Neural Computing and Applications (2025) 37:7015–7048
https://doi.org/10.1007/s00521-025-11020-z

REVIEW

A comprehensive survey on Arabic text augmentation: approaches, challenges, and applications

Ahmed Adel ElSabagh[1] · Shahira Shaaban Azab[1] · Hesham Ahmed Hefny[1]

Received: 19 March 2024 / Accepted: 11 January 2025 / Published online: 7 February 2025

Abstract

Arabic is a linguistically complex language with a rich structure and valuable syntax that pose unique challenges for natural language processing (NLP), primarily due to the scarcity of large, reliable annotated datasets essential for training models. The varieties of dialects and mixtures of more than one language within a single conversation further complicate the development and efficacy of deep learning models targeting Arabic. Data augmentation (DA) techniques have emerged as a promising solution to tackle data scarcity and improve model performance. However, implementing DA in Arabic NLP presents its challenges, particularly in maintaining semantic integrity and adapting to the language's intricate morphological structure. This survey comprehensively examines various aspects of Arabic data augmentation techniques, covering strategies for model training, methods for evaluating augmentation performance, understanding the effects and applications of augmentation on data, studying NLP downstream tasks, addressing augmentation problems, proposing solutions, conducting in-depth literature reviews, and drawing conclusions. Through detailed analysis of 75 primary and 9 secondary papers, we categorize DA methods into diversity enhancement, resampling, and secondary approaches, each targeting specific challenges inherent in augmenting Arabic datasets. The goal is to offer insights into DA effectiveness, identify research gaps, and suggest future directions for advancing NLP in Arabic.

Keywords Text augmentation · Arabic text · Natural language processing · Deep learning

1 Introduction

Arabic is a linguistically complex language with limited resources and a less extensively explored syntax compared to English. While certain languages benefit from shared Latin representations because of similar vocabulary and structure, Arabic, with its unique morphological and syntactic structure, lacks many similarities with other widely spoken Latin-based languages, and as a result, it cannot leverage shared representations [1]. It is a Semitic language that varies syntactically, morphologically, and semantically from Indo-European languages [2]. The Arabic language consists of 28 letters and enjoys rich morphology. One challenge facing Arabic texts is the variety of different Arabic dialects, prompting various research studies to explore this linguistic diversity, such as in [3], where authors used DA for classifying five Arabic dialects through dialectal sentence generation. Random shuffling was used in [4] to augment a model that distinguished between twenty-five distinct Arabic dialects. Another challenge facing Arabic texts is code-switching, which refers to the practice of mixing more than one language within a single conversation. It is widespread in culture-mixed countries where people switch between their primary and other languages while talking [5]. For example: “deadline الـ meet عشان ـنprogress الـ monitor”نـ لازم (We should monitor the progress to meet the deadline). Data containing mixed languages in Arabic are limited in comparison with datasets consisting solely of pure Arabic text. This led the authors of [5] to investigate the impact of augmenting a named entity

✉ Shahira Shaaban Azab
shahiraazazy@cu.edu.eg

Ahmed Adel ElSabagh
ahmad_elsabagh@pg.cu.edu.eg

Hesham Ahmed Hefny
hehefny@cu.edu.eg

[1] Department of Computer Science, Faculty of Graduate Studies for Statistical Research, Cairo University, Giza 12613, Egypt

recognition (NER) NLP task, resulting in a 1.51% increase in the F-score.

Deep learning models have achieved exceptional results in various tasks, but their training typically requires a significant amount of time and human effort in labeling, which limits the range of problems that can be effectively addressed. Hence, it is crucial to develop efficient training techniques that leverage smaller labeled training datasets, allowing for more resource-efficient utilization. Currently, several methods aim to address this challenge, including DA, active learning, and N-Shot learning. DA creates synthetic samples, while active learning selects informative, unlabeled data for manually labeling it, saving computational resources [5–7]. Few-shot learning handles tasks with minimal labeled data [8].

NLP involves enabling computers to comprehend human language, focusing on techniques for constructing systems that interact with languages. Arabic natural language processing has recently gained research attention, leading to the creation of systems and applications for tasks such as text classification, machine translation, and sentiment analysis [9].

DA is the field concerned with techniques used for expanding data fed into machine learning and deep learning tasks. It involves slight modifications to existing data or the synthetic creation of similar data without directly collecting new data. These techniques aid in saving algorithms from failure and overfitting due to data scarcity, helping to increase models' performance. It has been widely adopted across different machine learning fields, such as improving the performance of automatic speech recognition [10], reducing overfitting in computer vision [11], alleviating NLP data scarcity [8] and balancing classes in imbalanced datasets [12]. However, the adoption of DA in NLP is a more difficult task compared with computer vision. This complexity arises from the diversity of natural languages worldwide, each with distinct vocabularies and grammar rules. Additionally, due to the nature of NLP data, which is discrete data and not real numeric data, generating augmented examples is less straightforward [13]. This led researchers to study a variety of techniques to apply them to NLP downstream tasks. These augmentation techniques must consider the validity and integrity of the data. For instance, equivalent meanings in machine translation and identical labels in text classification are the same as those in the original data [14].

In recent years, data augmentation techniques have demonstrated significant improvements in various Arabic text processing tasks. For instance, in [15], augmenting the corpus resulted in a substantial boost in the Transformer model's performance, elevating the BLEU score from 15.9 to 60. Similarly, in [12], the F1-Score of an ensemble model surged from 77 to 93% post-augmentation. The application of augmentation also led to notable accuracy improvements, such as a 14.06% and 12.57% increase for LSTM models on Bahraini and MSA dialects, respectively, as reported in [16]. Additionally, [17] showed that effective performance could be achieved with only 10% to 30% of the dataset being manually labeled, closely matching results obtained from larger labeled datasets. Integration of self-training with zero-shot techniques, as in [18], resulted in an approximately 10% F1 improvement and a 2% accuracy boost. Furthermore, [19] observed an accuracy increase from 88.89% to 96.72% on Bahraini dialects, while in [20], augmentation enhanced the model's accuracy from 77 to 96%. In another example, the recall of a BERT-based Arabic model improved from 0.580 to 0.708 following augmentation [21], and a remarkable 34% improvement was achieved in [22]. Notably, augmentation techniques have also been widely adopted in industry. For example, back-translation is used in Google Translate to improve fluency in low-resource languages,[1] and Grammarly employs augmentation to address biases in grammatical error correction systems [23]. These instances highlight the substantial role of data augmentation in significantly boosting the performance of text models across a wide range of applications.

DA has achieved varying levels of success in improving model performance, ranging from trivial [24] to significant increases [25], depending on multiple factors. Important factors include choosing the most appropriate DA techniques and making fine-tuning decisions based on different metrics and algorithms used for measuring similarity. Due to the absence of a universal golden rule for selecting DA techniques, some papers have explored multiple DA techniques and compared results [26]. These techniques range from simple, static rules such as applying pre-defined morphological and linguistic rules [27] or noise-based rules [28] to more advanced approaches that utilize machine learning and deep learning algorithms to dynamically expand text, like pre-trained models [29] and self-training [30]. Future directions include integrating DA algorithms with other techniques, such as N-shot learning [31] and active learning [7], and combining data from different languages for greater diversity [32]. One of the most promising emerging directions is the utilization of large language models (LLMs), such as generating text through prompting [33]. An important observation is the imbalance in the number of tasks that applied DA to Arabic text. For example, there are more papers on sentiment analysis than on text summarization. Additionally, different metrics were used to assess DA quality in the literature, varying by downstream task and applied framework. This makes comparing different DA techniques across tasks challenging

[1] https://research.google/blog/recent-advances-in-google-translate/.

and offers a good opportunity for researchers to address DA challenges. As we found, the number of papers applying DA to Arabic text is increasing, and results are improving with the evolution of algorithms. We will explore these topics in depth in this paper.

This research focuses on the literature related to DA in Arabic, which was analyzed and evaluated based on specific criteria for inclusion and exclusion. Firstly, to be considered and included, the primary study must have been published relatively recently, specifically between 2019 and 2023. Secondly, text augmentation techniques should be applied specifically to the Arabic language as a targeted language. Thirdly, the augmentation technique must be explicitly and clearly introduced. Fourthly, augmentation should be applied to NLP rather than other fields. Fifthly, the augmentation technique must go beyond simply adding extra data to the dataset. Papers that met these criteria were analyzed in this work.

Some papers met the criteria but relied solely on secondary augmentation methods, which increase the variety of training samples without explicitly collecting new data. Examples include merging datasets, translating similar datasets from other languages, or mixing different dialects. These approaches, however, are not commonly considered standard data augmentation techniques. We restricted such papers to Sect. 2 exclusively, excluding them from other sections. Some papers, not available as Open-Access papers, were initially excluded. However, to introduce subjectivity into our analysis, we decided to include some of them from various publishers.

Various papers presented diverse approaches to categorize augmentation techniques. The difficulty behind this is that there are a variety of augmentation techniques applied across distinct NLP downstream tasks. For instance, [13] categorized techniques, including rule-based, interpolation, and model-based techniques, while in [34], categories included data space and feature space. Another categorization method was employed in [8] which categorized techniques based on token-level, sentence-level, adversarial data, and hidden-space. These earlier examples illustrated how augmentation techniques can be categorized differently. We followed a categorization approach similar to that in [14]. However, we adjusted these categories to fit with what we studied in Arabic literature. For instance, we added secondary-based augmentation techniques and resampling-based techniques to our taxonomy. Resampling-based techniques are methods that address class imbalances by either oversampling the minority class or undersampling the majority class to ensure balanced representation in the dataset. Additionally, we replaced the Mixup category with interpolation-based augmentation techniques.

Based on our study of the literature, there is a notable gap in surveys focusing on DA in Arabic text. This paper aims to fill that gap by providing the first comprehensive survey on this topic and offering valuable guidance to practitioners in the field. Our objectives are as follows: (1) To provide clear insights into DA approaches by deeply studying and explaining the methods used in Arabic literature; (2) To illustrate the different steps in the DA pipeline, such as preprocessing, feature extraction, model training, and measuring DA efficiency, helping readers gain a consolidated understanding and strengthen their grasp of DA processes; (3) To analyze the benefits of DA, including expanding and balancing datasets and reducing the need for manually annotated data, while also discussing the major challenges faced by DA techniques and proposing future directions; and (4) To clarify how various NLP downstream tasks have applied DA by classifying these tasks and explaining how research papers have implemented DA on Arabic text for each task.

To provide a comprehensive overview of the structure of this paper, the subsequent sections are organized as follows: Sect. 2 examines various Arabic data augmentation techniques, including paraphrasing, noising, and sampling methods, along with resampling and secondary-based techniques. Section 3 discusses strategies for training models with augmented data, focusing on preprocessing, feature extraction, and performance measurement. Section 4 highlights the benefits of augmentation, such as dataset expansion and balancing, and explores the specific tasks to which these techniques are applied in Arabic NLP. Section 5 presents a detailed analysis of the study's findings, addresses challenges, and outlines future research directions. Finally, Sect. 6 summarizes the key contributions of the paper, reflects on the literature gaps, and suggests potential avenues for further research.

2 Arabic data augmentation techniques

In this section, we explore the diverse range of DA techniques applied specifically to Arabic text. These techniques are categorized based on their approach to modifying the original data to improve the performance of NLP models. The primary categories include diversity-based techniques and resampling techniques. Diversity-based techniques involve paraphrasing, noising, and sampling. Generally, paraphrasing-based methods aim to maintain the semantics of the original data with minimal modifications, while noising-based methods enhance the model's robustness with more semantical changes compared to paraphrasing-based methods. Sampling-based methods introduce greater diversity into the training data, resulting in even higher diversity compared to paraphrasing and noising, which may affect the original labels. Resampling-based techniques, such as oversampling and undersampling, address class imbalances without affecting the original labels. Typically, techniques involving the direct addition of extra data are not considered augmentation techniques. However, due to their usage in several papers, we have included them in the secondary techniques subsection. These secondary techniques excel at expanding data easily and quickly. They often involve integrating external data sources, translating similar datasets to address data scarcity, and combining different dialects to enhance the dataset's diversity.

Easy data augmentation (EDA) techniques are among the simplest approaches. In a study by the authors of [35], they boosted the performance of text classification tasks by implementing simple yet effective EDA techniques using WordNet [36] as a synonym thesaurus. These techniques included synonym replacement (SR), random insertion (RI), random swap (RS), and random deletion (RD). [5] adopted modifications of EDA techniques to better suit Arabic text. This involved addressing diacritics and handling the various ways in which Arabic words can be combined.

Figure 1 illustrates the goals of augmentation techniques. Figure 2 represents a taxonomy of NLP DA methods, and Fig. 3 expresses the distribution of DA techniques applied to Arabic.

2.1 Paraphrasing

Generally, to paraphrase means to rephrase a passage of writing using alternate vocabulary and sentence structures to make it more comprehensible [37]. These techniques enrich datasets while preserving similar information to the original form [14].

2.1.1 Thesauruses and dictionaries

Certain papers replaced words with other semantically similar words, such as synonyms, to retain the original context's meaning [14]. In [38], the authors applied synonym replacement, stating its ability to maintain embedded context information and sustain the sentence label. EDA SR methods were modified in [5], and Arabic word synonyms were obtained through WikiSynonyms and Arabic WordNet [2], for inserting and replacing synonyms of random words, employing lemmatizing, and stemming techniques. [39] employed synonym substitution and sentence rephrasing while preserving meanings. Monolingual Arabic sentences were augmented [22] through a dictionary-based random replacement of words with their English glossary entries using the MADAMIRA [40] tool. WordNet was employed [27] for crafting sentences with similar meanings using synonyms. **Algorithm. 1** presents a conceptual overview, excluding detailed implementation steps.

2.1.2 Semantic embeddings

These methods effectively address challenges associated with the replacement range in the thesaurus-based approach. They utilize pre-trained word embeddings like Glove, Word2Vec, FastText, etc., choosing to substitute the original word in the sentence with its nearest neighbor within the embedding space [14]. [5] introduced two semantic embedding variants using FastText, replacing NER entities with similar alternatives. The first variant replaced all words in sentences, while the second exclusively substituted entities. In [41], AraVec served as the source for pre-trained word embedding-based similar synonym substitution in tweets. The authors extended this approach in [42] by creating additional training data through word embedding-based methods. [43] leveraged a synonym dictionary with pre-trained AraVec embeddings for frequent words. [44] used GloVe word vectors and the Gensim library to augment word embeddings, creating instances by substituting each keyword with semantically similar words based on the corresponding word vectors' similarity. [45] utilized both FastText and AraVec embeddings to create aspect synonym lists based on cosine similarity, enriching aspects, and sentiment terms with high-quality constrained synonyms. In

