applied sciences

Article

Enhancing Detection of Arabic Social Spam Using Data Augmentation and Machine Learning

Abdullah M. Alkadri , Abeer Elkorany and Cherry Ahmed *

Faculty of Computers and Artificial Intelligence, Cairo University, Giza 12613, Egypt
* Correspondence: c.ahmed@fci-cu.edu.eg

Abstract: In recent years, people have tended to use online social platforms, such as Twitter and Facebook, to communicate with families and friends, read the latest news, and discuss social issues. As a result, spam content can easily spread across them. Spam detection is considered one of the important tasks in text analysis. Previous spam detection research focused on English content, with less attention to other languages, such as Arabic, where labeled data are often hard to obtain. In this paper, an integrated framework for Twitter spam detection is proposed to overcome this problem. This framework integrates data augmentation, natural language processing, and supervised machine learning algorithms to overcome the problems of detection of Arabic spam on the Twitter platform. The word embedding technique is employed to augment the data using pre-trained word embedding vectors. Different machine learning techniques were applied, such as SVM, Naive Bayes, and Logistic Regression for spam detection. To prove the effectiveness of this model, a real-life data set for Arabic tweets have been collected and labeled. The results show that an overall improvement in the use of data augmentation increased the macro F1 score from 58% to 89%, with an overall accuracy of 92%, which outperform the current state of the art.

Keywords: data augmentation; machine learning; spam detection; online social networks; Arabic spam

check for updates

Citation: Alkadri, A.M.; Elkorany, A.; Ahmed, C. Enhancing Detection of Arabic Social Spam Using Data Augmentation and Machine Learning. *Appl. Sci.* **2022**, *12*, 11388. https://doi.org/10.3390/app122211388

Academic Editors: Huan Wang and Wen Zhang

Received: 2 October 2022
Accepted: 4 November 2022
Published: 10 November 2022

1. Introduction

Social networks such as Facebook, Twitter, and Amazon have become important platforms for satisfying people's needs for social interactions, information acquisition, and dissemination. However, such unparalleled convenience supports the operations of malicious entities, such as virus distribution, fraud, falsifying product reviews, and link farming.

In social networks, spam is defined as unwanted, malicious, or inappropriate content or behavior that can take many forms, such as microblogs, malicious links, messages, fake friends, fake reviews, etc. Spammers can earn money directly (e.g., by writing fake reviews to advertise their products) or achieve popularity via spamming (e.g., by building connections with many other users) [1]. