, and Persian. Each contains instances labeled as sarcastic or non-sarcastic. Table 2 summarizes the datasets, including their sources, annotation methods, class distributions, and training and test sizes. Before splitting, empty texts and exact duplicates were removed, and texts that occurred with both labels were removed entirely (English news headlines: 116 duplicate and 0 conflicting-label rows; English tweets: 0 duplicate and 0 conflicting-label rows; Arabic tweets (ArSarcasm): 43 duplicate and 35 conflicting-label rows; Urdu tweets: 3,256 duplicate and 2,131 conflicting-label rows; Persian tweets: 22 duplicate and 0 conflicting-label rows). None of the English tweets contained the collection hashtags #irony, #sarcasm, or #not, which would reveal the label; an automatic check removing such hashtags was nevertheless applied before any processing. In the Urdu file, 56 of 20,060 rows had no label and were discarded before cleaning; of the remaining 20,004 rows, 27.2% were empty, duplicated, or carried conflicting labels. Without this step, identical tweets would appear in both the training and test sets. For the news headlines, version 2 of the dataset [21] was used, because combining versions 1 and 2, as in the original submission, places near-identical headlines in both the training and test sets. For the English tweets, the official SemEval-2018 Task 3 (Task A) release [44]was used; its training (3,834) and test (784) files were combined and re-split in the same way as the other datasets, with the test texts taken from the official test input file, in which the collection hashtags had been removed. The official split was used additionally for the comparison with published results (Table 8).

Table 2. Details of textual datasets for sarcasm detection (class distribution and splits after cleaning)

No.	Dataset [ref.]	Language / domain	Source and annotation method	Raw → after cleaning	Sarcastic n (%)	Train / test
1	News Headlines Dataset for Sarcasm Detection, v2 [21]	English / news headlines	Headlines from *The Onion* (sarcastic) and *HuffPost* (non-sarcastic); labels assigned by source (distant supervision)	28,619 → 28,503	13,552 (47.5%)	22,802 / 5,701
2	SemEval-2018 Task 3 irony tweets (Task A) [44]	English / tweets	Tweets collected with #irony, #sarcasm and #not, balanced with non-ironic tweets from the same users, and manually annotated by trained annotators (Fleiss' κ = 0.72); collection hashtags removed	4,618 → 4,618	2,222 (48.1%)	3,694 / 924
3	ArSarcasm [45]	Arabic / tweets	Tweets from ASTD and SemEval-2017 sentiment corpora, re-annotated by crowdsourcing (Figure-Eight; ≥3 annotators	10,547 → 10,469	1,659 (15.8%)	8,375 / 2,094

			per tweet; 89.3% agreement)			
4	Urdu Sarcastic Tweets Dataset [33][46]	Urdu / tweets	Tweets collected from X via trending hashtags; annotated by three annotators with majority vote	20,004 → 14,568	5,907 (40.5%)	11,654 / 2,914
5	Persian Irony Detection dataset [47]	Persian / tweets	Tweets reposted on the Telegram channel OfficialPersianTwitter (Dec 2021–Aug 2022); labels assigned automatically from members' reactions	14,946 → 14,924	7,006 (46.9%)	11,939 / 2,985

Preprocessing: Data were preprocessed before being passed to the model (Figure 2), and every instance was processed along two parallel branches. In the sequence branch, missing values in the text column were replaced with empty strings and the text was Unicode-normalised (NFKC). English text was then converted to lowercase; this step does not apply to Arabic-script text. URLs, user mentions, and emojis were removed from the sequence input, and Arabic diacritics were stripped. Stop words were then removed using the NLTK lists for English and Arabic and manually compiled lists for Urdu and Persian, except for negation words such as "not" and "never" (and their Arabic, Urdu, and Persian equivalents, e.g., نہیں, لا, and نہ), which are important for recognizing sarcasm. After punctuation removal, the text was tokenized with a tokenizer fitted on the training data only (unseen words are mapped to an out-of-vocabulary token), and each sequence was padded or truncated to a fixed length of 20 tokens. In the lexical-cue branch, the four cues described below were extracted from the raw text (lower-cased for English) before stop-word and punctuation removal, so that these signals were preserved. Finally, labels were encoded as 1 (sarcastic) or 0 (non-sarcastic).



Labels are encoded as 1 = sarcastic and 0 = non-sarcastic. Every fitted component (tokenizer, phrase list, scaler) is learned from training data only and reused unchanged on validation and test data.

Figure 2. Data preprocessing steps for the sequence branch and the lexical-cue branch

Lexical Cues: Four types of lexical cues were extracted from the raw text of each instance, before stop-word and punctuation removal, so that these signals were preserved:

Sarcastic phrases: a separate phrase list was compiled for each language from the training set only, to avoid information leakage from the test set; during cross-validation the list was re-mined inside each fold from the four training folds. An n-gram g (n = 1–3) was retained as a sarcastic phrase if it satisfied Eq. (1):

$$df_s(g) \geq 5 \text{ and } df_s(g) / (df_s(g) + df_n(g)) \geq 0.75 \quad (1)$$

where df_s(g) and df_n(g) are the numbers of sarcastic and non-sarcastic training instances containing g. The 200-qualifying n-grams with the highest df_s(g) forms the phrase list P_L for language L. The three values were fixed a priori and were not tuned on test data. The minimum support df_s(g) ≥ 5 excludes rare n-grams whose sarcastic proportion is unreliable: an n-gram that occurs once in a sarcastic instance would otherwise obtain a proportion of 1.0 by chance. The proportion threshold of 0.75 requires an odds ratio of at least 3:1 in favour of the sarcastic class, which retains only strongly class-associated n-grams in the balanced datasets; for the imbalanced ArSarcasm corpus (about 16% sarcastic) it corresponds to almost five times the base rate and is therefore deliberately conservative. The cap of 200 keeps the list compact relative to the vocabulary and limits the influence of individual topical n-grams. The number of n-grams satisfying Eq. (1) on the full training sets was English news headlines: 2,137; English tweets: 204; Arabic tweets (ArSarcasm): 22; Urdu tweets: 2,609; Persian tweets: 1,494; when fewer than 200 n-grams qualify, all of them are used. Emojis: detected using the Python emoji library, which matches characters against the Unicode emoji standard.

Punctuation patterns: repeated exclamation or question marks ("!!", "?!", including the Arabic question mark "؟"), ellipses ("...", "…", and the repeated Urdu full stop "۔۔"), and short spans enclosed in quotation marks used for emphasis, detected using regular expressions.
Interjections: taken from manually compiled lists for each language (e.g., "wow" and "oh great" in English, "يا سلام" in Arabic, "واہ" in Urdu, and "وای" in Persian), complemented by regular expressions for written laughter (e.g., "haha", "هههه", and "خخخ").
Each instance x is then represented by a four-dimensional cue vector in Eq. (2):

$$c(x) = [c_phr(x), c_emo(x), c_pun(x), c_int(x)] \quad (2)$$

where c_phr, c_emo, c_pun, and c_int are the numbers of sarcastic phrases from P_L, emojis, punctuation patterns, and interjections in x, respectively. Each element is scaled to [0, 1] using min–max normalization fitted on the training set:

$$\hat{c}_j(x) = (c_j(x) - min_j) / (max_j - min_j) \quad (3)$$

where min_j and max_j are the minimum and maximum of cue j over the training set; test values outside [0, 1] are clipped to this range. Eq. (3) has a zero denominator when cue j is constant in the training data (max_j = min_j), for example when no training instance contains an emoji. In that case $\hat{c}_j(x)$ is set to 0 for every instance, so the constant cue carries no information and cannot cause a division by zero; this is the behaviour of the scikit-learn MinMaxScaler used in the implementation, which replaces a zero range by 1. The Architecture paragraph describes how $\hat{c}$ is combined with the sequence features.

Data Splitting and Prevention of Data Leakage:

After cleaning, each dataset was split once, with stratification by class and a fixed random seed (42), into a training set (80%) and a held-out test set (20%). Stratified 5-fold cross-validation [48]was applied to the training set only: each fold preserves the class proportions of the training data, and in each iteration the model is trained on four folds and validated on the fifth. Cross-validation was used for two purposes: to select the learning rate (10^{-3} or 10^{-4}) and the number of training epochs (1–20) of each neural model, and the smoothing parameter $\alpha \in \{0.1, 0.3, 1.0\}$ of the two Naive Bayes classifiers, by maximising the mean validation macro-F1 over the five folds; and to estimate the stability of each model (mean and standard deviation across folds, Table 9). After cross-validation, each model was retrained from scratch on the complete training set with its selected configuration, and this final model was evaluated once on the held-out test set. Final performance is reported on the held-out test set, which was never used during training, cross-validation, preprocessing, or hyperparameter selection. Algorithm 1 describes the fold generation, which was implemented with the StratifiedKFold class of scikit-learn (shuffling enabled, seed 42). For each class c, the

instances are shuffled and divided into k folds of size $\lfloor |D_c| / k \rfloor$, with the r_c leftover instances assigned one each to the first r_c folds, so that every fold preserves the class proportions of the training set. In each of the k iterations, the model is trained on $k - 1$ folds and validated on the remaining fold.

