Arabic Sentiment Analysis Using Mixup Data Augmentation Mixup
تحليل المشاعر في اللغة العربية باستخدام تقنية توسعة البيانات

Alia Hamwi: Faculty of Information Technology Engineering - Damascus University - Damascus - Syria.
Maisaa Aboukassem: Faculty of Information Technology Engineering - Damascus University - Damascus - Syria.
Nada Ghneim: Faculty of Informatics & Communication Engineering - Arab International University - Damascus - Syria.
Corresponding: alia.hamwi@damascusuniversity.edu.sy

ABSTRACT

Mixup, as a technique for augmenting data within the feature space, operates by applying linear interpolation to input instances and their associated modeling targets derived from randomly selected samples. The efficacy of this method in substantially enhancing the predictive accuracy of cutting-edge networks has been established across both image and text classification tasks. Despite its demonstrated success in various contexts, its application within the context of the Arabic language remains an unexplored area of research. This study employed three strategies to adapt Mixup for application in Arabic sentiment analysis. Experimental evaluations were conducted to assess the effectiveness of these strategies, utilizing a range of benchmark datasets. Our studies demonstrate that these interpolation strategies effectively function as domain-independent methods for augmenting data, in the context of text classification. Furthermore, these strategies have the potential to lead to enhancements in performance for both convolutional neural network (CNN) and long short-term memory (LSTM) models.

Keywords: Text Classification, Sentiment Analysis, Data Augmentation, Mixup Augmentation.

الملخص

تعمل تقنية Mixup، كونها تقنية لزيادة البيانات داخل فضاء السمات، من خلال تطبيق الاستيفاء الخطي على مثيلات الإدخال وأهداف النمذجة المرتبطة بها المستمدة من عينات مختارة عشوائياً. أثبتت فاعلية هذه الطريقة في تعزيز الدقة التنبؤية للشبكات المتطورة بشكل كبير عبر مهام تصنيف الصور والنصوص. وعلى الرغم من نجاحه الواضح في سياقات مختلفة، لا يزال تطبيقها في سياق اللغة العربية مجالاً غير مستكشف من البحث. استخدمت هذه الدراسة ثلاث استراتيجيات لتكييف Mixup للتطبيق في تحليل المشاعر العربية. وأجريت تقييمات تجريبية لتقييم فاعلية هذه الاستراتيجيات، وذلك باستخدام مجموعة من مجموعات البيانات المرجعية. توضح دراساتنا أن استراتيجيات الاستيفاء هذه تعمل على نحو فاعل بوصفها طرائق مستقلة عن المجال لزيادة البيانات، في سياق تصنيف النص. إضافة إلى ذلك، فإن هذه الاستراتيجيات لديها القدرة على أن تؤدي إلى تحسينات في الأداء لكل من نماذج الشبكة العصبية التلافيفية (CNN) والذاكرة القصيرة المدى المطولة (LSTM).

