AbjadGenEval: Abjad AI Generated Text Detection Shared Task for Languages Using Arabic Script at AbjadNLP 2026

Saad Ezzini[1], Irfan Ahmad[1], Salmane Chafik[2], Shadi Abudalfa[1], Mo El-Haj [3], Ahmed Abdelali[4], Mustafa Jarrar[5,6], Nadir Durrani [5], Hassan Sajjad[7], Farah Adeeba[8]
[1]King Fahd University of Petroleum & Minerals,
[2]Mohammed VI Polytechnic University, [3]VinUniversity, [4]Humain,
[5]Hamad Bin Khalifa University, [6]Birzeit University, [7]Dalhousie University,
[8]University of Engineering & Technology

Abstract

We present the findings of the AbjadGenEval shared task, organized as part of the AbjadNLP workshop at EACL 2026, which benchmarks AI-generated text detection for Arabic-script languages. Extending beyond Arabic to include Urdu, the task serves as a binary classification platform distinguishing human-written from AI-generated news articles produced by varied LLMs (e.g., GPT, Gemini). Twenty teams participated, with top systems achieving F1 scores of 0.93 for Arabic and 0.89 for Urdu. The results highlight the dominance of multilingual transformers-specifically XLM-RoBERTa and DeBERTa-v3-and reveal significant challenges in cross-domain generalization, where naive data augmentation often yielded diminishing returns. This shared task establishes a robust baseline for authenticating content in the Abjad ecosystem.

1 Introduction

The increasing accessibility and fluency of large language models (LLMs) has fundamentally altered the landscape of digital text production. Content generated by AI systems is now pervasive across news media, social platforms, educational settings, and online communication more broadly. While this development offers clear benefits, it also raises serious concerns related to misinformation, academic integrity, authorship attribution, and trust in digital content. As a result, automatic detection of AI-generated text has emerged as a critical task within contemporary Natural Language Processing (NLP), and recent surveys underline both the urgency of the problem and the methodological diversity of current approaches (Wu et al., 2025).

Early work on AI-generated text detection has largely concentrated on English and other high-resource languages written in Latin scripts. Languages using the Arabic script have received comparatively limited attention, despite their wide geographic spread and increasing support within modern generative models. Detection in these languages is complicated by rich morphology, orthographic ambiguity, limited standardisation, and uneven availability of annotated data. Similar challenges have been documented across a range of Arabic NLP tasks, including dialect identification, corpus construction, and domain-specific modelling (El-Haj et al., 2018; El-Haj and Ezzini, 2024). Recent evidence suggests that off-the-shelf detectors can struggle substantially in Arabic settings, with orthographic phenomena such as diacritics further amplifying brittleness and lowering reliability (Alshammari and Ahmed, 2023; Alshammari and Elleithy, 2024). These challenges motivate the need for dedicated benchmarks that explicitly target Arabic-script languages.

The AbjadGenEval shared task builds directly on the foundations established by the AraGenEval shared task introduced at the Third Arabic Natural Language Processing Conference in 2025 (Abudalfa et al., 2025). AraGenEval represented the first large-scale benchmark for Arabic authorship analysis, including AI-generated text detection, and demonstrated both the feasibility of the task and the strong performance of transformer-based approaches on curated Arabic data. At the same time, its findings highlighted important limitations, particularly with respect to cross-domain robustness and the generalisation of detection methods beyond Arabic alone.

AbjadGenEval extends this line of work in two key directions. First, it broadens the scope from Arabic-only evaluation to a wider Abjad ecosystem by explicitly incorporating multiple languages written in the Arabic script, starting with Arabic and Urdu. Second, it places a stronger emphasis on comparative, language-aware evaluation, encouraging participants to explore both language-specific and transferable detection strategies. In doing so, AbjadGenEval aims to complement broader multi-

Proceedings of the 2nd Workshop on NLP for Languages Using Arabic Script (AbjadNLP 2026), pages 530–537

domain and multilingual detection efforts by focusing on script-sharing languages with substantial structural diversity, building on insights from prior multilingual corpus and benchmark development efforts (Macko et al., 2025).