Algorithm 1: Stratified fold generation

Input:

- Dtrain: training set (80% of the dataset)
- C: set of classes in Dtrain, C = {sarcastic, non-sarcastic}
- k: number of folds (k = 5)

Output:

- F_i for i=0, 1, ...,k−1: k disjoint folds that together cover Dtrain, each with approximately the same class proportions as Dtrain

Procedure:

1. Initialize $F_i \leftarrow \emptyset$ for $i = 0, 1, \ldots, k - 1$
2. for each class $c \in C$ do
3. $D_c \leftarrow$ all instances of class c in Dtrain
4. Randomly shuffle D_c
5. $nc \leftarrow \lfloor |D_c| / k \rfloor$
6. $rc \leftarrow |D_c| - k \,.\, nc$ (leftover instances)
7. for $i = 0$ to $k - 1$ do
8. $si \leftarrow nc + 1$ if $i < rc$, otherwise $si \leftarrow nc$
9. Move the first si instances of D_c into F_i
10. end for
11. end for
12. return $F_0, F_1, \ldots, F_{(k-1)}$

Data leakage was prevented at four points. (i) Exact duplicates and conflicting-label texts were removed before splitting, and the absence of shared texts between the training and test sets was verified programmatically (zero overlap for every dataset). (ii) The tokenizer vocabulary, the sarcastic-phrase list P_L (Eq. 1), the min–max scaler (Eq. 3), and the TF-IDF vectorizer of the classical baselines were fitted on training data only; inside cross-validation they were refitted in every fold on the four training folds. (iii) The test set was not used for early stopping, threshold selection, or model selection; the decision threshold was fixed at 0.5. (iv) Label-revealing collection hashtags were removed from the English tweets. Random seeds were fixed (Python, NumPy, and TensorFlow seed 42) for splitting, fold generation, and weight initialisation. All experiments were implemented in Python with TensorFlow 2.21 and scikit-learn; the neural models were trained on a GPU in Google Colab, except for the Urdu experiments, which were run on a CPU with the same code and seeds. An independent second run of the complete Colab notebook reproduced every reported value exactly, including the configurations selected by cross-validation. Apart from the learning rate and the number of epochs, which were selected by cross-validation on the training set, the hyperparameters in Table 3 were fixed before the experiments on the basis of preliminary runs and common settings for Bi-LSTM text classifiers, and they were kept identical for all datasets; no hyperparameter was ever selected on the test sets.

Architecture:

The input sequence passes through an embedding layer (dimension 32), a spatial dropout layer (rate 0.3), a first Bi-LSTM layer (128 units per direction), a second spatial dropout layer (0.3), and a second Bi-LSTM layer (64 units per direction). Given the embedded input sequence ($e_1, e_2, \ldots, e_T$), with T = 20, each Bi-LSTM layer processes the sequence in

both directions, and its output at time step t is the concatenation of the forward and backward hidden states (Eq. 4):

$$h_t = [h_t(forward) ; h_t(backward)] \quad (4)$$

A global max pooling layer then keeps the strongest activation of each feature k across the sequence:

$$z_k = \max(h_{1,k}, h_{2,k}, \ldots, h_{T,k}) \quad (5)$$

The normalized cue vector $\hat{c}$ (Eq. 3) is projected by a dense layer of 16 units with ReLU activation and concatenated with the pooled sequence features z (128 dimensions), giving a 144-dimensional fused vector:

$$u = \mathrm{ReLU}(W_c \cdot \hat{c} + b_c),\ v = [z ; u] \quad (6)$$

The fused vector v then passes through two dense layers (256 and 128 units, ReLU activation, L2 regularization), each followed by batch normalization (BN) and dropout (rate 0.5):

$$a_l = \mathrm{Dropout}(\mathrm{BN}(\mathrm{ReLU}(W_l \cdot a_{(l-1)} + b_l))), \text{ with } a_0 = v \text{ and } l = 1, 2 \quad (7)$$

Finally, a dense output layer with sigmoid activation gives the probability that x is sarcastic:

$$\hat{y} = \sigma(w_o \cdot a_2 + b_o), \text{ where } \sigma(s) = 1 / (1 + e^{(-s)}) \quad (8)$$

Eqs. (4)–(8) together define the forward pass of Bi-LSTM-LC; an instance is classified as sarcastic if $\hat{y} \geq 0.5$. The model is trained with the Adam optimizer to minimize the binary cross-entropy loss with L2 regularization on the dense-layer weights (Eq. 9):

$$L = -(1/N) \Sigma_i [\, y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)\,] + \lambda\, (\|W_1\|^2 + \|W_2\|^2) \quad (9)$$

where N is the number of training instances, $y_i \in \{0, 1\}$ is the true label of instance i, $\hat{y}_i$ is the predicted probability, and $\lambda = 0.01$ is the regularization factor. Figure 3 shows the architecture, and Table 3 lists the hyperparameters. The parameter values were chosen as follows. The sequence length T = 20 was chosen to cover short texts without padding most inputs heavily; it covers almost all headlines and English tweets and most Arabic and Urdu tweets, but fewer than half of the Persian tweets, whose remaining tokens are truncated (share of training instances with at most 20 tokens after stop-word removal: English news headlines: 100.0%; English tweets: 100.0%; Arabic tweets (ArSarcasm): 96.8%; Urdu tweets: 84.4%; Persian tweets: 42.8%; see Limitations). A small embedding dimension (32) and two Bi-LSTM layers of decreasing width (128 and 64 units per direction) were preferred because several datasets are small (Table 2) and no pre-trained embeddings are used, so a compact model is less prone to overfitting. Global max pooling (Eq. 5) retains the most salient activation of each feature regardless of its position, which suits short texts in which the sarcastic signal is often a single phrase. The cue projection has only 16 units because its input has four dimensions. Spatial dropout (0.3), dropout (0.5), batch normalization, and L2 regularization ($\lambda = 0.01$) on the two dense layers counter overfitting, and the batch size was 64. The learning rate and the number of epochs were selected for each dataset and model by cross-validation (Table 9), because preliminary runs showed that, at a learning rate of 10^{-3}, the models fit the training data almost perfectly within a few epochs while validation performance stopped improving, so the stopping point matters more than the step size. All other values were identical for every dataset.

The Bi-LSTM baseline consists of an embedding layer (dimension 32), a single Bi-LSTM layer (64 units), and a dense output layer with sigmoid activation, and uses the same preprocessing, batch size, and cross-validation-based selection of the learning rate and number of epochs. The four classical classifiers (Decision Tree, Nearest Centroid, Complement NB, and Multinomial NB, with default scikit-learn settings) use TF-IDF features of word unigrams and bigrams computed on the same cleaned text and fitted on the training set. Because Bi-LSTM-LC contains additional recurrent, pooling, dense, normalization, and lexical-cue components compared with the simple Bi-LSTM baseline, the observed performance

difference reflects the contribution of the complete proposed architecture rather than lexical cues alone. The separate contribution of the cues is therefore isolated by an architecture-matched ablation study, described next.

Ablation Study:

To quantify how much the lexical cues contribute, six variants were compared on every dataset: the full Bi-LSTM-LC; Bi-LSTM-LC without the lexical-cue branch (the cue input, the 16-unit cue projection, and the concatenation removed, so that the pooled sequence representation z feeds the dense layers directly); and four leave-one-cue-out variants in which one of the four cues (sarcastic phrases, emojis, punctuation patterns, or interjections) is removed from the cue vector of Eq. (2). Every other component, the preprocessing, the data split and folds, and the training configuration (the learning rate and number of epochs selected by cross-validation for the full model) were kept identical, so that any difference can be attributed to the removed cues. Each variant was evaluated by stratified 5-fold cross-validation on the training set and once on the held-out test set. The full model was compared with each variant using the fold-wise differences in macro-F1 and an exact McNemar test on the paired test-set predictions (two-sided, $\alpha = 0.05$).

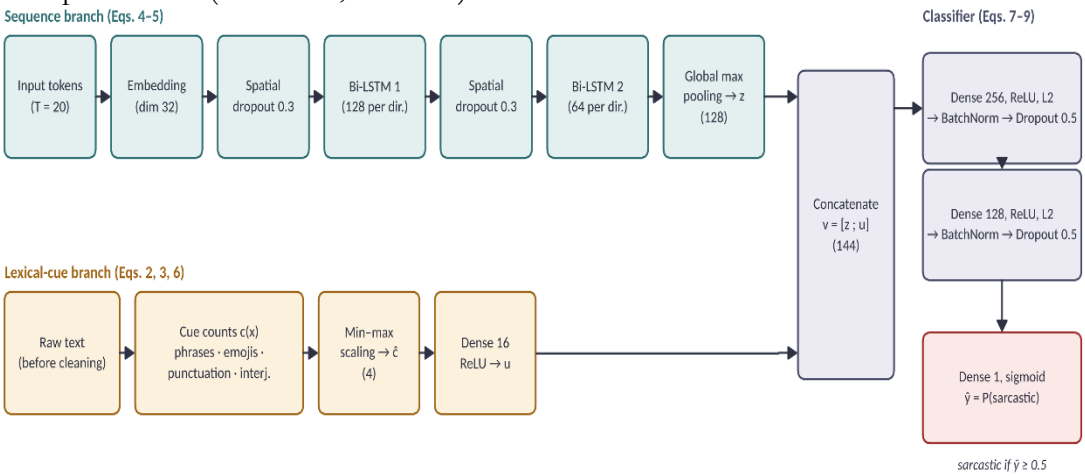


Figure 3. Architecture of Bi-LSTM-LC (sequence branch, lexical-cue branch, and classifier)

Table 3. Hyperparameters of the proposed Bi-LSTM-LC Model

Hyperparameter	Value
Maximum input sequence length	20
Embedding dimension	32
Bi-LSTM units (layers 1, 2; per direction)	128, 64
Spatial dropout rate (after embedding and after Bi-LSTM layer 1)	0.3
Dense layer units	256, 128
Dropout rate (after each dense layer)	0.5
L2 regularization factor (dense layers)	0.01
Epochs	1–20, selected by 5-fold CV
Batch size	64
Activation function (hidden layers)	ReLU
Activation function (output layer)	Sigmoid
Loss function	Binary cross-entropy
Learning rate	10^{-3} or 10^{-4}, selected by 5-fold CV
Optimizer	Adam
Lexical-cue dense layer units	16

