Libyan Open University Journal of Applied Sciences (LOUJAS)
Volume 2 - Issue 2 - 2026 - Pages 22-46

Leakage Aware Evaluation of Arabic Text Classification: Quantifying Document Level Leakage and the Sufficiency of Linear TF IDF Baselines

Albahlool Abood [1*] , Walid Hasan [2]
[1]Faculty of Information Technology, University of Gharyan, Gharyan, Libya
[2]Libyan National Authority for Scientific Research, Tripoli, Libya
[*]**Email for reference researcher:** a.aboud@academy.edu.ly

تقييم تصنيف النصوص العربية مع مراعاة تسرّب البيانات: قياس التسرّب على مستوى الوثيقة وكفاية خطوط الأساس الخطية المعتمدة على TF-IDF

البهلول عبود [1*]، وليد حسن [2]
[1] كلية تقنية المعلومات، جامعة غريان، غريان، ليبيا
[2] الهيئة الليبية للبحث العلمي ، طرابلس ، ليبيا

Received: 07-05-2026; Accepted: 01-07-2026; Published: 05-07-2026

Abstract

The volume of Arabic text online keeps growing, and with it the need for systems that sort it into meaningful categories. A quieter problem often passes unnoticed: when long documents are split into shorter segments and pieces of the same document fall on both sides of the boundary between training and testing, the reported scores can look far better than the model deserves. This study treats that document level leakage as its central subject. Using a purpose built multidomain Arabic corpus of 1,000 segments drawn from 183 source documents across seven categories, we first measure how much leakage distorts the numbers. Under a naive random split, an optimized Linear SVM appears to reach 0.9800 accuracy; under a grouped split keyed to the document identifier, the same model reaches 0.8528. The gap of about 12.7 accuracy points is attributable to leakage, and across models the inflation ranges from roughly 11.9 to 30.4 points. Leakage also flattens the ranking, so a weak Naive Bayes looks almost as strong as the best model. Under the honest protocol, an ablation shows that character level TF IDF alone matches the combined word and character representation and slightly exceeds it on Macro F1, 0.7801 against 0.7713, while word features mainly refine precision. McNemar tests and a document clustered bootstrap show that the strongest linear models are statistically indistinguishable. A fastText baseline reaches 0.6497 accuracy, and two pretrained Arabic transformers, AraBERT and MARBERT, perform comparably to the linear model rather than surpassing it. A class level analysis exposes the General category as a structural confound whose removal raises accuracy to 0.9349. Overall, leakage aware evaluation paired with subword rich TF IDF and a Linear SVM forms a strong, honest, and practical baseline for Arabic text classification.

Keywords: Arabic text classification; data leakage; grouped evaluation; character level TF IDF; Linear SVM; statistical testing; reproducibility.

الملخص.

يتزايد حجم النصوص العربية المنشورة على الإنترنت بصورة مستمرة، مما يزيد الحاجة إلى أنظمة فعّالة لتصنيفها ضمن فئات ذات دلالة. ومع ذلك، غالباً ما تُغفَل مشكلة منهجية دقيقة تتمثل في تسرّب البيانات على مستوى الوثيقة؛ إذ قد تُقسَّم الوثائق الطويلة إلى مقاطع أقصر، فتنتقل أجزاء من الوثيقة نفسها إلى كلٍّ من مجموعتي التدريب والاختبار، الأمر الذي يؤدي إلى تقديرات متفائلة وغير واقعية لأداء النماذج. تتناول هذه الدراسة هذه الظاهرة بوصفها القضية الرئيسة للبحث.
اعتمدت الدراسة على متن عربي متعدد المجالات جرى بناؤه خصيصاً لهذا الغرض، ويتكون من 1000 مقطع نصي مستخرج من 183 وثيقة مصدرية موزعة على سبع فئات موضوعية. وجرى أولاً قياس مقدار التشوه الذي يسببه التسرّب في نتائج التقييم. فقد حقق نموذج Linear SVM المُحسَّن دقة ظاهرية بلغت 0.9800 عند استخدام تقسيم عشوائي ساذج

للبيانات، في حين انخفضت الدقة إلى 0.8528 عند استخدام تقسيم مُجمَّع يعتمد على معرّف الوثيقة. ويشير هذا الفرق، البالغ نحو 12.7 نقطة مئوية، إلى مقدار التضخم الناتج عن التسرّب، بينما تراوح هذا التضخم بين نحو 11.9 و30.4 نقطة مئوية عبر النماذج المختلفة.
كما أظهرت النتائج أن التسرّب يؤدي إلى تقليص الفروق الحقيقية بين النماذج، بحيث يبدو مصنّف Naive Bayes الضعيف أقلَّ بعداً عن أفضل النماذج مما هو عليه في الواقع. وعند اعتماد بروتوكول التقييم النزيه، كشفت دراسة الاستئصال أن تمثيل TF-IDF القائم على المحارف وحدها يضاهي التمثيل المدمج المعتمد على الكلمات والمحارف، بل ويتفوق عليه قليلاً في مقياس Macro F1 محققاً 0.7801 مقابل 0.7713، في حين أسهمت سمات الكلمات بصورة رئيسة في تحسين الإحكام.(Precision)
وأظهرت اختبارات McNemar وتقنية Bootstrap المجمّعة على مستوى الوثيقة أن الفروق بين أقوى النماذج الخطية ليست ذات دلالة إحصائية. كما حقق خط أساس fastText دقة بلغت 0.6497، في حين قدّم نموذجا المحولات العربية المدربان مسبقاً AraBERT و MARBERTأداءً مقارباً للنموذج الخطي دون أن يتفوّقا عليه. كذلك كشف التحليل على مستوى الفئات أن فئة «عام» تمثل عاملاً بنيوياً مُربكاً، حيث أدى استبعادها إلى رفع الدقة إلى 0.9349.
وبصورة عامة، تُظهر النتائج أن اعتماد تقييم يراعي تسرّب البيانات، إلى جانب استخدام تمثيل TF-IDF الغني بالوحدات الجزئية ومصنّفLinear SVM ، يوفر خط أساس قوياً ونزيهاً وعملياً لتصنيف النصوص العربية.

الكلمات المفتاحية: تصنيف النصوص العربية؛ تسرّب البيانات؛ التقييم المُجمَّع؛ TF-IDF على مستوى المحارف؛ Linear SVM؛ الاختبارات الإحصائية؛ قابلية إعادة الإنتاج.

1. Introduction

The amount of Arabic text produced every day has grown beyond what any reader, or any newsroom, could sort by hand. News portals, government sites, scientific repositories, commercial pages, and social platforms all feed the same flood, and somewhere inside it sits a practical question. How does one route an Arabic document to the right thematic category automatically and reliably? The task has become a cornerstone of Arabic natural language processing and of any serious attempt at large scale content management (Wahdan et al., 2024; Einea et al., 2019).

What makes the problem interesting is that it resists the easy framing of just picking a good classifier. Arabic carries a rich morphology, forms words with unusual flexibility, and tolerates a good deal of orthographic variation, so a single underlying meaning can surface in several different written shapes. Arabic documents also rarely respect tidy topic boundaries. A piece nominally about technology may drift into economics or politics, and a general interest article can read almost exactly like a news report. Performance, then, turns on much more than the algorithm. It depends on how the text is cleaned, how features are represented, how the corpus is built, and how clearly the target labels are drawn. Earlier work has shown repeatedly that Arabic results move with preprocessing strategy, representation model, dataset structure, and the nature of the task itself (Albalawi et al., 2021; Elnagar et al., 2020; El-Rifai et al., 2022; Masadeh et al., 2024).

The methods people have tried form a broad spectrum. Classical learners such as Support Vector Machines, Logistic Regression, and Naive Bayes remain popular precisely because they are fast, interpretable, and comfortable with sparse textual features. More recent efforts reach for depth, including convolutional networks, recurrent models, hybrid neural designs, and transformer based models built for Arabic understanding (Alammary, 2022; Alsaleh & Larabi-Marie-Sainte, 2021; Alnagi et al., 2025; Minaee et al., 2021). Still others have explored data augmentation, word embeddings, and hybrid representations to cope with linguistic variation and scarce data (Abdhood et al., 2025; Alayba & Altamimi, 2025; Louail et al., 2024). Taken together, this body of work shows a field that has matured into many directions at once, and it also quietly suggests that no single model family wins everywhere, across every dataset and every evaluation setup.

There is one issue, though, that tends to slip past the spotlight, and it concerns whether the evaluation itself can be trusted. A reported number means little if the way the data was divided let related texts sit in both the training and the testing sets. The danger sharpens when long documents are first chopped into smaller segments. Should some segments of a

document land in training while its siblings land in testing, the classifier can lean on shared vocabulary, a recognizable writing style, and repeated turns of phrase rather than on anything that would transfer to a document it has never seen. The model looks as if it generalizes; in truth it is partly remembering. The wider machine learning community has named this hazard, data leakage, and has traced to it a great many overoptimistic and hard to reproduce results (Kapoor & Narayanan, 2023; Starcke et al., 2025).

Most discussions of leakage in Arabic classification, however, stop at naming the risk. They recommend a grouped split and move on, without showing how large the distortion really is on a concrete Arabic corpus. This study takes the opposite stance. It treats the size of the leakage effect as a quantity to be measured, not a caution to be asserted. Working with a curated multidomain Arabic corpus spanning seven categories, namely News, Economy, Cars, Technology, Science, General, and Sports, it runs every model twice, once under a conventional random row split and once under a split grouped by the original document identifier, and reports the difference. That difference, it turns out, is the most important number in the paper.