Hosted as part of the AbjadNLP workshop at EACL 2026 (El-Haj, 2025b, 2026), AbjadGenEval serves as a continuation and expansion of earlier efforts, providing new datasets, standardised evaluation protocols, and an open competition framework. The shared task is intended to support reproducible research, foster methodological diversity, and offer empirical insight into the behaviour of AI-generated text detectors in underrepresented language settings.

2 Related Work

Arabic AI-Generated Text Detection is commonly formulated as a binary classification problem, where the objective is to determine whether a given text originates from a human author or has been generated by an automated system. Existing approaches to this task can be broadly categorised into four families (Wu et al., 2025). First, *statistics-based methods* rely on surface-level distributional cues, such as entropy, perplexity, or n-gram statistics, to identify regularities that often arise in machine-generated text (Shen et al., 2023; Mitchell et al., 2023). Second, *neural-based approaches* typically fine-tune pre-trained transformer models, including BERT- and RoBERTa-style architectures, achieving strong performance in controlled settings but exhibiting notable sensitivity to domain shifts and adversarial manipulation (Ippolito et al., 2020; Li et al., 2025). Third, *watermarking techniques* introduce detectable signals during text generation, either at the token level or within hidden representations, enabling proactive identification of machine-generated content (Kirchenbauer et al., 2023; Zhao et al., 2023). Finally, *LLM-as-detector paradigms* employ large language models themselves as classifiers or reasoning agents to assess text origin, often providing richer explanations at the cost of increased computational complexity (Wang et al., 2024b; Su et al., 2025).

In the context of Arabic and other Arabic-script languages, recent studies have highlighted additional challenges stemming from complex morphology, orthographic variation, and limited annotated resources. Similar issues have been observed across Arabic corpus development and evaluation tasks, including literary text collections and large-scale domain-specific datasets (El-Haj, 2020, 2025a; El-Haj and Rayson, 2025). Arabic-specific transformer models have been explored for generative text detection, revealing both the potential benefits of language-aware pre-training and persistent weaknesses under cross-domain evaluation (Alshammari and Elleithy, 2024). To encourage more systematic comparison, several recent benchmarks have focused on cross-domain robustness, including MultiSocial (Macko et al., 2025), XDAC (Go et al., 2025), and M4GT-Bench (Wang et al., 2024b). In parallel, a number of shared tasks have been organised to assess detector performance in diverse settings, such as SemEval-2024 Task 8 (Wang et al., 2024a), the GenAI Content Detection Task on academic essay authenticity (Chowdhury et al., 2024), the M-DAIGT challenge (Lamsiyah et al., 2025), and GenAI Content Detection Task 3, which examined detection in scenarios involving a large but fixed set of known domains and generation models (Dugan et al., 2025).

Despite these efforts, large-scale and standardised evaluation resources for Arabic remain scarce. Empirical analysis on the AIRABIC dataset demonstrates that widely used detectors, including GPTZero and OpenAI's Text Classifier, perform poorly on Arabic text, particularly when diacritics are present, with reported accuracies dropping significantly (Alshammari and Ahmed, 2023). These findings expose fundamental limitations of detectors designed primarily for English and motivate the development of Arabic-centred evaluation frameworks. In response, AraGenEval introduced the ARATECT subtask as the first multi-genre benchmark dedicated to Arabic AI-generated text detection, providing a foundation that the AbjadGenEval shared task further extends.

3 Task Description

AbjadGenEval is formulated as a binary text classification task, where systems are required to determine whether a given input text is human-written or AI-generated. The task is designed to reflect realistic detection scenarios, covering both full-length news articles and shorter text snippets across multiple topical domains.