Phrase-mining parameters (Eq. 1)	df_s ≥ 5; proportion ≥ 0.75; top 200; n = 1–3
Decision threshold	0.5
Random seed	42
Cross-validation	Stratified 5-fold, training set only (selection criterion: mean macro-F1)

Performance Evaluation:

The trained model was evaluated on the held-out test set using accuracy, precision, recall, and F1-score (Eqs. 10–12), together with balanced accuracy and the Matthews correlation coefficient (MCC) (Eqs. 13–14), which remain informative under class imbalance. Confusion matrices are presented to show the distribution of correct and incorrect predictions per class.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad (10)$$

$$\text{Precision} = TP / (TP + FP),\ \text{Recall} = TP / (TP + FN) \quad (11)$$

$$F1 = 2 \times \text{Precision} \times \text{Recall} / (\text{Precision} + \text{Recall}) \quad (12)$$

$$\text{Balanced accuracy} = \tfrac{1}{2}\,[TP / (TP + FN) + TN / (TN + FP)] \quad (13)$$

$$\text{MCC} = (TP \cdot TN - FP \cdot FN) / \sqrt{[(TP + FP)(TP + FN)(TN + FP)(TN + FN)]} \quad (14)$$

Precision, recall, and F1 were computed for each class and macro-averaged across the two classes for the results reported in Tables 4–7. For binary classification, balanced accuracy equals macro-averaged recall. Sarcastic-class precision, recall, and F1 are reported separately for every dataset, and the cross-validation results on the training sets are reported as mean ± standard deviation over the five folds (Table 9).

Results and Discussion:

Performance Comparison Across Datasets:

Tables 4–7 compare Bi-LSTM-LC with four classical classifiers (Decision Tree, Nearest Centroid, Complement NB, and Multinomial NB) and the Bi-LSTM baseline on the held-out test set of each dataset. All values were recomputed with the corrected protocol described in Materials and Methods, with each model in the configuration selected by cross-validation on the training set. For each dataset, the tables report accuracy, balanced accuracy, macro-averaged precision, recall, and F1, the precision, recall, and F1 of the sarcastic class, and the MCC; the best value in each column is shown in bold, and the majority-class accuracy is given for reference. Table 9 reports the cross-validation results on the training sets, Table 10 reports a controlled comparison in which both neural models use identical fixed training settings, and Table 11 reports an ablation study of the lexical cues.

Table 4. Performance on English news headlines and English tweets (held-out test sets, %; MCC as coefficient)

Model	Acc.	Bal. acc.	Macro P	Macro R	Macro F1	Sarc. P	Sarc. R	Sarc. F1	MCC
English news headlines (test n = 5,701; majority-class accuracy = 52.4%)									
Decision Tree	71.8	71.7	71.7	71.7	71.7	70.6	69.6	70.1	0.434
Nearest Centroid	75.6	75.4	75.7	75.4	75.5	76.1	71.0	73.5	0.511
Complement NB	**80.5**	**80.5**	**80.5**	**80.5**	**80.5**	79.4	79.9	79.6	**0.610**
Multinomial NB	80.4	80.2	80.5	80.2	80.3	**81.3**	76.4	78.8	0.607
Bi-LSTM (baseline)	79.7	79.8	79.8	79.8	79.7	76.4	**82.9**	79.5	0.596
Proposed Bi-LSTM-LC	79.8	80.0	79.9	80.0	79.8	76.7	82.8	**79.6**	0.599
English tweets (test n = 924; majority-class accuracy = 51.8%)									

Model	Acc.	Bal. acc.	Macro P	Macro R	Macro F1	Sarc. P	Sarc. R	Sarc. F1	MCC
Decision Tree	60.5	60.3	60.4	60.3	60.3	59.6	55.7	57.6	0.207
Nearest Centroid	63.6	63.2	**64.0**	63.2	62.9	**65.6**	51.5	57.7	0.272
Complement NB	63.1	63.1	63.1	63.1	63.1	61.5	62.7	**62.1**	0.261
Multinomial NB	**63.9**	**63.6**	63.8	**63.6**	**63.6**	63.7	58.0	60.7	**0.275**
Bi-LSTM (baseline)	60.2	60.4	60.4	60.4	60.1	57.6	65.4	61.3	0.208
Proposed Bi-LSTM-LC	57.6	57.9	58.0	57.9	57.4	55.0	**65.8**	59.9	0.159

Table 5. Performance on Arabic tweets (ArSarcasm; held-out test set)

Model	Acc.	Bal. acc.	Macro P	Macro R	Macro F1	Sarc. P	Sarc. R	Sarc. F1	MCC
Arabic tweets (ArSarcasm) (test n = 2,094; majority-class accuracy = 84.1%)									
Decision Tree	80.1	59.7	61.1	59.7	60.3	35.1	29.8	32.2	0.208
Nearest Centroid	76.8	69.0	63.3	69.0	64.7	35.7	**57.5**	44.1	0.318
Complement NB	79.9	**69.6**	65.5	**69.6**	66.9	40.1	54.5	**46.2**	0.348
Multinomial NB	**85.0**	58.8	**71.9**	58.8	60.9	**57.1**	20.5	30.2	0.277
Bi-LSTM (baseline)	83.6	62.6	67.6	62.6	64.3	47.3	31.9	38.1	0.298
Proposed Bi-LSTM-LC	83.0	67.0	67.8	67.0	**67.4**	46.2	43.7	44.9	**0.349**

Table 6. Performance on Urdu tweets (held-out test set)

Model	Acc.	Bal. acc.	Macro P	Macro R	Macro F1	Sarc. P	Sarc. R	Sarc. F1	MCC
Urdu tweets (test n = 2,914; majority-class accuracy = 59.4%)									
Decision Tree	77.2	76.2	76.4	76.2	76.3	72.3	71.0	71.6	0.526
Nearest Centroid	76.6	77.0	76.1	77.0	76.2	68.3	78.8	73.2	0.530
Complement NB	78.9	78.3	78.1	78.3	78.2	73.2	75.5	74.4	0.564
Multinomial NB	79.2	77.8	78.7	77.8	78.2	**76.5**	70.5	73.4	0.565
Bi-LSTM (baseline)	**80.3**	**80.1**	**79.5**	**80.1**	**79.8**	73.9	**79.4**	**76.6**	**0.597**
Proposed Bi-LSTM-LC	78.5	78.6	77.9	78.6	78.1	71.0	**79.4**	75.0	0.565

Table 7. Performance on Persian tweets (held-out test set)

Model	Acc.	Bal. acc.	Macro P	Macro R	Macro F1	Sarc. P	Sarc. R	Sarc. F1	MCC
Persian tweets (test n = 2,985; majority-class accuracy = 53.1%)									
Decision Tree	66.9	66.9	66.8	66.9	66.8	64.1	67.0	65.5	0.337
Nearest Centroid	70.1	70.9	72.0	70.9	69.9	63.8	**83.9**	72.5	0.429
Complement NB	75.6	76.0	**76.1**	76.0	75.6	70.6	82.3	**76.0**	**0.521**
Multinomial NB	**75.8**	**76.0**	75.9	**76.0**	**75.8**	**72.0**	79.3	75.5	0.520
Bi-LSTM (baseline)	74.8	75.2	75.3	75.2	74.8	69.8	81.5	75.2	0.505

Model	Acc.	Bal. acc.	Macro P	Macro R	Macro F1	Sarc. P	Sarc. R	Sarc. F1	MCC
Proposed Bi-LSTM-LC	71.5	72.2	73.1	72.2	71.3	65.3	83.8	73.4	0.452

Three findings stand out. First, the TF-IDF Naive Bayes classifiers were the strongest models on English news headlines, English tweets and Persian tweets, where Complement or Multinomial NB achieved the highest macro-F1, and the tuned Bi-LSTM baseline was the best model on Urdu tweets; Bi-LSTM-LC achieved the highest macro-F1 only on Arabic tweets (ArSarcasm). Linear models over word n-grams are known to be strong baselines for short-text classification, particularly when, as here, the neural models are trained from randomly initialised embeddings on corpora of a few thousand to a few tens of thousands of instances. Second, against the tuned Bi-LSTM baseline the macro-F1 differences of Bi-LSTM-LC were small and inconsistent in direction (English news headlines: 79.8% vs. 79.7% (+0.2 points); English tweets: 57.4% vs. 60.1% (−2.7 points); Arabic tweets (ArSarcasm): 67.4% vs. 64.3% (+3.1 points); Urdu tweets: 78.1% vs. 79.8% (−1.7 points); Persian tweets: 71.3% vs. 74.8% (−3.5 points)). Third, the absolute scores are considerably lower than those in the original submission; the earlier values were inflated by overlap between training and test data, which has now been removed.