INTRODUCTION

In recent years, deep learning models have exhibited remarkable performance in numerous Natural Language Processing (NLP) tasks, such as parsing [1], text classification [2], [3] and machine translation [4]. These models are typically characterized by their substantial parameter count, often reaching millions, necessitating extensive data for training to prevent overfitting and enhance generalization capabilities. However, collecting a sizable annotated dataset proves to be a laborious and costly endeavour. To mitigate the data-hungry nature of deep learning models, an approach known as automatic data augmentation has emerged. This technique involves generating synthetic data samples to augment the training dataset, effectively serving as regularization for the learning models. Data augmentation has been actively and successfully employed in computer vision [5], [6], [7] and speech recognition tasks [8], [9]. In these domains, methods frequently rely on human knowledge to apply label-invariant data transformations, such as image scaling, flipping, and rotation. However, natural language processing presents a different challenge, as there are no straightforward rules for label-invariant transformations in textual data. Even slight changes in a word within a sentence can drastically alter its meaning. Consequently, popular data augmentation techniques in NLP focus on transforming text through word replacements, either using synonyms from manually curated ontologies, such as WordNet [10] or leveraging word similarity measures [11], [12]. Nonetheless, this synonym-based approach can only be applied to a limited portion of the vocabulary since finding words with precisely or nearly identical meanings is rare. Furthermore, certain NLP data augmentation methods are specifically designed for particular domains, rendering them less adaptable to other domains [13]. As a result, developing more versatile and effective data augmentation techniques remains a significant research challenge in the field of NLP. In recent researches, a straightforward yet highly impactful data augmentation technique called Mixup [7] has been introduced, demonstrating remarkable effectiveness in improving the accuracy of image classification models. This method operates by linearly interpolating the pixels of randomly paired images along with their corresponding training targets, thereby generating synthetic examples for the training process. The application of Mixup as a training strategy has proven to be highly effective in regularizing image classification networks, leading to notable performance improvements. Mixup methodologies can be classified into input-level Mixup [14], [15], [16] and hidden-level Mixup [17] depending on where the mix operation occurs. In the context of natural language processing (NLP), applying Mixup poses greater challenges compared to computer vision due to the discrete nature of text data and the variability in sequence lengths. As a result, prior efforts in Mixup for textual data [18], [19] have put forth two strategies for its application in text classification: one involves performing interpolation on word embedding, while the other applies it to sentence embedding. This incentive drives us to explore Mixup text techniques for low-resource languages, specifically concentrating on Arabic sentiment classification. Our study involves a comparative analysis of basic LSTM classification models, both with and without the incorporation of Mixup techniques. Furthermore, we conduct experiments on diverse datasets, spanning sample sizes varying from hundreds to thousands per class. Additionally, we perform an ablation study to investigate the effects of different Mixup parameter values. To the best of our knowledge, this represents the pioneering research utilizing Mixup in the context of Arabic text classification.

RELATED WORKS

Data augmentation is a methodology employed to enhance the diversity and quality of data without the need to collect additional samples directly. The concept of data augmentation was initially applied in computer vision [20], where the manipulation of individual pixels in an image allows for modifications such as adding noise, cropping, padding, or flipping without compromising the underlying information. In various domains, such as image classification [21], [22]. [23] and sound classification [24], augmenting datasets with perturbed replicas of their samples has proven to be highly effective. However, the application of data augmentation techniques in Natural Language Processing remains relatively unexplored. Unlike image-related techniques that can generate new images with preserved semantic information through flipping and rotation, such approaches cannot be directly applied to text due to potential disruptions in syntax, grammar, and changes in the original sentence's meaning. Moreover, while noise injection is commonly used to enhance audio signal data [8], [25], [26], its direct suitability for text is limited by the categorical nature of word and character tokens. Text data augmentation can be categorized into two main approaches: Feature space including Mixup [7] and data space [27]. In the data space, augmentation involves transforming raw data into readable textual data. The data space, as described in [27], is further divided into four categories: Character Level, word Level, phrase sentence Level, and document Level. In the feature space, Mixup opts for virtual embeddings rather than generating augmented samples in natural language form. This methodology leverages existing data to sample points in the virtual vector space, potentially resulting in sampled data with labels distinct from those of the original dataset. The inception of Mixup traces back to the domain of image processing, which originated from the work of [7]. Expanding upon this foundation, two variations of Mixup tailored introduced specifically for sentence classification by [18]. Mixup has witnessed widespread adoption in recent Natural Language Processing (NLP) research. In the field of neural machine translation, [28] pioneered the construction of adversarial samples, utilizing the methodology introduced by [29]. They subsequently implemented two Mixup strategies, namely Padv and Paut. Padv involves interpolation between adversarial samples, while Paut interpolates between the two corresponding original samples. Concurrently, [30] incorporated Mixup into Named Entity Recognition (NER), presenting both Intra-LADA and InterLADA approaches. In [31] introduced Mixup-Transformer, a novel approach integrating Mixup with transformer-based pre-trained architecture. The researchers evaluated its efficacy by assessing its performance across various text classification datasets. While the proposed method Saliency-Based Span Mixup in [32], SSMix, distinguishes itself by performing operations directly on input text rather than on hidden vectors, as seen in previous approaches. From the available literature, it appears that only a limited number of recent Arabic research studies have primarily focused on data space. For instance, Duwairi et al. [33] employed a set of rules to modify or swap branches of parse trees according to Arabic syntax, generating new sentences with the same labels. Similarly, in Named Entity Recognition, Sabty et al. [34] explored various data augmentation techniques, such as Word random insertion, swap and deletion within sentences, sentence back-translation, and word embedding substitution, which have also been utilized in other research, like [35], for spam detection.