The shared task is organised into two primary language-specific subtasks, each evaluated independently via the Codabench platform:

Team	Track / Lang	Backbone model(s)	System idea / contribution	Representation / aggregation	Training strategy	Inference strategy
kickitlikeshika	Arabic	multilingual-e5-large	Compare pooling variants; mean pooling best under limited data	Mean pooling over token embeddings (also tried WLP, attention pooling, gated fusion)	Focal loss; AdamW; layer-wise LR decay; multi-sample dropout; cosine warmup; 2 epochs	Single model; standard probability output (no special calibration described)
HCMUS_TheFangs	Arabic	DeBERTa-v3-base	Dual pooling to capture multi-level artifacts: WLP (layers) + attention pooling (tokens)	WLP over [CLS] from layers 4–12 + attention pooling over tokens; concat → 1536-d vector	5-fold stratified CV; FP16; 5 epochs; batch 8; AdamW + cosine + warmup; discriminative fine-tuning; cross-entropy (no class weights)	No ensembling/thresholding procedure specified beyond CV evaluation
U-RoCX	Urdu	Frozen XLM-RoBERTa embeddings + CNN + xLSTM	Hybrid efficient architecture: CNN for local patterns + xLSTM for sequence modeling; freeze backbone to reduce trainable params	Embeddings → CNN → xLSTM stack; last hidden state → dense (256) → logits	Cross-entropy; AdamW; OneCycleLR; dropout 0.3 (head); trained on Tesla T4	Single model; standard softmax output
mohannad_hendi	Urdu	XLM-RoBERTa-base	Data-centric approach: sliding-window segmentation + doc-level aggregation + threshold tuning	Segment-level classifier; document score = mean of segment probabilities	Freeze encoder 1st epoch then unfreeze; differential LRs; early stopping; threshold optimized on validation for F1	Mean aggregation of segment probabilities + tuned decision threshold
LoRAD	Arabic + Urdu	XLM-RoBERTa	Low-resource baseline; emphasize multilingual transformer; different data handling per language	Standard sequence classification (pooled rep + classifier)	AdamW (lr 2e-5); 3 epochs; dynamic padding; batch: Arabic 16, Urdu 8	Single model per track; no ensemble/thresholding described
Kashif-AI	Arabic	CAMeLBERT-Mix and MARBERT (best: CAMeLBERT-Mix on official data)	Transformer fine-tuning baseline + ablation on external augmentation effects	[CLS] embedding → linear classifier	Stratified split; 3 epochs; batch 16; AdamW; lr 3e-5; warmup 0.1; weight decay 0.01; save best	Single best model (no ensemble/thresholding described)
se7s0	Arabic	AraBERT + XLM-RoBERTa	Supervised contrastive learning + stacking ensemble; 3-stage training	Two-head models (classifier + 256-d projection); stacking via logistic regression over model outputs	Stage 1 CE fine-tune; Stage 2 add supervised contrastive loss; Stage 3 fine-tune on abstracts (lr reduced to 1e-5); AdamW; wd 0.01; batch 16	Stacking (logistic regression) over AraBERT/XLM-R probabilities
REGLAT	Arabic	AraBERTv2 + BERT-base-arabic	Full pipeline: Arabic normalization + augmentation + CV + weighted ensemble + threshold optimization	Weighted avg of probabilities (0.6 AraBERTv2 / 0.4 BERT-base-arabic)	5-fold stratified CV; average fold preds; class-weighted cross-entropy; label smoothing 0.05; dropout 0.25; lr 1.5e-5; warmup 0.15; eff. batch 36; wd 0.05; FP16; early stopping	Average CV predictions; tuned threshold (0.69); weighted ensemble
AyahVerse	Arabic + Urdu	Arabic mono: AraBERTv2, CAMeLBERT-DA, ArabicBERT; Multi: mBERT, XLM-R	Monolingual vs multilingual comparison; feature isolation preprocessing; cross-lingual transfer experiments	Standard fine-tuned transformers; submitted best per track (Arabic: AraBERTv2; Urdu: mBERT)	AdamW; lr 2×10^{-5}; batch 32; epochs 2–6; layer-freezing ablations (bottom 6)	Per-track best model (no ensemble described); cross-lingual evaluation reported
saeedanabtawi	Urdu	Linear SVM (via SGD)	Hybrid stylometric-statistical pipeline	Feature Union: TF-IDF Character N-grams (2-4 range) concatenated with 4 custom Urdu stylometric features (repeated words ratio, punctuation ratio, Urdu formal/function markers counts) combined using SVM as a classifier	Grid search with 3-fold CV; StandardScaler (mean centering disabled); Hinge loss with L2 penalty; early stopping; optimal learning rate	Single model with best setting that achieved highest F1 score