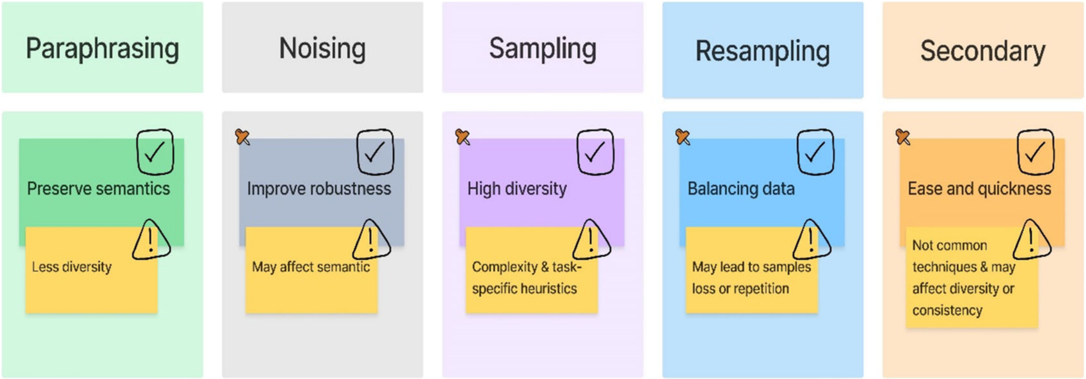


Fig. 1 Goals of augmentation techniques

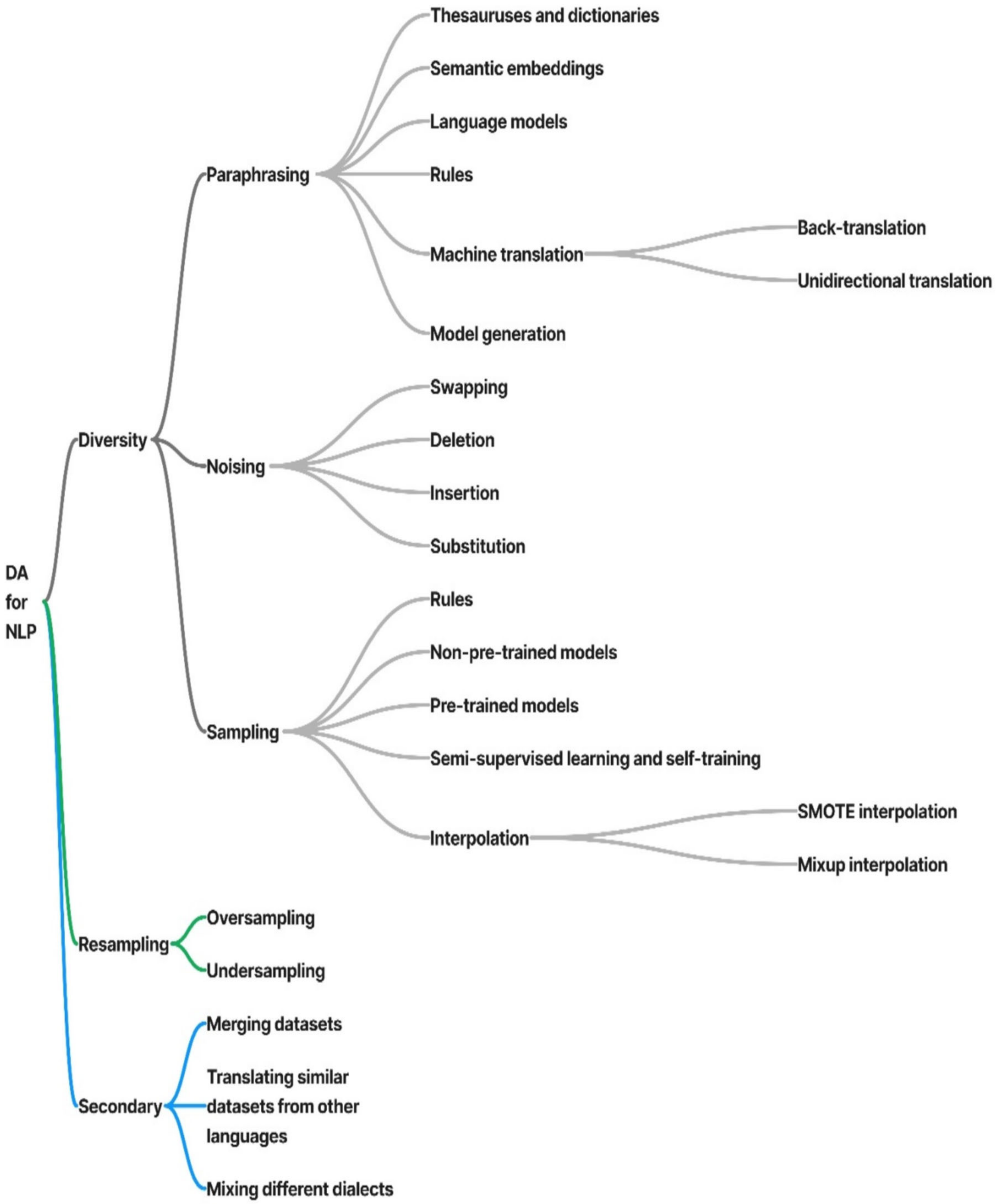


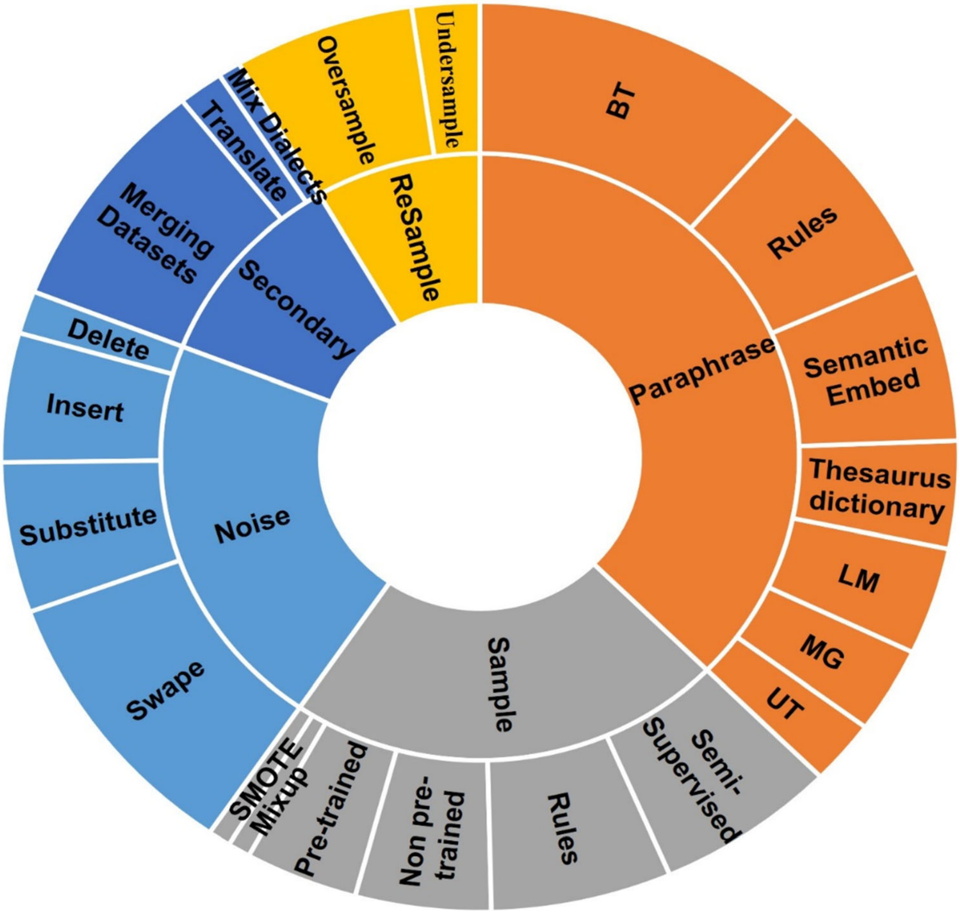
Fig. 2 Taxonomy of NLP DA methods

[24], words were randomly replaced with the most semantically similar words according to a Word2Vec model. Finally, [46] measured word vector similarity with GloVe embeddings and cosine similarity for word replacement. **Algorithm. 2** presents a conceptual overview, excluding detailed implementation steps.

2.1.3 Language models (LM)

A language model, such as BERT or BiGRU, predicts words based on context, enabling diverse data generation. In [31], a BiGRU-based model trained on 70% of a dataset used a one-shot technique to triple the examples by predicting new sentence vocabularies. The authors in [5] employed BERT and KERMIT for entity replacement, masking, and

Fig. 3 Distribution of DA techniques applied to Arabic



augmenting one entity at a time. AraBERTV02 was used by [45] to mask four randomly chosen words in the sentence, subsequently predicting, and introducing new replacement words guided by contextual modeling and tokenization. Multilingual BERT [47] masked Anaphoric Zero Pronouns (AZP) antecedents, replacing them with the most semantically similar tokens. [48] proposed semantics-preserving data augmentation, substituting unimportant tokens for improved understanding. Experiments on selectively perturbed masking were conducted to expand the training data without altering meanings. The authors concatenated auxiliary sentences and sentimental reviews using a special token, selectively replacing unimportant words with masked tokens. Auto-encoding and sequence-to-sequence (seq2seq) models, initialized with pre-trained BERT and BART weights, were employed to predict, and replace the masked words, thereby preserving the original context meanings. **Algorithm. 3** presents a conceptual overview, excluding detailed implementation steps.

2.1.4 Rules

Some studies leveraged language morphology and linguistics. [27] built an Arabic grammar-based framework that involved converting a labeled sentence into its parse tree, generating equivalent sentences with Arabic WordNet [2] synonyms, and creating variant-based transformation rules. A Negation module optionally introduced negation, substantially increasing output sentences with opposite labels. In [47], three augmentation rules were introduced for AZP generation and detection. These rules included changing subject-verb agreement, such as transforming the singular AZP verb and its reference to either dual or plural form, applying a two-word window on AZP samples' part of speech using t-test scores, and dropping subjects of verbs. Following linguistic rules, the authors of [49] manually transliterated the Tunisian Dialect (TD)-Modern Standard Arabic (MSA) corpus into Latin to address dual writing methods in TD texts. For example, ل was represented as L and م as M, resulting in an equivalent number of TD sentences in Latin.

While some surveys, such as [14], treated stemming as a data cleaning method, [50] utilized stemming to generate new words synthetically by extracting their root. They emphasized the efficacy of this technique for data augmentation, particularly in social media text, where grammar rules are often ignored, and people typically use stemmed or root words for posting. [51] addressed unpunctuated parallel text in a Tunisian dialectal corpus through manual segmentation and applied a rule-based approach in [15] by deleting all adjectives to increase sentence count while maintaining original context.

Some works relied on heuristic approaches to augment data. [52] augmented a model assessing the consistent meanings of shared words between sentences. They paired sentences with the same target word but different second sentences, labeling them as positive if both were positive (indicating the same meaning) and negative if one was positive and the other negative (indicating a different meaning). In [53], a two-step approach was followed based on emoji augmentation by removing emojis initially and subsequently adding positive and negative emojis to corresponding positive and negative tweets. Differing from typical DA techniques, [54] addressed the issue of models neglecting semantic meaning in class labels. They innovated a unique approach—label-semantic augmentation within a meta-learner framework. This entailed appending labels' semantic information to input sentences, effectively addressing implicit hidden semantic meanings and metaphors.

2.1.5 Machine translation

- *Back-Translation (BT)* A paraphrasing technique that involves translating sentences from a language into one or more other languages and then re-translating them back into the original language. This process typically results in the introduction of new vocabulary while preserving the semantics of the original source sentences [5, 14]. Translation services played a crucial role for BT. For instance, Google's translation service was employed to translate Arabic to English and back to Arabic in [47, 55, 56]. [5], utilized it along with a deep learning library to facilitate BT for code-switching data among English, Arabic, French, and German. Similarly, AWS's translation service was used by [57] for the BT of Arabic data, using English as an intermediate language. [21] used the same service to augment minority classes through BT of multigenre language data, while [58] used it for BT to enrich minority classes in Arabic and other languages using BERT-based models for subjectivity detection. In several studies, researchers utilized BT to translate Arabic sentences into English and back into Arabic. For example, [20, 59] implemented this approach, creating new samples that exhibit differences while maintaining some similarity. Additionally, researchers explored the use of BT with Arabic dialects in studies involving Egyptian and Levantine Arabic dialects, as seen in [60]. Furthermore, studies like [15] leveraged BT between TD and MSA. The authors in [49] took a similar approach, employing BT between TD and MSA in both directions. Addressing sentence alignment loss in TD augmentation, [51] resolved the issue by applying BT to MSA target sentences and aligning them with corresponding TD sentences. In the context of sentiment analysis, [61] applied BT for aspect-based tasks, while [62] used BT to enhance a multilabel sentimental model involving the application of metaheuristic genetic algorithms with ensemble learning. **Algorithm. 4** presents a conceptual overview, excluding detailed implementation steps.
- *Unidirectional translation (UT)* In contrast to BT, unidirectional translation involves translating the source text into other languages without the need to translate it back to the original language, typically in multilingual contexts [14]. In [32], a cross-lingual augmentation approach was adopted to enrich each language's training set, including Arabic. The authors translated each training sample into English, French, and German and appended both the original and translated samples to the training dataset. They trained a classifier using the mBERT model, aiming to enhance the model's ability to predict labels using translated contexts when the original context was insufficient. In [63], a multilingual model was implemented and trained on various languages, including Arabic, translated from the primary English language datasets. In [64], a model was trained on five languages, including Arabic, and machine translation via Google Translate was employed to translate tweets from each language to others.

2.1.6 Model generation (MG)

Given a word vocabulary, [3] created sequences where the probability of selecting a word (t+1) relied on previously generated words. They adopted a model similar to Senti-GAN, initially developed based on GANs. To control the generated words, authors used a penalty function rather than a reward for the discriminator, leading to more varied dialectal sentences of similar high quality to the originals. In [25], an augmentation system was devised through paraphrasing and model generation using transformer-based models, such as the AraGPT-2-Base model, along with defining similarity functions. [33] utilized OpenAI's GPT-3.5 to generate paraphrases and translations in English, Spanish, and Arabic, leveraging the multilingual pre-trained XLM-RoBERTa-large model. Based on contextual word embedding, [65] used statistical metrics and pre-trained AraBERT to generate similar words. Classifier chains were used for multilabel augmentation to address label dependencies neglected in usual approaches.

Algorithm. 1 Augmentation using thesauruses and dictionaries.

Input: An Arabic sentence
Output: A new sentence with synonyms replacing some words

```
cleaned_sentence = clean_text(sentence)
thesaurus = load_arabic_thesaurus()
augmented_sentence = []
for word in cleaned_sentence.split():
  synonyms = thesaurus.get_synonyms(word)
  if synonyms:
    augmented_sentence.append(random.choice(synonyms))
  else:
    augmented_sentence.append(word)
return ' '.join(augmented_sentence)
```

Algorithm. 2 Augmentation using semantic embeddings and cosine similarity.

Input: An Arabic sentence, and word embeddings
Output: The sentence with some words replaced by their closest semantic match from the embeddings

```
for token in sentence.split():
  token_vector = word_embeddings.get(token)
  if token_vector is not None:
    nearest_neighbor_vector =
      max(word_embeddings.values(), key=lambda x: cosine_similarity(token_vector, x))
    nearest_neighbor_token =
      [word for word, vector in word_embeddings.items() if vector is nearest_neighbor_vector][0]
    sentence = sentence.replace(token, nearest_neighbor_token, 1)
return sentence
```

Algorithm. 3 Augmentation using masking and AraBERTV02 prediction.

Input: An Arabic dataset
Output: A dataset with newly generated sentences where masked words are replaced by predictions from the AraBERTV02 model

```
model = AraBERTV02()
for original_sentence in dataset:
  masked_sentence = original_sentence.copy()
  random_word_idx = random.randint(0, len(original_sentence) - 1)
  masked_sentence[random_word_idx] = '[MASK]'
  predicted_word = model.predict_word(masked_sentence)
  new_sentence = original_sentence.copy()
  new_sentence[random_word_idx] = predicted_word
  dataset.append(new_sentence)
return dataset
```

Algorithm. 4 Augmentation using back-translation.

Input: An Arabic dataset, source language src_lang, and targeted language tgt_lang
Output: A new dataset with newly generated sentences through back translation

```
translated_dataset = []
for sentence in dataset:
  translated_sentence = translate(sentence, src_lang, tgt_lang)
  back_translated_sentence = translate(translated_sentence, tgt_lang, src_lang)
  translated_dataset.append(back_translated_sentence)
return translated_dataset
```

2.2 Noising

Noising-oriented techniques introduce minor variations that do not significantly alter the semantics, ensuring a suitable deviation from the original data [14].

2.2.1 Swapping

Swapping can double data, as in [66]. It can occur at the word level by randomly selecting words and changing their positions within sentences [14]. Additionally, it can be executed at the sentence level, as seen in various studies. For example, [46, 67] randomly swapped sentences, while [68] mixed TRAIN tweets, noting an issue of neglecting minority labels in tweet mixing augmentation. [69] generated synthetic sequences based on subparts of sentences. They created one record per technique tag and randomly mixed the produced sequences.

Word-level swapping involves techniques such as randomly swapping words [16, 28] and shuffling within small context windows [70]. The authors of [4] shuffled words to create variations of five new sentences from each original sentence. [19] applied consecutive swaps on reviews. [71] shuffled words for creating new samples, and [39] rearranged words and sentences randomly. In addition to swapping words, another approach was adopted in [26] by reversing the order of input words on the target side from right-to-left to left-to-right. **Algorithm. 5** presents a conceptual overview, excluding detailed implementation steps.

2.2.2 Deletion

Using this strategy, words are chosen at random and removed from sentences. In order to address concerns about an imbalanced dataset, the authors of [28] deleted words at random depending on whether a produced number between 0 and 1 fell below a certain threshold. The writers of [39] also used arbitrary word insertion and deletion. **Algorithm. 6** presents a conceptual overview, excluding detailed implementation steps.

2.2.3 Insertion

In order to insert and replace words through embeddings, the authors of [12] employed NLPAug and AraBERT. Through the contextual embeddings, [53] randomly selected words and inserted them into tweets using the NLPAug BERT model. Using MARBERT embeddings, [72] incorporated word insertion with transformers to add words to tweets at random. [5], researchers chose a word at random, looked up its synonym, and added it to the sentence. Words were added and removed at random [39]. After selecting a word at random, the authors of [15] used BERT pre-trained contextual embeddings to identify a similar word, which they then added to the original sentence without breaking the sentence's meaning. **Algorithm. 7** presents a conceptual overview, excluding detailed implementation steps.

2.2.4 Substitution

Substitution differs from paraphrasing in that it uses random replacement without guaranteeing equivalent meanings [14]. [22] randomly injected target-side words into a source-side Arabic-English corpus. This method ensured replacements occurred at random locations while maintaining alignment between selected target words and the source-target intersection. Furthermore, [26] employed one-to-one word alignment, replacing a specific number of target words in the source sentence with random entries from the training vocabulary.

The NLPAug Library played a key role in various studies, such as [12] with the utilization of AraBERT [53, 73] with the incorporation of BERT for substituting contextual embeddings. While [59] employed AraBERTv0.2 in conjunction with NLPAug to select suitable words for augmentation, [56] utilized NLPAug with AraBERTv0.2, resulting in the generation of sentences that exhibited significant differences in meaning in certain instances and semantically similar sentences in others. **Algorithm. 8** presents a conceptual overview, excluding detailed implementation steps.

Algorithm. 5 Augmentation by swapping two words.

```
Input: An Arabic sentence
Output: A new sentence with two swapped words
    words = sentence.split()
    idx1, idx2 = random.sample(range(len(words)), 2)
    words[idx1], words[idx2] = words[idx2], words[idx1]
    return ' '.join(words)
```

Algorithm. 6 Augmentation by random words deletion.

```
Input: An Arabic sentence, and a deletion threshold
Output: A new sentence with some words removed based on the given threshold
    words = sentence.split()
    for i in range(len(words)):
      if random.random() < threshold:
        del words[i]
    return ' '.join(words)
```

Algorithm. 7 Augmentation by word insertion.

```
Input: An Arabic sentence
Output: A new sentence with one extra word inserted at a random position
    words = sentence.split()
    idx = random.randint(0, len(words)-1)
    word_to_insert = get_similar_word(words[idx])
    words.append(word_to_insert)
    return ' '.join(words)
```

Algorithm. 8 Augmentation by word substitution.

```
Input: An Arabic sentence
Output: A new sentence with one word substituted by a similar word
    words = sentence.split()
    idx = random.randint(0, len(words)-1)
    word_to_substitute = get_similar_word(words[idx])
    words[idx] = word_to_substitute
    return ' '.join(words)
```

2.3 Sampling

Sampling-based techniques are similar to paraphrasing but for more specific tasks [59]. They work by mastering the data distribution and extracting fresh data from it. Much like paraphrasing-based models, they utilize rules and trained models for generating augmented data [14].

2.3.1 Rules

In contrast to rule-based paraphrasing, this approach produces novel samples that may not necessarily be similar to the original data, potentially having different labels [14]. Several studies employed symmetry, reflexivity, and transitivity relations in the context of question generation. In [74], these relations were utilized for question generation, with explanations: A) Symmetry—a question duplicating itself. B) Reflexivity: if question A duplicates B, then B duplicates A. C) Transitivity: if A does not duplicate B and B duplicates C, then A does not duplicate C. For augmenting models focused on semantic similarity of questions, these relations were utilized both in [55] to expand an Arabic question dataset and predict semantic similarity of question pairs and in [75] to generate examples of questions.