Arabic (Table 5):

ArSarcasm is the most difficult dataset because only 15.8% of its tweets are sarcastic, so a classifier that always predicts the majority class already reaches 84.1% accuracy. Accuracy is therefore uninformative here: Multinomial NB obtained the highest accuracy (85.0%) while detecting only 20.5% of the sarcastic tweets (balanced accuracy 58.8%). Bi-LSTM-LC obtained the highest macro-F1 (67.4%) and MCC (0.349), while Complement NB obtained the highest balanced accuracy (69.6%) and sarcastic-class F1 (46.2% vs. 44.9% for Bi-LSTM-LC). Bi-LSTM-LC recovered 145 of 332 sarcastic test tweets, compared with 106 for the tuned Bi-LSTM baseline (Figure 4c), but more than half of the sarcastic tweets were still missed by every neural model. Four factors explain these weak results. (i) The loss is unweighted and the decision threshold is fixed at 0.5, so the severe imbalance pushes predictions towards the majority class. (ii) Only 22 n-grams satisfied Eq. (1), and many of them are political campaign hashtags or insults tied to specific events rather than general sarcasm markers; the phrase cue fired in 11.7% of the sarcastic and 0.4% of the non-sarcastic training tweets, so it is precise but has low coverage. (iii) ArSarcasm mixes Modern Standard Arabic with several dialects, which fragments the vocabulary of a model trained without pre-trained Arabic representations. (iv) Much Arabic sarcasm in this corpus is expressed through world knowledge and political context without any surface marker, which lexical cues cannot capture. Remedies are discussed in the Research Implications section.

Figure 4 shows the confusion matrices of Bi-LSTM-LC on the held-out test sets of all five datasets. The error profile differs by dataset: English news headlines (a): 22.8% of non-sarcastic instances labelled sarcastic and 17.2% of sarcastic instances missed; English tweets (b): 50.1% of non-sarcastic instances labelled sarcastic and 34.2% of sarcastic instances missed; Arabic tweets (ArSarcasm) (c): 9.6% of non-sarcastic instances labelled sarcastic and 56.3% of sarcastic instances missed; Urdu tweets (d): 22.2% of non-sarcastic instances labelled sarcastic and 20.6% of sarcastic instances missed; Persian tweets (e): 39.5% of non-sarcastic instances labelled sarcastic and 16.2% of sarcastic instances missed. The model over-predicts sarcasm on English news headlines, English tweets and Persian tweets; it under-predicts sarcasm on Arabic tweets (ArSarcasm), where the sarcastic class is the minority. Errors on the remaining datasets are approximately balanced between the classes.

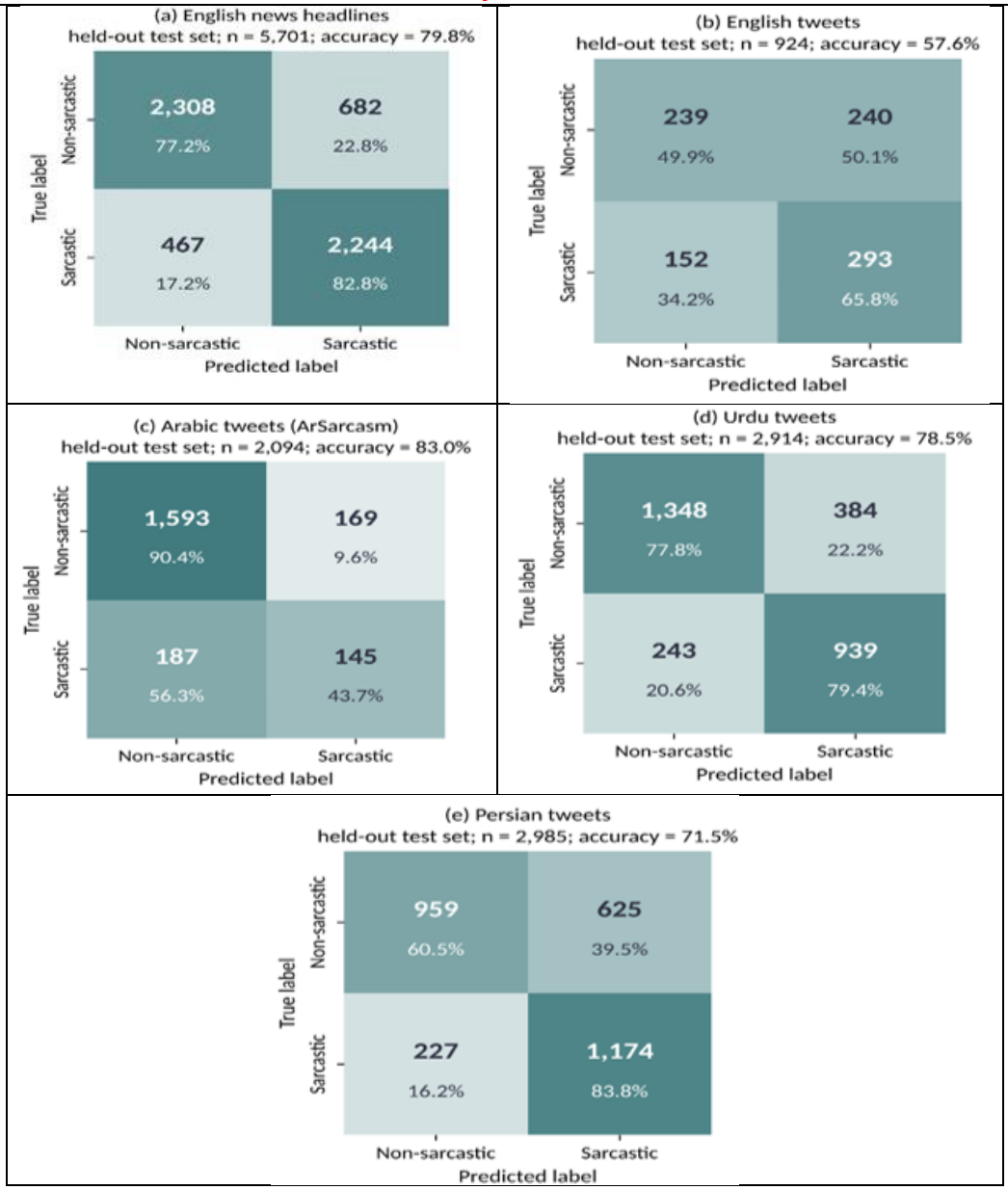


Figure 4. Confusion matrices of Bi-LSTM-LC for (a) English news headlines, (b) English tweets, (c) Arabic tweets (ArSarcasm), (d) Urdu tweets, (e) Persian tweets. Each cell shows the number of instances and the percentage of the true class (row-normalized). All panels are computed on the held-out test sets (n = 5,701, 924, 2,094, 2,914, 2,985, respectively).

Comparison with Previous Studies:

Table 8 compares Bi-LSTM-LC with previously published results on the same datasets. Because published studies differ in dataset version, preprocessing, data split, and evaluation metric, Table 8 records these conditions and indicates whether each comparison is direct or only indicative. To make direct comparisons possible, two additional experiments were run on the official train/test splits released with ArSarcasm and with SemEval-2018 Task 3, using the same cross-validation-based model selection on the official training sets. On News Headlines v2, [21]reported 89.7% accuracy with an attention-based hybrid network using pre-trained Word2Vec embeddings, whereas Bi-LSTM-LC reached 79.8% with randomly initialised embeddings, stop-word removal, and a 20-token input; the comparison is indicative because the splits differ, but it suggests that pre-trained embeddings contribute substantially on this dataset. The results reported for News Headlines v1 [20] and for combinations of the dataset

with other corpora [27][41] are not comparable with ours. On the official ArSarcasm split, Bi-LSTM-LC obtained a sarcastic-class F1 of 0.39 (precision 0.44, recall 0.36), compared with 0.46 (precision 0.62, recall 0.38) for the Bi-LSTM of [45], 0.42 for our tuned Bi-LSTM baseline, and 0.46 for Complement NB; on this split, Bi-LSTM-LC therefore did not improve on the published Bi-LSTM. On the official SemEval-2018 Task 3 split, the ironic-class F1 of Bi-LSTM-LC (0.516) and of our Bi-LSTM baseline (0.583) were below both the official unigram SVM baseline (0.589) and the best participating system (0.705) [44], confirming that irony in tweets depends heavily on context and world knowledge that surface cues capture only partially. On the Urdu Sarcastic Tweets dataset, [33]reported accuracies between 0.81 and 0.89 for classical classifiers with textual features in different parts of their paper, with an 80:20 split; Bi-LSTM-LC reached 78.5% on our stratified split, which is indicative only. For the Persian Irony Detection dataset, the dataset authors report 84.7% accuracy and 84.6% F1 with a fine-tuned XLM-RoBERTa-Large model on their own train/test split [47]Bi-LSTM-LC reached 71.5% accuracy on our split, and the gap reflects the advantage of large pre-trained multilingual encoders over models trained from scratch. Overall, these comparisons should be interpreted cautiously because the studies differ in dataset versions, preprocessing procedures, and experimental settings.