A second thread runs through the work, and it concerns how the text is represented. Word level TF IDF is good at the obvious lexical signals that mark a topic. Character level TF IDF reaches into the subword layer of affixes, internal patterns, and orthographic regularities, which matters a great deal in Arabic, where the clue to a category may live inside a word rather than in the word itself. We put both representations through an ablation under the honest protocol, and the result reframes the usual hybrid story. The character level component, on its own, carries almost all of the signal. The combination does not so much beat the stronger single view as match it while rounding out precision.

The contributions of this study can be stated plainly:

1) It quantifies document level leakage in multidomain Arabic text classification, showing on a single corpus that a naive random split inflates accuracy by roughly 11.9 to 30.4 points relative to a grouped split, and that leakage also collapses the ranking among models.

2) It shows, through a controlled ablation under the honest protocol, that character level TF IDF is sufficient for this task. It matches the full word plus character representation and slightly exceeds it on Macro F1, while word level features mainly refine precision.

3) It compares several classical learners against a fair neural baseline, fastText with subword features, and against two pretrained Arabic transformers, AraBERT and MARBERT, under one grouped protocol, with statistical testing, namely McNemar exact tests and a document clustered bootstrap. The strongest linear models are statistically indistinguishable from one another, and neither pretrained transformer surpasses them.

4) It looks past aggregate accuracy to class level F1, the composition of the test set, and a confusion matrix, and it isolates the General category as a structural confound whose removal raises overall accuracy and sharply improves the neighboring News category.

The pipeline moves from corpus preparation and Arabic cleaning, through leakage aware grouped splitting, into feature extraction, model training, and a final evaluation on standard metrics. The remainder of the paper proceeds as follows. Section 2 surveys related work on Arabic classification, representation strategies, deep learning, and evaluation reliability. Section 3 lays out the methodology, including corpus construction, preprocessing, grouped partitioning, feature extraction, the models, the statistical procedures, and the metrics. Section 4 presents the results and reads them closely. Section 5 draws the threads together and points to where the work might go next.

2. Literature Review and Theoretical Background

The early literature was largely a search for the right classifier and a few extra points of accuracy. The more recent literature reads differently. It treats classification as the product of several interacting choices, namely corpus design, preprocessing, representation, model selection, and the reliability of the evaluation. That shift matters especially for Arabic, where

rich morphology, spelling variation, and overlap between categories make the problem harder than a first glance suggests (Wahdan et al., 2024; Elnagar et al., 2020; Alammary, 2022).

This section follows the strands most relevant to the present work. It looks at how Arabic corpora are represented, where classical machine learning and hybrid sparse features stand, what deep learning and transformers have added, and the rising unease about data leakage and reproducible evaluation.

2.1 Arabic Text Classification and Corpus Representation

Almost everything in this task traces back to two things, the quality of the dataset and the way the text is turned into features before a model ever sees it. Survey work has documented how widely Arabic classification has spread across domains and how many machine learning and deep learning methods have been applied to it (Wahdan et al., 2024; Abdulghani & Abdullah, 2022), confirming the place of the task at the center of Arabic NLP and of any effort to organize large collections automatically.

Building the dataset is itself a substantial part of the problem. Einea et al. (2019) introduced SANAD, a large single label Arabic news corpus that has since become a common benchmark, and its value lies in offering organized news documents across clean categories. A well structured dataset, however, does not dissolve every evaluation worry. Where the corpus came from, how documents were segmented, and how the partition was drawn all keep their grip on the trustworthiness of the numbers, which is precisely the concern this study foregrounds.

Elnagar et al. (2020) examined Arabic classification with deep models and showed that they can learn useful structure given suitable data, helping to cement the role of neural methods in the field. As in many such studies, though, the spotlight fell on model performance rather than on the quieter question of document level dependency leaking between training and testing.

Category overlap is its own recurring theme. El-Rifai et al. (2022) argued that Arabic news classification may actually call for multilabeling, since real Arabic documents so often braid more than one topic together, politics with economy or technology with science. That observation lands squarely on the present corpus, where News, General, Economy, and Technology can share vocabulary and tone. In those cases the difficulty is not purely technical. It reaches back to how sharply the labels were defined in the first place.

Preprocessing leaves deep fingerprints too. Albalawi et al. (2021) studied how preprocessing and pretrained embeddings shape Arabic health information classification on social media, and their results caution against treating cleaning as a routine reflex. Stripping noise helps, yet aggressive normalization or stemming can quietly delete the very signal a model needs, a risk that grows sharper once character level features enter the picture, because the patterns inside words are exactly what get sanded away.

Data augmentation has been floated as a remedy for scarce data. Abdhood et al. (2025) reviewed Arabic augmentation methods and weighed their promise against their hazards, including more variety in training but also the chance of unnatural samples or shifted meaning when the process runs unchecked. The lesson reinforces the case for transparent dataset preparation and an evaluation one can believe.

2.2 Classical Machine Learning, TF IDF, and Hybrid Sparse Representation

For all the momentum behind deep learning, classical models have not left the stage, and for Arabic text they remain genuinely competitive. SVMs, Logistic Regression, and Naive Bayes endure because they are efficient, interpretable, and at home with the high dimensional sparse features that text produces. They are also simply practical when a dataset is too small to feed a hungry neural network.

TF IDF is still among the most serviceable representations in this classical toolkit. It weighs how much a term matters within a document and surfaces the words that distinguish one category from another, and in Arabic it works especially well when paired with careful

cleaning and well chosen n gram ranges. Muaad et al. (2022) and Allam et al. (2025) both showed that machine learning can still solve document classification convincingly when the features are designed with care.

A repeated finding is that the representation can matter as much as the classifier behind it. Masadeh et al. (2024) traced how preprocessing and representation choices move Arabic classification results, supporting the view that how text is prepared often decides the outcome before training begins. In the same spirit, Louail et al. (2024) proposed Tasneef, a hybrid representation that fuses different sources of textual information and improves effectiveness, which is direct encouragement for the combined representation examined here.

Hybrid representation is a natural fit for Arabic because the useful signal lives at two depths at once. Word level TF IDF captures the obvious topical vocabulary, the words tied to sport, science, economy, or cars, while character level TF IDF picks up shorter patterns, affixes, spelling variants, and morphological regularities. For a language in which a single word may appear in many forms yet keep a recognizable internal shape, that second layer is not a luxury. The present study revisits this idea critically, asking through an ablation how much each layer truly contributes rather than assuming the fusion is what helps.

SVMs in particular suit sparse text well. Alayba and Altamimi (2025) showed that they perform strongly for Arabic when matched with a meaningful representation. Dense embeddings can capture semantic relations, but sparse representations like TF IDF are easier to interpret and often the better bet on smaller datasets, which is why this study settles on a Linear SVM over word and character level TF IDF.

Feature selection has its own Arabic literature. Alhaj et al. (2022), Larabi-Marie-Sainte and Alalyani (2020), Hadni and Hjiaj (2023), and Hijazi et al. (2023) all explored optimization based selection and showed that pruning noisy or redundant features can help. The same work carries a warning that bears directly on this study. If those feature decisions are allowed to peek at validation or test data, the evaluation tilts in the favor of the model, a leakage in everything but name.

Kowsari et al. (2019) offered a wider survey of classification algorithms across both classical and deep paradigms. Though not Arabic specific, it is useful for situating model choice in a larger frame, and for its core point that the success of a classifier depends on the interplay between data characteristics, representation quality, and model design. That framing supports the comparison drawn here, where a Linear SVM is set against a neural baseline under a leakage aware split.

2.3 Deep Learning and Transformer Based Arabic Classification

Deep learning opened a different route into Arabic classification by learning distributed and contextual representations straight from text. Convolutional networks, recurrent models, hybrid neural designs, attention based models, and transformers have all been turned on Arabic, and their appeal is real. They can model nonlinear patterns and contextual relations that sparse lexical features may never see.

Alsaleh and Larabi-Marie-Sainte (2021) paired convolutional networks with genetic algorithms for Arabic classification, showing that neural architectures can profit from optimization, though at the cost of heavier computation and thinner interpretability. Alnagi et al. (2025) proposed a hybrid framework joining an Inception style CNN with an LSTM, capturing both local patterns and sequence, which is useful but, like most hybrid neural systems, dependent on larger datasets, more tuning, and more compute than a linear model asks for.

Transformers have likewise become central to Arabic NLP. Evidence from low resource settings suggests that pretrained transformers can do well when adapted with care to the target language (Agbesi et al., 2024). Alammary (2022) reviewed BERT based models for Arabic and charted the rising use of pretrained Arabic language models. AraBERT from Antoun et al. (2020) and ARBERT and MARBERT from Abdul-Mageed et al. (2021) are prominent

examples that read contextual meaning more deeply than sparse features can. Alzanin et al. (2023) push the same frontier from another side, tackling Arabic multilabel classification with optimization and ensemble learning and underscoring the appetite for models that can handle complex label structures.