Abbreviations: WLP = weighted layer pooling; CV = cross-validation; CE = cross-entropy.

Table 1: Key characteristics of the submitted systems.

Task 1: AI-Generated Arabic Text Detection This subtask focuses exclusively on Arabic news text. Participants are provided with a balanced dataset consisting of human-authored articles sourced from verified news outlets and AI-generated articles produced using a range of contemporary LLMs under diverse prompting strategies. Systems must learn to distinguish between human and machine-generated content while remaining robust to variation in article length, topic, and generation style.

Task 2: AI-Generated Urdu Text Detection This subtask mirrors the Arabic track but targets Urdu news text. As a comparatively lower-resource language in the context of AI-generated text detection, the Urdu track presents additional challenges related to data scarcity and orthographic variation. The dataset composition and task formulation are aligned with the Arabic track to enable comparative analysis across languages.

Input and Output For both subtasks, the input consists of a single text instance, typically a news article or excerpt. Systems are required to output a binary label indicating whether the text is *human-written* or *AI-generated*. Submissions are made in the form of prediction files uploaded to the Codabench evaluation system.

Evaluation Metrics System performance is evaluated primarily using the macro-averaged F1 score, which accounts for potential class imbalance and provides a balanced view of precision and recall. Accuracy, precision, and recall are reported as secondary metrics to support more detailed analysis of system behaviour.

Participation and Tracks Participants may submit systems to one or both language-specific subtasks. Each track is evaluated independently, allowing teams to explore language-specific modelling strategies as well as transfer and multilingual approaches. The task design deliberately avoids assumptions about model architecture, encouraging a wide range of solutions including fine-tuned multilingual encoders, language-specific models, and hybrid approaches.

Through this formulation, AbjadGenEval aims to provide a controlled yet realistic evaluation setting for AI-generated text detection in Arabic-script languages, supporting both methodological innovation and deeper empirical understanding of detection challenges beyond high-resource, Latin-script contexts.

4 Data and Evaluation Protocol

Data Construction and Anonymization We constructed a balanced dataset of human-written and AI-generated news articles for both Arabic and Urdu. Human-written articles were sourced from diverse, reputable news outlets, ensuring coverage of various topics including politics, sports, and culture. We applied strict filtering to remove author names, social media handles, and direct source references to prevent model bias based on metadata.

Evaluation Metrics Systems were evaluated using the macro-averaged F1 score as the primary metric to account for class balance. Secondary metrics included Accuracy, Precision, Recall, and Balanced Accuracy. The evaluation was conducted on the Codabench platform with a blind test phase where participants submitted predictions on held-out data.

5 Data Generation

5.1 Arabic AI-Generated Text Creation

To generate the Arabic segment of the dataset, we adopted an iterative, feedback-driven pipeline similar to the approach used in the AraGenEval shared task (Abudalfa et al., 2025). We collected authentic human-written news articles from sources such as Al Jazeera and Hespress. Titles from these articles served as prompts for AI generation.