For enhancing grammatical error correction (GEC), [26] utilized compressed bitext, consisting of co-occurring biword pairs. They leveraged linguistic knowledge from Bitext for detailed sources in the target annotation. One-to-many word alignment, then target-side word reordering, induced non-fluent target sentences, boosting encoder attention during word generation. Additionally, a technique was proposed involving synthetic data generation by introducing a spelling error confusion function, emphasizing the target side during training to reinforce the encoder's influence on decoding.

Constraint propagation rules based on Allen's interval algebra rules [76] and transitivity and symmetry rules were used in [77] to augment temporal relations between entities. [78] augmented data through character substitution and word generation by replacing ة (Taa-marbuta) with ـه (Haa) and ز (Zain) with ذ (Zaal) and using prefixes and suffixes for morphological context-independent forms of words. [79] applied augmentation to a neural model by duplicating and fixing sentences that match linguistic error detector rules or that contain errors such as missing words or incorrect words' orders. In [80], TF-IDF vectorization extracted 15 keywords for each category, and manual generation of new examples by substituting these keywords into sentence templates did not yield significant improvements.

2.3.2 Non-pre-trained models

Several DA methods have utilized non-pre-trained models. Machine translation serves as a method of paraphrasing, such as simple back-translation, and it can also be employed in more advanced data sampling scenarios [14]. The bootstrapping technique, introduced in [60], addressed the challenge of direct back-translation with Arabic dialects due to a lack of sufficient monolingual data. The authors trained a reverse dialect-English machine translation system, iteratively back-translating with varied decoding parameters to generate diverse synthetic source sentences. In [26], an encoder-decoder transformer-based model with an attention mechanism was used for GEC and neural machine translation (NMT). The authors incorporated a masking token< UNK>during training to weaken the decoder and compel the model to generate accurate predictions by relying on the latent variable. In [81], language models were employed to generate text, followed by back-translation through transformer models, rule-based, and semantic filtering.

In [22], authors augmented monolingual Arabic sentences by injecting words from the English side, using a predictive sequence-to-sequence model to learn replacement points. They developed an algorithm for identifying code-switched segments and fine-tuned an mBERT model for predicting target words. [82] introduced a method merging pseudo-relevance feedback, deep averaging networks, and word embeddings to augment Arabic information retrieval. This approach identified and added semantically related terms to the original query, demonstrating superior performance, particularly on specific topics, by selecting relevant terms from the top pseudo-relevant documents. In [83], dataset augmentation occurred through a multistep process involving the generation of offensive and hateful tweets by sampling 500K unlabeled tweets containing "Ya" trigger words and seed lexicon words. Negative sentiment labeling was done using AraNet.

2.3.3 Pre-trained models

GPT-based models and BERT-based models are valuable for generating synthetic data. [29] employed AraGPT2-base to create synthetic data, integrating it into a transformer summarization pipeline with real data to generate synthetic summaries with the AraBART model fine-tuned on both real and synthetic data. In [80], GPT-3 was used to augment text, using prompts to produce sarcastic phrases. However, the results showed limitations in generating diverse sentences due to insufficient fine-tuning data. For English and Arabic, [81] leveraged GPT-J and mGPT pre-trained language models to generate in-domain synthetic segments. Authors in [59] utilized the AraGPT2-medium model for new text generation, while [84] employed AraBERT and MARBERT transformer models for classification and augmented instance generation in tweets, involving text replacement and word injection.

2.3.4 Semi-supervised learning (SSL) and self-training

Machine learning can be categorized into three primary groups: supervised learning models, which rely on massively labeled training data; unsupervised learning models, which can be trained without the need for labeled data; and semi-supervised models, suitable for situations with limited labeled data while the majority remains unlabeled [19]. Self-training, an inductive, semi-supervised technique [85], is considered a form of sampling DA technique [14]. Semi-supervised learning proved successful in reducing the manual annotation requirement for NLP tasks [8]. However, some studies exclude it from DA techniques [34]. In the scenario outlined by [86], self-training involves using a small number of labeled samples to train a classifier, which then classifies the unlabeled sample set. Samples with confidently predicted labels are appended to the labeled set based on the classifier's output. Retraining can continue until all samples are included or a specified stopping criterion is met.

In [85], a semi-supervised learning strategy was implemented, utilizing self-training with Support Vector Machine (SVM). This involved iteratively expanding the labeled data through the addition of newly labeled data. A similar self-learning process, following a semi-supervised approach, was utilized in [30] to iteratively enhance training datasets for hate speech detection. Self-training, employing BERT with softmax, was also explored in [86] to augment the training set for the Arabic dialect identification task. However, [68] highlighted the limitations of this approach, such as “catastrophic forgetting” where the model may forget initial data, and the potential increase in training and initial prediction errors. To address these challenges, a modified version of the semi-supervised method was adopted.

Additionally, [18] integrated zero and few-shot learning with self-training to address data scarcity in multidialectal Arabic data. They utilized labeled MSA data to self-train dialectal Arabic data. In [17], authors employed semi-supervised learning with GANs on a limited labeled dataset to classify tweet sentiments. LSTM models were used for both the discriminator and generator, with the discriminator trained semi-supervised on labeled and unlabeled data. Furthermore, [15] implemented a semi-supervised augmentation method for generating MSA sentences from TD using a statistical machine translation model and further improved results by incorporating a 3-g language model.

Data distillation is an omni-supervised learning approach that leverages both labeled and unlabeled data for self-training, aiming to improve model performance [87]. Self-training, distillation, and teacher models can be leveraged to label unlabeled data in various scenarios [14]. A cross-lingual machine reading comprehension strategy using multi-teacher distillation and zero-shot transfer was presented in [88]. After training models in numerous languages, it distilled them to create a final model for all target languages. English questions and passages were translated into different languages. Then, a method built the model dataset for every language.

2.3.5 Interpolation

In the domain of numerical analysis, interpolation is a method used to generate new data points based on existing ones. However, combining two different text instances isn’t straightforward. A novel sentence can be created by interpolating the hidden states of two sentences, resulting in encapsulating the meaning of the original sentences [34].

Synthetic Minority Oversampling Technique (SMOTE) interpolation: This technique helps in scenarios such as the balancing of classes. It works only with numeric features as inputs for oversampling [65] and involves searching for close neighbors in the feature space of a specific instance to perform interpolation [34]. Only instances belonging to the same class are interpolated, which enhances the safety of the technique. Originally, it was designed within its context to oversample the minority class. However, this approach results in limited diversity and novelty in the generated instances. It was defined in [34] as:

$$\tilde{x} = x_i + \lambda * dist\left(x_i, x_j\right) \tag{1}$$

where (*xi*,*yi*) is the original instance and (*xj*, *yi*) is a close neighbor instance sharing the same class label. *dis*(*a*, *b*) is a distance measure and λ ∈ [0,1]. To overcome the class imbalance issue [7], augmented data utilized the SMOTE technique with an active learning approach to reduce the human intervention required for manually labeling samples.

Mixup interpolation: This technique augments samples using virtual embeddings rather than text generated from natural language. Samples in the virtual vector space are taken from the existing data, which may or may not have the same labels as the original data [14]. The need for continuous input was a challenge to applying this method to NLP tasks. A workaround for this is the mixing of hidden layers or embeddings [13]. The authors of [89] extended Mixup, which randomly selected samples without taking the spatial distribution of dataset samples into account. They introduced DMIX, an adaptive distance-aware interpolative Mixup that took into account the similarity between latent representations of samples in the embedding space, specifically in the hyperbolic space.

2.4 Resampling-based techniques

These techniques can help with some issues, such as addressing class imbalances [13]. They should be used cautiously, though, as oversampling could produce redundancy in the minority class samples, while undersampling could cause a considerable loss of important information [65]. It is very important to mention that some surveys, such as [14], considered them other data manipulation techniques instead of data augmentation techniques. They were, however, employed as data augmentation techniques in some works, such as [90].

2.4.1 Oversampling

Techniques are employed to address data imbalance by increasing minority class samples, is common in imbalanced scenarios [14]. In [90], tweets were duplicated to augment the minor sarcastic class in an imbalanced dataset. Similarly, [91] duplicated negative and neutral samples for code-switched data. Although [80] copied and pasted sentences to enhance sarcastic keyword understanding, results showed only trivial improvement. In [26], authors introduced a copy DA strategy to strengthen the encoder and weaken the decoder. They leverage noise features in augmented data by transcribing source sentences into targets, enhancing synthetic data distribution. However, [71] found replicating words ineffective. To overcome overfitting from minority class duplication, [92] balanced minority labels with majority classes. Additionally, [44] repeated instances for class and label balance, while [81] used mixed fine-tuning by combining synthetic data with oversampling.

2.4.2 Undersampling

Techniques address class imbalance by reducing samples in the majority class to align with the minority class. In [90], non-sarcastic samples were removed, and in [84], 32% of non-sarcastic tweets were down-sampled to maintain topic diversity. In order to provide a fair comparison across languages and prevent the removal of many tweets while addressing the issue of imbalanced label distribution, [64] fixed a 30% check-worthy claim ratio between languages.

2.5 Secondary-based techniques

DA uses methods to increase the variety of training samples without explicitly collecting new data [13]. However, some studies added external data, merged databases, and collected more data. These approaches are not common DA techniques. Merging data from several sources may cause inconsistencies. However, they excel at being simple and straightforward to apply.

2.5.1 Merging datasets

The authors of [93] implemented two sub-tasks: sarcasm detection and sentiment analysis. Their dataset was annotated with sarcasm Boolean labels and sentimental labels (NEU, NEG, or POS). Recognizing that sarcastic sentences often imply negative sentiment, they sourced data from another sentimental dataset, replacing “negative” with “TRUE” and “positive” with “FALSE” for sarcasm detection. In [78], data was merged to create a profanity list. Augmentation rules and a morphological approach were applied, manually expanding an online bad word list by adding words from the training dataset through inspection.

In [94], a combination of existing datasets with in-house manually labeled data and crawled tweets from users’ timelines was utilized. [44] expanded the initial dataset by incorporating additional instances from external datasets following a thorough data filtering process. Similarly, for dialect identification, the authors of [39] expanded the initial dataset by incorporating external datasets in the same domain representing various dialects, including tweets and a collection of songs’ lyrics.

The authors of [95] added YouTube comments. Similarly, [71] added offensive and non-offensive YouTube comment samples to the dataset. In [96], data from social media were processed and added to the dataset after selecting Facebook pages, collecting comments, and annotating the data manually. [97] improved models by randomly adding 2/3 Dev data to the training data. [98] collected data from various resources to balance the original dataset, while [99] augmented sentiment-related datasets with additional manually annotated samples from the Twitter API.

2.5.2 Translating similar datasets from other languages

This subsection explores studies that translated datasets between languages without additional manipulations. In [100], authors translated an English dataset to Arabic to address data scarcity and imbalance. [101] augmented data for predicting intimacy across several languages, including Arabic, by translating an additional dataset from English using Google Translate. Nevertheless, the model’s translation augmentation performance suffered due to inaccurate translations and cultural differences, as noted by the authors.

2.5.3 Mixing different dialects

The integration with other dialects approach was followed in [49] by adding Moroccan and Algerian dialectal sentences to the original Tunisian dialect corpus. However, BLEU scores dropped as the added dialects lacked sufficient sentences, causing performance to decline.

In summary, the diverse array of Arabic data augmentation techniques, including diversity, resampling, and secondary methods, plays a crucial role in enriching datasets and enhancing model performance. These methods not only introduce variety but also address specific challenges such as data imbalance and semantic preservation. Moving forward to the next section, we will focus on studying the training models’ pipeline and then provide insights into assessing data augmentation performance.

3 Strategies for training models and measuring augmentation performance

Building on the insights from Sect. 2, where we delved into various Arabic DA methods, this section begins by examining the NLP task pipeline where these augmentation techniques are applied. Understanding the pipeline is crucial for optimizing DA’s impact. Following this, we will discuss the metrics and techniques used to assess the effectiveness of different augmentation strategies. Evaluating these frameworks is essential for determining the practical impact of augmentation methods on NLP tasks.

Typically, augmentation serves as a component within the model pipeline that involves data preprocessing, feature extraction, training, and evaluation. Techniques vary based on language and task, influencing classification outcomes. Its versatility across machine learning tasks leads to diverse model architectures and training and evaluation methods.

This section illustrates these concepts with a few selected examples from Arabic literature.

3.1 Preprocessing

The purpose of this step is to prepare data for model input, and its requirements vary across different tasks. Here are just a few illustrative examples.

- *Removing diacritics* Diacritics in Arabic, are symbols that indicate short vowels and other pronunciation features, crucial for distinguishing word meanings. For example, “عَلَم” means “flag,” while “عِلْم” means “knowledge.” However, both Modern Standard Arabic and colloquial Arabic usually omit diacritics in writing, relying on context for understanding. This omission leads to lexical ambiguity but simplifies written communication. In text augmentation research, including papers [15, 17, 24, 41, 70, 80], diacritics are often removed during preprocessing to normalize text and focus on contextual understanding.
- *Removing punctuation* In Arabic, punctuation marks (التّرقيم) such as the comma (،), period (.), question mark (؟), and exclamation mark (!) play a crucial role in structuring sentences and clarifying meaning. However, they do not alter the fundamental meaning of the words themselves. Some papers, like [3, 15, 24, 41, 45, 46, 53, 74, 90], opted to remove punctuation during preprocessing to reduce noise and simplify the data. This approach helps minimize the size of vector representations and does not break the meaning of certain NLP tasks.
- *Removing numbers* Similarly, some papers [3, 45, 46] removed numbers, which are symbols used to represent quantities or values in written form, to streamline the data and focus on linguistic elements more relevant to certain NLP tasks.
- *Removing stop words* Some papers [41, 46] removed stop words, which are common, low-information words such as “و” (and), “في” (in), and “من” (from), to reduce noise and focus on the more meaningful content of the text.
- *Removing special characters* Some papers [45, 46, 53, 70] removed special characters, such as (‘@’, ‘#’, ‘%’, ‘&’, ‘*’, ‘?’, ‘∧’), and sequences of English letters, as these characters often have no phonetic value and can introduce noise into the data. By eliminating unnecessary characters, including those with no direct impact on linguistic meaning, the papers aimed to simplify the text and improve the performance of various NLP tasks. However, some papers retained specific characters like ‘/’ and ‘-‘ when they were essential for extracting meaningful entities, such as dates.
- *Removing duplications* Removing duplicates helps eliminate redundancy and ensure that each piece of data contributes unique, non-repetitive information. In papers [41, 53, 70], this involved eliminating repeated instances of text, such as duplicate tweets or redundant characters and punctuation marks, which reduces data noise and enhances the quality and efficiency of the analysis.
- *Removing non-Arabic alphanumeric* This involves eliminating Latin characters, numbers, and any non-Arabic alphabets from the text. This step can significantly impact the text by removing sequences of English letters and numbers that may not be relevant to the context of Arabic text processing. Papers such as [3, 41, 45, 70, 74, 90] removed these elements to focus exclusively on Arabic script, thereby reducing noise and simplifying the data.
- *Removing noise and spam* This process involves eliminating elements like short vowels, symbols, and irrelevant content such as advertisements or inappropriate links from the text. In Arabic text processing, noise removal can significantly impact word meaning and computational efficiency by filtering out unnecessary characters and symbols that interfere with text manipulation. Papers such as [15, 46, 70] removed noise and spam to enhance the quality and relevance of the data.
- *Removing other dialects* Filtering out texts in different dialects reduces variability and noise, enhancing dataset uniformity. [70] removed tweets containing other dialects, such as Arabic Gulf dialects.
- *Removing/adding spaces/inputs* Adding or removing spaces is a critical step due to the language's script and word formation. Incorrect spacing can change word meanings or create non-words, impacting readability and comprehension. [37, 46, 70, 90] focused on this aspect to normalize texts, ensuring words are correctly segmented or merged.
- *Handling links/URLs* Handling links and URLs is crucial as they do not contribute to the semantic content and can disrupt text analysis. Links can be lengthy and contain characters that confuse language models. Papers [37, 44, 53] address this by replacing URLs with unique tokens or removing them altogether, thereby reducing noise and ensuring cleaner, more relevant data.
- *Handling emojis and icons* Handling emojis and icons is essential, as they can convey emotions and context not captured by words alone. However, their presence can complicate text analysis and model training. [17, 44, 90]

addressed this with approaches such as removing emojis or converting them to their textual descriptions using tools like the “emoji” Python library.

- *Limiting the length of words* Limiting the length of words helps to standardize input data and improve model efficiency. Extremely long words can introduce noise and irregularities, complicating analysis and slowing down processing. [3, 80] addressed this by restricting word lengths, ensuring texts remain concise and manageable.
- *Normalization* Normalization involves standardizing characters, removing diacritics, eliminating repeating characters, and unifying different forms of the same letter to create a consistent and clean dataset. This process reduces noise and enhances the clarity of the text, ensuring that variations in writing do not confuse the language models [15, 41, 46, 53, 80].
- *Segmentation* Segmentation involves dividing text into meaningful units, such as words or phrases, to facilitate better analysis and understanding. This process is crucial for languages with complex morphology and script, as it helps in accurate tokenization and improves model performance. [15, 37, 74] applied the segmentation step to break down text and enhance data quality using tools such as Farasa.
- *Lemmatization* Lemmatization involves reducing words to their base or root form, which helps in understanding their core meaning and improving the consistency of text data [5, 15].
- *Tokenization* Tokenization involves splitting text into individual units, like words or sub-words, aiding in clearer analysis and better model performance. Various techniques have been applied [15, 41, 55, 74], such as relying on whitespace tokenization for simplicity and utilizing the AutoTokenizer from the Transformers library.
- *Replacing character* (e.g., Alif): In Arabic text preprocessing, replacing characters involves standardizing different forms of letters to reduce inconsistencies and improve text uniformity. This process addresses variations such as replacing (ى) with (ي) or different forms of Alif (آ ,إ ,أ) with bare Alif (ا). Such replacements are crucial for maintaining the semantic integrity of words [24, 70].
- *Stemming* Stemming involves reducing words to their root forms, which helps handle the language’s morphological complexity [5, 41]. This may involve stripping away elements like prefixes or suffixes to reveal the word’s stem, which is essential for understanding the core meaning of words and finding their synonyms.
- *Canonicalization* In NLP, canonicalization involves reducing words to their base or root form using techniques like stemming or lemmatization. In the context of Arabic, this process helps manage the language’s morphological complexity. For instance, [27] converted input sentences into parse trees to facilitate the application of morphological and linguistic rules for text augmentation.
- *Mapping Hindi numbers to Arabic* [37] replaced Hindi numbers with their corresponding Arabic ones.
- *Correcting misspelled words* [70] manually corrected words with missing letters, incorrect replacements, or improper spellings.