MATERIALS AND METHODS

Mixup Concept

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable. The concept of Mixup involves creating a synthetic sample through linear interpolation of a pair of training samples and their corresponding model targets. To elaborate, let us consider a pair of samples denoted as (xi, yi) and (xj, yj), where x represents the input data, and y is the one-hot encoding representation of the respective class label for each sample. The process of generating the synthetic sample is as follows:

$$\tilde{x}^{ij} = \lambda x^i + (1-\lambda)\, x^j \quad (1)$$

$$\tilde{y}^{ij} = \lambda y^i + (1-\lambda)\, y^j \quad (2)$$

where λ could be either fixed value in [0; 1] or it is sampled from Beta distribution with a hyper-parameter Beta (α; α).

The synthetic data generated using this approach are subsequently introduced into the model during training, aiming to minimize the loss function, such as the cross-entropy function typically employed in supervised classification tasks. To achieve computational efficiency, the mixing process involves randomly selecting one sample and pairing it with another sample drawn from the same mini-batch.

Mixup for text classification

In contrast to images that comprise pixels, sentences are composed of sequences of words. Consequently, constructing a meaningful sentence representation involves aggregating information from this word sequence. In typical CNN or LSTM models, a sentence is initially represented as a sequence of word embedding and then processed through a sentence encoder. Commonly used sentence encoders include CNN and LSTM architectures. The resulting sentence embedding, generated by either CNN or LSTM, is subsequently passed through a softmax layer to generate the predictive distribution encompassing the possible target classes for making predictions. In [18], Guo introduced two variations of Mixup tailored for sentence classification. The first variant, referred to as wordMixup, employs sample interpolation within the word embedding space. The second variant, known as senMixup, performs interpolation on the final hidden layer of the network just before it is fed into the standard softmax layer to generate the predictive distribution across classes. Specifically, in the wordMixup technique, all sentences are first zero-padded to a uniform length. Subsequently, interpolation is performed for each dimension of every word within a sentence. Let us consider a given text, such as a sentence consisting of N words, which can be represented as a matrix B in an N × d form. Here, each row t of the matrix corresponds to an individual word, denoted as Bt, and is represented by a d-dimensional vector obtained either from a pre-trained word embedding table or randomly generated. To formalize the process, let (Bi, yi) and (Bj, yj) be a pair of samples, where Bi and Bj represent the embedding vectors of the input sentence pairs, and yi and yj correspond to their respective class labels, represented in a one-hot format. For a specific word at the t-th position in the sentence, the interpolation procedure is applied. The process can be formulated as:

$$\tilde{B}_t^{ij} = \lambda B_t^i + (1-\lambda) B_t^j \quad (3)$$

$$\tilde{y}^{ij} = \lambda y^i + (1-\lambda)\, y^j \quad (4)$$

The obtained novel sample⟦(B ˜⟧^ij;y ˜^ij) is subsequently employed for training purposes. As for senMixup, the hidden embeddings for both sentences, having identical dimensions, are initially generated using an encoder like CNN or LSTM. Following this, the pair of sentence embeddings, f(Bi) and f(Bj), is linearly interpolated. In more detail, let f represent the sentence encoder; thus, the sentences Bi and Bj are first encoded into their corresponding sentence embedding, f(Bi) and f(Bj), respectively. In this scenario, the mixing process is applied to each kth dimension of the sentence embedding as follows:

$$\tilde{B}_{\{k\}}^{ij} = f\left(B^i\right)_{\{k\}} + (1-\lambda)\, f\left(B^j\right)_{\{k\}} \quad (5)$$

$$\tilde{y}^{ij} = \lambda y^i + (1-\lambda)\, y^j \quad (6)$$

The sentMixup usually applies Mixup directly before the softmax while we experimented with additional Mixup type that work on the hidden layers output similar to [17] applying Mixup before the final linear layer. Th proposed models structures are represented in Fig. 1.