More recent work has aimed transformers and attention at Arabic news and fake news detection. Othman et al. (2024), Fouad et al. (2022), and Hossain et al. (2024) all report strong results on Arabic textual decisions, yet all of them remain hostage to the dataset, the fine tuning recipe, the model size, and the evaluation design. As Minaee et al. (2021) put it in their broad review, complexity by itself buys no guarantee of better performance.

That caveat is exactly why a strong classical baseline deserves a fair fight under identical conditions, and why the neural side of the comparison must itself be fair. A recurrent model trained from randomly initialized embeddings on a moderate corpus is not a meaningful stand in for modern Arabic deep learning. For that reason the present study treats such a model only as a degenerate reference and instead places its neural comparison on fastText, a standard and efficient classifier that learns from subword character information without external pretraining, and it completes the contextual comparison with two pretrained Arabic transformers, AraBERT and MARBERT, each fine tuned for four epochs with a learning rate of 2e-5 and a maximum sequence length of 256 under the identical grouped protocol, so that the comparison with modern Arabic deep learning is conducted on equal footing.

2.4 Evaluation Reliability, Data Leakage, and Reproducibility

Evaluation reliability is one of the load bearing questions in machine learning. A model can look excellent simply because the experimental design let information from the training data seep into validation or testing. Kapoor and Narayanan (2023) named leakage as a leading cause of overoptimistic and irreproducible results, and Starcke et al. (2025) showed how leakage and feature selection choices can warp the very performance estimates researchers rely on.

In text classification the leak is often subtle. The classic case arrives when long documents are cut into segments. Split them at random, and pieces of one document scatter across both training and testing. The classifier then learns to recognize a document, its vocabulary, its cadence, its favored phrases, rather than its category, and accuracy inflates while generalization quietly stays where it was.

Arabic makes this especially live. Documents from the same source tend to share distinctive vocabulary, expressions, morphology, and style, so scattering their segments leaves a test set that no longer represents anything truly unseen. Grouping the split by the original document identifier closes that gap, holding every segment of a source within a single subset. The present study not only adopts this discipline but measures what is at stake when it is ignored, by comparing the grouped protocol directly against the random one on the same data and models.

Leakage aware evaluation, then, is more than a technicality. It changes how the results should be read. A model judged under a grouped split may post lower numbers than the same model under a random row split, but the grouped figure is the more honest one, because it asks the harder question of generalizing to new sources. That conviction is the methodological spine of this study, and Section 4 gives it an empirical backbone. Table 1 summarizes a representative set of recent studies and their relation to the present work.

Table 1. Summary of selected recent studies related to Arabic text classification and evaluation reliability

Authors and Year	Research Focus	Methods or Models	Limitations or Open Issues	Relevance to the Present Study
Wahdan et al. (2024)	Systematic review of Arabic text classification	Review of Arabic classification areas, applications, and future directions	As a review, it offers no controlled experimental comparison under a leakage aware protocol.	Frames the wider landscape and supports the motivation of this study.
Abdulghani and Abdullah (2022)	Survey of Arabic text classification (ML and DL)	Review of ML and DL algorithms for Arabic classification	As a survey, it introduces no new leakage aware protocol or empirical dataset.	Supports the broader positioning of this study.
Einea et al. (2019)	Arabic news corpus construction	SANAD single label Arabic news dataset	Focuses on news and does not address grouped leakage aware splitting.	Informs corpus design, category organization, and benchmarking.
Albalawi et al. (2021)	Arabic health info classification on social media	Preprocessing, pretrained embeddings, KNN, SVM, MNB, Logistic Regression	Centered on health social media text; may not generalize to broad multidomain corpora.	Supports controlled cleaning and careful representation before training.
Elnagar et al. (2020)	Arabic classification with deep learning	DL models on single and multilabel Arabic datasets	Emphasis on DL performance; leakage aware partitioning is not central.	A strong DL reference; motivates a fair neural baseline comparison.
El-Rifai et al. (2022)	Arabic news classification and multilabeling	Logistic Regression and XGBoost, one vs rest multilabel	Multilabel annotation complicates direct single label comparison.	Explains why General and News overlap and confuse.
Masadeh et al. (2024)	Preprocessing and representation effects	ML based Arabic classification	Findings tied to specific datasets, settings, and models.	Supports the focus on controlled preprocessing and representation.
Louail et al. (2024)	Hybrid representation for Arabic	Tasneef hybrid representation	Hybrid design adds complexity and needs careful validation.	Motivates the word and character TF IDF ablation here.
Muaad et al. (2022)	Arabic document classification with ML	ML based document classification	Tied to one dataset; does not target document level leakage.	Supports including classical baselines.
Alayba and Altamimi (2025)	Arabic classification with SVM and embeddings	SVM with Word2Vec, GloVe, fastText	Dense embeddings are less transparent and depend on dataset and protocol.	Directly supports a Linear SVM for Arabic.
Alhaj et al. (2022)	Arabic classification with feature selection	Improved ML and feature selection	Selection is sensitive to the protocol unless restricted to training data.	Supports careful feature design under a controlled protocol.

Authors and Year	Research Focus	Methods or Models	Limitations or Open Issues	Relevance to the Present Study
Larabi-Marie-Sainte and Alalyani (2020)	Feature selection for Arabic	Firefly algorithm feature selection	Optimization based selection adds cost and needs validation control.	Reinforces pruning noisy or redundant features.
Hadni and Hjiaj (2023)	Arabic categorization with feature selection	Chaotic firefly feature selection	Depends on tuning, data, and validation strategy.	Supports feature selection as an active direction.
Hijazi et al. (2023)	Arabic classification with wrapper selection	Artificial bee colony optimization	Wrapper selection is costly and sensitive to validation design.	Supports careful representation and model selection.
Alammary (2022)	BERT models for Arabic	Systematic review of Arabic BERT models	Does not compare them under a document grouped leakage aware protocol.	Positions contextual models as a step beyond sparse comparison.
Antoun et al. (2020)	Arabic language understanding	AraBERT	Needs fine tuning, suitable data, and compute downstream.	Provides AraBERT, one of the two pretrained transformers evaluated under the protocol of this study.
Abdul-Mageed et al. (2021)	Arabic transformer modeling	ARBERT and MARBERT	Performance depends on fine tuning, task data, and compute.	Provides MARBERT, the second pretrained transformer evaluated in this study.
Alsaleh and Larabi-Marie-Sainte (2021)	Neural and optimization methods	CNN and Genetic Algorithms	Adds computational and tuning complexity; less interpretable.	A neural optimization reference point for comparison.
Alnagi et al. (2025)	Hybrid deep learning for Arabic	Inception CNN and LSTM on SANAD and NADiA	Hybrid neural models need more compute, tuning, and data.	Supports comparing hybrid neural and sparse models.
Othman et al. (2024)	Arabic fake news detection	DL based fake news classification	Fake news cues differ from multidomain topic classification.	A recent Arabic DL reference for news related work.
Hossain et al. (2024)	Arabic news classification (transformer)	Hybrid attention transformer with embeddings	Needs more compute and careful fine tuning vs sparse models.	A recent attention based Arabic model that motivates the contextual comparison conducted here.
Alzanin et al. (2023)	Arabic multilabel classification	Genetic Algorithm, ensemble, multilabel design	Multilabel differs methodologically and needs richer annotation.	Supports discussion of label overlap and semantic complexity.

Authors and Year	Research Focus	Methods or Models	Limitations or Open Issues	Relevance to the Present Study
Minaee et al. (2021)	Deep learning text classification	Review of CNN, RNN, LSTM, attention, transformers	General review; not specific to Arabic linguistic properties.	Supports a balanced view; complexity alone is no guarantee.
Kapoor and Narayanan (2023)	Leakage and reproducibility in ML science	Cross domain analysis of leakage failures	Cross domain, not specific to Arabic NLP.	Provides the core rationale for leakage aware splitting.
Starcke et al. (2025)	Leakage and feature selection effects	Empirical analysis of leakage effects	Context is clinical ML, not Arabic text.	Supports blocking information flow into test evaluation.

2.5 Research Gap and Position of the Present Study

Read together, the literature shows a field explored from many angles, from classical learning and deep learning to transformers, feature selection, and hybrid representation, yet a few gaps persist.

The first is a matter of emphasis. Much of the work chases accuracy and spends less attention on whether the evaluation protocol itself is sound. Document level leakage in particular is often named but rarely measured, even though it can dominate the reported numbers the moment documents are segmented. This study answers that gap by quantifying the effect directly.

The second concerns the baselines on both sides. Strong classical models are not always optimized, nor compared under strict leakage aware conditions, and neural baselines are not always fair. A Linear SVM with carefully designed word and character level TF IDF may perform strongly on a moderate corpus where interpretability carries weight, and a fair neural comparison should use a model that can actually learn from the available data.

The third is about granularity and evidence. Many studies report a single aggregate accuracy and stop there, which can mask trouble in semantically broad or overlapping categories, and few support their model rankings with statistical testing. For Arabic multidomain text, where General, News, Technology, and Economy trade vocabulary and style, class level analysis and significance testing are not optional.

Against those gaps, this study evaluates Arabic classification under a grouped split keyed to the original document identifier, measures the inflation that a random split would have produced, examines word and character level TF IDF with a Linear SVM through an ablation, compares it with a fair fastText baseline, and tests the differences for significance. Beyond aggregate metrics, it reports class level F1, the composition of the test set, and a confusion matrix to show, in detail, how the model behaves across Arabic categories.