3.2 Feature extraction

Representing data extensively involves extracting essential information and characteristics through feature extraction [9]. Below are just a few illustrative examples.

Patterns and rule-based techniques [27] defined grammar-based rules for nominal sentences, verbal sentences, questions, verbs, adjectives, pronouns, prepositions, conjunctions, and numbers defined using the Stanford Arabic parser tagset. Additionally, the author built a set of parse trees of the Arabic sentences using Stanford Arabic Parser. Using these rules, the author was able to generate equivalent sentences to augment the original dataset.

- *Word Embedding and Pre-Trained Models* NLP tasks involve processing textual data, but machine learning algorithms necessitate numerical representations. To address this, various word embedding techniques were employed. Word2vec and GloVe, for instance, were used to convert textual data into real-valued vectors.
- *Word2vec* employs Continuous Bag of Words (CBOW) and skip-gram models [70], as seen in [45], for aspect vector generation.
- *GloVe* word embeddings were also used for vectorization [46], and the glove-twitter-100 model, which was trained on two billion tweets with 100-dimensional word vectors, was used for word embeddings [44].
- *FastText*, a pre-trained continuous word embedding by Facebook, is capable of learning vectors for complete words and sub-components like character N-grams [5], providing an advantage in producing embeddings based on character N-gram features, even if not processed during training [5]. FastText was used as a word embedding technique in [45, 46].

- *ELMo*, a deep contextualized, pre-trained word representation, was employed in [74] to map textual questions into a vector space.
- *AraVec*, a pre-trained distributed word representation offering six models for Arabic collected from Wikipedia, Twitter, and Common Crawl webpage crawl data, provides CBOW and skip-gram models for each resource. The embeddings from AraVec were used in [27] for classifiers, [70] as a pre-trained CBOW model, and [45] as a word embedding technique. Additionally, AraVec-TWI, a 300-dimensional CBOW vector, was utilized in [41] for social spam detection.
- *Generative Pre-Training Transformer (GPT) models* function as unidirectional language models. GPT-2, as referenced in [29], generated texts from in-domain sentences, while AraGPT2 [102] served as an Arabic version of the GTP2 model in [59] for the GPT-2 model. In [80], GPT-3 produced new sentences.
- *Bidirectional Encoder Representations from Transformers (BERT)* are pre-trained deep bidirectional transformer models, featured in [25] for computing embedded words from text, [5] for code-switching Arabic-English NER embeddings, [44] as CAMeLBERT-Mix, [61] for pre-trained language models, [17] for contextualized embeddings through CAMeLBERT, [15] for pre-trained contextual embeddings, and for AraBERT in [54, 90].

3.3 Training models

Augmentation can be used with various NLP downstream tasks. As a result, there is no one way to train the augmented model. This subsection illustrates this with just a few selected examples from Arabic literature.

- *SVM* was employed in various studies, including [3, 27, 39, 65, 74, 85]. It was employed with stochastic gradient descent [7]. Linear Support Vector Classification (LinearSVC) was employed in [41].
- *Logistic Regression* in [41, 65], decision tree (DT) in [39, 65], multilayer perceptron (MLP) in [39], *random f*orest (RF) as an ensemble learning method in [90], and Naïve Bayes (NB) in [27, 41].
- *Multinomial Naïve Bayes (MNB)* in [3, 4], K-nearest neighbor (KNN) in [27, 39].
- *General Deep Neural Network (DNN)* in [3], convolutional neural network (CNN) in [3, 70]. Recurrent convolution neural network (RCNN) with LSTM in [70], and long short-term memory (LSTM) in [3, 16, 17].
- *Transformer architectures* were employed in different studies: the RoBERTa classifier (12 layers, 768 hidden, 12 self-attention heads, 125 M parameters) and the XLM-RoBERTa classifier (12 layers, 768 hidden, 8 self-attention heads, 125 M parameters). In [80] the BERT-Base model was used for multilabel classification, while RoBERTa and XLM-RoBERTa were used for binary classification. Additionally, BERT architecture for English data and fine-tuned multilingual BERT (mBERT) for other languages were described in [32, 89]. Ensemble voting module with AraBERT, MARBERT, and ARA-ELECTRA in [12]. AraBERT vr2 model in [24], pre-trained AraBERT base v0.2 model in [55], and AraBERT twitter-base model in [50, 56].

3.4 Measuring similarities and distances

A common practice when employing similarity metrics such as Euclidean and cosine is to convert textual data into numerical representations, like word embeddings. This facilitates the calculation of distances between word vectors and enables the mathematical determination of semantic similarity between sentences [25]. The cosine metric focuses on the direction of two sentence points, regardless of their size, resulting in similar scores for small and large sentences [37]. In contrast, the Euclidean metric is influenced by sentence size, potentially leading to dissimilar similarity results for small and large sentences. Additionally, Jaccard and BLEU similarities assess novelty and diversity between two sentences [37]. Furthermore, experiments [25] indicated that when dealing with large imbalanced Arabic datasets, BLEU was the ideal similarity metric to utilize, whereas Jaccard was the recommended metric to use for smaller datasets.

- *Euclidean distance* calculates the distance between two points by measuring the length of the path between them using the Pythagorean Theorem. A shorter distance between word vectors yields a higher score, indicating similarity [37]. It has been utilized for tasks such as selecting informative data [85] and calculating distances between embeddings in sentiment analysis [48].
- *Cosine similarity* measures the similarity between two-word vectors based on the cosine of the angle between them. In applications like finding similar vectors [82], selecting informative data [85], or identifying synonyms for word replacement [5], cosine similarity has proven valuable. It has also been employed to create unique aspect vectors in sentiment analysis [45] and measure word vector similarity for replacement using embeddings [46]. In back-translation assessment [58], average cosine similarity and weighted average sentiment scores were used.

- *Jaccard similarity* operates on objects as sets instead of operating on them as vectors. It is the intersection cardinality divided by the union cardinality of object sets [37]. Used for measuring novelty and diversity [3], it served as an indicator of overlap between predicted and correct labels [7].
- *Hyperbolic distance* as used in [89], provides a solution for defining similarity between latent representations, particularly for capturing complex hierarchical information not effectively handled by standard Euclidean space.
- *Bilingual Evaluation Understudy (BLEU) score*, an algorithm measuring similarity between sentences, calculates common words using unigrams and n-grams [37]. It has been employed as a similarity metric in certain studies, including [25].

3.5 Measuring augmentation quality and impact on performance

Text augmentation is used in various NLP tasks, and different methods are used to assess its quality and impact on tasks. In this subsection, we discuss some of these methods. They work together, allowing for a better evaluation of the impact of DA. Using one or more of these methods does not conflict with using others. You don't have to rely on just one; using one or more at the same time is possible.

3.5.1 Novelty and diversity evaluation

These terms are used by different NLP researchers to describe text variability, such as word generation [25]. To calculate them, there is no uniform equation or algorithm. They could instead be assessed heuristically according to the task. The degree to which the generated sentences are unique and not merely duplicates from the training corpus is measured by novelty. It cannot be assessed using the indicators of sentence logic or fluency. An example of it was applied to a modified SentiGAN model [3] as follows:

$$Novelty(S_i) = 1 - max\left\{\phi\left(S_i, C_j\right)\right\}_{j=1}^{j=|C|} \tag{2}$$

where C is the sentence set of the training corpus, ϕ is a similarity function, the Jaccard function in this case, and Si is a generated sentence. Diversity measures how much the generated sentences differ from the original sentences in the same dataset. It was applied to the modified SentiGAN [3] model as:

$$Diversity(S_i) = 1 - max\left\{\phi\left(S_i, S_j\right)\right\}_{j=1}^{j=|S|,j\neq i} \tag{3}$$

where S is the generated sentence collection, ϕ is a similarity function, the Jaccard function in this case, and Si is a generated sentence.

3.5.2 Human, intrinsic, and extrinsic evaluation

Human evaluation assesses language system performance using people as judges. It's manual and costly, involving experts to evaluate outputs. Automatic evaluation is more efficient compared to references or measuring impact on other systems. Intrinsic evaluation checks the quality of NLP system outputs against ground truth, like a reference text. It focuses on the system's direct task performance. For example, in paraphrase evaluation, it asks if the generated paraphrase conveys the original text's meaning. For example, using the BLEU metric to evaluate the quality of paraphrases. Extrinsic evaluation assesses NLP system outputs based on their impact on other NLP systems' performance. It examines the contributions of outputs to broader tasks, such as improving question-answering models or improving text classification models by using generated paraphrases [103].

[22] evaluated augmentation techniques using intrinsic, extrinsic, and human assessments. They conducted intrinsic evaluations on predictive models for code-switching (CS) by comparing CS predictions with actual points, analyzing distribution, and evaluating synthetic CS data against natural occurrences according to how similar they were. For the extrinsic evaluation, they measured the achieved DA performance on several NLP tasks they implemented, which included DA on language models, automatic speech recognition, machine translation, and speech translation tasks. With the help of human annotators, they performed human evaluations of the quality of the generated sentences. Bilingual Egyptian Arabic-English speakers judged 900 sentences for understandability and naturalness. Authors faced subjectivity in human evaluation due to user-dependent code-switching that differs from one user to another. They addressed this issue by using three annotators and averaging ratings. [5] conducted an extrinsic evaluation, evaluating the impact of the generated data on an external task. They appraised the quality of the produced data by measuring its contribution to enhancing NER on CS data.

3.5.3 Model's performance-based evaluation

Various methods are available for assessing the performance of augmented models, and their suitability may vary depending on the specific model and task. Superior end model results are a sign of successful DA, whereas inadequate end model results are a sign of DA failure. To provide a clearer understanding, we will discuss a few scoring metrics commonly used to evaluate model performance with a few examples.

- *BLEU score* was used in [51] to measure the performance improvement after augmenting a machine translation task. In [26], the BLEU-4 score was utilized to evaluate the performance of machine corrections, considering not only accuracy but also fluency and naturalness. [60] employed the BLEU score to assess a new augmentation technique's performance. The evaluation of a task involving the translation of Tunisian dialect texts in social networks in [49] and the assessment of augmentation results in an NMT task in [15] also employed the BLEU score.
- *Recall-Oriented Understudy for Gisting Evaluation (ROUGE) score*, widely used in text summarization tasks, measures the quality of summaries by counting the overlap of n-grams between the system and reference summary [29], with the mentioned study utilizing ROUGE 1 F1, ROUGE 2 F1, ROUGE L F1, and ROUGE-L-sum to score the generated abstractive summary with augmentation.
- *Precision* was used by [26] to measure the performance of a grammatical error correction system in correcting errors and was described as the proportion of true positives to the total of false negatives and true positives. The number of relevant results in the top k claims in the ranked list was defined as Precision @ k in [57], where the authors used Precision @ 30 (P@30) as the metric for fact-checking claims in Arabic. [82] employed Precision @ 10 (P@10) for performance measurement.
- *Accuracy* is one of the most commonly used metrics, as applied in several studies such as [3, 16, 27, 41, 55].
- *F1-Score* was utilized in [65], where the impact of augmentation on imbalanced datasets was studied, stating that for nearly balanced classes, accuracy is preferred, while for imbalanced data, F1-score is more suitable. It was also used in [25, 41].
- *Average Precision (AP)* was used in [64] to evaluate a model trained on five different languages and the impact of various augmentation methods.
- *Mean Average Precision (mAP)* was utilized in [59], where claim detection was treated as a ranking task and mAP was used as the average of the AP's (average precisions) across classes. The same metric was applied in [73], where tweets were classified as check-worthy claims and ranked in priority for fact-checking. mAP was also used in [82], considering the top 100 documents.

3.5.4 Ablation experiment-based evaluation

Ablation experiments involve systematically removing components from the pipeline to evaluate their impact. For instance, one might assess the influence of excluding DA by omitting it from the pipeline and recalculating model scores. [59] performed ablation experiments by excluding DA from a few-shot learning model. The outcomes demonstrated that the augmented model outperformed the non-augmented one by 1% after augmentation was implemented. In [18], an NER task ablation study compared self-training using unlabeled MSA and equally sized DA data. Results showed a 2.67-point F1 score improvement with unlabeled DA, indicating its efficacy in adapting the model during testing. [32] conducted an ablation study on a cross-lingual augmentation technique, showing improved results. Arabic achieved an accuracy of 0.8734 with DA and 0.8684 without it. [4] studied how adding DA affected the model. Results showed a small improvement, maybe because of how the classifier works.

In conclusion, the strategies discussed in this section for training models and measuring the performance of augmentation techniques form the foundation for optimizing NLP tasks. By understanding the impact of preprocessing, feature extraction, and various model architectures, we can better evaluate the effectiveness of data augmentation methods. The insights gained from these evaluations pave the way for exploring the broader benefits of augmentation, which will be the focus of the next section.

4 Augmentation benefits and applied tasks

Building on the detailed strategies for training models and evaluating augmentation techniques discussed in the previous section, this section delves into the tangible benefits and practical applications of augmentation. We will examine how augmentation not only expands and improves datasets but also enhances model performance across various NLP tasks. By exploring specific examples from Arabic literature, we aim to highlight the significant impact augmentation can have on overcoming data limitations and achieving better results in language processing tasks.

4.1 Augmentation benefits

Augmentation has various benefits, as shown in Fig. 4, and we will explore a few examples within the context of Arabic literature.

4.1.1 Expanding datasets

Researchers have explored different techniques, employing DA, to increase datasets for languages like Arabic with limited data. Notably, the use of text generation [3], specifically targeting dialectal Arabic datasets, was crucial in overcoming the challenges of having limited annotated data. Additionally, [56] used back-translation and synonym replacement, resulting in a significant increase in the religious hate class data (1666.6%) and social hate class data (579.7%). F1 was increased by 13% in [25] through paraphrasing.

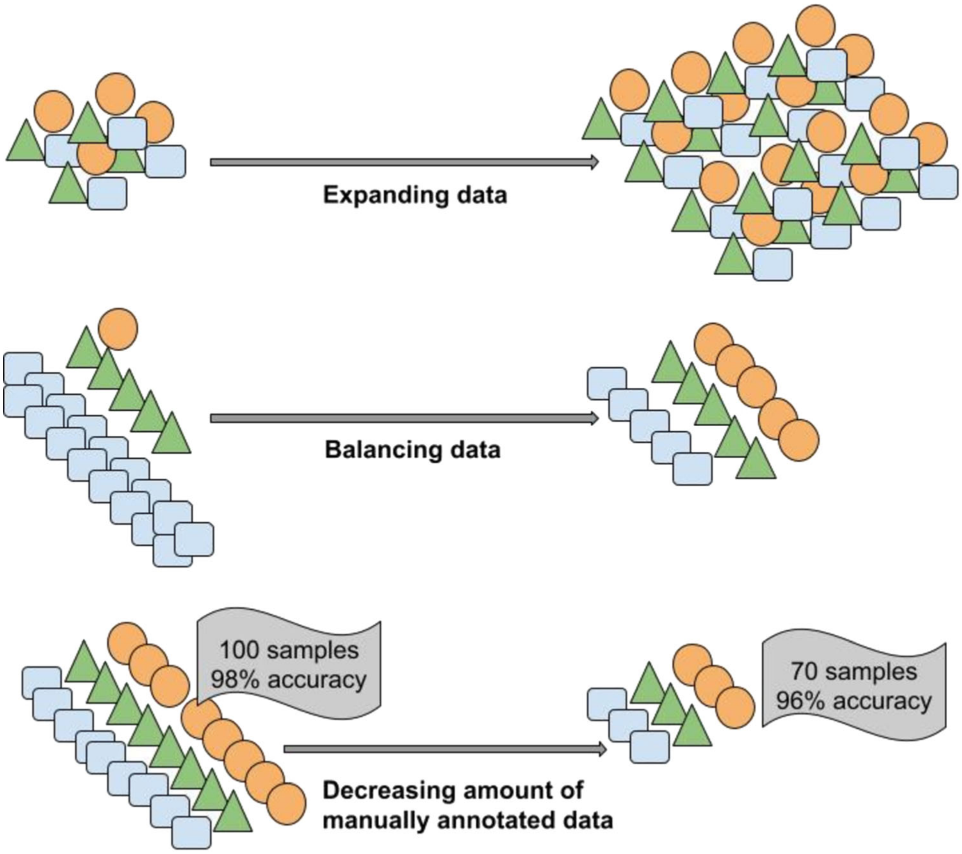
Improving datasets through rule-based strategies has proven significant in overcoming data scarcity. In [27], defining Arabic grammatical rules led to a tenfold increase in initial datasets, along with an impressive 42% average accuracy improvement. Moreover, [79] applied a set of rules, resulting in a notable 25.5% increase in sentences. Training data were increased four times over [75] by using transitive, symmetric, and reflexive rules. The number of samples increased from 504 before augmentation to 1512 after augmentation by stemming words [50].

Enhancing model performance through embedding-based methods has been a focus. Researchers, as seen in [41], doubled the number of spam tweets four times through substitutions based on embeddings. As shown in [15], utilizing BERT pre-trained contextual embeddings in the Transformer model's baseline corpus raised its BLEU score from 15.9 to 60. Furthermore, [45] achieved a significant accuracy boost from 91.5% to 93.18% by using semantic embeddings, particularly relying on FastText-based augmentation.

Using the language branch knowledge distillation model in [88], an evaluation metric was used, and the results showed more than 4% improvement on one dataset and more than 2 points increase on another. Randomization techniques offer a way to introduce variability to enhance model robustness. Studies such as [4] used random shuffling to expand the training data six times, while [16], random swap noising augmentation led to improvements of 14.06% and 12.57% in Bahraini and MSA dialects, respectively. [69] improved the micro-F1 score by 5% via a mixture of subparts of the sequences. The LSTM model's accuracy [70] increased by 8.3% using random shuffling.