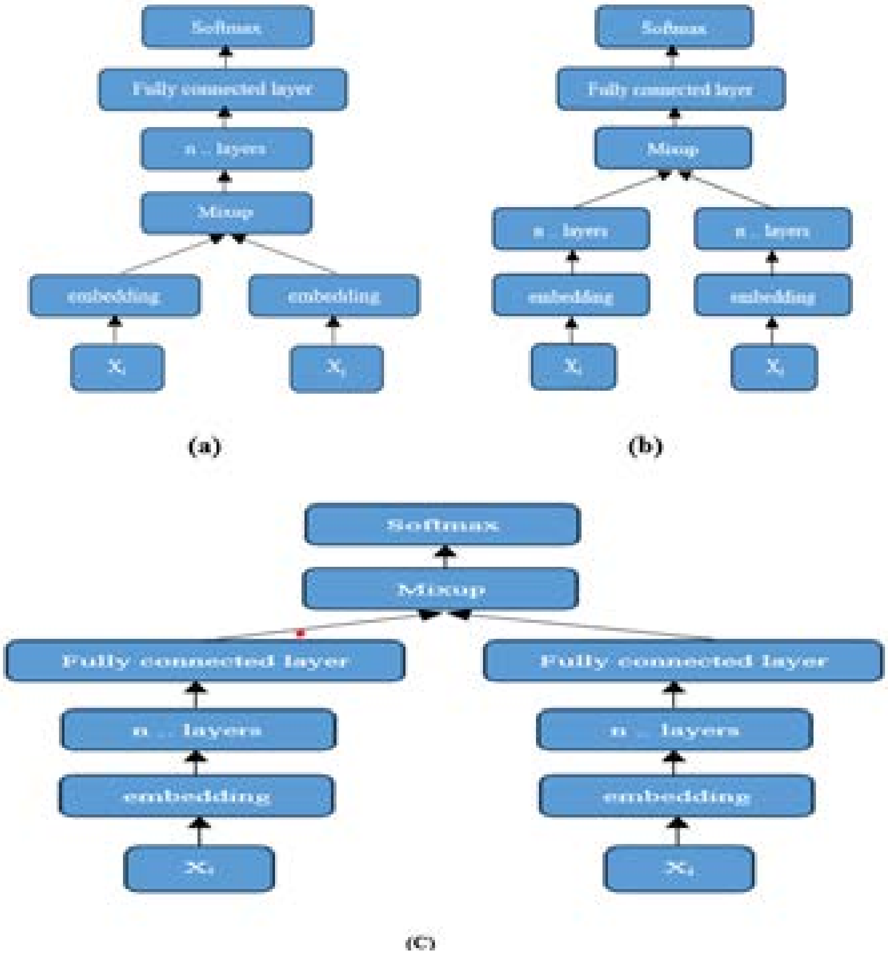

Figure. 1 Models structure: (a) Mix-embed (Mixup with word embedding), (b) Mix-encoder (Mixup at the encoder level) and (c) Mix-output (Mixup at the output level)

Datasets

We performed experiments using 8 Arabic sentiment classification benchmark datasets: ArSarcasm v[36] 1 & v[37] 2, SemEval[38] 2017, ArSenTD-LEV [39], AJGT [40], ASTD-3C [41], MOV [42]. The training sets differ in size (from 12548 to 1524), and in number of labels (2 to 5). The used datasets are summarizedin Table 1.

Preprocessing

The effectiveness of sentiment analysis models greatly depends on the quality of data

Table 1. The datasets distribution

Dataset	Train	Test	Labels
ArSarcasmv1	8437	2110	3
ArSarcasmv2	12548	3000	3
SemEval2017	3353	6098	3
ArSenTD-LEV	4000	C	5
AJGT	1800	C	2
ASTD-3C	3315	C	3
MOV	1524	C	3

preprocessing, which is equally critical as the model›s architectural design. Preprocessing involves cleaning and preparation of the text data for the classification process. Textual data, particularly when sourced from the internet, tends to be unstructured, necessitating additional processing steps for proper classification.