3. Materials and Methods

This section walks through how the Arabic corpus was prepared, cleaned, divided, turned into features, used to train models, and finally evaluated. One concern shaped every decision, namely to build a pipeline that is reliable and, in particular, one that does not quietly leak information across the document boundary. That is why a simple random row based split was rejected as the evaluation protocol from the outset in favor of a grouped strategy keyed to the original document identifier, so that related segments from the same source never drifted into different subsets. The random split is not discarded entirely, however. It is retained as a controlled point of comparison in Section 4.1, where its purpose is to expose, rather than to hide, the cost of leakage.

3.1 Dataset Construction and Category Organization

The corpus was assembled specifically for this study rather than borrowed from an existing benchmark. It holds 1,000 Arabic text segments spread across seven thematic categories, namely News, Economy, Cars, Technology, Science, General, and Sports, chosen to span genuinely different kinds of Arabic writing, from general news style prose to domain specific, technical, and public information content. The balanced distribution appears in Table 2.

The raw material was gathered from a mix of publicly accessible Arabic web sources rather than a single outlet, including general news portals, domain specific sites, and topical blogs. Mixing source types was deliberate. A corpus drawn from one outlet tends to share a house style and vocabulary that a classifier can exploit, so spanning several kinds of sources keeps the categories from collapsing into a single editorial voice and makes the classification task more representative of real Arabic content. Because the corpus is purpose built rather than a published benchmark, it could be designed from the ground up to support a controlled, leakage aware evaluation, which was the whole point.

Underneath those 1,000 segments sit 183 unique Arabic source documents, selected from a larger pool of 336 candidates. Labeling combined two sources of evidence. For documents drawn from a clearly themed section of their source, such as a sports page or a science portal, the section provided a natural initial category. For the rest, and wherever the source signal was weak or absent, the author assigned the category manually by reading the document and judging its dominant topic. Each document was placed in exactly one category. After segmentation, the resulting labels were reviewed manually to confirm that the segments still reflected the category of their parent document and that the chunking had not produced fragments whose topic had drifted. Ambiguous cases were re examined and corrected against the source document. This review was the safeguard against propagating a mislabeled source into many segments at once. Annotation was carried out by a single annotator, which is acknowledged as a limitation in Section 4.9.

The documents were then segmented into fixed length chunks of roughly 80 words, with the final chunk of each document keeping whatever remained, typically between 41 and 79 words. This produced anywhere from 1 to 38 segments per document, with a mean of about 5.5 and a median of 3, reflecting how unevenly long the original documents were. The number of source documents per category was itself uneven, ranging from only 10 for Technology to 44 for Science, with News 32, Economy 30, Cars 21, General 12, and Sports 34 in between. Left unaddressed, that imbalance would have biased both training and evaluation, so sampling was applied at the segment level after chunking to produce a balanced corpus of 143 segments per category, and 142 for Sports, for 1,000 in total. Table 2 reports both the balanced segment counts and the underlying source document counts, since the latter, not the former, determine how much truly independent evidence each category carries.

The original document identifier was the linchpin of the whole design. Every segment carries the identifier of the document it came from, and every segment from a given document is bound to one partition only. Treating sibling segments as fully independent samples, and then dividing them at random, would have let a model train on part of a document and be tested on the rest, the textbook recipe for document level leakage. Tying segments to their source, and splitting on that tie, is what prevents it.

Table 2. Dataset distribution by category, in balanced segments and underlying source documents

Category	Segments	Source documents
News	143	32
Economy	143	30
Cars	143	21

Category	Segments	Source documents
Technology	143	10
Science	143	44
General	143	12
Sports	142	34
Total	**1000**	**183**

3.2 Arabic Text Cleaning and Normalization

Before any feature was extracted, the Arabic text was cleaned, but cleaned with restraint. The goal was to clear away noise without scrubbing off the linguistic detail that classification depends on. The pipeline removed HTML tags, web links, Arabic diacritics, tatweel elongation, stray symbols, repeated line breaks, and excess whitespace, and stopped there.

What it deliberately did not do was reach for heavy stemming or aggressive normalization. The morphology of Arabic is informative. Prefixes, suffixes, inflections, and spelling variants can each carry a clue about the category of a document. Flattening them away might tidy the text, but it would also blunt the signal, an especially costly trade once character level TF IDF is in play, since that representation feeds on exactly the internal patterns such normalization would erase. Light cleaning, in other words, was a modeling decision as much as a hygiene one, and Section 4.3 shows that it was the right one.

3.3 Leakage Aware Grouped Partitioning

The partitioning step is where the central commitment of the study becomes concrete. As Section 2 argued, leakage in text classification often hides in segmented documents. Divide a long document and scatter its parts, and a model can learn the document rather than its category. The grouped split is the direct antidote.

Concretely, the data was divided by the original document identifier, with every segment sharing an identifier assigned to the same subset and no further. No source document, therefore, could appear in training, validation, and testing at once. To keep the categories proportionate, the division was performed within each category, shuffling the documents of a category and allocating roughly seventy percent to training, fifteen percent to validation, and fifteen percent to testing, all at the document level. The procedure yielded the three subsets in Table 3, and a direct check confirmed that no document identifier was shared across any two subsets. The resulting test set holds 197 segments drawn from 34 source documents.

Each subset had a distinct job. Training fit the models and let them learn classification patterns. Validation drove model selection and parameter tuning. Testing was sealed off and touched only for the final evaluation. Because that test set was carved out by source document, it stands in for genuinely unseen material, a stricter bar than any random row based split would set.

For the controlled comparison in Section 4.1, a second partition was also produced, this time the conventional one, a random row level split stratified by category in the same seventy, fifteen, fifteen proportions, ignoring the document identifier. This is the protocol that a study unaware of leakage would normally use. Holding the data, the models, and the preprocessing fixed, the only difference between the two protocols is whether segments of one document may straddle the boundary between training and testing. Under the random split, 83 of the 183 documents do exactly that, which is what makes the comparison so informative.

Table 3. Experimental partition after grouped splitting by the original document identifier

Partition	Segments	Purpose
Training	682	Fitting the models and learning classification patterns from the prepared Arabic corpus.
Validation	121	Model selection, parameter tuning, and comparison of alternative configurations.
Testing	197	Final, independent evaluation of the trained models only, on 34 unseen source documents.
Total	**1000**	**Complete corpus after grouped splitting.**

3.4 Feature Representation

TF IDF carried the representation, chosen because it is effective, interpretable, and well matched to high dimensional sparse text. Two views were extracted and, in the main model, used together. The word level view used unigrams and bigrams with a minimum document frequency of two and sublinear term frequency scaling, capped at fifteen thousand features. The character level view used the char_wb analyzer over character three to five grams, again with a minimum document frequency of two and sublinear scaling, capped at fifteen thousand features. Word level TF IDF captures the explicit lexical signals, the words and phrases that announce a topic such as sports, science, economy, or cars. Character level TF IDF works a layer down, catching subword sequences, affixes, spelling variants, and morphological regularities. For Arabic that second view earns its place, because the tell for a category often lives inside a word, in a repeated character pattern, rather than in the whole token.

To keep feature extraction itself free of leakage, the TF IDF vectorizers were fit on the training subset alone, and validation and testing were then transformed using those already fitted vectorizers. No statistic from validation or test data was ever allowed to shape the feature space, the same principle as the grouped split, applied one level down.

3.5 Classification Models and Model Selection

Several models were evaluated so that different modeling philosophies could be compared on exactly the same leakage aware footing. The classical contingent, namely Multinomial Naive Bayes, Logistic Regression, and a Linear Support Vector Machine over word level TF IDF, was included because these methods are staples of text classification and set honest baselines for sparse features. Logistic Regression and the Linear SVM used balanced class weighting so that no category was favored simply for being larger in the training fold. These classical baselines used a word level TF IDF representation with raw term frequency capped at ten thousand features, which distinguishes them from the word restricted variant of the proposed pipeline that uses sublinear weighting and a fifteen thousand feature cap.

The proposed model was a Linear SVM over the combined word and character level TF IDF representation. Tuning concentrated on the regularization parameter, with candidate values of 0.5, 1.0, 2.0, and 5.0 compared on the validation set by Macro F1. The configuration that won used word unigrams and bigrams, character three to five grams, and a regularization value of C equal to 0.5. Crucially, the selection was made on validation alone, and the test set played no part in choosing the model, the features, or any hyperparameter.

To learn which view of the text actually carries the signal, two reduced variants of this model were also trained under the identical split and classifier settings, one using only the word level features and one using only the character level features. Their comparison with the full representation forms the ablation in Section 4.3.

All experiments used a single fixed random seed of 42 for the document level splitting, the model fitting, the fastText training, and the bootstrap resampling, so that the classical pipeline is fully deterministic and reproducible. The transformer fine tuning was run on a GPU, where

a small amount of run to run variation, on the order of a single test segment, remains even under a fixed seed, so each transformer result is reported from a single fixed seed run. The pipeline was implemented in Python, using scikit-learn for the classical models, the TF IDF features, and the evaluation metrics, statsmodels for the McNemar test, the fastText library for the neural baseline, and the Transformers library for AraBERT and MARBERT. The exact library versions are pinned in the released code, as noted in the data and code availability statement. Table 4 lists the full configuration of every model evaluated in this study.