Morphological changes, merging with online lists, and incorporating additional training data [78] led to a substantial expansion of the seed list of bad words from 87 to 5497. [19] obtained ten thousand reviews by augmenting the initial five thousand review datasets.

Fig. 4 Augmentation benefits



4.1.2 Balancing datasets

In addressing the issue of imbalanced data classes, studies revealed that classifiers trained on imbalanced datasets tended to exhibit decreased recall for minority classes [71]. The challenge of class imbalance might have involved strategies like increasing minority class examples through oversampling or reducing majority class examples through undersampling. Widely used techniques, such as SMOTE, generated synthetic minority class examples [13].

Augmentation proved effective in solving class imbalances, as demonstrated by the authors of [12]. The dataset initially had an imbalance, with 2357 non-sarcastic tweets and 745 sarcastic ones. After augmentation using noising techniques, the number of sarcastic tweets became equal to that of non-sarcastic tweets. Another approach, presented in [65], offered a DA framework based on statistical metrics and word similarity generation to address imbalanced data. Furthermore, in [90], oversampling and undersampling techniques were employed to balance an imbalanced tweet dataset, originally comprising 10,380 non-sarcastic tweets and 2168 sarcastic tweets, resulting in an equal representation of both classes following augmentation.

Addressing imbalances in biased datasets across multiple languages, including Arabic, the authors of [73] tackled the issue where check-worthy Arabic tweets initially constituted only 22% of the Arabic data, and they addressed this discrepancy by generating 2748 additional check-worthy Arabic samples through augmentation. [83] applied contextual embedding to balance datasets initially featuring only 20% offensive and 5% hateful samples. Following augmentation, the dataset achieved a more balanced distribution, comprising 265,413 offensive samples and 10,489 hateful samples. In [71], addressing a highly imbalanced training set with 19% offensive and 81% inoffensive content involved utilizing word shuffling, replication, and merging external data.

Before augmentation, the dataset in [92] had an imbalance: 36,082 neutral samples, 8533 positive samples, and 8674 negative samples. After augmentation, each category reached 36,082 samples through random oversampling. [57] employed back-translation to increase the representation of the minority class and achieve balance. [44] followed paraphrasing, resampling, and secondary methods for class and label balance. In [28], noising techniques were implemented to address the issue of imbalance where the sarcastic class was more than two times smaller than the non-sarcastic class.

For code-switched data, the authors of [91] duplicated negative and neutral samples in minority classes. [64] balanced the data for a model trained on different languages by undersampling, making the check-worthy claim ratio 30% of the train set. In [21], back-translation was utilized to balance multigenre language data for a check worthiness task. [39] achieved balance in Arabic dialect datasets through paraphrasing, noising, and merging external data techniques, resulting in the dataset expanding from 8437 to 24,104 instances after augmentation. [84] employed downsampling, removing 32% of non-sarcastic tweets for balance.

4.1.3 Decreasing the amount of manually annotated data

DA can decrease the amount of manual work yet obtain close results from a largely labeled dataset. In [17], authors achieved strong performance with just 10% labeled data, closely matching results obtained with a fully labeled dataset. The model attained an F-score of 83.17% with 10% labeled samples, compared to 87.06% when all data were labeled. In [7], SMOTE augmentation significantly reduced the number of training samples needed for both active and passive learning. Before SMOTE, active learning achieved 99% accuracy using 31.9% of training samples, while passive learning required 100% of training samples to reach the same accuracy. Using SMOTE, active learning reached 99% accuracy with only 17.8% of training samples, while passive learning required 86.6% of samples to reach the same accuracy, showcasing the substantial impact of SMOTE on minimizing training samples.

4.2 Tasks applied DA to Arabic

In this subsection, we discuss some tasks that employed augmentation to train models. Figure 5 shows tasks applied DA to Arabic.

- *Sarcasm detection* Requires careful discernment of sarcastic elements within a given text. The authors in [12] used noising augmentation to address the imbalance between sarcastic and non-sarcastic tweets, while [80] employed sampling and resampling methods. In [44], paraphrasing, resampling, and secondary techniques were adopted for the same purpose. Other approaches included resampling [90], random swapping, and deletion noising [28] to overcome data imbalances. Additionally, back-translation was used in [62] to enhance a multilabel model, and semantic augmentation was experimented with in a few/zero-shot learning model for sarcasm detection [54].
- *Social spam detection* To distinguish between spam and non-spam categories, the authors of [41] classified tweets and expanded the dataset by incorporating synonyms through AraVec word embedding substitution.

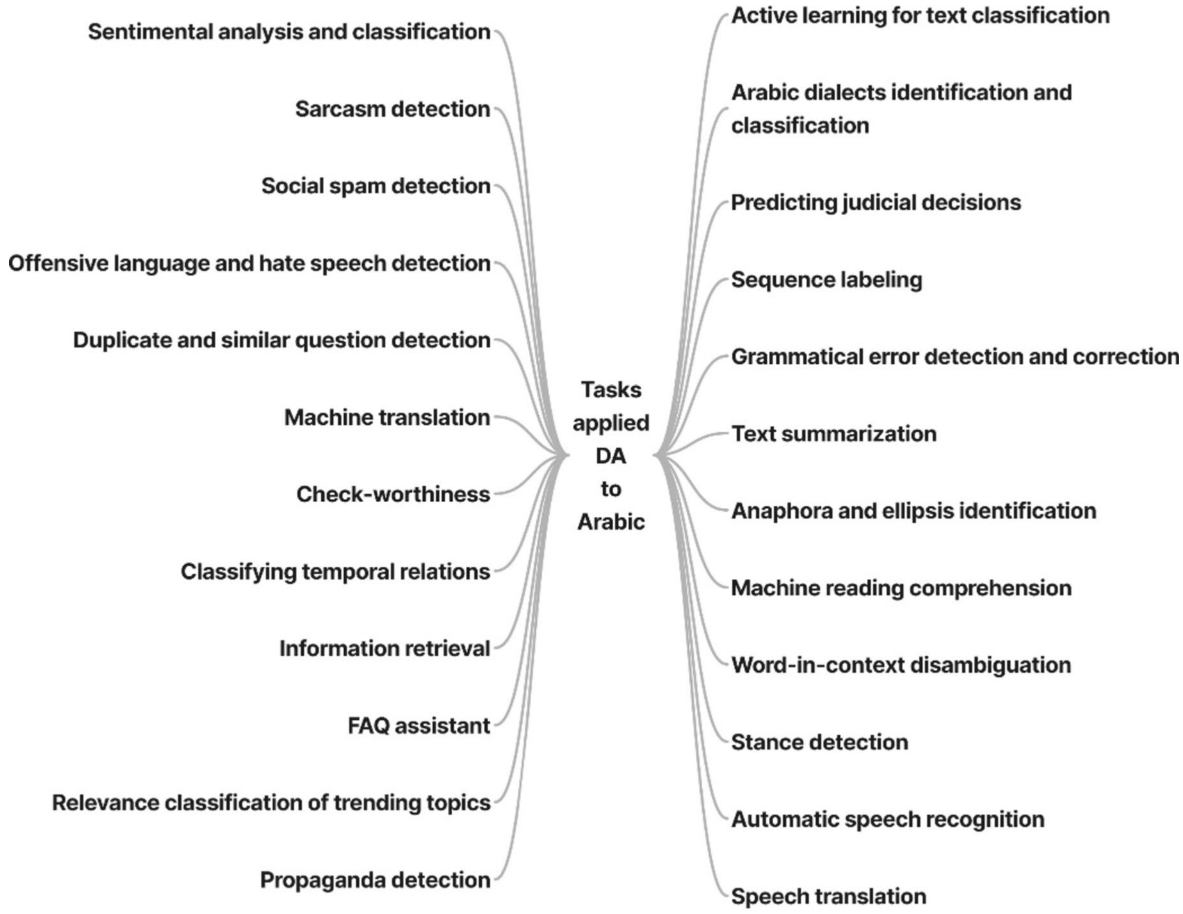

Tasks applied DA to Arabic
Sentimental analysis and classification
Sarcasm detection
Social spam detection
Offensive language and hate speech detection
Duplicate and similar question detection
Machine translation
Check-worthiness
Classifying temporal relations
Information retrieval
FAQ assistant
Relevance classification of trending topics
Propaganda detection
Active learning for text classification
Arabic dialects identification and classification
Predicting judicial decisions
Sequence labeling
Grammatical error detection and correction
Text summarization
Anaphora and ellipsis identification
Machine reading comprehension
Word-in-context disambiguation
Stance detection
Automatic speech recognition
Speech translation

Fig. 5 Tasks applied DA to Arabic

- *Propaganda detection* The authors in [50] employed word stemming to derive the root forms in the context of an Arabic propaganda detection task. In a similar task, [69] generated synthetic sequences by leveraging subparts.
- *Offensive language and hate speech detection* To balance offensive and hateful content datasets, [83] employed a multistep process to augment their dataset. The authors in [78] expanded the profanity dataset through morphological changes and appending extra data. In [32], a cross-lingual augmentation approach was implemented across five different languages, including Arabic, by translating each training sample into three other languages. In [43], authors utilized a synonym dictionary and AraVec embeddings to enhance Arabic data within a multilingual offensive tweet identification system. Addressing a highly imbalanced training set, [71] implemented techniques such as word shuffling, replication, and appending external data. The authors of [30] employed a semi-supervised self-learning technique to augment datasets for hate speech detection. Additionally, [56] utilized back-translation and synonym replacement to augment data for a hate speech detection task.
- *Sequence labeling* NER, a sub-task of information extraction involving the recognition and classification of named entities in unstructured text, was addressed by the authors of [5] through the application of techniques such as word embedding substitution (a modification of Easy EDA) and back-translation. In addition to NER, part of speech (POS) tagging was addressed by [18] in a sequence labeling system that combined self-training augmentation with zero and few-shot learning models.
- *Duplicate and similar question detection* Identifying duplicated questions entails approaching the task as paraphrase identification, framing all sentences as questions, as demonstrated in [74], where augmentation strategies, including the utilization of symmetry, reflexivity, and transitivity relations, were applied. Likewise, [55] utilized these rules to predict the semantic similarity of pairs of Arabic questions. Furthermore, in [75], these relations were employed to enhance a semantic question similarity model.

- *Grammatical error detection and correction* Grammatical error correction involves automatically detecting and correcting monolingual grammatical, spelling, punctuation, and word-positioning errors, such as in [26], where the authors applied seven different augmentation techniques. [79] augmented neural network models by duplicating and fixing sentences that matched a set of rules they defined.
- *Machine Translation (MT)* [15, 49] implemented a NMT architecture to translate between Tunisian dialect (TD) and MSA. The authors of [15] considered MSA as a bridge or a pivot from dialects to foreign languages, and they augmented the model with paraphrasing, noising, and sampling techniques. [49] used paraphrasing and secondary augmentation techniques. Another MT model that translates TD into MSA was introduced in [51], where authors augmented data based on segmentation and back-translation. In [60], the authors developed a sampling augmentation technique and compared its affection with back-translation. [81] utilized pre-trained language models to generate text. They then implemented a process involving back-translation and a filtration step to enhance the performance of a MT model that operates with both English and Arabic. The authors of [22] trained a transformer model on a MT system that operated on code-switched data. They studied various augmentation techniques, including paraphrasing, noising, and sampling.
- *Text summarization* Automatic text summarization involves generating a brief and coherent summary while retaining essential information and the overall meaning, as in [29], where a transformer-based abstract text summarizer model was implemented and augmented through the generation of synthetic summaries in the public health domain.
- *Check-worthiness* The check-worthiness model prioritizes sentences based on their need for checking to evaluate claims [59]. This ranking challenge requires the model to order sentences, with the top sentences expected to contain verifiable claims. In their work [59], the authors have introduced an innovative model incorporating features for few-shot learning and DA. In a different approach, [57] employed back-translation augmentation in a fact-checker and subsequently utilized contextual word embedding substitution augmentation in [73]. This involved classifying claims based on their check-worthiness and ranking them in priority order for the fact-checker. Furthermore, [64] employed machine translation and undersampling to augment a check-worthy claims detection model, while [21] utilized back-translation to enhance multigenre language data for a check-worthiness task. In a similar approach, [33] enhanced a multigenre, multilingual check-worthiness model through the use of paraphrasing and translation across various languages, including Arabic.
- *Anaphora and ellipsis identification* AZP, explained by [47], are absent yet significant pronouns in specific languages, functioning as implicit references requiring contextual understanding. In their study, they implemented several distinct paraphrasing augmentation techniques to create and identify AZPs. Meanwhile, in [85], researchers focused on identifying and resolving pronominal anaphors and ellipses. They employed self-training with SVM and iteratively expanded the labeled dataset with newly annotated data.
- *Classifying temporal relations* To make explicit what was implicit, authors in [77] utilized rules for augmenting temporal relations such as “After,” “Before,” and “During” between entities.
- *Machine Reading Comprehension (MRC)* Retrieves the proper response span from a text paragraph and an inquiry, such as in the [88] study that introduced a cross-lingual method that created language-specific groups of passages and questions for training models.
- *Information retrieval* The authors of [82] used query expansion through the pseudo-relevance feedback method and deep averaging networks to facilitate Arabic information retrieval.
- *Word-in-context disambiguation* To determine whether a word shared by two sentences had the same meaning in both contexts, the authors of [52] used sentence pairing to augment a multilingual and cross-lingual system, while [67] treated disambiguation as question answering and classification and implemented a span prediction model augmented by swapping sentences.
- *FAQ assistant* To answer users’ queries [63] machine translation was used to augment an innovative technique where a single model was jointly trained on up to 15 languages, including Arabic.
- *Stance detection* To explore users’ opinions on women’s empowerment, [62] utilized back-translation to enhance a multilabel model encompassing both positive and negative stances.
- *Relevance classification of trending topics* To determine the relevance of Arabic tweets to specific trending topics on social media, [42] used word embeddings to generate additional training data.
- *Automatic Speech Recognition (ASR)* The ASR baseline system was one of several baselines implemented by [22] and trained on Egyptian Arabic data that was

augmented using techniques such as dictionary-based replacement and predictive model replacement.

- *Speech Translation (ST)* As previously noted, [22] studied several replacement augmentation methods and used them on several systems, one of which was a cascaded ST system that made use of ASR and MT models.

In conclusion, the application of data augmentation techniques has demonstrated substantial benefits in various NLP tasks related to Arabic language processing. These methods have effectively expanded datasets, balanced data classes, and reduced the necessity for extensive manual annotation, leading to significant improvements in model performance across diverse tasks. However, while these advancements mark significant progress, they also introduce a set of challenges that must be addressed to fully harness the potential of data augmentation in Arabic NLP. The subsequent section delves into these challenges, highlighting the obstacles and offering insights into future directions for overcoming them.

5 Analysis, challenges, and future directions

Following the discussion of augmentation benefits and practical applications in the previous section, it is crucial to address the inherent challenges of implementing these techniques. These challenges include linguistic issues, such as the lack of proper synonyms and handling of diacritics, as well as technical limitations like insufficient data quantity and context preservation problems. By understanding these obstacles, we can better navigate the complexities of Arabic NLP and explore innovative solutions to improve data augmentation strategies. In this section, we first analyze our findings and emphasize the significance of data augmentation. We then discuss the challenges associated with Arabic text augmentation and outline future directions for advancing more effective approaches.

5.1 Comparison between DA types

In the previous chapters, we explored various DA techniques. In Table 1, we compare these techniques to help readers better understand their differences and make more informed decisions when selecting a DA method.

5.2 How metrics and augmentation techniques influence each other

The relationship between metrics and augmentation techniques is complex and reveals a research gap, as some studies do not clearly explain why specific metrics are chosen with certain augmentation methods for different NLP tasks, as noted in 0. Given the importance of this area, we offer initial insights and examples that reveal common patterns and overlaps across studies. These examples help set a foundation for understanding how metrics can guide the choice of augmentation techniques, offering a starting point for further research.

5.2.1 Class imbalance and data expansion

Metrics like F1-score, precision, recall, and accuracy were essential in guiding the selection of augmentation methods tailored to manage class imbalance, especially in domains where minority classes (such as sarcasm or rare sentiment categories) were underrepresented. For example, the oversampling technique improved minority class detection in sarcasm detection [90] and in identifying code-switched sentiment tasks [91], achieving better F1-score in classes by rebalancing datasets with duplicated samples. Techniques such as SMOTE [7] and synonym replacement [27] [41] significantly improved the model's ability to detect minority classes and expanded dataset size, reducing the need for manual annotation. These methods led to marked enhancements in accuracy, macro F1, and precision scores, particularly for imbalanced and small datasets. In sentiment analysis and sarcasm detection, models enhanced by balanced class training and targeted augmentations—such as category duplication, text generation based on statistical metrics, TF-IDF vectorizers [80] [65], and similarity-based techniques—demonstrate improved precision and recall, benefiting minority class performance. Techniques including transformer-based text generation, cosine and Jaccard similarity evaluations, and targeted keyword replacements help mitigate the challenges of class imbalance, thereby strengthening model robustness and reliability in imbalanced classification tasks [25].