We employed a diverse set of Large Language Models (LLMs), including GPT-4, and Gemini-3-Pro, to generate synthetic articles. The generation pipeline incorporated a detection-based feedback loop:

1. **Generation**: An LLM generated an article based on a provided title and a specific persona (e.g., "Write as a professional news reporter").

2. **Detection Check**: The generated text was passed to a preliminary AI detection model.

3. **Refinement**: If the text was easily detected as AI-generated, the generator was prompted to revise the content to sound more human-like. This process repeated until the text passed the detection threshold or a maximum number of iterations was reached.

This adversarial generation process ensured that the resulting dataset contained high-quality, challenging examples of AI-generated text.

5.2 Urdu AI-Generated Text Creation

For the Urdu subtask, we curated a corpus of approximately 6,000 human-written articles from BBC Urdu (2019–2021), filtered to include only those by Pakistani reporters to ensure linguistic consistency. AI-generated counterparts were produced using GPT-4o and GPT-3.5, conditioned on

the style of specific reporters. The prompting strategy involved providing the model with a "few-shot" example of a reporter's writing style and asking it to generate a new article on a given topic in that specific style. A subsequent validation step involved using different LLMs as judges to filter out generated texts that were easily distinguishable, resulting in a final balanced dataset of 1,826 AI-generated and 1,826 human-written articles.

6 System Overview

The AbjadGenEval shared task attracted diverse approaches, primarily leveraging transformer-based architectures. Table 1 summarizes the key characteristics of participating systems.

Transformer Dominance The majority of submitted systems relied on fine-tuning pre-trained transformer models. Multilingual models like **XLM-RoBERTa** and **DeBERTa-v3** were particularly popular and effective, often outperforming or matching monolingual Arabic models.

Advanced Pooling Strategies Several top-performing teams, such as HCMUS_TheFangs (Paper 77), moved beyond simple [CLS] token classification. They implemented sophisticated pooling mechanisms like **Weighted Layer Pooling**, which aggregates representations from multiple layers to capture both surface-level syntax and deep semantic features, and **Attention Pooling** to focus on salient parts of the input.

Ensembling and Hybrid Architectures Ensemble methods proved robust. Team se7s0 (Paper 88) employed a stacking ensemble of AraBERT and XLM-RoBERTa, while REGLAT (Paper 89) used weighted averaging of two Arabic BERT variants. Innovative hybrid architectures also appeared; notably, U-RoCX (Paper 79) integrated a Convolutional Neural Network (CNN) and an **xLSTM** (Extended LSTM) block on top of frozen XLM-RoBERTa embeddings to capture sequential dependencies more effectively.

Classical and Stylometric Approaches While transformer-based models dominated, Team AnonAI (saeedanabtawi) demonstrated the continued relevance of classical approaches. They employed a hybrid pipeline combining TF-IDF character n-grams with custom stylometric features (e.g., repeated word ratio, punctuation density, formal markers), classified using a Linear SVM. This lightweight approach achieved competitive performance (F1=0.88) in the Urdu track, highlighting the potential of interpretable, linguistically-motivated features for low-resource languages.

7 Results

This section reports the official leaderboard results for the AbjadGenEval shared task as obtained from the Codabench evaluation platform. We report results separately for the Arabic and Urdu subtasks to reflect their independent evaluation settings.

7.1 Arabic AI-Generated Text Detection Results

Table 2 presents the top-performing systems for the Arabic track. The competition was fierce, with the top team achieving an F1 score of nearly 0.93. This suggests that, for the released dataset, current approaches are able to effectively distinguish between human-written and AI-generated Arabic news text under the provided conditions.

A total of 20 teams registered for the Arabic subtask, submitting 12 valid system runs during the evaluation phase.

7.2 Urdu AI-Generated Text Detection Results

The Urdu subtask results are presented in Table 3. The top teams achieved very high consistency, with F1 scores clustering around 0.88. While top systems again achieve very strong performance, there is a much narrower performance spread compared to Arabic, and capped below 90% F1 score. This reflects the additional challenges posed by the Urdu task, including limited training resources.