Table 4. Main experimental model configurations

Model	Feature representation	Main configuration	Purpose
Multinomial Naive Bayes	Word TF IDF	Word n gram sparse lexical representation	Classical probabilistic baseline
Logistic Regression	Word TF IDF	Balanced class weighting, maximum iteration control	Linear discriminative baseline
Linear SVM	Word TF IDF	Balanced class weighting, linear decision boundary	Strong classical text baseline
Linear SVM (proposed)	Word and character TF IDF	Word unigram and bigram features, character 3 to 5 gram features, C equal to 0.5	Final model under the honest protocol
fastText	Subword character n grams	Character 3 to 5 subwords, word bigrams, 100 dimensions, validation tuned	Fair self contained neural baseline
AraBERT	Pretrained transformer	aubmindlab/bert-base-arabertv02, fine tuned 4 epochs, learning rate 2e-5, max length 256	Pretrained contextual comparison
MARBERT	Pretrained transformer	UBC-NLP/MARBERT, fine tuned 4 epochs, learning rate 2e-5, max length 256	Pretrained contextual comparison
BiLSTM	Token sequences, random embeddings	Vocabulary 15,000; length 120; embedding 128; bidirectional LSTM; Adam	Degenerate recurrent reference

3.6 Statistical Analysis and Residual Leakage Check

Because the categories are backed by a limited number of independent documents, reporting point estimates alone would overstate how precisely the models are known, and would invite unjustified claims that one configuration beats another. Two procedures guard against this. First, pairwise differences between the strongest model and its closest competitors were tested with the McNemar exact test on the grouped test set, which compares the cases that one model classifies correctly while the other does not. Second, confidence intervals were estimated with a bootstrap that resamples whole documents rather than individual segments, with three thousand resamples. The clustering matters, because the 197 test segments are not independent. They cluster within 34 documents, and a naive segment level bootstrap would report intervals that are far too narrow. The same document clustered procedure was used to place an interval on the difference in accuracy between the proposed model and its closest rivals.

A separate concern is whether near duplicate documents might defeat the grouped split, for instance a wire story republished across outlets and landing in different partitions. To rule this out, a word level TF IDF profile was built for each source document, and the cosine similarity was computed between all pairs of distinct documents. The check looks for any pair whose

similarity is high enough to suggest near duplication, and in particular for any such pair that straddles the boundary between subsets.

3.7 Evaluation Metrics

Five measures framed the evaluation, namely Accuracy, Macro Precision, Macro Recall, Macro F1, and Weighted F1. Accuracy gives the overall share of correct predictions. The macro averaged metrics matter because they weigh every category equally, regardless of how many samples it holds, which is useful here, where some categories are simply easier to separate than others. Weighted F1 was reported as well, since it accounts for class size, but it was never trusted on its own, because a model can post a flattering Weighted F1 by excelling on the large, easy categories while quietly failing the hard ones. Macro F1 and the per class F1 scores were included precisely to keep that failure mode visible. A confusion matrix completed the picture, since it shows not only how well the model did but where, and why, it stumbled, which earns its keep in Arabic, where categories blur, with General drifting toward News and Technology brushing against Science.

4. Results and Discussion

The discussion moves from the cost of leakage, to honest performance under the grouped protocol, to an ablation that isolates each feature view, to a statistical reading of the differences, then to class level behavior, the General category, and a residual leakage check, before drawing the findings together. Throughout, it helps to remember that, except where the random split is named explicitly for comparison, every number reflects behavior on unseen source documents rather than on familiar segments wearing a new label.

4.1 The Cost of Leakage: Random versus Grouped Evaluation

The most important result in this study is not which model wins, but how much the answer changes with the evaluation protocol. Table 5 reports every model twice, once under the conventional random split and once under the grouped split, with the data, the preprocessing, and the model settings held fixed. The only thing that changes is whether segments of one document may appear on both sides of the boundary between training and testing.

Table 5. Effect of document level leakage. The same models under a random split and a grouped split

Model	Random accuracy	Random Macro F1	Grouped accuracy	Inflation
Naive Bayes	0.9667	0.9661	0.6650	+30.2
Logistic Regression	0.9667	0.9657	0.8071	+16.0
Linear SVM (word)	0.9733	0.9728	0.8223	+15.1
Linear SVM (word and char)	0.9800	0.9796	0.8528	+12.7
Character only SVM	0.9800	0.9796	0.8528	+12.7
fastText (subword)	0.9533	0.9536	0.6497	+30.4
AraBERT	0.9667	0.9660	0.8477	+11.9
MARBERT	0.9667	0.9660	0.8426	+12.4

Inflation is the gain in accuracy points from leakage, computed as random accuracy minus grouped accuracy.

Note. The character only and combined rows coincide on accuracy and on random split Macro F1 because the two models agree on all but four of the 197 test segments; under the grouped split they differ on Macro F1, as reported in Table 6.

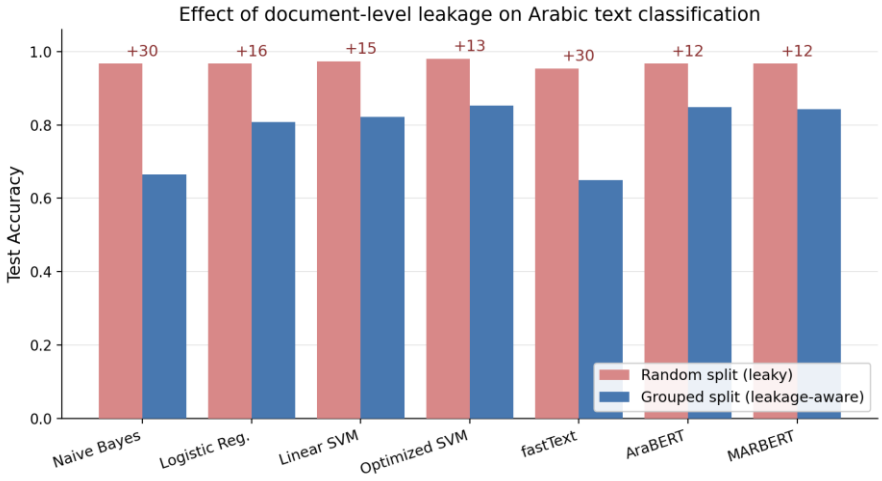

Figure 1. Effect of document level leakage.

Under a random split every model appears far stronger than it is, and the inflation in accuracy points is annotated above each pair. In the figure, Optimized SVM denotes the proposed Linear SVM over word and character TF IDF.

Two things stand out. The first is the sheer size of the distortion. Under the random split the proposed model appears to reach 0.9800 accuracy, a number that would not look out of place in a strong results table. Under the grouped split the very same model, trained and tested on the very same corpus, reaches 0.8528. The 12.7 point gap is attributable to leakage, and it is the smallest gap in the table. The simpler and more memory reliant models inflate far more, with Naive Bayes and fastText each gaining about thirty points, because models that lean hardest on memorized vocabulary benefit most when fragments of a remembered document reappear in the test set. The two pretrained transformers are not immune either, inflating by 11.9 points for AraBERT and 12.4 points for MARBERT, so leakage misleads contextual models much as it does sparse ones.

The second observation is subtler and, for practice, more dangerous. Leakage does not merely lift the scores. It flattens the ranking among models. Under the random split, Naive Bayes reaches 0.9667 and sits almost level with the proposed model at 0.9800, a difference of about one point. Under the honest grouped split the same two models stand 18.8 points apart. A researcher who selected a model under the leaky protocol could therefore choose a far weaker classifier while believing it to be competitive. Leakage, in other words, is not only a scoring hazard but a model selection hazard, and that is the strongest practical reason to take it seriously.

4.2 Honest Performance under the Grouped Protocol

With the protocol settled, Table 6 reports the full set of metrics under the grouped split. These are the honest numbers, and the rest of the paper builds on them.

Table 6. Performance under the grouped protocol, with full averaged metrics

Model	Accuracy	Macro P	Macro R	Macro F1	Weighted F1
Naive Bayes	0.6650	0.6844	0.6799	0.6443	0.6923
Logistic Regression	0.8071	0.7422	0.7287	0.7094	0.7926
Linear SVM (word)	0.8223	0.7675	0.7407	0.7272	0.8092
Word only TF IDF SVM	0.8223	0.7641	0.7407	0.7265	0.8079
Character only TF IDF SVM	0.8528	0.8303	0.7921	0.7801	0.8373

Model	Accuracy	Macro P	Macro R	Macro F1	Weighted F1
Linear SVM (word and char)	**0.8528**	**0.8432**	**0.7880**	**0.7713**	**0.8361**
fastText (subword)	0.6497	0.6185	0.5983	0.5820	0.6653
AraBERT	0.8477	0.7322	0.7879	0.7428	0.8058
MARBERT	0.8426	0.7815	0.7901	0.7611	0.8145
BiLSTM (reference)	0.1624	n/a	n/a	0.0393	0.0454

Note. Linear SVM (word) and Word only TF IDF SVM are not identical. The former is the classical word level baseline, capped at 10,000 word features with raw term frequency and the default regularization. The latter is the word restricted variant of the proposed pipeline, capped at 15,000 word features with sublinear term frequency and C equal to 0.5. They differ only in these settings, which is why their scores are close but not the same.