5.2.2 Text quality and semantic fidelity

Metrics evaluating semantic quality, such as, ROUGE score, BLEU score, mean average precision (MAP), perplexity, and similarity measures (e.g., Euclidean, cosine, Jaccard), enriched the use of augmentation methods that maintained semantic coherence. In machine translation, techniques such as back-translation and contextual embeddings have proven effective at maintaining semantic fidelity, contributing to an improved BLEU score [15]. Additionally, in text summarization, the use of synthetic data alongside real data has shown improvements in semantic coherence, with models like GPT-2 enhancing the quality of summaries in terms of ROUGE scores [29]. Perplexity assesses how accurately a model can predict upcoming words in a sequence, with lower scores indicating higher predictive

Table 1 Comparison between DA types

Technique	Description	Strengths	Weaknesses	Resource requirements
Thesauruses and dictionaries	Substituting words or phrases based on entries from thesaurus and dictionaries (e.g., synonyms)	Easy to implement; Enhances lexical diversity	Limited by thesauruses and dictionaries quality; Context mismatches	Low; Requires comprehensive thesauruses and dictionaries
Semantic embeddings	Using pre-trained word embeddings to capture semantic relationships	Maintains semantic integrity; Effective for complex tasks	Struggles with words with multiple meanings; Context-specific complexities	Moderate; Needs pre-trained embeddings
Language models (LM)	Predicting contextually appropriate words or phrases using models like BERT	Produces fluent, coherent, and contextually accurate text	High computational demands; Dependent on model quality	High; Requires high-quality pre-trained models
Paraphrasing rules	Using pre-defined rules to generate paraphrases of text	Generates diverse expressions; Easy to implement	Limited by rule comprehensiveness; Can be less natural	Low; Requires well-defined paraphrasing rules
Machine translation	Translating text into another language and back	Generates diverse paraphrases; Effective for diverse expressions	Quality depends on the translation tool's accuracy	Moderate; Needs translation tools
Model generation (MG)	Generating text using trained models like GPT	Produces high-quality, diverse text	Very high computational demands; Dependent on model training	Very High; Requires advanced models and training
Swapping	Swapping words or phrases within a sentence	Simple to implement; Increases text variability	Can disrupt grammatical structure; May alter meaning	Low; No special resources needed
Deletion	Deleting words or phrases from text	Simplifies text; Reduces noise	May lose important information; Can alter meaning	Low; No special resources needed
Insertion	Inserting additional words or phrases into text	Adds diversity; Can introduce new context	Can disrupt flow; Risk of introducing irrelevant information	Low; No special resources needed
Substitution	Replacing words or phrases with alternatives	Increases diversity; Maintains semantic meaning if done well	Risk of context mismatch; Requires good alternatives	Low; Needs a list of suitable substitutions
Sampling rules	Applying specific rules to sample data subsets	Enhances diversity; Can target specific aspects of data	May introduce bias; Requires well-defined rules	Low to moderate; Depends on rule complexity
Non-pre-trained models	Using models trained from scratch for augmentation	Customizable to specific tasks; No dependency on pre-trained data	Requires large datasets; High computational cost	High; Requires extensive training data and computation
Pre-trained models	Using models pre-trained on large datasets for augmentation	High-quality text; Reduces training time and resources	Limited by pre-training data; May not fit all tasks	Moderate to high; Requires high-quality pre-trained models
Semi-supervised learning (SSL) and Self-training	Combining labeled and unlabeled data for training	Improves model performance with less labeled data	Requires careful tuning; Can be complex to implement	Moderate to high; Needs a mix of labeled and unlabeled data
Interpolation	Combining data from multiple sources to create new data	Enhances diversity and robustness	Requires careful blending; Risk of creating unrealistic data	Moderate; Needs multiple data sources
Oversampling	Repeatedly sampling minority class data to balance datasets	Addresses class imbalance; Simple to implement	Can lead to overfitting; Redundant information	Low; No special resources needed
Undersampling	Reducing majority class data to balance datasets	Addresses class imbalance; Reduces dataset size	Risk of losing valuable information; May underrepresent majority class	Low; No special resources needed

Table 1 (continued)

Technique	Description	Strengths	Weaknesses	Resource requirements
Merging datasets	Combining multiple datasets to enhance size and diversity	Increases data size and diversity; Can cover more variations	Requires data compatibility; Risk of inconsistencies	Moderate; Needs multiple datasets
Translating similar datasets from other languages	Translating datasets in other languages into Arabic	Increases data availability; Leverages multilingual resources	Quality depends on translation accuracy;	Moderate; Needs translation models
Mixing different dialects	Combining data from different Arabic dialects	Enhances robustness to dialectal variations	Requires careful handling of dialect differences; Can introduce noise	Moderate; Needs diverse dialectal datasets

accuracy and fluency. For example, a 34% reduction in perplexity in Arabic-English code-switched data highlighted the effectiveness of lexical replacements in generating coherent outputs [22]. MAP is commonly applied in ranking tasks and other areas to measure system accuracy in retrieving relevant items. This metric has been applied to various domains, including check-worthiness [59], claim retrieval [73], and information retrieval query expansion [82]. Euclidean, cosine, and Jaccard distances play a crucial role in guiding the selection of augmentation techniques by evaluating how closely generated text aligns with the original content. For instance, techniques that demonstrate lower Euclidean distances or higher cosine and Jaccard similarities are favored, as they indicate better semantic coherence. Emphasizing evaluation metrics ensures that methods like AraGPT-2-based and GAN-based text generation and other data augmentation techniques effectively enhance fluency and alignment in tasks such as sentiment and sarcasm detection, leading to improved model performance [3] [25].

5.2.3 Data novelty and diversity

Novelty and diversity metrics (see Sect. 3.5.1) are pivotal in enhancing dataset variety and model robustness across Arabic dialects. [3] employed SentiGAN model to introduce novel linguistic variations by generating unique sentences in Arabic dialects, thereby expanding dialectal coverage and supporting tasks like dialect classification by exposing models to a wider range of language patterns. Similarly, [25] utilized AraGPT-2 with similarity measures like Euclidean and cosine distances to maintain coherence while increasing lexical diversity. This approach enriched datasets with semantically varied expressions, benefiting sentiment classification by providing models with a broader spectrum of sentences. Together, these strategies highlight how novelty and diversity metrics guide the selection of augmentation techniques, showing that increased linguistic diversity and coherence can be achieved through carefully chosen methods, enhancing the effectiveness of augmentation metrics.

5.2.4 Error reduction and convergence stability

Metrics such as cross-entropy loss and error rate are crucial for evaluating error reduction and model convergence stability in augmented models. Data augmentation techniques, like back-translation, have demonstrated effectiveness in improving dataset balance and generalization. For example, [55] employed data augmentation techniques—including back-translation, transitive, symmetric, and reflexive rules—to expand and diversify the training dataset. This approach reduced the error rate from the previous best of 4.08% to 3.12%, demonstrating a significant improvement in predicting semantic similarity for Arabic questions and enhancing the model's generalization capability. In addition, [22] emphasized the effectiveness of lexical replacements in generating more coherent outputs in code-switched Arabic-English data. By employing lexical replacements, the study achieved a notable 34% reduction in perplexity during model evaluation, highlighting how these augmentative techniques can address challenges posed by linguistic variations and improve overall model accuracy. In [74], data augmentation for Arabic question-answering systems was conducted by generating duplicate and non-duplicate question pairs using symmetry, reflexivity, and transitivity relations. This expanded dataset helped the model learn diverse question patterns, improving generalization. The model employed cross-entropy loss as part of its training objective, with structured attention and contextual embeddings enabling the model to focus on critical parts of each question, thereby enhancing accuracy in duplicate question detection. In [44], lower cross-entropy loss scores served as an indicator of improved learning stability in the sarcasm detection model. By applying data augmentation techniques using word embeddings and instance repetition, the model could better distinguish between sarcastic and non-sarcastic instances, particularly in imbalanced datasets.

5.3 Supplementary research materials

5.3.1 Popular datasets

As discussed in the earlier sections, data augmentation (DA) can be applied to a wide range of NLP tasks, resulting in no single dataset being universally applicable. However, in Table 2 we have compiled a selection of popular datasets used in the research papers reviewed in this survey. This collection is intended to help readers gain a more comprehensive understanding of the field.

5.3.2 Code repositories and resources

Examining code and scripts with full implementation is essential for gaining a technical understanding that complements theoretical knowledge. Unfortunately, many published papers do not provide their experimental code publicly. Therefore, in Table 3, we have compiled a few available code repositories.

5.4 Augmentation challenges

Data augmentation in Arabic text presents several unique challenges that impact the effectiveness and accuracy of NLP models. These challenges can be categorized into specific issues related to Arabic text and general text augmentation difficulties.

5.4.1 Arabic text augmentation challenges

One significant challenge in Arabic text augmentation is the *lack of proper Arabic synonyms*. When compared to other WordNets, Arabic WordNet performs poorly. To be more precise, it only covers 9.7% of the Arabic lexicon, whereas the English WordNet covers 67.5% of the English lexicon. Furthermore, the English WordNet makes use of seven semantic relations, while the Arabic WordNet only uses hyponymy, synonymy, and equivalency to link synsets together [27]. Another issue is the *lack of tools that can handle Arabic text effectively*. Lack of Arabic text representation tools that can handle Arabic text with diacritics may lead researchers to remove diacritics in preprocessing steps. This causes changes in the syntax and semantics of text, leading to wrong labeling [17]. Lack of Tunisian dialect segmentation tools and standard syntax rules led authors [51] to segment sentences manually based on Arabic stop words and punctuation marks, resulting in sentence alignment loss. To remedy this issue, they used back-translation. [33] adopted paraphrasing and translating between English, Spanish, and Arabic using GPT-3.5 provided by OpenAI. They noted trivial improvement in Arabic, suggesting GPT-3.5 struggles with Arabic due to its complex morphology and right-to-left script, impacting synthetic text generation compared to English and Spanish.

Heterogeneity in Arabic and Latin scripts also presents a challenge. Researchers [49] researchers, there was a decrease in BLEU score when combining Latin script with the Tunisian dialect. This indicates that incorporating diverse script types into the corpus increases DA ambiguity. The authors suggested converting Latin-script comments to Arabic before employing the NMT model. Additionally, *inadequate data quantity during integration with other Arabic dialects* led to a marginal decrease in BLEU scores when Moroccan and Algerian dialect sentences were merged with Tunisian dialect data [49]. The insufficient number of sentences from these dialects, which face similar challenges to the Tunisian dialect (such as the absence of spelling standards and resources), might be responsible for the observed decline.

5.4.2 General text augmentation challenges

Deviation from the original context is a notable challenge with data augmentation. Although DA can increase NLP tasks' performance, it may cause some issues if applied without caution. For example, if a sentence before augmentation was "I like something" with a positive label, then after augmentation it became "I don't like something" with a negative label, then this is a contradiction of the original context's meaning [37]. It can also generate sentences with very different meanings that are not related to the original context's meaning if not used correctly, as in a few cases [56]. *Limitation on the number of possible generated sentences with different dialects* can also be problematic. The author of [3] was limited by the number of sentences in the dataset due to using independent generators and discriminators in the modified SentiGAN model to avoid the possibility of mixing common features between the different Arabic dialects. *Shuffling word augmentation* may not suit some models. The authors of [70] combined CNN, LSTM, and RCNN sentimental data classification models with random shuffling. Data augmentation improved their LSTM model's accuracy by 8.3%, but CNN and RCNN models had no effect. Random shuffling augmentation contributed to this consequence. CNN and RCNN focus on word features, not sentence order, unlike LSTM. Thus, these models need more real data to improve results.

Some techniques did not gain a significant improvement after augmentation. Augmenting datasets based on Word2Vec for embeddings and AraBERT for classification [24] did not yield improved results, prompting the authors to suggest exploring alternative models to enhance

Table 2 Datasets from the literature

Dataset	Description
ArSarcasm [104]	The dataset comprises 10,547 tweets, with 16% identified as sarcastic. Along with sarcasm, the data is also annotated for sentiment and dialects https://github.com/iabufarha/ArSarcasm
Arabic sentiment tweets dataset (ASTD) [105]	This dataset for Arabic social sentiment analysis, collected from Twitter, includes approximately 10,000 tweets. The tweets are categorized into four sentiment types: objective, subjective positive, subjective negative, and subjective mixed https://github.com/mahmoudnabil/ASTD
Multi Arabic dialect applications and resources (MADAR) [106]	This dataset features a comprehensive parallel corpus covering 25 Arabic city dialects related to travel, alongside a lexicon with 1,045 concepts, each represented by an average of 45 terms from 25 cities https://sites.google.com/nyu.edu/madar/
The United nations parallel Corpus v1.0 (UNCorpus) [106]	This dataset consists of a parallel corpus created from United Nations documents, encompassing translations for the six official UN languages: Arabic, Chinese, English, French, Russian, and Spanish, from 1990 to 2014 https://huggingface.co/datasets/Helsinki-NLP/un_pc
Corpus on Arabic Egyptian tweets [108]	This corpus contains 40,000 tweets, evenly divided between 20,000 positive and 20,000 negative tweets. The tweets cover a variety of general topics commonly discussed on Twitter https://doi.org/10.7910/DVN/LBXV9O
Arabic speech-act and sentiment (ArSAS) [109]	A collection of 21,000 Arabic tweets in dialectal Arabic was compiled for sentiment analysis. The dataset spans various topics and is categorized into four sentiment classes: positive, negative, neutral, and mixed https://homepages.inf.ed.ac.uk/wmagdy/resources.htm
ANERsys [110]	An Arabic named entity recognition system based on maximum entropy https://camel.abudhabi.nyu.edu/anercorp/
NADA [111]	NADA is a newly created Arabic dataset for text categorization, covering 10 categories from diverse fields such as social sciences (e.g., economics, law), religious studies (e.g., Islam), applied sciences (e.g., healthcare), pure sciences (e.g., technology), literature, and arts (e.g., sports). It combines data from two existing corpora, OSAC and DAA https://huggingface.co/datasets/arbml/NADA
Large-SCale Arabic book reviews dataset (LABR) [112]	This dataset comprises over 63,000 Arabic book reviews, making it the most extensive dataset for Arabic sentiment analysis so far. The reviews were gathered from www.goodreads.com in March 2013 and include the review ID, user ID, book ID, rating (on a scale of 1 to 5), and the review text https://github.com/mohamedadaly/LABR
propaganda-detection [113]	The dataset contains Arabic tweets along with an annotation of 21 different propaganda techniques applied within the texts https://gitlab.com/araieval/propaganda-detection

Table 3 Papers and repos

Paper	Repo
[44]	https://github.com/mosab-shaheen/iSarcasm-SemEval-2022-Task-6
[26]	https://github.com/aimanmutasem/GECDA
[48]	https://github.com/Quant-NLP/SPDAug-ABSA
[50]	https://colab.research.google.com/drive/19zAYftPaXcNDZ6N6Pyj8K8BJXtkEgglx
[59]	https://github.com/amanisa/AraCWA
[18]	https://github.com/mukhal/low-resource-seq-labeling
[78]	https://github.com/AMR-KELEG/offenseval-2020-ASU_OPTO
[79]	https://gist.github.com/iwan-rg/4e7f522a53e664607c2a3e664f4c076a
[75]	https://github.com/AliOsm/semantic-question-similarity
[67]	https://github.com/davletov-aa/mcl-wic
[81]	https://github.com/ymoslem/MT-LM

predictive capabilities. Additionally, [71] found replicating words in the minority class ineffective for achieving balance. Some suggested solutions included using WordNet for meaningful augmentation and employing SMOTE for oversampling offensive samples, along with random undersampling for the majority class to enhance model flexibility. In [4], the authors suggested that the observed trivial performance improvement might be attributed to the generative nature of the MNB classifier and its assumption of independence between features. *Multilingual challenges* while training a model on various languages caused challenges in [64], where augmentation through translation between languages yielded low scores, indicating claim check-worthiness effectiveness variations across nations. However, undersampling was found to give better results in the Arabic language within the same study.

Noisy AraBERT masked language model predictions was another noted issue. AraBERT-based DA performed poorly compared to FastText (word embedding), contrary to the authors' expectations [45]. This issue resulted from the prediction of words with diverse meanings, introducing noise into the results. In [74], *duplication and syntax similarity in augmented data* were discussed. The authors employed symmetry, reflexivity, and transitivity relations to generate questions as an augmentation technique. Nevertheless, subsequent analysis by [114] indicated that this augmentation approach did not significantly enhance performance. The lack of improvement may be attributed to the possibility that the augmented data duplicated existing examples or introduced questions with similar syntax, resulting in a similar performance achieved by another model without augmentation. *Insufficient training data limits fine-tuning* was faced while using GPT-3 in [80] as a text-augmenting tool, generating sarcastic phrases through prompts. The technique ultimately failed because the results, which may have been caused by a lack of training data for GPT-3 fine-tuning, produced repetitive sentences with little structural or stylistic variation.

In [90], oversampling and undersampling produced a fake high accuracy of 0.93. The authors suggested that *augmenting the dataset before splitting it* may have caused this misleading result. They advised avoiding augmenting data before splitting. Similarly, *how much of the original text should be replaced* was an important question. In data augmentation studies, a common challenge is deciding how much data to replace to boost diversity without losing the original meaning. There's no specific formula for this challenge. In tackling it, researchers have proposed various approaches. For instance, in [45], 30% of the words in each review were randomly replaced with synonyms. Moreover, they retained at least 50% of words in short sentences, particularly those with less than five words. *Choosing a suitable metric* was another area that requires careful consideration. Finding the suitable metric for assessing augmented sentence similarities is a challenge, and [37] was one of the trials conducted to explore potential solutions. In their experiments, they suggested that the cosine metric, by emphasizing sentence direction, provides consistent scores irrespective of sentence size. Conversely, the Euclidean metric, influenced by sentence size, may yield dissimilar similarity results for small and large sentences. Additionally, [37] proposed that Jaccard and BLEU similarities offer insights into the novelty and diversity between sentences. Lastly, *masking challenges (balancing semantics and morphology)* were discussed in [47]. The authors highlighted a concern related to masking models such as BERT. The process of augmenting data by masking and predicting similar tokens should keep a balance between semantics and morphology. There was a risk that the model might predict tokens that were semantically similar but differed morphologically. For instance, replacing "teacher" with "teachers" disrupted training accuracy in certain models.

5.5 Future directions of augmentation

DA creates synthetic training data, while active learning identifies informative, unlabeled samples for labeling. Few-shot learning addresses tasks without sufficient labeled data by leveraging existing information. Some studies have implemented augmentation techniques across different languages, enhancing Arabic data or testing DA methods. This subsection will provide a concise overview of these aspects. Figure 6 illustrates the evolving application of DA to Arabic literature.