Experimental environment and hardware

The experiments were developed using Python 3.9.7. The experiments, including their development, implementation, execution, and analysis, were conducted on an ASUS ROG G531GT Notebook. This machine runs Windows 11 and is equipped with a 9th generation Intel Core i7 processor, 32GB of RAM, a 512GB NVMe SSD, and an NVIDIA GeForce GTX 4 1650GB graphics card. The software libraries used in this study include PyTorch, Scikit-learn, Pandas, Gensim, and NumPy.

Model

We conducted an evaluation of wordMixup and senMixup using both CNN and LSTM architectures for sentence classification. In our setup, we employ filter sizes of 4 ,3, and 5, each configured with 100 feature maps, and a dropout rate of 0.5 for the baseline CNN. For the LSTM model, we utilize 3 hidden layers, each comprising 100 hidden state dimensions, with the activation function set to tanh. Additionally, the mixing policy parameter is set to the default value of one. In cases where datasets lacked a standard test set, we adopted cross-validation with a k-fold value of 5 and reported the average performance metrics. Our training process utilized the Adam optimizer [36] with mini-batches of size 32 with 30 epochs and learning rate of 1e-3. For word embedding, we employed 100-dimensional Aravec embedding.

RESULTS

The four variations of models evaluated are None (without Mixup), Mix-embed (Mixup with word embedding), Mix-encoder (Mixup at the encoder level), and Mix-output (Mixup at the output level). Table 2 and Table 3 present the results of the different experiments using LSTM and CNN model respectively. We can observe a general improvement on the performance of different Mixup models on various datasets.

TABLE 2. The result of lstm model (results are formatted as accuracy/ macrof1); none is the basic model with no mixup

Acc/ MacroF1	LSTM			
Dataset	**None**	**wordMixup**	**encoderMixup**	**senMixup**
ArSarcasmv1	61.8/56.8	60.8/55.5	62.5/58	62.9/58.3
ArSarcasmv2	53.5/48.5	51.3/47.1	55.4/50.4	54.8/50.8
SemEval2017	50.7/47.7	49.3/47.7	51/48.1	51.3/48.8
ArSenTD-LEV	47.4/44.3	46.8/43.3	48.8/45.7	48.9/45.8
AJGT	84.6/84.5	82.4/82.4	85.2/85.2	85.2/85.2
ASTD-3C	56.1/53.4	54.6/49.1	56.3/52.5	56.5/53.1
MOV	57.9/43.5	58/42.3	58.6/43.4	58.9/43.8

Table 3. The result of cnn model (results are formatted as accuracy/ macrof1); none is the basic model with no mixup

Acc/ MacroF1	CNN			
Dataset	**None**	**wordMixup**	**encoderMixup**	**senMixup**
ArSarcasmv1	62.1/55.3	62.6/56.8	63.1/57.5	63.5/57.6
ArSarcasmv2	53.2/49.0	55.4/50.6	56.7/52.1	57.2/52.4
SemEval2017	51.5/47.4	51.4/50.3	52.0/49.3	52.2/49.5
ArSenTD-LEV	49.7/45.9	48.0/45.3	49.3/46.0	49.2/45.8
AJGT	85.6/85.5	85.3/85.3	84.0/83.9	83.9/83.8
ASTD-3C	58.6/52.6	57.7/51.6	60.6/51.4	60.2/50.9
MOV	70.7/43.4	71.0/43.2	69.2/40.2	68.8/39.4