The proposed Linear SVM over word and character TF IDF leads the honest table with 0.8528 accuracy, 0.7713 Macro F1, and 0.8361 Weighted F1. The plain Linear SVM over word features is not far behind at 0.8223 accuracy, and Logistic Regression stays competitive at 0.8071, confirming that linear classifiers remain useful when the features are informative. Naive Bayes trails at 0.6650, most likely because its strong independence assumptions miss the relationships among Arabic textual features.

The two pretrained Arabic transformers, fine tuned under the same grouped protocol, do not change this picture. AraBERT reaches 0.8477 accuracy with 0.7428 Macro F1, and MARBERT reaches 0.8426 accuracy with 0.7611 Macro F1, both a little below the proposed model on accuracy and on Macro F1. Both contextual models perform comparably to the character level Linear SVM on this corpus rather than surpassing it, and the character only model retains the highest Macro F1 of any model tested. This is consistent with the central reading of the study, that once leakage is controlled, a sparse subword representation paired with a linear classifier is sufficient for this task, and on the point estimates reported here, the added cost of a pretrained transformer brings no gain over the linear model, though the transformer comparison was not subjected to the same significance testing.

The fastText baseline reaches 0.6497 accuracy and 0.5820 Macro F1. This places a fair neural method clearly below the sparse linear models on this moderate corpus, yet far above the recurrent reference, which is the point of including it. The bidirectional LSTM with random embeddings sits at 0.1624 accuracy, barely above the level a seven class problem would reach by chance. That collapse should be read carefully. It is not evidence that recurrent models are unfit for Arabic. It is evidence that, starting from randomly initialized embeddings on a corpus of this size, such a model cannot generalize, and that a fair neural comparison needs either subword features, as fastText has, or pretrained representations, which the recurrent reference lacked.

4.3 Ablation: Character Level Features Carry the Signal

A natural question hangs over the leading result. The proposed model fuses two kinds of features, but which kind is actually doing the work? To answer it, the representation was dismantled and rebuilt one piece at a time, holding everything else fixed, namely the same Linear SVM with C equal to 0.5 and balanced weights, and the same grouped split. Table 7 collects the outcome.

Table 7. Ablation. The effect of word and character level TF IDF on grouped test performance

Representation	Accuracy	Macro P	Macro F1	Weighted F1
Word level TF IDF only	0.8223	0.7641	0.7265	0.8079
Character level TF IDF only	0.8528	0.8303	0.7801	0.8373
Word and character TF IDF	**0.8528**	**0.8432**	**0.7713**	**0.8361**

The breakdown is clarifying, and a little humbling for the hybrid framing. The character level features are the engine of the result. On their own they reach 0.8528 accuracy and 0.7801 Macro F1, matching the full representation on accuracy and, in fact, edging past it on Macro F1, 0.7801 against 0.7713. Word level features alone manage 0.8223 accuracy and 0.7265 Macro F1, essentially the behavior of a lexical overlap baseline. Almost all of the lift over the word only model comes from the character side, where Arabic affixes, clitics, and morphological variants evidently carry the bulk of the discriminative signal.

That does not make the word level features pointless, but it does reposition them. The combined representation posts the highest Macro Precision of the three, 0.8432 against 0.8303 for character only, which says the word level view contributes complementary topical cues that refine precision even when it adds little to the aggregate F1. The honest summary, then, is this. The combination does not beat the stronger single view so much as match it while rounding out precision, and the character level component is what makes the model work. For a morphologically rich language like Arabic, that is a result worth stating plainly rather than hiding behind the word hybrid. It is also why the combined model is retained, not because the fusion is magic, but because it equals the best single view, gives the most balanced precision, with only a negligible difference in recall.

4.4 Are the Differences Real? Statistical Analysis

The honest table shows the proposed model a few points above the plain Linear SVM and level with the character only model. Before reading anything into those gaps, they must be tested, because the effective sample is small. Table 8 brings together the McNemar exact tests against the closest competitors and the document clustered bootstrap intervals.

Table 8. Statistical analysis on the grouped test set. McNemar tests and document clustered bootstrap intervals

Comparison or model	McNemar p	Accuracy, 95 percent interval	Macro F1, 95 percent interval
Proposed vs character only	1.000	n/a	n/a
Proposed vs Linear SVM (word)	0.180	n/a	n/a
Proposed vs Logistic Regression	0.049	n/a	n/a
Proposed (word and char)	n/a	0.855 [0.716, 0.945]	0.750 [0.644, 0.880]
Character only	n/a	0.856 [0.732, 0.942]	0.752 [0.645, 0.862]
Linear SVM (word)	n/a	0.825 [0.676, 0.924]	0.704 [0.590, 0.813]
Logistic Regression	n/a	0.809 [0.669, 0.913]	0.689 [0.578, 0.793]

Note. The accuracy and Macro F1 entries are bootstrap point estimates over document level resamples and may differ slightly from the observed values in Table 6. The McNemar tests use the observed predictions on the grouped test set.

The statistical picture is consistent and, for the framing of the paper, clarifying. The proposed model is statistically indistinguishable from the character only model, with a McNemar p value of 1.000, since the two disagree on only four of the 197 test segments, two in each

direction. It is also not significantly better than the plain Linear SVM over word features, where the p value is 0.180. Only against Logistic Regression does the difference approach significance, at a borderline 0.049. Because three pairwise comparisons were carried out, even this borderline value should be read with caution, since it would not remain significant under a conservative correction for multiple comparisons. The bootstrap tells the same story from another angle. The intervals are wide, spanning roughly eleven accuracy points on either side of the estimate, because the effective sample is 34 documents rather than 197 segments, and the intervals of the strongest models overlap almost entirely. The paired difference in accuracy between the proposed model and the plain Linear SVM is about 0.031 with a ninety five percent interval from negative 0.010 to positive 0.078, which includes zero.

The right conclusion is therefore not that one configuration wins, but that a family of strong linear models over TF IDF features, namely the Linear SVM over word features, the character only model, and the combined model, perform at a statistically indistinguishable level, all clearly above the fair neural baseline. On a test set backed by 34 documents, the small gaps among them are within the noise, and the value of the proposed representation lies in its balanced precision rather than in any decisive margin. By the same logic, the more parsimonious character only model is an equally defensible choice on this corpus, since it attains the highest Macro F1 and is statistically indistinguishable from the combined model; the combined representation is reported as the primary configuration only for its more balanced precision and recall, not for any advantage in accuracy. This restraint is itself a contribution, because it is exactly the kind of claim that a leakage aware study should be willing to make.

4.5 Class Level Behavior and the Fragility of Per Class Numbers

Aggregate metrics hide a wide spread across categories, and reading that spread honestly requires knowing how much independent evidence each category carries in the test set. Table 9 reports, for the proposed model, the per class precision, recall, and F1, alongside the number of test documents and test segments behind each figure.

Table 9. Per class results of the proposed model and the composition of the test set

Category	Test documents	Test segments	Precision	Recall	F1
News	6	31	0.674	1.000	0.805
Economy	5	15	0.545	0.800	0.649
Cars	4	60	0.983	0.950	0.966
Technology	2	13	0.700	0.538	0.609
Science	8	24	1.000	1.000	1.000
General	3	22	1.000	0.227	0.370
Sports	6	32	1.000	1.000	1.000

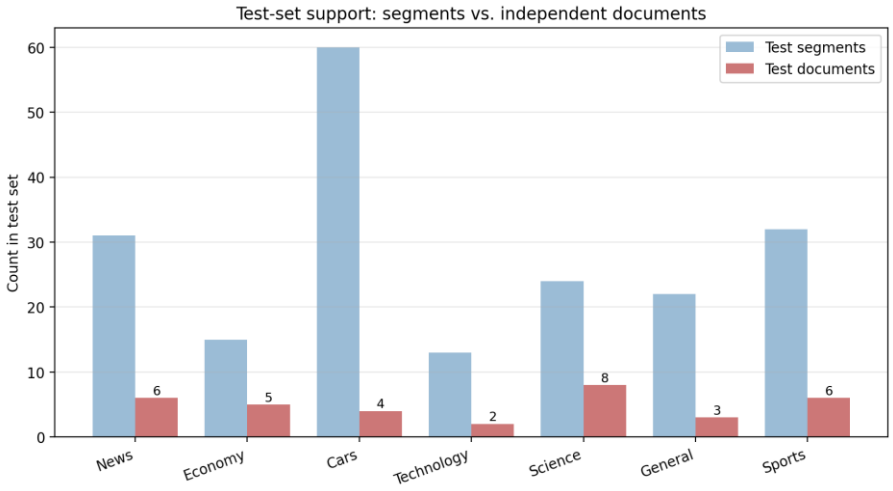

Figure 2. Test set support by category.

The number of independent documents behind each score, shown against the number of segments, makes clear how few sources some categories rest on.

The spread is wide. Science and Sports reach an F1 of 1.000, with Cars close behind at 0.966, which shows that the character rich representation thrives where lexical and subword patterns are sharp and repeated. Economy and Technology land in the middle at 0.649 and 0.609, both trading vocabulary and tone with neighbors. News reaches 0.805 but with a revealing imbalance, a perfect recall of 1.000 against a precision of only 0.674, which means other categories are being misclassified into News rather than News being missed.