5.5.1 Integrating augmentation with N-Shot learning

In the context of N-Shot model training, a support set comprising only N samples is supplied to the model. Few-shot learning provides a straightforward method for handling tasks characterized by a lack of labeled data,

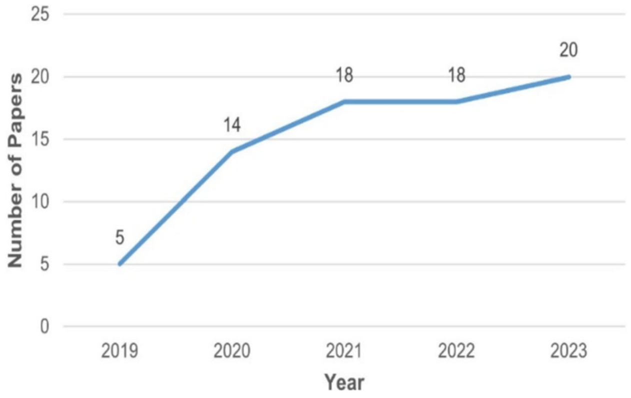


Fig. 6 Application of DA to Arabic over the years

leveraging existing information [54]. Despite the highly restricted training data, few-shot learning models encounter the challenge of achieving performance levels comparable to those of traditional machine learning models processing large datasets. To address this data scarcity issue, DA proves to be a valuable solution [14].

In the field of few/zero-shot learning for text classification, the authors of [54] implemented label-semantic augmentation within a meta-learner framework. Additionally, [59] conducted a cross-topic check-worthy claim detection task, keeping a few instances for use in the few-shot settings. Subsequently, DA, utilizing paraphrasing and sampling techniques, was employed to generate new synthetic samples. Furthermore, [18] integrated zero/few-shot learning and self-training for sequence labeling applied to NER and POS tagging tasks.

5.5.2 Integrating augmentation with active learning

While data augmentation can waste computational resources by randomly generating non-informative samples, [6] suggested using active learning to save computational resources and time by selecting the most informative unlabeled training data subset. The number of training samples required for both active and passive learning was greatly decreased [7] by SMOTE augmentation. Prior to SMOTE, 100% of training samples were needed for passive learning to reach the same accuracy as active learning, which only needed 31.9% of training samples to reach 99% accuracy. SMOTE significantly reduced the number of training samples needed for active learning, which was able to achieve 99% accuracy with just 17.8% of training samples and 86.6% for passive learning.

5.5.3 Cross-lingual, multigenre, and multilingual models

Augmentation approaches were applied to different languages, including Arabic, to enrich low-resource Arabic data or to test the proposed augmentation strategy on different languages. An innovative approach by training a single model on up to 15 languages using multilingual pretrained language models was introduced in [63]. When training data were lacking for a language, a translate-train paradigm was used with machine translations. Translated English data were used in Arabic. Instead of training different models for each language, training one model saved time and effort and improved performance.

Cross-lingual augmentation approaches were applied to five languages [32] by translating each training sample into three other languages. A cross-lingual MRC model was introduced in [88] that created language-specific groups of passages and questions for training separate MRC models, facilitating multilingual comprehension through multiteacher distillation models. Sentence pairing was used to augment a system of five multilingual and four cross-lingual sub-tracks [52].

An offensive multilingual tweet identification system was implemented by researchers [43]. However, only the Arabic language was augmented using a synonym dictionary and AraVec embeddings. Sentence swapping was employed to augment a multilingual model [67]. In [73], five different languages, including Arabic, were augmented using a contextual embedding substitution model to classify check-worthy and rank in priority.

Researchers in [48] performed Arabic aspect-based sentiment analysis. Semantics-preserving data augmentation replaced unimportant tokens for better understanding. They translated auxiliary sentences into seven languages to test the proposed augmentation strategy. Back-translation was utilized in [21] to augment multigenre language data from a check worthiness model that concerned English, Arabic, and Spanish languages. In [33], a multigenre, multilingual model was utilized for paraphrasing and translation across English, Spanish, and Arabic, and the obtained results were compared.

5.5.4 Large language models

Large language models (LLMs) are sophisticated AI systems engineered to comprehend and generate human language. These models rely on deep learning techniques and are trained on extensive datasets, enabling them to perform various NLP tasks, including text generation, summarization, and translation. The architecture of LLMs is predominantly based on transformers, which allow for efficient text processing and generation by using self-attention mechanisms that understand the relationships between words in context. Notable examples of LLMs include OpenAI's GPT series (GPT-3, GPT-4), Google's Bard and Gemini, and Meta's Llama. Hugging Face is a platform that offers easy access to a wide array of pre-trained LLMs and other machine learning models, serving as a key resource for developers and researchers to share, discover, and utilize these models. LLMs hold significant promise in data augmentation, particularly for the Arabic language, where few studies have applied these methods. For instance, in [29], the authors used the AraBART model from Hugging Face for text summarization as part of DA. Another example is the use of masking models, such as AraBERTV02 available on Hugging Face, where authors in [45] masked four random words in a sentence and predicted new replacement words to augment text. A growing area of interest is the application of prompting models for DA, as seen in [81], where Hugging Face's mGPT was employed to generate in-domain synthetic segments. Prompting encompasses not only text generation but also the exploration of advanced techniques like chain-of-thought and the use of chatbots like ChatGPT, with future ideas extending this approach by proposing methods such as knowledge distillation or merging different LLM models to leverage the strengths of multiple LLMs [115]. While these approaches have yet to be widely applied in Arabic research, they open the door to exciting opportunities for future exploration and innovation in this field.

5.6 Key research gaps

Although significant efforts have been made in Arabic literature to augment Arabic text, several critical research questions remain. This subsection highlights a few of these open research areas to guide Arabic researchers in identifying potential research opportunities.

5.6.1 Evaluating augmentation effectiveness

As discussed in subSect. 3.5, various methods and metrics for assessing augmentation quality have been explored, including novelty and diversity evaluations, human assessments, intrinsic and extrinsic evaluations, and performance metrics such as BLEU score, F1 score, and accuracy. However, our study indicates that there is no universally stable method for evaluating the success of data augmentation across different downstream tasks. Hence, we advise researchers to go further with a comparative analysis of different metrics and evaluation methods.

5.6.2 Scarcity of LLMs for Arabic

Despite the promising future directions involving LLMs, there remains a significant gap in LLMs capable of processing Arabic text. Many models are trained solely on English or multilingual data, while those trained specifically for Arabic are less accessible. Addressing this challenge may involve fine-tuning existing LLMs, combining different LLMs, employing knowledge distillation, and incorporating translation layers into pre-trained models. These approaches warrant further research.

5.6.3 Diffusion models in text augmentation

Diffusion models, a generative AI technique that adds noise to data and then reconstructs it, have shown considerable success in computer vision. In text augmentation, these models generate diverse and realistic text variations by iteratively adding and removing noise. However, their effectiveness in NLP remains under investigation. Further research is needed to determine whether diffusion models can achieve similar success in text augmentation as they have in computer vision or if their application in NLP will face limitations.

6 Conclusion

Arabic is a data-scarce language with a complex linguistic structure. DA is one strategy used to get around the lack of data. Certain Arabic works employed techniques like diversity-based techniques, resampling techniques, and secondary techniques. DA is the application of methods to artificially generate data without the need to add new data directly from external sources. However, some researchers added data directly, and we classified their approach as secondary techniques. In addition to helping to balance and grow datasets, DA typically reduces the number of manually labeled samples needed for model training. Some works, however, failed to meet the desired performance increase because of a number of issues that we covered in this survey. Additionally, in some works, DA was combined with other approaches like N-Shot learning and active learning. In certain works, various Arabic dialects were combined, and in other works, Arabic data were trained in cross-lingual, multigenre, and multilingual models utilizing data from other languages.

Our investigation for this work discovered certain gaps in the literature in general, along with those in Arabic research specifically. There are still not enough papers that discuss the appropriate steps for measuring DA impact and the quality of the synthesized text, despite the fact that few of them address these topics. Furthermore, a range of downstream NLP tasks have been used to test DA; however, we can observe that the number of papers that addressed each task is biased. For instance, there are more studies on sentiment analysis and sarcasm detection than there are on text summarization and information retrieval.

Funding Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB). This study was funded by Cairo University. Open access agreement for Egypt: All the authors of this paper are researchers at Cairo University, so we should be eligible to publish open access (OA) in Springer Nature’s portfolio of journals with fees covered under the transformative agreement plus fully OA agreement. For more details about the agreement: https://www.springernature.com/gp/open-research/oa-agreements/egypt

Data availability This article is classified as a comprehensive survey/review that studied the literature review. There is not publicly or privately available data. There are no generated data. All papers studied and mentioned in this review have been clearly cited and listed in the References.

Declarations

Conflict of interest All authors certify that they have no conflicts of interest. The authors have no relevant financial or non-financial interests to disclose. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Antoun W, Baly F, Hajj H (2020) AraBERT: Transformer-based Model for Arabic Language Understanding
2. Elkateb S, Black W, Vossen P, et al (2006) Building a WordNet for Arabic. In: Proceedings of the 5th international conference on language resources and evaluation, LREC 2006, pp 29–34
3. Carrasco XA, Elnagar A, Lataifeh M (2021) A generative adversarial network for data augmentation: the case of arabic regional dialects. Procedia Comput Sci 189:92–99. https://doi.org/10.1016/J.PROCS.2021.05.072
4. Talafha B, Fadel A, Al-Ayyoub M, et al (2019) Team JUST at the madar shared Task on arabic fine-grained dialect identification. In: ACL 2019 - 4th Arabic natural language processing workshop, WANLP 2019—Proceedings of the Workshop, pp 285–289
5. Sabty C, Omar I, Wasfalla F et al (2021) Data augmentation techniques on arabic data for named entity recognition. Procedia CIRP 189:292–299. https://doi.org/10.1016/J.PROCS.2021.05.092
6. Tran T, Do TT, Reid I, Carneiro G (2019) Bayesian generative active deep learning. In: 36th international conference on machine learning, ICML 2019 2019-June, pp 10969–10978
7. Al-Tamimi AK, Bani-Isaa E, Al-Alami A (2021) Active Learning for Arabic Text Classification. In: Proceedings of 2nd IEEE international conference on computational intelligence and knowledge economy, ICCIKE 2021 pp 123–126. https://doi.org/10.1109/ICCIKE51210.2021.9410758
8. Chen J, Tam D, Raffel C et al (2023) An empirical survey of data augmentation for limited data learning in NLP. Trans Assoc Comput Linguist 11:191–211. https://doi.org/10.1162/TACL_A_00542
9. Rahma A, Azab SS, Mohammed A (2023) A Comprehensive survey on Arabic sarcasm detection: approaches, challenges and future trends. IEEE Access 11:18261–18280. https://doi.org/10.1109/ACCESS.2023.3247427
10. Park DS, Chan W, Zhang Y, et al (2019) Specaugment: a simple data augmentation method for automatic speech recognition. In: Proceedings of the annual conference of the international speech communication association, INTERSPEECH 2019-September: pp 2613–2617. https://doi.org/10.21437/INTERSPEECH.2019-2680
11. Shorten C, Khoshgoftaar TM (2019) A survey on image data augmentation for deep learning. J Big Data 6:1–48. https://doi.org/10.1186/S40537-019-0197-0/FIGURES/33
12. Kamr AM, Mohamed EH (2022) akaBERT at SemEval-2022 Task 6: An Ensemble Transformer-based Model for Arabic Sarcasm Detection. In: SemEval 2022—16th international workshop on semantic evaluation, proceedings of the workshop, pp 885–890. https://doi.org/10.18653/v1/2022.semeval-1.124
13. Feng SY, Gangal V, Wei J, et al (2021) A survey of data augmentation approaches for NLP. Findings of the association for computational linguistics: ACL-IJCNLP 2021, pp 968–988. https://doi.org/10.18653/v1/2021.findings-acl.84
14. Li B, Hou Y, Che W (2022) Data augmentation approaches in natural language processing: a survey. AI Open 3:71–90. https://doi.org/10.1016/J.AIOPEN.2022.03.001
15. Kchaou S, Boujelbane R, Belguith LH (2022) Hybrid pipeline for building Arabic tunisian dialect-standard Arabic neural machine translation model from scratch. ACM Trans Asian Low-Resour Lang Inf Process. https://doi.org/10.1145/3568674
16. Omran T, Sharef B, Grosan C, Li Y (2023) The Impact of Data Augmentation on Sentiment Analysis of Translated Textual Data. In: 2023 international conference on IT innovation and knowledge discovery, ITIKD 2023. https://doi.org/10.1109/ITIKD56332.2023.10099851
17. Asma AlNashash Supervisor Ahmad T Al-Taani Co-Supervisor Saleh M Abu-Soud BJ (2022) Annotated data augmentation for arabic sentiment analysis using semi-supervised GANs
18. Khalifa M, Abdul-Mageed M, Shaalan K (2021) Self-training pre-trained language models for zero- And few-shot multi-dialectal Arabic sequence labeling. In: EACL 2021 - 16th conference of the European chapter of the association for computational linguistics, proceedings of the conference, pp 769–782. https://doi.org/10.18653/v1/2021.eacl-main.65
19. Omran TM, Sharef BT, Grosan C, Li Y (2023) Transfer learning and sentiment analysis of Bahraini dialects sequential text data using multilingual deep learning approach. Data Knowl Eng 143:102106. https://doi.org/10.1016/J.DATAK.2022.102106
20. Chouikhi H, Jarray F (2023) BERT-based ensemble learning approach for sentiment analysis. Commun Comput Inf Sci 1718:118–128. https://doi.org/10.1007/978-3-031-35924-8_7
21. Tran S, Rodrigues P, Strauss B, Williams EM (2023) Accenture at CheckThat! 2023: identifying claims with societal impact using NLP data augmentation. CEUR Workshop Proc 3497:518–525
22. Hamed I, Habash N, Abdennadher S, Vu NT (2023) Investigating lexical replacements for Arabic-English code-switched data augmentation. In: 6th workshop on technologies for machine translation of low-resource languages, LoResMT 2023 – Proceedings, pp 86–100. https://doi.org/10.18653/v1/2023.loresmt-1.7
23. Lund G, Omelianchuk K, Grammarly IS (2023) Gender-Inclusive Grammatical Error Correction through Augmentation. https://doi.org/10.18653/v1/2023.bea-1.13. https://aclanthology.org/2023.bea-1.13/
24. Almasre MA (2022) Enhance the aspect category detection in Arabic language using AraBERT and text augmentation. In: 2022 fifth national conference of Saudi computers colleges (NCCC), pp 1–4. https://doi.org/10.1109/NCCC57165.2022.10067648
25. Refai D, Abu-Soud S, Abdel-Rahman MJ (2023) Data augmentation using transformers and similarity measures for improving Arabic text classification. IEEE Access 11:132516–132531. https://doi.org/10.1109/ACCESS.2023.3336311
26. Solyman A, Zappatore M, Zhenyu W, Mahmoud Z, Alfatemi A, Ibrahim AO, Gabralla LA (2023) Optimizing the impact of data