The Urdu subtask attracted 10 participating teams, with a total of 12 submissions evaluated during the training phase.

8 Discussion

Architecture Impact The results underscore the efficacy of large multilingual transformers like DeBERTa-v3 and XLM-RoBERTa. Team HCMUS_TheFangs' success ($F1 \approx 0.93$) in the Arabic track with DeBERTa-v3 suggests that models with disentangled attention mechanisms may better capture the subtle structural incoherence often found in AI-generated text.

Data Augmentation Pitfalls A counter-intuitive finding from multiple participants (e.g., Kashif-AI, se7s0) was that naive data augmentation often

Team	F1	Acc	Prec	Rec	Bal. Acc
HCMUS_TheFangs	0.9271	0.9300	0.9674	0.8900	0.9300
chisboizhoigay	0.9005	0.9050	0.9451	0.8600	0.9050
alizain157 (LoRAD)	0.8867	0.8850	0.8738	0.9000	0.8850
se7s0	0.7819	0.7350	0.6643	0.9500	0.7350
mariamlabib90 (REGLAT)	0.7626	0.6950	0.6242	0.9800	0.6950
AyahVerse	0.7534	0.7250	0.6829	0.8400	0.7250
kickitlikeshika	0.7500	0.7900	0.9265	0.6300	0.7900
songohan	0.7300	0.7300	0.7300	0.7300	0.7300
HCMUS_RepeatedGame	0.6667	0.5000	0.5000	1.0000	0.5000
astral_fate (Kashif-AI)	0.6629	0.7050	0.7733	0.5800	0.7050
WinnerHere	0.5824	0.6200	0.6463	0.5300	0.6200
michaelibrahim	0.3931	0.4750	0.4658	0.3400	0.4750

Table 2: Official Leaderboard for the Arabic Subtask.

Team	F1	Acc	Prec	Rec	Bal. Acc
alizain157	0.8878	0.8878	0.8834	0.8922	0.8879
basilh	0.8873	0.8875	0.8839	0.8907	0.8875
mohannad_hendi	0.8868	0.8871	0.8844	0.8891	0.8871
rabeeqasem93	0.8804	0.8802	0.8747	0.8861	0.8803
saeedanabtawi	0.8781	0.8779	0.8725	0.8838	0.8780
ibad-ur-rehman	0.8655	0.8627	0.8439	0.8884	0.8629
salmane	0.8398	0.8403	0.8379	0.8417	0.8403

Table 3: Official Leaderboard for the Urdu Subtask.

degraded performance. Adding external datasets, such as the Arabic Generated Abstracts, led to domain shifts where models overfitted to specific artifacts (e.g., academic writing style) rather than generalizing to the domain of the shared task, i.e., news. This highlights the importance of domain alignment in training data.

Precision-Recall Trade-offs While some systems achieved balanced performance, others like REGLAT prioritized Recall (0.98) at the expense of Precision (0.62). In safety-critical applications where missing AI-generated misinformation is costly, high recall is desirable; however, for automated content moderation, low precision could lead to legitimate content being flagged, emphasizing the need for tunable decision thresholds (as explored by Team mohannad_hendi).

Cross-Lingual Capabilities The strong performance of the same architectures (e.g., XLM-RoBERTa used by alizain157) across both Arabic and Urdu tracks demonstrates the viability of language-agnostic approaches. This is crucial for low-resource Abjad languages where dedicated monolingual models may not exist.

Efficiency of Classical Models The success of Team AnonAI's SVM-based system in the Urdu track challenges the notion that heavy neural models are strictly necessary. By leveraging domain-specific stylometric markers, they achieved high accuracy with a fraction of the computational cost of transformer models. This is particularly relevant for deployment in resource-constrained environments.