Table 9 also issues a clear warning about precision in the statistical sense. Several of these scores rest on very few independent documents. Technology, with an F1 of 0.609, is backed by only two test documents. Cars, with its strong 0.966, draws on just four documents that happen to contribute sixty highly correlated segments. Even the perfect Science and Sports scores rest on eight and six documents. These numbers describe behavior on a handful of held out sources, not stable, dataset independent class accuracies, and they should be read in that spirit. Reporting the document counts beside the scores is what keeps the reader from over interpreting them.

The confusion matrix in Table 10 shows where the errors actually fell. The dominant pattern is the entanglement of General with News. Of the 22 General test segments, 14 were predicted as News, which simultaneously explains the low recall of General, at 0.227, and the depressed precision of News, since those General segments arrive as false positives in the News column. The remaining errors are minor by comparison, with a few Technology segments drifting into Economy. Science, Sports, and the bulk of Cars and News separate cleanly.

Table 10. Confusion matrix of the proposed model on the grouped test set. Rows are true categories, columns are predictions

True \ Pred	News	Econ	Cars	Tech	Sci	Gen	Sports
News	31	0	0	0	0	0	0
Economy	1	12	1	1	0	0	0
Cars	0	3	57	0	0	0	0
Technology	0	6	0	7	0	0	0
Science	0	0	0	0	24	0	0
General	14	1	0	2	0	5	0

True \ Pred	News	Econ	Cars	Tech	Sci	Gen	Sports
Sports	0	0	0	0	0	0	32

4.6 The General Category as a Structural Confound: A Six Class Experiment

The class level analysis points to a single culprit. General is not merely a difficult category. It behaves like a miscellaneous bin that absorbs and confuses news style writing, and the cost is paid by its neighbors as much as by itself. To test this directly, the General category was removed and the proposed model was retrained and evaluated under the same grouped protocol on the remaining six categories. If General were simply a hard but well defined class, removing it would delete a low score and little else. If it were a structural confound, removing it would also lift the categories it had been contaminating.

The second prediction is what happens. With General removed, overall accuracy rises from 0.8528 to 0.9349, and Macro F1 climbs from 0.7713 to 0.8852. The gain is not evenly spread, which is the telling part. The News F1 jumps from 0.805 to 0.984, since the segments that General had been bleeding into News are no longer there to create false positives. Technology improves from 0.609 to 0.667 and Economy from 0.649 to 0.686, while Cars, Science, and Sports stay at or near their already high levels. Table 11 collects these figures.

Table 11. Removing the General category. Effect on the proposed model under the grouped protocol

Metric or category	Seven classes	Six classes	Change
Overall accuracy	**0.8528**	**0.9349**	**+0.082**
Overall Macro F1	**0.7713**	**0.8852**	**+0.114**
News F1	0.805	0.984	+0.179
Technology F1	0.609	0.667	+0.058
Economy F1	0.649	0.686	+0.037
Cars F1	0.966	0.975	+0.009
Science F1	1.000	1.000	0.000
Sports F1	1.000	1.000	0.000

This is strong evidence that the weakness of General is a label design problem rather than a modeling one. As currently defined, General admits text tethered to no single domain and overlapping in register with News, Economy, and Technology, which makes it far less lexically stable than a tight category like Sports or Cars. The practical implication is concrete. For this corpus, the cleaner six class problem is both a stronger and an equally honest result, and the longer term remedy is to redraw the General label, split it into sharper subcategories, or move to a multilabel setting for documents that genuinely span more than one topic.

4.7 Ruling Out Residual Leakage: Near Duplicate Check

A grouped split closes the obvious leak, but it would still be defeated by near duplicate documents, for example a wire story republished across outlets and assigned to different subsets. To rule this out, the cosine similarity between the word level TF IDF profiles of all pairs of distinct documents was examined. No pair of documents reached a cosine similarity of 0.7, and therefore none reached the higher thresholds of 0.8 or 0.9 either, and no near duplicate pair straddled the boundary between subsets. The grouped test set is therefore free of near duplicate contamination, which means its difficulty is genuine and the honest scores are not quietly inflated by hidden source level repetition.

4.8 Discussion of Main Findings

Stepping back, the findings form a coherent picture. The first and most important is methodological. On this Arabic corpus, document level leakage inflates accuracy by between roughly 11.9 and 30.4 points and, worse, collapses the ranking among models, so a study that

splits at random risks both an inflated score and a wrong model choice. The grouped split keyed to the original document identifier removes that distortion and asks the harder, more honest question of generalizing to unseen sources. This aligns with the wider literature that names leakage a prime source of overoptimistic and irreproducible results (Kapoor & Narayanan, 2023; Starcke et al., 2025), and it turns that general warning into a measured effect on a concrete Arabic task.

The second finding is about representation. Under the honest protocol, the work of classification is done at the subword level. Character level TF IDF alone matches the combined representation and slightly exceeds it on Macro F1, while word level features mainly refine precision. The takeaway is less to combine everything than to represent Arabic at the subword level, a reading consistent with work emphasizing the weight of preprocessing and representation in Arabic classification (Masadeh et al., 2024; Louail et al., 2024).

The third finding is a counsel of restraint. The strongest linear models are statistically indistinguishable from one another, so the paper does not crown a single winner. It reports instead a family of strong, cheap, interpretable linear models, all clearly above a fair neural baseline. The recurrent reference faltered not because neural models are unfit for Arabic but because it lacked the prerequisites of subword features or pretrained representations, a reading in line with the broader text classification literature on what deep learning needs to succeed (Alnagi et al., 2025; Minaee et al., 2021), and with the fact that fastText, which does use subword features, performed far better than the recurrent reference.

The fourth finding is about the data itself. The task is not uniformly hard. Science, Sports, and Cars separate cleanly on distinctive lexical and subword patterns, while General is a structural confound whose removal lifts the whole system and, in particular, rescues News, an echo of earlier Arabic work on overlapping category boundaries in news and general domains (El-Rifai et al., 2022). Taken as a whole, the results suggest that leakage aware evaluation, paired with subword rich TF IDF and a Linear SVM, makes a strong and practical baseline for Arabic text classification, accurate enough to be useful, interpretable enough to be trusted, and cheap enough to run anywhere. This conclusion already withstands a contextual test. Two pretrained Arabic transformers, AraBERT and MARBERT, evaluated under the same grouped protocol, perform comparably to the linear model rather than surpassing it, with the linear model matching them while remaining far cheaper and fully interpretable, so the added capacity of a pretrained transformer yields no measurable advantage on this corpus once leakage is removed.

4.9 Limitations

For all its controls, the study has limits worth stating openly. The corpus holds 1,000 segments drawn from 183 documents, ample for a focused and carefully controlled experiment but still narrow for sweeping claims across every Arabic domain. More pointedly, the grouped test set rests on only 34 documents, which is why the bootstrap intervals are wide and why the per class scores, especially the perfect ones, should be read as behavior on a handful of sources rather than as stable accuracies. Broadening the corpus with more verified sources and a wider topic range, and on the scale of established benchmarks, is the obvious next step.

Annotation relied on a single annotator, supported by source section signals and a manual post segmentation review. While ambiguous cases were corrected against the source document, a second independent annotator and a measure of inter annotator agreement would strengthen confidence in the labels, particularly for the broad General category.

The neural comparison was strengthened with fastText as a self contained subword baseline and with two pretrained Arabic transformers, AraBERT and MARBERT, fine tuned under the same grouped protocol, while a bidirectional LSTM from randomly initialized embeddings is reported only as a degenerate reference. The contextual comparison nevertheless remains limited to these two models under a single fine tuning configuration, and because the

transformers were fine tuned on a GPU their results carry minor run to run variation, so each is reported from a single fixed seed run. Extending the comparison to further Arabic models and fine tuning strategies is left to future work.

Finally, the General category remained a structural difficulty. The six class experiment shows that removing it lifts the whole system, but the more durable remedy is a redrawn label, sharper subcategories, or a multilabel formulation for documents that genuinely span topics, which this study identifies but does not yet implement.

5. Conclusion and Future Work

This study set out to evaluate Arabic text classification honestly, not merely to crown a model but to ask how much more trustworthy the evaluation becomes once document level overlap is brought under control, and how large the distortion is when it is not. The answer ran through a grouped partitioning strategy keyed to the original document identifier, which confined every segment of a source document to a single subset and so cut off the leakage that quietly flatters segmented document experiments. Measured directly against a conventional random split on the same data and models, that leakage was shown to inflate accuracy by between roughly 11.9 and 30.4 points and, more insidiously, to collapse the ranking among models, so that a weak classifier could be mistaken for a strong one.

Under the honest protocol, a Linear SVM over word and character level TF IDF gave the strongest performance of the models tested, with a test accuracy of 0.8528, a Macro F1 of 0.7713, and a Weighted F1 of 0.8361. A feature level ablation refined that story usefully, showing that the character level component carries most of the signal, matching the full representation on its own and slightly exceeding it on Macro F1, while the word level features mainly improve precision. Statistical testing then counselled restraint, since the strongest linear models proved statistically indistinguishable from one another, all comfortably above a fair fastText baseline that itself far outperformed a degenerate recurrent reference. The class level picture was just as instructive as the aggregate one. Science, Sports, and Cars separated cleanly, while General behaved as a structural confound whose removal raised overall accuracy to 0.9349 and lifted the News F1 from 0.805 to 0.984.