Table 8. Comparison of the proposed model with previously published results on the same datasets under stated experimental conditions

Study	Dataset (version)	Split / evaluation	Model	Reported performance	Comparability
[21]	News Headlines v2	Own split (not stated in detail)	Hybrid NN with attention, Word2Vec	Acc. 89.7%	Indicative (same version, different split)
[20]	News Headlines v1 + Sarcasm Corpus V2	70:30 split	GMP-LSTM	Acc. 92.54%	Not comparable (different data)
[27]	News Headlines (version not stated)	Not stated	Multi-layer Bi-LSTM + GloVe	Acc. 95.41%	Indicative
Proposed Bi-LSTM-LC	News Headlines v2 (deduplicated)	Stratified 80/20; CV on training set	Bi-LSTM + lexical cues	Acc. 79.8%; macro-F1 79.8%	—
[45]	ArSarcasm	Official 80/20 split	Bi-LSTM	Sarcastic-class F1 0.46 (P 0.62, R 0.38)	Direct (official split row below)
Proposed Bi-LSTM-LC	ArSarcasm (deduplicated)	Official split; CV on official training set	Bi-LSTM + lexical cues	Sarcastic-class F1 0.39 (P 0.44, R 0.36)	Direct
Proposed Bi-LSTM-LC	ArSarcasm (deduplicated)	Stratified 80/20; CV on training set	Bi-LSTM + lexical cues	Acc. 83.0%; macro-F1 67.4%; sarcastic-class F1 0.45	—
[44]	SemEval-2018 Task 3 (Task A)	Official split (3,834 / 784)	Unigram SVM baseline; best system (THU_NGN)	Ironic-class F1 0.589; 0.705	Direct

Study	Dataset (version)	Split / evaluation	Model	Reported performance	Comparability
Proposed Bi-LSTM-LC	SemEval-2018 Task 3 (Task A)	Official split; CV on official training set	Bi-LSTM + lexical cues	Ironic-class F1 0.516	Direct
[33]	Urdu Sarcastic Tweets	80:20 split	Classical classifiers + textual features	Acc. 0.81–0.89 (best LR/RF/SVM)	Indicative
Proposed Bi-LSTM-LC	Urdu Sarcastic Tweets (deduplicated)	Stratified 80/20; CV on training set	Bi-LSTM + lexical cues	Acc. 78.5%; macro-F1 78.1%	—
[47]	Persian Irony Detection	Authors' train/test split	Fine-tuned XLM-RoBERTa-Large	Acc. 84.7%; F1 84.6%	Indicative
Proposed Bi-LSTM-LC	Persian Irony Detection (deduplicated)	Stratified 80/20; CV on training set	Bi-LSTM + lexical cues	Acc. 71.5%; macro-F1 71.3%	—

Cross-Validation and Effect of Hyperparameter Selection:

Table 9 reports the cross-validation performance on the training sets at the configuration selected for each model (macro-F1 of Bi-LSTM-LC vs. the Bi-LSTM baseline, mean ± SD over five folds: English news headlines: 79.4 ± 0.8 vs. 80.4 ± 0.7; English tweets: 60.3 ± 1.2 vs. 60.5 ± 1.0; Arabic tweets (ArSarcasm): 63.4 ± 3.4 vs. 63.5 ± 2.5; Urdu tweets: 79.5 ± 0.9 vs. 79.6 ± 1.1; Persian tweets: 72.5 ± 1.9 vs. 75.0 ± 0.9). In cross-validation, Bi-LSTM-LC did not exceed the Bi-LSTM baseline on any dataset, and on ArSarcasm the two were equal within one standard deviation; the advantage of Bi-LSTM-LC on the ArSarcasm test set should therefore be interpreted with caution. The configurations selected by cross-validation were: English news headlines: learning rate 0.0001, 13 epochs (Bi-LSTM baseline: learning rate 0.001, 2 epochs); English tweets: learning rate 0.001, 8 epochs (Bi-LSTM baseline: learning rate 0.0001, 12 epochs); Arabic tweets (ArSarcasm): learning rate 0.0001, 20 epochs (Bi-LSTM baseline: learning rate 0.0001, 9 epochs); Urdu tweets: learning rate 0.0001, 9 epochs (Bi-LSTM baseline: learning rate 0.001, 1 epoch); Persian tweets: learning rate 0.0001, 10 epochs (Bi-LSTM baseline: learning rate 0.001, 1 epoch). On English news headlines, Urdu tweets and Persian tweets, the Bi-LSTM baseline reached its best validation performance after only one or two epochs at the larger learning rate, whereas Bi-LSTM-LC, which has more parameters, preferred the smaller learning rate and more epochs.

Table 9. Stratified 5-fold cross-validation results on the training sets at the selected configurations (mean ± SD, %)

Dataset	Model	Accuracy	Balanced accuracy	Macro F1	Sarcastic F1
English news headlines	Bi-LSTM (baseline)	80.5 ± 0.7	80.3 ± 0.7	80.4 ± 0.7	78.9 ± 1.2
	Proposed Bi-LSTM-LC	79.4 ± 0.8	79.4 ± 0.8	79.4 ± 0.8	78.4 ± 0.9
English tweets	Bi-LSTM (baseline)	60.8 ± 0.9	60.9 ± 0.9	60.5 ± 1.0	60.9 ± 3.8
	Proposed Bi-LSTM-LC	61.4 ± 1.0	60.9 ± 1.1	60.3 ± 1.2	54.7 ± 4.4
Arabic tweets (ArSarcasm)	Bi-LSTM (baseline)	82.1 ± 1.5	62.5 ± 2.5	63.5 ± 2.5	37.4 ± 4.6

Dataset	Model	Accuracy	Balanced accuracy	Macro F1	Sarcastic F1
	Proposed Bi-LSTM-LC	79.0 ± 6.5	64.7 ± 2.9	63.4 ± 3.4	39.7 ± 4.3
Urdu tweets	Bi-LSTM (baseline)	80.4 ± 0.8	79.6 ± 1.5	79.6 ± 1.1	75.7 ± 2.2
	Proposed Bi-LSTM-LC	80.2 ± 0.7	79.5 ± 1.1	79.5 ± 0.9	75.6 ± 1.4
Persian tweets	Bi-LSTM (baseline)	75.1 ± 0.9	75.2 ± 0.9	75.0 ± 0.9	74.5 ± 1.1
	Proposed Bi-LSTM-LC	72.7 ± 1.6	73.0 ± 1.2	72.5 ± 1.9	72.9 ± 0.9

Before model selection was introduced, both neural models were trained with the same fixed settings (learning rate 10^{-4}, 20 epochs), with all other aspects of the protocol unchanged. Table 10 reports this controlled comparison. Under identical settings, Bi-LSTM-LC outperformed the Bi-LSTM baseline in test macro-F1 on four of the five datasets (English news headlines: 79.2% vs. 77.1% (+2.1 points); English tweets: 59.2% vs. 56.8% (+2.5 points); Arabic tweets (ArSarcasm): 67.4% vs. 64.6% (+2.8 points); Urdu tweets: 72.6% vs. 73.1% (−0.5 points); Persian tweets: 71.8% vs. 70.6% (+1.3 points)), and in cross-validation on four of the five datasets. When each model was instead given the learning rate and number of epochs that maximise its own cross-validation performance, the Bi-LSTM baseline improved considerably (e.g., on Persian and on the news headlines) and the advantage of Bi-LSTM-LC disappeared on most datasets. The gain observed under fixed settings therefore reflects mainly the fact that the fixed schedule suited the larger Bi-LSTM-LC model better than the small baseline, rather than an intrinsic benefit of the lexical cues. This finding illustrates why baselines must be tuned with the same effort as the proposed model. Because the two models also differ in architecture, Table 10 cannot isolate the contribution of the cues; this is the purpose of the ablation study below.

Table 10. Controlled comparison of the two neural models with identical fixed training settings (learning rate 10^{-4}, 20 epochs; %)

Dataset	Model	Test macro F1	Test bal. acc.	Test sarc. F1	CV macro F1 (mean ± SD)
English news headlines	Bi-LSTM (baseline)	77.1	77.1	74.3	77.0 ± 0.2
	Proposed Bi-LSTM-LC	79.2	79.2	78.2	78.4 ± 0.9
English tweets	Bi-LSTM (baseline)	56.8	57.0	58.4	59.5 ± 2.6
	Proposed Bi-LSTM-LC	59.2	59.7	61.9	58.8 ± 2.2
Arabic tweets (ArSarcasm)	Bi-LSTM (baseline)	64.6	63.2	39.0	62.3 ± 1.1
	Proposed Bi-LSTM-LC	67.4	67.0	44.9	63.4 ± 3.4
Urdu tweets	Bi-LSTM (baseline)	73.1	73.9	70.0	75.7 ± 0.6
	Proposed Bi-LSTM-LC	72.6	74.9	72.1	77.7 ± 0.9
Persian tweets	Bi-LSTM (baseline)	70.6	70.7	69.7	68.5 ± 1.5
	Proposed Bi-LSTM-LC	71.8	72.3	72.6	71.2 ± 1.4

Ablation Study:

Table 11 reports the ablation study described in Materials and Methods. For each dataset, the full Bi-LSTM-LC was compared with five variants that differ from it only in the lexical-cue input: one without the cue branch and four in which a single cue was removed. All variants were trained with the learning rate and number of epochs selected for the full model (Table 9) and were evaluated on the same cross-validation folds and the same held-out test

set; Δ denotes the macro-F1 of the full model minus that of the variant, so positive values indicate that the removed cues helped.