DISCUSSION

Across the datasets, it is evident that applying Mixup techniques generally leads to slight improvements in accuracy compared to the baseline None model. However, the effectiveness of Mixup varies depending on the dataset. For instance, on the AJGT dataset, all Mixup variants consistently outperform the None model, with Mix-encoder and Mix-output achieving the highest accuracy of %85.2. On the other hand, for the SemEval2017 and ArSenTD-LEV datasets, Mixup provides only marginal gains, suggesting that the impact of Mixup might be more prominent in certain scenarios. Additionally, while Mixup seems to be beneficial in some cases, it does not necessarily lead to performance improvements across all datasets. For instance, on the MOV dataset, the Mixup variants show comparable or slightly worse results compared to the None model. Furthermore, it is worth noting that the Mix-encoder and Mix-output models tend to perform better than the Mix-embed model in most cases. This could be attributed to the advantage of applying Mixup at the higher levels of the model architecture, which allows the model to capture more abstract and meaningful patterns. Mixup augments data by interpolating sequences, which can create new variations that capture a broader range of sequential patterns. LSTMs, with their capability to understand and generalize sequences over long contexts, can leverage these variations more effectively than CNNs, which focus more on local patterns and may not fully utilize the sequential nature of the augmented data. Overall, these results demonstrate that Mixup techniques can be advantageous for sentiment analysis tasks, but their effectiveness is influenced by the dataset characteristics and the specific Mixup strategy used.

CONCLUSIONS AND RECOMMENDATIONS

Taking inspiration from the promising results of Mixup, a data augmentation technique based on sample interpolation used in image recognition and text classification, we conducted an investigation into three variations of Mixup for Arabic sentiment classification task which is the first study on Mixup in Arabic to our knowledge. Our experiments demonstrate that the application of Mixup leads to improvements in accuracy and Macro F1 scores for both CNN and LSTM text classification models. Notably, our findings highlight the effectiveness of interpolation strategies as domain-independent regularizer, effectively mitigating the risk of overfitting in sentence classification. These results underscore the potential of Mixup as a valuable tool in the field of NLP for enhancing model generalization and performance across various sentence classification tasks. In our future research endeavors, we have outlined our intentions to explore and examine further proposed variations of Mixup. Among these variants are AutoMix [44], a method that adaptively learns a sample mixing policy by leveraging discriminative features, SaliencyMix [32], which synthesizes sentences while maintaining the contextual structure of the original texts through span-based mixing and EMTCNN [45], an Enhanced Mixup that leverage transfer learning to address challenges in Twitter sentiment analysis. We are also interested by questions related to the visual appearance of mixed sentences and the underlying mechanisms responsible for the efficacy of interpolation in sentence classification. These inquiries will provide valuable insights into the potential applications and benefits of various Mixup techniques, contributing to the advancement of NLP tasks, particularly those focused on sentence classification.

REFERENCES

1. Socher R, Lin CCY, Ng AY, Manning CD. Parsing Natural Scenes and Natural Language with Recursive Neural Networks. In: Proceedings of the 28th International Conference on International Conference on Machine Learning; 2011; Bellevue, Washington, USA. p. 136–129.
2. Kim Y. Convolutional Neural Networks for Sentence Classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2014. p. 1751–1746.
3. Tai KS, Socher R, Manning CD. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers); 2015. p. 1566–1556.
4. Sutskever I, Vinyals O, Le Q. Sequence to Sequence Learning with Neural Networks. In: Advances in Neural Information Processing Systems. 4;2014.
5. Simard P, LeCun Y, Denker JS, Victorri B. Transformation Invariance in Pattern Recognition Tangent Distance and Tangent Propagation. In: Neural Networks: Tricks of the Trade. This Book is an Outgrowth of a 1996 NIPS Workshop. 1998. p. 27–239.
6. Krizhevsky A, Sutskever I, Hinton GE. ImageNet Classification with Deep Convolutional Neural Networks. Commun ACM. 90–84:(6)60;2017.
7. Zhang H, Cissé M, Dauphin YN, Lopez-Paz D. mixup: Beyond Empirical Risk Minimization. In: 6th

International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 2018 ,3, Conference Track Proceedings. 2018.

8. Jaitly N, Hinton G. Vocal Tract Length Perturbation (VTLP) improves speech recognition. In: ICML Workshop on Deep Learning for Audio, Speech and Language Processing. 2013.

9. Ko T, Peddinti V, Povey D, Khudanpur S. Audio augmentation for speech recognition. In: INTERSPEECH. 2015. p. 3589–3586.