The central contribution is the way these pieces fit together, namely controlled Arabic preprocessing, a critically examined word and character level TF IDF representation, a leakage aware grouped evaluation with the cost of leakage actually measured, fair baselines on both the classical and neural sides, and statistical testing that keeps the claims modest. That combination yields a more credible read on Arabic classification performance, and it shows that a well built classical model can stand as a strong, interpretable, and efficient baseline for multidomain Arabic text, provided the evaluation is honest about what it measures.

Several roads lead onward. The corpus can grow, drawing on additional verified Arabic sources to widen category diversity and strengthen external generalization, ideally to the scale of established benchmarks. The contextual comparison can be widened further, for instance to ARBERT, to domain adapted or larger Arabic models, and to longer fine tuning budgets, to confirm whether the parity observed here for AraBERT and MARBERT persists for stronger pretrained representations (Alammary, 2022; Antoun et al., 2020; Abdul-Mageed et al., 2021). Multilabel classification invites exploration for the many Arabic documents that genuinely carry more than one topic, especially in news and general content (El-Rifai et al., 2022). And the most stubborn categories, General and News above all, warrant a closer look at label design, which may be the surest route to better separability in multidomain Arabic text classification.

Data and Code Availability

The dataset, model configurations, and source code used in this study are publicly available at https://doi.org/10.6084/m9.figshare.32751555. The repository contains the cleaned dataset, experimental protocols, and implementation scripts required to reproduce the experiments and

results reported in this paper. All materials are provided to support transparency, reproducibility, and future research in Arabic text classification .

References

[1]Abdhood, S. F., Omar, N., & Tiun, S. (2025). Data augmentation for Arabic text classification: A review of current methods, challenges and prospective directions. PeerJ Computer Science, 11, Article e2685. https://doi.org/10.7717/peerj-cs.2685

[2]Abdul-Mageed, M., Elmadany, A., & Nagoudi, E. M. B. (2021). ARBERT and MARBERT: Deep bidirectional transformers for Arabic. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (pp. 7088–7105). https://doi.org/10.18653/v1/2021.acl-long.551

[3]Abdulghani, F. A., & Abdullah, N. A. Z. (2022). A survey on Arabic text classification using deep and machine learning algorithms. Iraqi Journal of Science, 63(1), 409–419. https://doi.org/10.24996/ijs.2022.63.1.37

[4]Agbesi, V. K., Chen, W., Yussif, S. B., Hossin, M. A., Ukwuoma, C. C., Kuadey, N. A., Agbesi, C. C., Abdel Samee, N., Jamjoom, M. M., & Al-antari, M. A. (2024). Pre-trained transformer-based models for text classification using low-resourced Ewe language. Systems, 12(1), Article 1. https://doi.org/10.3390/systems12010001

[5]Alammary, A. S. (2022). BERT models for Arabic text classification: A systematic review. Applied Sciences, 12(11), Article 5720. https://doi.org/10.3390/app12115720

[6]Alayba, A. M., & Altamimi, M. (2025). Optimization of Arabic text classification using SVM integrated with word embedding models on a novel dataset. International Journal of Advanced and Applied Sciences, 12(9), 140–151. https://doi.org/10.21833/ijaas.2025.09.013

[7]Albalawi, Y., Buckley, J., & Nikolov, N. S. (2021). Investigating the impact of preprocessing techniques and pretrained word embeddings in detecting Arabic health information on social media. Journal of Big Data, 8, Article 95. https://doi.org/10.1186/s40537-021-00488-w

[8]Alhaj, Y. A., Dahou, A., Al-qaness, M. A. A., Abualigah, L., Abbasi, A. A., Almaweri, N. A. O., Abd Elaziz, M., & Damaševičius, R. (2022). A novel text classification technique using improved particle swarm optimization: A case study of Arabic language. Future Internet, 14(7), Article 194. https://doi.org/10.3390/fi14070194

[9]Allam, H., Makubvure, L., Gyamfi, B., Graham, K. N., & Akinwolere, K. (2025). Text classification: How machine learning is revolutionizing text categorization. Information, 16(2), Article 130. https://doi.org/10.3390/info16020130

[10]Alnagi, E., Ghnemat, R., & Abu Al-Haija, Q. (2025). Boosting Arabic text classification using hybrid deep learning approach. Discover Applied Sciences, 7, Article 540. https://doi.org/10.1007/s42452-025-07025-x

[11]Alsaleh, D., & Larabi-Marie-Sainte, S. (2021). Arabic text classification using convolutional neural network and genetic algorithms. IEEE Access, 9, 91670–91685. https://doi.org/10.1109/ACCESS.2021.3091376

[12]Alzanin, S. M., Gumaei, A., Haque, M. A., & Muaad, A. Y. (2023). An optimized Arabic multilabel text classification approach using genetic algorithm and ensemble learning. Applied Sciences, 13(18), Article 10264. https://doi.org/10.3390/app131810264

[13]Antoun, W., Baly, F., & Hajj, H. (2020). AraBERT: Transformer based model for Arabic language understanding. In Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools (pp. 9–15).

[14]Einea, O., Elnagar, A., & Al Debsi, R. (2019). SANAD: Single label Arabic news articles dataset for automatic text categorization. Data in Brief, 25, Article 104076. https://doi.org/10.1016/j.dib.2019.104076

[15]El-Rifai, H., Al Qadi, L., & Elnagar, A. (2022). Arabic text classification: The need for multi-labeling systems. Neural Computing and Applications, 34(2), 1135–1159. https://doi.org/10.1007/s00521-021-06390-z

[16]Elnagar, A., Al-Debsi, R., & Einea, O. (2020). Arabic text classification using deep learning models. Information Processing and Management, 57(1), Article 102121. https://doi.org/10.1016/j.ipm.2019.102121

[17]Fouad, K. M., Sabbeh, S. F., & Medhat, W. (2022). Arabic fake news detection using deep learning. Computers, Materials and Continua, 71(2), 3647–3665. https://doi.org/10.32604/cmc.2022.021449

[18]Hadni, M., & Hjiaj, H. (2023). New model of feature selection based chaotic firefly algorithm for Arabic text categorization. The International Arab Journal of Information Technology, 20(3A), 461–468. https://doi.org/10.34028/iajit/20/3A/3

[19]Hijazi, M., Zeki, A. M., & Ismail, A. R. (2023). Utilizing artificial bee colony algorithm as feature selection method in Arabic text classification. The International Arab Journal of Information Technology, 20(3A), 536–547. https://doi.org/10.34028/iajit/20/3A/11

[20]Hossain, M. M., Hossain, M. S., Safran, M., Alfarhood, S., Alfarhood, M., & Mridha, M. F. (2024). A hybrid attention based transformer model for Arabic news classification using text embedding and deep learning. IEEE Access, 12, 198046–198066. https://doi.org/10.1109/ACCESS.2024.3522061

[21]Kapoor, S., & Narayanan, A. (2023). Leakage and the reproducibility crisis in machine-learning-based science. Patterns, 4(9), Article 100804. https://doi.org/10.1016/j.patter.2023.100804

[22]Kowsari, K., Jafari-Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L. E., & Brown, D. E. (2019). Text classification algorithms: A survey. Information, 10(4), Article 150. https://doi.org/10.3390/info10040150

[23]Larabi-Marie-Sainte, S., & Alalyani, N. (2020). Firefly algorithm based feature selection for Arabic text classification. Journal of King Saud University, Computer and Information Sciences, 32(3), 320–328. https://doi.org/10.1016/j.jksuci.2018.06.004

[24]Louail, M., Kara-Mohamed Hamdi-Cherif, C., & Hamdi-Cherif, A. (2024). Tasneef: A fast and effective hybrid representation approach for Arabic text classification. IEEE Access, 12, 120804–120826. https://doi.org/10.1109/ACCESS.2024.3450507

[25]Masadeh, M., Moustapha, A., Sharada, B., Hanumanthappa, J., Hemachandran, K., Chola, C., & Muaad, A. Y. (2024). Investigating the impact of preprocessing techniques and representation models on Arabic text classification using machine learning. International Journal of Advanced Computer Science and Applications, 15(1), 1115–1123. https://doi.org/10.14569/IJACSA.2024.01501110

[26]Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep learning based text classification: A comprehensive review. ACM Computing Surveys, 54(3), Article 62. https://doi.org/10.1145/3439726

[27]Muaad, A. Y., Kumar, G. H., Hanumanthappa, J., Benifa, J. V. B., Mourya, M. N., Chola, C., Pramodha, M., & Bhairava, R. (2022). An effective approach for Arabic document classification using machine learning. Global Transitions Proceedings, 3(1), 267–271. https://doi.org/10.1016/j.gltp.2022.03.003

[28]Othman, N. A., Elzanfaly, D. S., & Elhawary, M. M. M. (2024). Arabic fake news detection using deep learning. IEEE Access, 12, 122363–122376. https://doi.org/10.1109/ACCESS.2024.3451128

[29]Starcke, J., Spadafora, J., Spadafora, J., Spadafora, P., & Toma, M. (2025). The effect of data leakage and feature selection on machine learning performance for early Parkinson's disease detection. Bioengineering, 12(8), Article 845. https://doi.org/10.3390/bioengineering12080845

[30]Wahdan, A., Al-Emran, M., & Shaalan, K. (2024). A systematic review of Arabic text classification: Areas, applications, and future directions. Soft Computing, 28(2), 1545–1566. https://doi.org/10.1007/s00500-023-08384-6