Table 11: Ablation study of the lexical cues (macro-F1, %; Δ = full model − variant, in percentage points; McNemar test on the paired test-set predictions)

Dataset	Variant	Test macro F1	Δ test (full − variant)	CV macro F1 (mean ± SD)	Δ CV (mean ± SD)	McNemar p
English news headlines	**Full Bi-LSTM-LC**	**79.8**	—	79.4 ± 0.8	—	—
	w/o all lexical cues	78.9	+0.9	78.8 ± 1.3	+0.6 ± 0.7	0.030
	w/o sarcastic phrases	79.7	+0.2	79.3 ± 0.4	+0.1 ± 0.5	0.610
	w/o emojis	79.3	+0.5	79.2 ± 0.8	+0.2 ± 0.1	0.029
	w/o punctuation patterns	79.4	+0.5	78.7 ± 0.4	+0.7 ± 0.7	0.035
	w/o interjections	79.5	+0.3	79.1 ± 0.8	+0.3 ± 0.1	0.152
English tweets	**Full Bi-LSTM-LC**	**57.4**	—	60.3 ± 1.2	—	—
	w/o all lexical cues	57.5	−0.1	54.9 ± 8.6	+5.4 ± 8.3	1.000
	w/o sarcastic phrases	58.1	−0.7	56.3 ± 2.9	+4.0 ± 3.7	0.650
	w/o emojis	55.8	+1.7	56.7 ± 5.8	+3.6 ± 6.4	0.516
	w/o punctuation patterns	58.4	−1.0	57.8 ± 2.5	+2.5 ± 2.9	0.456
	w/o interjections	58.3	−0.9	57.7 ± 4.4	+2.6 ± 4.4	0.582
Arabic tweets (ArSarcasm)	**Full Bi-LSTM-LC**	**67.4**	—	63.4 ± 3.4	—	—
	w/o all lexical cues	55.8	+11.7	56.0 ± 4.9	+7.4 ± 6.3	0.006
	w/o sarcastic phrases	65.2	+2.2	62.2 ± 3.1	+1.2 ± 1.8	< 0.001
	w/o emojis	63.9	+3.6	61.1 ± 4.0	+2.2 ± 4.1	< 0.001
	w/o punctuation patterns	65.7	+1.7	62.9 ± 2.8	+0.5 ± 2.0	< 0.001
	w/o interjections	67.7	−0.3	63.9 ± 2.2	−0.6 ± 1.5	0.228
Urdu tweets	**Full Bi-LSTM-LC**	**78.1**	—	79.5 ± 0.9	—	—
	w/o all lexical cues	78.7	−0.7	77.0 ± 2.1	+2.4 ± 1.7	0.061
	w/o sarcastic phrases	75.8	+2.3	77.4 ± 1.4	+2.0 ± 0.8	< 0.001
	w/o emojis	78.0	+0.0	79.1 ± 1.0	+0.3 ± 0.9	1.000
	w/o punctuation patterns	78.6	−0.5	79.6 ± 1.0	−0.1 ± 0.5	0.103
	w/o interjections	77.9	+0.2	79.3 ± 1.0	+0.2 ± 0.3	0.755
Persian tweets	**Full Bi-LSTM-LC**	**71.3**	—	72.5 ± 1.9	—	—

Dataset	Variant	Test macro F1	Δ test (full − variant)	CV macro F1 (mean ± SD)	Δ CV (mean ± SD)	McNemar p
	w/o all lexical cues	73.2	−1.9	72.0 ± 1.0	+0.5 ± 1.6	0.017
	w/o sarcastic phrases	72.4	−1.0	72.1 ± 1.1	+0.4 ± 2.0	0.150
	w/o emojis	71.0	+0.3	72.5 ± 1.2	+0.0 ± 1.0	0.498
	w/o punctuation patterns	70.4	+0.9	72.3 ± 1.0	+0.2 ± 1.1	0.161
	w/o interjections	71.9	−0.6	73.1 ± 0.7	−0.6 ± 1.7	0.165

Removing all cues: Removing the complete cue branch lowered the mean cross-validation macro-F1 on all five datasets and the test macro-F1 on two of the five datasets. The test-set difference was statistically significant in favour of the full model on English news headlines and Arabic tweets (ArSarcasm) and in favour of the variant without cues on Persian tweets. The effect was by far the largest on ArSarcasm: without the cue branch, the model recovered only 38 of the 332 sarcastic test tweets instead of 145 (sarcastic-class recall 11.4% vs. 43.7%; balanced accuracy 55.3% vs. 67.0%), so on this imbalanced corpus the cue branch is what prevents the model from collapsing towards the majority class. An independent repetition of the ArSarcasm ablation on a different platform (CPU instead of GPU) reproduced the drop in macro-F1 (+11.6 points) and in sarcastic-class recall (31.6% vs. 9.3%); in that run, however, the overall error rate, which McNemar's test compares and which is dominated by the majority class, did not differ ($p = 1.000$). On the news headlines, removing the cues lowered macro-F1 by 0.9 points, a small difference (see the control below). On Urdu tweets, the full model was better in all five folds (+2.4 points) but not on the test set (−0.7 points, not significant); on English tweets, the fold-wise differences were highly variable (SD 8.3 points) and the test-set difference was negligible; and on Persian tweets, the variant without cues was better on the test set (−1.9 points, $p = 0.017$), although not in cross-validation (+0.5 points), so on this noisily labelled corpus the cues appear to be redundant with, or slightly harmful to, the sequence representation.

Removing individual cues: On ArSarcasm, removing emojis, sarcastic phrases and punctuation patterns each lowered the test macro-F1 significantly (by 3.6, 2.2 and 1.7 points, respectively), mainly by lowering the sarcastic-class recall (to 26.5%, 30.4% and 29.8%, vs. 43.7% for the full model); the cues therefore complement each other, and none of them alone accounts for the effect of the complete branch. Interjections, which occur in only 4.3% of the sarcastic Arabic training tweets, had no significant effect. In the CPU repetition, these three differences had the same sign but were smaller (0.7 to 1.1 points) and not significant, and removing interjections raised the macro-F1 (−1.1 points); the ranking of the individual cues on ArSarcasm is therefore less certain than the effect of the complete branch. On Urdu, sarcastic phrases were the only cue with a significant effect, which is consistent with their coverage: the mined Urdu phrases occur in 71.6% of the sarcastic and 12.3% of the non-sarcastic training tweets. The news headlines provide a useful control: emojis occur in almost none of the headlines, so the variant without emojis differs from the full model essentially only in its random weight initialisation (the cue projection has one input fewer). Nevertheless, it differed from the full model by +0.5 points on the test set ($p = 0.029$). Differences of about half a point, even when nominally significant, can therefore arise from initialisation alone, and the small effects observed on the news headlines and English tweets should not be read as effects of the cues.

Interpretation: The ablation shows that the lexical cues are not uniformly useful, but help where the sequence model alone is weakest: on the imbalanced Arabic corpus they are essential

for detecting the sarcastic minority class, and on Urdu the mined phrase list carries real signal. On the English datasets and on Persian, the cues did not improve on the same architecture without them. The ablation also clarifies the comparison with the Bi-LSTM baseline: on ArSarcasm, the Bi-LSTM-LC architecture without its cue branch performed worse than the simple tuned Bi-LSTM (55.8% vs. 64.3% macro-F1), so the advantage of Bi-LSTM-LC on this dataset is due to the cues rather than to its additional layers, whereas on the other datasets the tuned simple baseline was comparable to or better than both variants. Because every variant was trained with the configuration selected for the full model and a single random seed, the ablation estimates the contribution of the cues within that configuration; it does not exclude that a variant would perform better with separately tuned settings.

Limitations:

The study has several limitations. First, the datasets differ substantially in size, domain, language, annotation method, and class distribution, which limits direct performance comparisons across languages; in particular, the Persian labels were assigned automatically from user reactions and are therefore noisier than the manually annotated corpora. Second, Bi-LSTM-LC differs from the simple Bi-LSTM baseline in several architectural components in addition to the lexical-cue branch; the comparison with the Bi-LSTM baseline (Tables 4–7 and 10) therefore reflects the complete architecture, and the contribution of the cues is assessed only by the ablation study (Table 11), which used a single training configuration per dataset (the one selected for the full model) and a single random seed; as the emoji control on the news headlines shows, differences of about half a macro-F1 point between variants are within run-to-run variability. Third, lexical resources such as sarcastic phrase lists and manually compiled interjections may be dataset- and language-dependent and may not generalize equally well to unseen domains; the mined phrase lists contain topical n-grams (e.g., named entities in news headlines and political hashtags in ArSarcasm). Fourth, the fixed input length of 20 tokens truncates many Persian tweets (only 42.8% of the Persian training tweets have at most 20 tokens after preprocessing), and no pre-trained embeddings were used, which disadvantages both neural models relative to linear n-gram classifiers. Fifth, the hyperparameter search covered only the learning rate and number of epochs, and each final model was trained with a single random seed; the cross-validation standard deviations in Table 9 indicate the expected variability. Finally, the ArSarcasm results show that severe class imbalance remains a significant limitation, particularly for minority-class detection.

Research Implications:

Practical applications: Bi-LSTM-LC is a compact model (two Bi-LSTM layers, no pre-trained encoder) whose lexical-cue branch is transparent: for every prediction, the four cue counts can be inspected, and the language-specific phrase lists produced by Eq. (1) show analysts how sarcasm is expressed in a given community. Such models, and especially the TF-IDF Naive Bayes classifiers that performed best here, are inexpensive sarcasm filters that can be placed in front of sentiment-analysis and opinion-mining pipelines, used to flag potentially sarcastic customer feedback or reviews for human review, or used in content moderation for languages in which large pre-trained models are unavailable or too costly to deploy.

Implications for Evaluation Practice: The corrected experiments show that leakage between training and test data can inflate reported accuracy by more than 20 percentage points, and that a proposed model can appear superior simply because its baseline was trained with unsuitable settings (Table 10). We therefore recommend that sarcasm-detection studies (i) deduplicate datasets and fit all preprocessing on training data only, (ii) tune baselines with the same procedure and budget as the proposed model, (iii) include linear n-gram baselines, and (iv) report results on official splits where they exist.

Recommendations for Class Imbalance: On imbalanced corpora such as ArSarcasm, minority-class detection, not accuracy, should drive model design. The ablation study (Table

11) shows that explicit cues that are precise for the minority class are an inexpensive way to raise minority-class recall: on ArSarcasm, the cue branch raised sarcastic-class recall from 11.4% to 43.7%. We further recommend class-weighted or focal loss functions or resampling of the training folds, selecting the decision threshold on the validation folds rather than fixing it at 0.5, and always reporting balanced accuracy, MCC, and sarcastic-class precision, recall, and F1 alongside accuracy and the majority-class rate.