10. Zhang X, Zhao J, LeCun Y. Character-Level Convolutional Networks for Text Classification. In: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1. Montreal, Canada; 2015. p. 657–649.

11. Wang WY, Yang D. That's So Annoying!!!: A Lexical and Frame-Semantic Embedding Based Data Augmentation Approach to Automatic Categorization of Annoying Behaviors using #petpeeve Tweets. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. 2015. p. 2563–2557.

12. Kobayashi S. Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). June 2018. p. 457–452. doi: 10.18653/v1/N2072-18.

13. Şahin GG, Steedman M. Data Augmentation via Dependency Tree Morphing for Low-Resource Languages. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. 2018. p. 5009–5004. doi: 10.18653/v1/D1545-18.

14. Yun S, Han D, Oh SJ, Chun S, Choe J, Yoo YJ. CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). 2019. p. 6031–6022. Available: https://api.semanticscholar.org/CorpusID:152282661

15. Walawalkar D, Shen Z, Liu Z, Savvides M. Attentive Cutmix: An Enhanced Data Augmentation Approach for Deep Learning Based Image Classification. May 2020. p. 3646–3642. doi: 10.1109/ICASSP40776.2020.9053994.

16. Uddin AFMS, Monira MS, Shin W, Chung T, Bae SH. SaliencyMix: A Saliency Guided Data Augmentation Strategy for Better Regularization. 2021.

17. Verma V et al. Manifold Mixup: Better Representations by Interpolating Hidden States. In: Proceedings of the 36th International Conference on Machine Learning. June 2019. vol. 97. p. 6447–6438. Available: https://proceedings.mlr.press/v97/verma19a.html

18. Guo H, Mao Y, Zhang R. Augmenting data with mixup for sentence classification: An empirical study. arXiv preprint arXiv:2019 .1905.08941.

19. Chen J, Yang Z, Yang D. MixText: Linguistically-Informed Interpolation of Hidden Space for Semi-Supervised Text Classification. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. July 2020. p. 2157–2147. doi: 10.18653/v2020/1.acl-main.194.

20. Li B, Hou Y, Che W. Data augmentation approaches in natural language processing: A survey. AI Open. 90–3:71;2022. doi: https://doi.org/10.1016/j.aiopen.2022.03.001.

21. Krizhevsky A, Sutskever I, Hinton GE. ImageNet Classification with Deep Convolutional Neural Networks. Commun ACM. 90–84:(6)60;2017. doi: 3065386/10.1145.

22. Tran T, Pham T, Carneiro G, Palmer L, Reid I. A Bayesian Data Augmentation Approach for Learning Deep Models. In: Advances in Neural Information Processing Systems. 30;2017. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/076023edc9187cf1ac1f1163470e479a-Paper.pdf

23. Al-Dhabyani W, Gomaa M, Khaled H, Fahmy A. Deep Learning Approaches for Data Augmentation and Classification of Breast Masses using Ultrasound Images. Int J Adv Comput Sci Appl. (5)10;2019. doi: 10.14569/IJACSA.2019.0100579.

24. Salamon J, Bello JP. Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification. IEEE Signal Process Lett. 283–279:(3)24;2017. doi: 10.1109/LSP.2017.2657381.

25. Hannun A et al. Deep speech: Scaling up end-to-end speech recognition. arXiv preprint arXiv:2014.1412.5567.

26. Tzirakis P, Trigeorgis G, Nicolaou M, Schuller B, Zafeiriou S. End-to-End Multimodal Emotion Recognition Using Deep Neural Networks. IEEE J Sel Top Signal Process. 2017;PP. doi: 10.1109/JSTSP.2017.2764438.

27. Bayer M, Kaufhold MA, Reuter C. A Survey on Data Augmentation for Text Classification. ACM Comput Surveys. 39–1:(7)55;2022. doi: 3544558/10.1145.

28. Cheng Y, Jiang L, Macherey W, Eisenstein J. AdvAug: Robust Adversarial Augmentation for Neural Machine Translation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020. p. 5970–5961. doi: 10.18653/v2020/1.acl-main.529.