9 Limitations

One primary limitation of this shared task is the specific domain focus on news articles. News text has a distinct, formal structure that may make detection easier compared to informal social media text or creative writing. Additionally, the set of

generator models (GPT-4, etc.) is fixed; real-world detectors must contend with a continuously evolving landscape of new models. Finally, the "blind" nature of the test set, while ensuring fair evaluation, revealed significant generalization gaps for many teams, indicating that current models are still brittle to distribution shifts.

10 Future Work

Future iterations of AbjadGenEval should expand to include:

1. **More Dialects and Genres**: moving beyond MSA news to cover Dialectal Arabic tweets, comments, and literary works.

2. **Adversarial Evaluation**: Testing against "jailbroken" or adversarially prompted LLMs explicitly trying to evade detection.

3. **Explainability**: Encouraging submissions that not only detect but also explain *why* a text is flagged, highlighting specific linguistic markers.

4. **Language Expansion**: Including other Abjad-script languages such as Farsi, Pashto, and Sindhi.

11 Conclusion

AbjadGenEval successfully established a benchmark for AI-generated text detection in Arabic and Urdu. The participation of diverse teams and the high performance of top systems ($F1 > 0.90$ for Arabic, $F1 \approx 0.88$ for Urdu) demonstrate that automated detection is feasible with current technology. However, the reliance of top systems on specific architectures and the observed sensitivity to training data domains suggest that "solving" detection requires more than just better models—it requires robust, diverse, and evolving datasets. We hope this task serves as a catalyst for further research into trustworthy AI for the Abjad languages ecosystem.

Acknowledgements

We thank all the participating teams for their hard work and contributions. We also acknowledge the support of the AbjadNLP workshop organizers and the EACL 2026 conference for hosting this shared task.

Ethics Statement

The datasets used in this shared task were constructed with privacy and ethics in mind. Human-written texts were sourced from public news outlets, and all personal identifiable information (PII) was removed or anonymized. The AI-generated texts were produced using commercial and open-source models in accordance with their usage policies. We emphasize that AI detection tools should be used as decision-aids, not absolute arbiters of truth, given the potential for false positives.

References

Shadi Abudalfa, Saad Ezzini, Ahmed Abdelali, Hamza Alami, Abdessamad Benlahbib, Salmane Chafik, Mo El-Haj, Abdelkader El Mahdaouy, Mustafa Jarrar, Salima Lamsiyah, and Hamzah Luqman. 2025. The AraGenEval shared task on Arabic authorship style transfer and AI generated text detection. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 1–13, Suzhou, China. Association for Computational Linguistics.

Hamed Alshammari and EI-Sayed Ahmed. 2023. Airabic: Arabic dataset for performance evaluation of ai detectors. In *2023 International Conference on Machine Learning and Applications (ICMLA)*, pages 864–870. IEEE.

Hamed Alshammari and Khaled Elleithy. 2024. Toward robust arabic ai-generated text detection: Tackling diacritics challenges. *Information*, 15(7):419.

Shammur Absar Chowdhury, Hind Almerekhi, Mucahid Kutlu, Kaan Efe Keles, Fatema Ahmad, Tasnim Mohiuddin, George Mikros, and Firoj Alam. 2024. Genai content detection task 2: Ai vs. human–academic essay authenticity challenge. *arXiv preprint arXiv:2412.18274*.

Liam Dugan, Andrew Zhu, Firoj Alam, Preslav Nakov, Marianna Apidianaki, and Chris Callison-Burch. 2025. Genai content detection task 3: Cross-domain machine generated text detection challenge. In *Proceedings of the 1stWorkshop on GenAI Content Detection (GenAIDetect)*, pages 377–388.

Mahmoud El-Haj. 2020. Habibi-a multi dialect multi national arabic song lyrics corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1318–1326.