Recommendations for Model Accuracy: The strong performance of the linear classifiers indicates that the neural models are limited by corpus size and randomly initialised embeddings. Initialising the embedding layer with pre-trained multilingual or language-specific representations (e.g., fastText, AraBERT, UrduBERT, or ParsBERT), combining the cue branch with such encoders, increasing the input length for long tweets (notably Persian), and ensembling neural models with linear n-gram classifiers are the most promising directions.

Recommendations for Performance Across Languages: Because the benefit of the cues differed strongly between languages (Table 11), their usefulness should be verified by an ablation for every new language or domain rather than assumed. Cue resources should be built per language from training data, as in Eq. (1), and complemented by language-specific normalisation (e.g., Arabic dialect normalisation and diacritic handling, Urdu and Persian character unification), by dialect-aware interjection lists, and by filters that remove topical named entities and campaign hashtags from the phrase lists. Cross-lingual transfer from high-resource languages and evaluation on official splits with identical metrics would make multilingual results directly comparable.

Conclusion:

This study investigated text-based sarcasm detection across two domains and four languages using classical machine learning models, a Bi-LSTM baseline, and the proposed Bi-LSTM-LC framework. The findings are summarised by research objective. **O1:** all five datasets were evaluated under one leakage-free protocol, with zero overlap between training and test texts and all fitted components learned from training data only. **O2:** Bi-LSTM-LC reached test macro-F1 scores of 79.8% (English news headlines), 57.4% (English tweets), 67.4% (Arabic tweets), 78.1% (Urdu tweets) and 71.3% (Persian tweets). It outperformed the Bi-LSTM baseline on four of the five datasets when both used identical fixed settings, but not consistently once each model was tuned by cross-validation, and TF-IDF Naive Bayes classifiers were the strongest models on English news headlines, English tweets and Persian tweets, and the tuned Bi-LSTM baseline was the best model on Urdu tweets. **O3:** on the imbalanced ArSarcasm corpus, Bi-LSTM-LC achieved the highest macro-F1 (67.4%) but a sarcastic-class F1 of only 44.9%, missed more than half of the sarcastic tweets, and did not exceed the majority-class accuracy. **O4:** on the official ArSarcasm and SemEval-2018 splits, Bi-LSTM-LC did not improve on the published Bi-LSTM and SVM baselines. **O5:** the architecture-matched ablation (Table 11) showed that the cues are essential on ArSarcasm, where removing them lowered the test macro-F1 by 11.7 points and the sarcastic-class recall from 43.7% to 11.4%, and that the mined phrase list was the decisive cue for Urdu (+2.3 points, $p < 0.001$); on the English datasets and on Persian, the cues gave no improvement. In summary, surface lexical cues do not make a Bi-LSTM trained from scratch a better sarcasm detector in general, but they are an effective and interpretable way to improve minority-class detection on imbalanced, low-resource data; the results also underline the importance of leakage-free protocols, equally tuned baselines, ablation studies, and minority-class metrics.

Future work:

The ablation study should be extended by tuning each variant separately and repeating all experiments with several random seeds, so that small effects can be separated from run-to-run variability. A second priority is to test whether the cue branch also benefits models built on pre-trained encoders, and to improve the cues for languages in which they did not help,

for example by context-aware phrase mining for English and Persian and by filtering topical n-grams from the phrase lists.

For Arabic sarcasm detection, where the cues were most useful, future work will combine the cue branch with Arabic pre-trained encoders (e.g., AraBERT or MARBERT), use class-weighted or focal losses and validation-based threshold selection to address the 16% sarcastic rate of ArSarcasm, and build dialect-aware interjection and phrase lists that exclude topical hashtags. Ensemble methods, transfer learning, and attention-based or transformer models should also be evaluated on the same datasets, together with whether lexical cues improve these models, especially on noisy multilingual text; a wider hyperparameter search and longer input sequences for Persian should also be explored. Another direction is to distinguish types of sarcasm, such as dramatic, self-deprecating, brooding, and deadpan sarcasm, which could help infer the writer's mood.

References:

[1] R. W. Gibbs, "On the Psycholinguistics of Sarcasm," *J. Exp. Psychol. Gen.*, vol. 115, no. 1, pp. 3–15, Mar. 1986, doi: 10.1037/0096-3445.115.1.3.

[2] S. Mane and V. Khatavkar, "Researchers eye-view of sarcasm detection in social media textual content," Apr. 2023, doi: 10.1729/Journal.31277.

[3] Ahmed Derbala Yacoub, Amal Elsayed Aboutabl, and Salwa O. Slim, "Multilingual Sarcasm Detection for Enhancing Sentiment Analysis using Deep Learning Algorithms," *J. Commun. Softw. Syst.*, vol. 20, no. 4, 2024, [Online]. Available: https://jcoms.fesb.unist.hr/pdfs/v20n4_2024-0071_Yacoub.pdf

[4] K. Sreelakshmi and P. C. Rafeeque, "An Effective Approach for Detection of Sarcasm in Tweets," *2018 Int. CET Conf. Control. Commun. Comput. IC4 2018*, pp. 377–382, Nov. 2018, doi: 10.1109/CETIC4.2018.8531044.

[5] A. C. Băroiu and Ştefan Trăuşan-Matu, "Automatic Sarcasm Detection: Systematic Literature Review," *Inf. 2022, Vol. 13, Page 399*, vol. 13, no. 8, p. 399, Aug. 2022, doi: 10.3390/INFO13080399.

[6] V. Govindan and V. Balakrishnan, "A machine learning approach in analysing the effect of hyperboles using negative sentiment tweets for sarcasm detection," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 8, pp. 5110–5120, Sep. 2022, doi: 10.1016/J.JKSUCI.2022.01.008.

[7] A. Kumar, V. T. Narapareddy, V. A. Srikanth, A. Malapati, and L. B. M. Neti, "Sarcasm Detection Using Multi-Head Attention Based Bidirectional LSTM," *IEEE Access*, vol. 8, pp. 6388–6397, 2020, doi: 10.1109/ACCESS.2019.2963630.

[8] A. D. Yacoub, S. O. Slim, and A. E. Aboutabl, "A Survey of Sentiment Analysis and Sarcasm Detection: Challenges, Techniques, and Trends," *Int. J. Electr. Comput. Eng. Syst.*, vol. 15, no. 1, pp. 69–78, Jan. 2024, doi: 10.32985/IJECES.15.1.7.

[9] Y. Mansour and A. Elnagar, "Sarcasm Detection in Arabic Text Using Contextualized Models," *Proc. 2023 Int. Conf. Asian Lang. Process. IALP 2023*, pp. 278–283, 2023, doi: 10.1109/IALP61005.2023.10337175.

[10] M. E. Hassan, M. Hussain, I. Maab, U. Habib, M. A. Khan, and A. Masood, "Detection of Sarcasm in Urdu Tweets Using Deep Learning and Transformer Based Hybrid Approaches," *IEEE Access*, vol. 12, pp. 61542–61555, 2024, doi: 10.1109/ACCESS.2024.3393856.

[11] Q. Abuein, R. M. Al-Khatib, A. Migdady, M. S. Jawarneh, and A. Al-Khateeb, "ArSa-Tweets: A novel Arabic sarcasm detection system based on deep learning model," *Heliyon*, vol. 10, no. 17, p. e36892, Sep. 2024, doi: 10.1016/J.HELIYON.2024.E36892.

[12] Y. Zhang, C. Zou, Z. Lian, P. Tiwari, and J. Qin, "SarcasmBench: Towards Evaluating Large Language Models on Sarcasm Understanding," *IEEE Trans. Affect.*

Comput., vol. 16, no. 4, pp. 2560–2578, 2025, doi: 10.1109/TAFFC.2025.3604806.

[13] Q. Yang, L. Yang, J. Zeng, K. Ma, and H. Lin, "Sarcasm-R1: Enhancing Sarcasm Detection through Focused Reasoning," *EMNLP 2025 - 2025 Conf. Empir. Methods Nat. Lang. Process. Find. EMNLP 2025*, pp. 10773–10785, 2025, doi: 10.18653/V1/2025.FINDINGS-EMNLP.570.

[14] R. J. Kreuz and R. M. Roberts, "Two Cues for Verbal Irony: Hyperbole and the Ironic Tone of Voice," *Metaphor Symb.*, vol. 10, no. 1, pp. 21–31, Mar. 1995, doi: 10.1207/S15327868MS1001_3.

[15] S. Rendalkar and C. Chandankhede, "Sarcasm Detection of Online Comments Using Emotion Detection," *Proc. Int. Conf. Inven. Res. Comput. Appl. ICIRCA 2018*, pp. 1244–1249, Dec. 2018, doi: 10.1109/ICIRCA.2018.8597368.

[16] D. Jain, A. Kumar, and G. Garg, "Sarcasm detection in mash-up language using soft-attention based bi-directional LSTM and feature-rich CNN," *Appl. Soft Comput.*, vol. 91, p. 106198, Jun. 2020, doi: 10.1016/J.ASOC.2020.106198.

[17] A. Reyes, P. Rosso, and D. Buscaldi, "From humor recognition to irony detection: The figurative language of social media," *Data Knowl. Eng.*, vol. 74, pp. 1–12, Apr. 2012, doi: 10.1016/J.DATAK.2012.02.005.