٢٩. Cheng Y, Jiang L, Macherey W. Robust Neural Machine Translation with Doubly Adversarial Inputs. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019. p. 4333–4324. doi: 10.18653/v1/P1425-19.

٣٠. Chen J, Wang Z, Tian R, Yang Z, Yang D. Local Additivity Based Data Augmentation for Semi-supervised NER. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2020. p. 1251–1241. doi: 10.18653/v2020/1.emnlp-main.95

٣١. Sun L, Xia C, Yin W, Liang T, Yu P, He L. Mixup-Transformer: Dynamic Data Augmentation for NLP Tasks. In: Proceedings of the 28th International Conference on Computational Linguistics. Dec. 2020. p. 3440–3436. doi: 10.18653/v2020/1.coling-main.305.

٣٢. Yoon S, Kim G, Park K. SSMix: Saliency-Based Span Mixup for Text Classification. arXiv preprint arXiv:2021 .2106.08062.

33. Duwairi R, Abushaqra F. Syntactic- and morphology-based text augmentation framework for Arabic sentiment analysis. PeerJ Comput Sci. 7;2021:e469. Available: https://doi.org/10.7717/peerj-cs.469.

34. Sabty C, Omar I, Wasfalla F, Islam M, Abdennadher S. Data Augmentation Techniques on Arabic Data for Named Entity Recognition. Procedia Comput Sci. –189:292;2021 299. doi: https://doi.org/10.1016/j.procs.2021.05.092.

35. Alkadri AM, Elkorany A, Ahmed C. Enhancing Detection of Arabic Social Spam Using Data Augmentation and Machine Learning. Appl Sci. (22)12;2022. doi: 10.3390/app122211388.

36. Abu Farha I, Magdy W. From Arabic Sentiment Analysis to Sarcasm Detection: The ArSarcasm Dataset. In:

Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection. May 2020. p. –32 39. Available: https://aclanthology.org/2020.osact-1.5.

37. Abu Farha I, Zaghouani W, Magdy W. Overview of the WANLP 2021 Shared Task on Sarcasm and Sentiment Detection in Arabic. In: Proceedings of the Sixth Arabic Natural Language Processing Workshop. Apr. 2021. p. 305–296. Available: https://aclanthology.org/2021.wanlp-1.36.
38. Rosenthal S, Farra N, Nakov P. SemEval-2017 Task 4: Sentiment Analysis in Twitter. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). Aug. 2017. p. 518–502. doi: 10.18653/v1/S2088-17.
39. Baly R, Khaddaj A, Hajj H, El-Hajj W, Shaban KB. ArSentD-LEV: A Multi-Topic Corpus for Target-based Sentiment Analysis in Arabic Levantine Tweets. 2019.
40. Alomari KM, Elsherif HM, Shaalan KF. Arabic Tweets Sentimental Analysis Using Machine Learning. 2017. Available: https://api.semanticscholar.org/CorpusID:37492873.
41. Nabil M, Aly M, Atiya A. ASTD: Arabic Sentiment Tweets Dataset. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Sep. 2015. p. 2519–2515. doi: 10.18653/v1/D1299-15.
42. ElSahar H, El-Beltagy SR. Building Large Arabic Multi-domain Resources for Sentiment Analysis. In: Computational Linguistics and Intelligent Text Processing. 2015. p. 34–23.
43. Kingma DP, Ba J. Adam: A method for stochastic optimization. CoRR. 2014;abs/1412.6980.
44. Liu Z et al. AutoMix: Unveiling the Power of Mixup for Stronger Classifiers. In: European Conference on Computer Vision. 2021.
45. Wang Q. Learning From Other Labels: Leveraging Enhanced Mixup and Transfer Learning for Twitter Sentiment Analysis. In: 2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI). 2021. p. 343-336. doi:10.1109/ICTAI52525.2021.00055.

Author Contributions: The first author conducted investigations, prepared data, executed the study, analyzed the findings, and prepared the initial draft. The second and third authors offered guidance, and conducted a review of the work.

Competing Interests: Authors declare that they have no competing interests.
Data And Materials Availability: All data are available in the main text