Mahmoud El-Haj, Paul Rayson, and Mariam Aboelezz. 2018. Arabic dialect identification in the context of bivalency and code-switching. In *Proceedings of the 11th International Conference on Language Resources and Evaluation, Miyazaki, Japan.*, pages 3622–3627. European Language Resources Association.

Mo El-Haj. 2025a. Arabjobs: A multinational corpus of arabic job ads. In *Proceedings of The Third Arabic Natural Language Processing Conference*, pages 16–25.

Mo El-Haj. 2025b. Proceedings of the 1st workshop on nlp for languages using arabic script (abjadnlp 2025). In *Proceedings of the 1st Workshop on NLP for Languages Using Arabic Script, held at COLING 2025*, Abu Dhabi, United Arab Emirates.

Mo El-Haj. 2026. Proceedings of the 2nd workshop on nlp for languages using arabic script (abjadnlp 2026). In *Proceedings of the 2nd Workshop on NLP for Languages Using Arabic Script, held at EACL 2026*, Rabat, Morocco.

Mo El-Haj and Saad Ezzini. 2024. The multilingual corpus of world's constitutions (mcwc). In *Proceedings of the 6th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT) with Shared Tasks on Arabic LLMs Hallucination and Dialect to MSA Machine Translation@ LREC-COLING 2024*, pages 57–66.

Mo El-Haj and Paul Rayson. 2025. Arafinnews: Arabic financial summarisation with domain-adapted llms. *arXiv preprint arXiv:2511.01265*.

Wooyoung Go, Hyoungshick Kim, Alice Oh, and Yongdae Kim. 2025. XDAC: XAI-driven detection and attribution of LLM-generated news comments in Korean. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22728–22750, Vienna, Austria. Association for Computational Linguistics.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822, Online. Association for Computational Linguistics.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *International Conference on Machine Learning*, pages 17061–17084. PMLR.

Salima Lamsiyah, Saad Ezzini, Abdelkader Elmahdaouy, Hamza Alami, Abdessamad Benlahbib, Samir El Amrany, Salmane Chafik, and Hicham Hammouchi. 2025. Shared task on multi-domain detection of ai-generated text (m-daigt). In *Proceedings of the 15th International Conference on Recent Advances in Natural Language Processing (RANLP)*.

Yuanfan Li, Zhaohan Zhang, Chengzhengxu Li, Chao Shen, and Xiaoming Liu. 2025. Iron sharpens iron: Defending against attacks in machine-generated text detection with adversarial training. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3091–3113, Vienna, Austria. Association for Computational Linguistics.

Dominik Macko, Jakub Kopál, Robert Moro, and Ivan Srba. 2025. MultiSocial: Multilingual benchmark of machine-generated text detection of social-media texts. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 727–752, Vienna, Austria. Association for Computational Linguistics.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International conference on machine learning*, pages 24950–24962. PMLR.

Lujia Shen, Xuhong Zhang, Shouling Ji, Yuwen Pu, Chunpeng Ge, Xing Yang, and Yanghe Feng. 2023. Textdefense: Adversarial text detection based on word importance entropy. *arXiv preprint arXiv:2302.05892*.

Zhixiong Su, Yichen Wang, Herun Wan, Zhaohan Zhang, and Minnan Luo. 2025. HACo-det: A study towards fine-grained machine-generated text detection under human-AI coauthoring. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22015–22036, Vienna, Austria. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, and Thomas Arnold. 2024a. SemEval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2057–2079, Mexico City, Mexico. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024b. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1369–1407, St. Julian's, Malta. Association for Computational Linguistics.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Lidia Sam Chao, and Derek Fai Wong. 2025. A survey on llm-generated text detection: Necessity, methods, and future directions. *Computational Linguistics*, 51(1):275–338.

Xuandong Zhao, Yu-Xiang Wang, and Lei Li. 2023. Protecting language generation models via invisible watermarking. In *International Conference on Machine Learning*, pages 42187–42199. PMLR.