[18] L. H. Son, A. Kumar, S. R. Sangwan, A. Arora, A. Nayyar, and M. Abdel-Basset, "Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network," *IEEE Access*, vol. 7, pp. 23319–23328, 2019, doi: 10.1109/ACCESS.2019.2899260.

[19] E. Sharma, N. K. Gondhi, M. Murugappan, G. Naik, and L. Murugan, "Toward Effective Sarcasm Detection in Social Media: A Review of Artificial Intelligence-Based Automated Approaches," *Expert Syst.*, vol. 43, no. 3, p. e70203, Mar. 2026, doi: 10.1111/EXSY.70203;REQUESTEDJOURNAL:JOURNAL:14680394;WGROUP:STRING:PUBLICATION.

[20] R. Ali *et al.*, "Deep Learning for Sarcasm Identification in News Headlines," *Appl. Sci. 2023, Vol. 13, Page 5586*, vol. 13, no. 9, p. 5586, Apr. 2023, doi: 10.3390/APP13095586.

[21] R. Misra and P. Arora, "Sarcasm detection using news headlines dataset," *AI Open*, vol. 4, pp. 13–18, Jan. 2023, doi: 10.1016/J.AIOPEN.2023.01.001.

[22] R. A. Potamias, G. Siolas, and A. G. Stafylopatis, "A transformer-based approach to irony and sarcasm detection," *Neural Comput. Appl. 2020 3223*, vol. 32, no. 23, pp. 17309–17320, Jun. 2020, doi: 10.1007/S00521-020-05102-3.

[23] S. Kaya and B. Alatas, "Sarcasm Detection with A New CNN+BiLSTM Hybrid Neural Network and BERT Classification Model," *Int. J. Adv. Netw. Appl.*, vol. 14, no. 03, pp. 5436–5443, 2022, doi: 10.35444/IJANA.2022.14304.

[24] S. Sinha and V. K. Yadav, "Sarcasm Detection in News Headlines Using Deep Learning," *Int. Conf. Recent Adv. Sci. Eng. Technol. ICRASET 2023*, 2023, doi: 10.1109/ICRASET59632.2023.10420344.

[25] S. Ili´cili´c, E. Marrese-Taylor, J. A. Balazs, and Y. Matsuo, "Deep contextualized word representations for detecting sarcasm and irony," *Brussels*, pp. 2–7, Jun. 2018, doi: 10.18653/V1/W18-6202.

[26] A. Onan and M. A. Tocoglu, "A Term Weighted Neural Language Model and Stacked Bidirectional LSTM Based Framework for Sarcasm Identification," *IEEE Access*, vol. 9, pp. 7701–7722, 2021, doi: 10.1109/ACCESS.2021.3049734.

[27] Z. N. Ardilla, T. I. Sari, N. Hayatin, and C. Fatichah, "Sarcasm detection on news headline using multilayer bidirectional-LSTM with glove embeddings," *AIP Conf. Proc.*, vol. 2927, no. 1, Mar. 2024, doi: 10.1063/5.0192254/3279275.

[28] M. M. Al-Ani, N. Omar, and A. A. Nafea, "A Hybrid Method of Long Short-Term

Memory and Auto-Encoder Architectures for Sarcasm Detection," *J. Comput. Sci.*, vol. 17, no. 11, pp. 1093–1098, Dec. 2021, doi: 10.3844/JCSSP.2021.1093.1098.

[29] M. A. Galal, A. Hassan Yousef, H. H. Zayed, and W. Medhat, "Arabic sarcasm detection: An enhanced fine-tuned language model approach," *Ain Shams Eng. J.*, vol. 15, no. 6, p. 102736, Jun. 2024, doi: 10.1016/J.ASEJ.2024.102736.

[30] A. Israeli, Y. Nahum, S. Fine, and K. Bar, "The IDC System for Sentiment Classification and Sarcasm Detection in Arabic," 2021. Accessed: Oct. 23, 2026. [Online]. Available: https://aclanthology.org/2021.wanlp-1.48/

[31] B. Talafha, M. E. Za'ter, S. Suleiman, M. Al-Ayyoub, and M. N. Al-Kabi, "Sarcasm Detection and Quantification in Arabic Tweets," *Proc. - Int. Conf. Tools with Artif. Intell. ICTAI*, vol. 2021-November, pp. 1121–1125, 2021, doi: 10.1109/ICTAI52525.2021.00177.

[32] M. Y. Khan, T. Ahmed, M. S. Siddiqui, and S. Wasi, "Cognitive Relationship-Based Approach for Urdu Sarcasm and Sentiment Classification," *IEEE Access*, vol. 11, pp. 126661–126690, 2023, doi: 10.1109/ACCESS.2023.3325048.

[33] S. Khan *et al.*, "An automated approach to identify sarcasm in low-resource language," *PLoS One*, vol. 19, no. 12, p. e0307186, Dec. 2024, doi: 10.1371/JOURNAL.PONE.0307186.

[34] S. Khan, I. Qasim, W. Khan, K. Aurangzeb, J. A. Khan, and M. S. Anwar, "A novel transformer attention-based approach for sarcasm detection," *Expert Syst.*, vol. 42, no. 1, p. e13686, Jan. 2025, doi: 10.1111/EXSY.13686;WGROUP:STRING:PUBLICATION.

[35] M. Islam and M. Azhar, "Sarcasm Detection in Multilingual Text through Embedding-Enhanced Language Models: BERT Variants," *Proc. IEEE Int. Multi Top. Conf. INMIC*, no. 2024, 2024, doi: 10.1109/INMIC64792.2024.11004369.

[36] "(PDF) Humor Detection in Persian: A Transformers-Based Approach | Academia.edu." Accessed: Oct. 23, 2026. [Online]. Available: https://www.academia.edu/104337275/Humor_Detection_in_Persian_A_Transformers_Based_Approach

[37] Z. B. Nezhad and M. A. Deihimi, "SARCASM DETECTION IN PERSIAN," *J. Inf. Commun. Technol.*, vol. 20, no. 1, pp. 1–20, Jan. 2021, doi: 10.32890/JICT2021.20.1.1.

[38] P. Golazizian, B. Sabeti, S. A. A. Asli, Z. Majdabadi, O. Momenzadeh, and R. Fahmi, "Irony Detection in Persian Language: A Transfer Learning Approach Using Emoji Prediction," 2020. Accessed: Oct. 23, 2026. [Online]. Available: https://aclanthology.org/2020.lrec-1.346/

[39] M. Panahandeh Nigjeh and S. Ghanbari, "Leveraging ParsBERT for cross-domain polarity sentiment classification of Persian social media comments," *Multimed. Tools Appl. 2023 834*, vol. 83, no. 4, pp. 10677–10694, Jun. 2023, doi: 10.1007/S11042-023-16067-5.

[40] P. Goel, R. Jain, A. Nayyar, S. Singhal, and M. Srivastava, "Sarcasm detection using deep learning and ensemble learning," *Multimed. Tools Appl. 2022 8130*, vol. 81, no. 30, pp. 43229–43252, May 2022, doi: 10.1007/S11042-022-12930-Z.

[41] R. Jamil, I. Ashraf, F. Rustam, E. Saad, A. Mehmood, and G. S. Choi, "Detecting sarcasm in multi-domain datasets using convolutional neural networks and long short term memory network model," *PeerJ. Comput. Sci.*, vol. 7, pp. 1–24, 2021, doi: 10.7717/PEERJ-CS.645.

[42] A. Y. Muaad *et al.*, "Artificial Intelligence-Based Approach for Misogyny and Sarcasm Detection from Arabic Texts," *Comput. Intell. Neurosci.*, vol. 2022, no. 1, p. 7937667, Jan. 2022, doi: 10.1155/2022/7937667.

[43] A. El Mahdaouy, A. El Mekki, K. Essefar, N. El Mamoun, I. Berrada, and A.

Khoumsi, "Deep Multi-Task Model for Sarcasm Detection and Sentiment Analysis in Arabic Language," 2021. Accessed: Oct. 23, 2026. [Online]. Available: https://aclanthology.org/2021.wanlp-1.42/

[44] C. van Hee, E. Lefever, and V. Hoste, "SemEval-2018 Task 3: Irony Detection in English Tweets," *NAACL HLT 2018 - Int. Work. Semant. Eval. SemEval 2018 - Proc. 12th Work.*, pp. 39–50, 2018, doi: 10.18653/V1/S18-1005.

[45] I. A. Farha and W. Magdy, "From Arabic Sentiment Analysis to Sarcasm Detection: The ArSarcasm Dataset," 2020. Accessed: Oct. 23, 2026. [Online]. Available: https://aclanthology.org/2020.osact-1.5/

[46] "Urdu Sarcastic Tweets Dataset." Accessed: Oct. 23, 2026. [Online]. Available: https://www.kaggle.com/datasets/shumailakhan/urdu-sarcastic-tweets-dataset

[47] "GitHub - fatemenajafi135/Irony-detection: Persian Irony Detection, include a Persian dataset, creating a dataset automatically, and finetuning transformer-based language models for the task · GitHub." Accessed: Oct. 23, 2026. [Online]. Available: https://github.com/fatemenajafi135/Irony-detection

[48] S. Szeghalmy and A. Fazekas, "A Comparative Study of the Use of Stratified Cross-Validation and Distribution-Balanced Stratified Cross-Validation in Imbalanced Learning," *Sensors 2023, Vol. 23, Page 2333*, vol. 23, no. 4, p. 2333, Feb. 2023, doi: 10.3390/S23042333.

Copyright © by authors and 50Sea. This work is licensed under Creative Commons Attribution 4.0 International License.