augmentation for low-resource grammatical error correction. J King Saud Univ-Comput Inf Sci 35(6):101572. https://doi.org/10.1016/J.JKSUCI.2023.101572
27. Duwairi R, Abushaqra F (2021) Syntactic- and morphology-based text augmentation framework for Arabic sentiment analysis. PeerJ Comput Sci 7:1–25. https://doi.org/10.7717/PEERJ-CS.469/SUPP-4
28. Al-Jamal WQ, Mustafa AM, Ali MZ (2022) Sarcasm detection in arabic short text using deep learning. In: 2022 13th international conference on information and communication systems, ICICS 2022, pp 362–366. https://doi.org/10.1109/ICICS55353.2022.9811153
29. Zakraoui J, Alja'am JM, Salah I (2022) Domain-specific text generation for Arabic text summarization. In: Proceedings of the international conference on computer and applications, ICCA 2022—Proceedings. https://doi.org/10.1109/ICCA56443.2022.10039630
30. Alsafari S, Sadaoui S (2021) Semi-supervised Self-learning for Arabic hate speech detection. In: Conf Proc IEEE Int Conf Syst Man Cybern, pp 863–868. https://doi.org/10.1109/SMC52423.2021.9659134
31. Beseiso M, Elmousalami H (2020) Subword attentive model for Arabic sentiment analysis: a deep learning approach. ACM Trans Asian Low-Resour Lang Inf Process (TALLIP) 19(2):1–7. https://doi.org/10.1145/3360016
32. Ghadery E, Moens MF (2020) LIIR at SemEval-2020 Task 12: a cross-lingual augmentation approach for multilingual offensive language identification. In: 14th international workshops on semantic evaluation, SemEval 2020—co-located 28th international conference on computational linguistics, COLING 2020, Proceedings, pp 2073–2079. https://doi.org/10.18653/v1/2020.semeval-1.274
33. Modzelewski A, Sosnowski W, Wierzbicki A (2023) DSHacker at CheckThat! 2023: Check-Worthiness in Multigenre and Multilingual Content With GPT-3.5 Data Augmentation. In: CEUR Workshop Proc, vol 3497, pp 383–393
34. Bayer M, Kaufhold MA, Reuter C (2022) A survey on data augmentation for text classification. ACM Comput Sur 55(7):1–39. https://doi.org/10.1145/3544558
35. Wei J, Zou K (2019) EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In: EMNLP-IJCNLP 2019—2019 conference on empirical methods in natural language processing and 9th international joint conference on natural language processing, proceedings of the conference, pp 6382–6388. https://doi.org/10.18653/v1/d19-1670
36. Miller GA (1995) WordNet: a lexical database for english. Commun ACM 38:39–41. https://doi.org/10.1145/219717.219748
37. Samer D, Supervisor R, Abu-Soud S, Abdullah K (2022) A new data augmentation approach for boosting arabic text classification performance using transformers and similarity measures
38. Omara E, Mosa M, Ismail N (2019) Emotion analysis in Arabic language applying transfer learning. In: ICENCO 2019—2019 15th international computer engineering conference: utilizing machine intelligence for a better world, pp 204–209. https://doi.org/10.1109/ICENCO48310.2019.9027295
39. AbuElAtta AH, Sobhy M, El-Sawy AA, Nayel H (2023) Arabic regional dialect identification (ARDI) using pair of continuous bag-of-words and data augmentation. Int J Adv Comput Sci Appl 14:258–264
40. Pasha A, Al-Badrashiny M, Diab M, et al (2014) MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic. https://aclanthology.org/L14-1479/
41. Alkadri AM, Elkorany A, Ahmed C (2022) Enhancing detection of Arabic social spam using data augmentation and machine learning. Appl Sci 12(22):11388. https://doi.org/10.3390/APP122211388
42. Alkadri AM, ElKorany A, Ezzat CA (2023) An integrated framework for relevance classification of trending topics in Arabic tweets. Int J Adv Comput Sci Appl 14:884–891. https://doi.org/10.14569/IJACSA.2023.0140796
43. Tawalbeh SK, Hammad M, AL-Smadi M (2020) KEIS@JUST at SemEval-2020 Task 12: identifying multilingual offensive tweets using weighted ensemble and fine-tuned BERT. In: 14th international workshops on semantic evaluation, SemEval 2020—co-located 28th international conference on computational linguistics, COLING 2020, proceedings, pp 2035–2044. https://doi.org/10.18653/v1/2020.semeval-1.269
44. Shaheen M, Nigam SK (2022) Plumeria at SemEval-2022 Task 6: sarcasm detection for english and Arabic using transformers and data augmentation. In: SemEval 2022—16th international workshop on semantic evaluation, proceedings of the workshop, pp 923–937. https://doi.org/10.18653/V1/2022.SEMEVAL-1.130
45. Fadel AS, Abulnaja OA, Saleh ME (2023) Multi-task learning model with data augmentation for Arabic aspect-based sentiment analysis. Comput, Mater Contin 75:4419–4444. https://doi.org/10.32604/CMC.2023.037112
46. Zahir J (2023) Prediction of court decision from Arabic documents using deep learning. Expert Syst. https://doi.org/10.1111/EXSY.13236
47. Aloraini A, Poesio M (2021) Data augmentation methods for anaphoric zero pronouns. In: 4th workshop on computational models of reference, anaphora and coreference, CRAC 2021—proceedings of the workshop, pp 82–93. https://doi.org/10.18653/v1/2021.crac-1.9
48. Hsu TW, Chen CC, Huang HH, Chen HH (2021) Semantics-preserved data augmentation for aspect-based sentiment analysis. In: EMNLP 2021—2021 conference on empirical methods in natural language processing, proceedings, pp 4417–4422. https://doi.org/10.18653/v1/2021.emnlp-main.362
49. Kchaou S, Boujelbane R, Belguith LH (2022) Bottom-up approach to translate Tunisian dialect texts in Social Networks. In: Proceedings of IEEE/ACS international conference on computer systems and applications, AICCSA 2022-December: https://doi.org/10.1109/AICCSA56895.2022.10017688
50. Laskar SR, Singh R, Khilji AFUR, et al (2022) CNLP-NITS-PP at WANLP 2022 shared task: propaganda detection in Arabic using data augmentation and AraBERT Pre-trained model. In: WANLP 2022—7th Arabic natural language processing—proceedings of the workshop, pp 541–544
51. Kchaou S, Boujelbane R, Belguith LH (2020) Parallel resources for Tunisian Arabic dialect translation. pp 200–206
52. Yuan Z, Strohmaier D (2021) Cambridge at SemEval-2021 Task 2: neural WiC-model with data augmentation and exploration of representation. In: SemEval 2021—15th international workshop on semantic evaluation, proceedings of the workshop, pp 730–737. https://doi.org/10.18653/v1/2021.semeval-1.96
53. Fsih E, Kchaou S, Boujelbane R, Belguith LH (2022) Benchmarking transfer learning approaches for sentiment analysis of Arabic dialect. In: WANLP 2022—7th Arabic Natural Language Processing - Proceedings of the Workshop, pp 431–435
54. Basabain S, Cambria E, Alomar K, Hussain A (2023) Enhancing Arabic-text feature extraction utilizing label-semantic augmentation in few/zero-shot learning. Expert Syst. https://doi.org/10.1111/EXSY.13329
55. Alawawdeh SM, Abandah GA (2021) Improving the accuracy of semantic similarity prediction of arabic questions using data augmentation and ensemble. In: 2021 IEEE jordan international joint conference on electrical engineering and information

technology, JEEIT 2021—Proceedings, pp 272–277. https://doi.org/10.1109/JEEIT53412.2021.9634095
56. Batarfi HA, Alsaedi OA, Wali AM, Jamal AT (2023) Impact of data augmentation on hate speech detection. Commun Comput Inf Sci 1876:187–199. https://doi.org/10.1007/978-3-031-40852-6_10
57. Williams E, Rodrigues P, Novak V (2020) Accenture at Check-That! 2020: If you say so: Post-hoc fact-checking of claims using transformer-based models. CEUR Workshop Proc 2696:2020
58. Tran S, Rodrigues P, Strauss B, Williams EM (2023) Accenture at CheckThat! 2023: impacts of back-translation on subjectivity detection. CEUR Workshop Proc, vol 3497, pp 507–517
59. Abumansour AS, Zubiaga A (2023) Check-worthy claim detection across topics for automated fact-checking. PeerJ Comput Sci 9:e1365. https://doi.org/10.7717/PEERJ-CS.1365
60. Abid W (2020) The SADID evaluation datasets for low-resource spoken language machine translation of Arabic dialects. In: COLING 2020—28th international conference on computational linguistics, proceedings of the conference, pp 6030–6043. https://doi.org/10.18653/v1/2020.coling-main.530
61. BensoltaneRajae ZakiTaher (2023) Combining BERT with TCN-BiGRU for enhancing Arabic aspect category detection. J Intell Fuzzy Syst 44:4123–4136. https://doi.org/10.3233/JIFS-221214
62. Alzanin SM, Gumaei A, Haque MA, Muaad AY (2023) An optimized Arabic Multilabel text classification approach using genetic algorithm and ensemble learning. Appl Sci (Switzerland) 13:10264. https://doi.org/10.3390/app131810264
63. Patidar M, Kumari S, Patwardhan M, et al (2021) From monolingual to multilingual faq assistant using multilingual co-training. In: DeepLo@EMNLP-IJCNLP 2019—Proceedings of the 2nd workshop on deep learning approaches for low-resource natural language processing—proceedings, pp 115–123. https://doi.org/10.18653/v1/d19-6113
64. Zengin MS, Kartal YS, Kutlu M (2021) TOBB ETU at Check-That! 2021: Data engineering for detecting check-worthy claims. CEUR Workshop Proc 2936:670–680
65. Ameur A, Hamdi S, Ben Yahia S (2023) Arabic aspect category detection for hotel reviews based on data augmentation and classifier Chains. In: Proceedings of the ACM symposium on applied computing, pp 942–949. https://doi.org/10.1145/3555776.3577746
66. Omran T, Sharef B, Grosan C, Li Y (2022) Ensemble learning for sentiment analysis of translation-based textual data. In: International conference on electrical, computer, communications and mechatronics engineering, ICECCME Male. https://doi.org/10.1109/ICECCME55909.2022.9988242
67. Davletov A, Arefyev N, Gordeev D, Rey A (2021) LIORI at SemEval-2021 Task 2: span prediction and binary classification approaches to word-in-context disambiguation. In: SemEval 2021—15th international workshop on semantic evaluation, proceedings of the workshop, pp 780–786. https://doi.org/10.18653/v1/2021.semeval-1.103
68. Gaanoun K, Benelallam I (2020) Arabic dialect identification: an Arabic-BERT model with data augmentation and ensembling strategy. pp 275–281
69. Gaanoun K, Benelallam I (2022) SI2M & AIOX Labs at WANLP 2022 shared task: propaganda detection in Arabic, a data augmentation and named entity recognition approach. In: WANLP 2022—7th Arabic natural language processing—proceedings of the workshop
70. Mohammed A, Kora R (2019) Deep learning approaches for Arabic sentiment analysis. Soc Netw Anal Min 9:52. https://doi.org/10.1007/S13278-019-0596-4
71. Orabe Z, Haddad B, Al-Abood A, Ghneim N (2020) DoTheMath at SemEval-2020 Task 12 : deep neural networks with self attention for Arabic offensive language detection. In: 14th international workshops on semantic evaluation, SemEval 2020—co-located 28th international conference on computational linguistics, COLING 2020, Proceedings, pp 1932–1937. https://doi.org/10.18653/v1/2020.semeval-1.254
72. Shammary F, Chen Y, Kardkovács ZT, et al (2022) TF-IDF or transformers for Arabic dialect identification? ITFLOWS participation in the NADI 2022 Shared Task. WANLP 2022—7th Arabic natural language processing—proceedings of the workshop, pp 420–424. https://doi.org/10.18653/v1/2022.wanlp-1.42
73. Williams E, Rodrigues P, Tran S (2021) Accenture at CheckThat! 2021: Interesting claim identification and ranking with contextually sensitive lexical training data augmentation. In: CEUR Workshop Proc 2936:659–669
74. Hamza A, El AlaouiOuatik S, Zidani KA, En-Nahnahi N (2022) Arabic duplicate questions detection based on contextual representation, class label matching, and structured self attention. J King Saud Univ—Comput Inf Sci 34:3758–3765. https://doi.org/10.1016/J.JKSUCI.2020.11.032
75. Fadel A, Tuffaha I, Al-Ayyoub M (2019) Tha3aroon at NSURL-2019 Task 8: semantic question similarity in Arabic. 50–58
76. Allen JF (2013) Maintaining knowledge about temporal intervals. Readings in Qualitative Reasoning About Physical Systems 361–372. https://doi.org/10.1016/B978-1-4832-1447-4.50033-X
77. Haffar N, Hkiri E, Zrigui M (2020) Using bidirectional LSTM and shortest dependency path for classifying Arabic temporal relations. Procedia Comput Sci 176:370–379. https://doi.org/10.1016/j.procs.2020.08.038
78. Keleg A, El-Beltagy SR, Khalil M (2020) ASU_OPTO at OSACT4—Offensive Language Detection for Arabic text. pp 66–70
79. Madi N, Al-Khalifa H (2020) Error detection for Arabic text using neural sequence labeling. Appl Sci (Switzerland) 10:5279. https://doi.org/10.3390/APP10155279
80. Lad R, Ma W, Vosoughi S (2022) Dartmouth at SemEval-2022 Task 6: detection of sarcasm. In: SemEval 2022—16th international workshop on semantic evaluation, proceedings of the workshop, pp 912–918. https://doi.org/10.18653/v1/2022.semeval-1.128
81. Moslem Y, Way A, Haque R, Kelleher JD (2022) Domain-specific text generation for machine translation. In: AMTA 2022—15th conference of the association for machine translation in the Americas, proceedings, vol 1, pp 14–30
82. Farhan YH, Noah SAM, Mohd M, Atwan J (2021) Word embeddings-based pseudo relevance feedback using deep averaging networks for Arabic document retrieval. J Inf Sci Theor Pract 9:1–17. https://doi.org/10.1633/JISTaP.2021.9.2.1
83. Elmadany A, Zhang C, Abdul-Mageed M, Hashemi A (2020) Leveraging Affective Bidirectional Transformers for Offensive Language Detection. pp 11–16
84. Israeli A, Nahum Y, Fine S, Bar K (2021) The IDC system for sentiment classification and sarcasm detection in Arabic. In: WANLP 2021—6th Arabic natural language processing workshop, proceedings of the workshop, pp 370–375
85. Bouzid SM, Zribi CBO (2021) Efficient learning approach for pronominal anaphora and ellipsis identification and resolution in Arabic texts. IEEE/ACM Trans Audio Speech Lang Process 29:3335–3348. https://doi.org/10.1109/TASLP.2021.3120649
86. Zhang C, Abdul-Mageed M (2019) No army, no navy: Bert semi-supervised learning of arabic dialects _. In: ACL 2019—4th Arabic natural language processing workshop, WANLP 2019—proceedings of the workshop, pp 279–284. https://doi.org/10.18653/v1/w19-4637
87. Miao L, Last M, Litvak M (2020) Twitter data augmentation for monitoring public opinion on COVID-19 intervention measures. https://doi.org/10.18653/v1/2020.nlpcovid19-2.19. https://aclanthology.org/2020.nlpcovid19-2.19/

88. Liu J, Shou L, Pei J, et al (2020) Cross-lingual machine reading comprehension with language branch knowledge distillation. In: COLING 2020—28th international conference on computational linguistics, proceedings of the conference, pp 2710–2721. https://doi.org/10.18653/v1/2020.coling-main.244
89. Sawhney R, Thakkar M, Pandit S, et al (2022) DMIX: adaptive distance-aware interpolative Mixup. In: Proceedings of the annual meeting of the association for computational linguistics, vol 2, pp 606–612. https://doi.org/10.18653/v1/2022.acl-short.67
90. Elagbry HE, Attia S, Abdel-Rahman A, et al (2021) A contextual word embedding for Arabic sarcasm detection with random forests. In: WANLP 2021—6th Arabic natural language processing workshop, proceedings of the workshop,pp 340–344
91. Adouane W, Touileb S, Bernardy JP (2020) Identifying sentiments in algerian code-switched user-generated comments. In: LREC 2020—12th international conference on language resources and evaluation, conference proceedings 2698–2705
92. Setyanto A, Laksito A, Alarfaj F et al (2022) Arabic language opinion mining based on long short-term memory (LSTM). Appl Sci (Switzerland) 12:4140. https://doi.org/10.3390/app12094140
93. Abuzayed A, Al-Khalifa H (2021) Sarcasm and sentiment detection in Arabic tweets using BERT-based models and data augmentation. pp 312–317
94. Zhang C, Abdul-Mageed M (2019) BERT-based Arabic social media author profiling. CEUR Workshop Proc 2517:84–91
95. Haddad B, Orabe Z, Al-Abood A, Ghneim N (2020) Arabic offensive language detection with attention-based deep neural networks. Pp 76–81
96. Dahou AH, Cheragui MA (2023) Impact of normalization and data augmentation in NER for algerian arabic dialect. Lecture notes in networks and systems 593 LNNS:249–262. https://doi.org/10.1007/978-3-031-18516-8_18/TABLES/8
97. Zhang C, Zhang Q, Hansen JHL (2019) Semi-supervised learning with generative adversarial networks for Arabic dialect identification. In: ICASSP, IEEE international conference on acoustics, speech and signal processing—proceedings, pp 5986–5990. https://doi.org/10.1109/ICASSP.2019.8682629
98. Soufan A (2019) Deep learning for sentiment analysis of Arabic text. In: ACM international conference proceeding series. https://doi.org/10.1145/3333165.3333185
99. Al-Azani S, El-Alfy ESM (2021) Early and late fusion of emojis and text to enhance opinion mining. IEEE Access 9:121031–121045. https://doi.org/10.1109/ACCESS.2021.3108502
100. Samir A, Soliman AB, Ibrahim M, et al (2022) NGU_CNLP at WANLP 2022 Shared Task: propaganda detection in Arabic. In: WANLP 2022—7th Arabic natural language processing—proceedings of the workshop, pp 545–550. https://doi.org/10.18653/v1/2022.wanlp-1.66
101. García-Díaz JA, Pan R, Zafra SMJ, et al (2023) UMUTeam and SINAI at SemEval-2023 Task 9: multilingual tweet intimacy analysis using multilingual large language models and data augmentation. In: 17th international workshop on semantic evaluation, SemEval 2023—proceedings of the workshop, pp 293–299. https://doi.org/10.18653/v1/2023.semeval-1.39
102. Antoun W, Baly F, Hajj H (2020) AraGPT2: Pre-Trained transformer for Arabic language generation. In: WANLP 2021—6th Arabic natural language processing workshop, proceedings of the workshop, pp 196–207
103. Hailu TT, Yu J, Fantaye TG (2020) Intrinsic and extrinsic automatic evaluation strategies for paraphrase generation systems. J Comput Commun 08:1–16. https://doi.org/10.4236/JCC.2020.82001
104. Farha IA, Magdy W (2020) From Arabic sentiment analysis to sarcasm detection: the ArSarcasm dataset. pp 32–39
105. Nabil M, Aly M, Atiya AF (2015) ASTD: Arabic sentiment tweets dataset. In: conference proceedings—EMNLP 2015: conference on empirical methods in natural language processing, pp 2515–2519. https://doi.org/10.18653/V1/D15-1299
106. Bouamor H, Habash N, Salameh M, et al (2019) The madar Arabic dialect corpus and lexicon. In: LREC 2018—11th international conference on language resources and evaluation, pp 3387–3396
107. Ziemski M, Junczys-Dowmunt M, Pouliquen B (2016) The United Nations Parallel Corpus v1.0. pp 3530–3534
108. Kora R, Mohammed A (2019) Corpus on Arabic Egyptian tweets
109. Elmadany A, Mubarak H, Magdy W (2018) Arsas: An arabic speech-act and sentiment corpus of tweets. Osact 3
110. Benajiba Y, Rosso P, Ruiz JMB (2007) ANERsys: an Arabic named entity recognition system based on maximum entropy. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 4394 LNCS, pp 143–153. https://doi.org/10.1007/978-3-540-70939-8_13
111. Alalyani N, Larabi S, Sainte M (2018) NADA: new Arabic dataset for text classification. Int J Adv Comput Sci Appl. https://doi.org/10.13140/RG.2.2.13606.01603
112. Aly M, Atiya A (2013) LABR: A large scale arabic book reviews dataset. Pp 494–498. https://aclanthology.org/P13-2088/
113. Alam F, Mubarak H, Zaghouani W, et al (2022) Overview of the WANLP 2022 shared task on propaganda detection in Arabic
114. Alshammari W, Alhumoud S (2022) TAQS: an Arabic question similarity system using transfer learning of BERT with BiLSTM. IEEE Access. https://doi.org/10.1109/ACCESS.2022.3198955
115. Zhou Y, Guo C, Wang X, et al (2024) A survey on data augmentation in large model Era. p 14
116. Reimers N, Gurevych I (2019) Sentence-BERT: Sentence embeddings using siamese BERT-networks. In: EMNLP-IJCNLP 2019—2019 conference on empirical methods in natural language processing and 9th international joint conference on natural language processing, proceedings of the conference, pp 3982–3992. https://doi.org/10.18653/v1/d19-1410
117. Soliman AB, Eissa K, El-Beltagy SR (2017) AraVec: a set of Arabic word embedding models for use in Arabic NLP. Procedia Comput Sci 117:256–265. https://doi.org/10.1016/J.PROCS.2017.10.117
118. Wang K, Wan X (2019) Automatic generation of sentimental texts via mixture adversarial networks. Artif Intell 275:540–558. https://doi.org/10.1016/J.ARTINT.2019.07.003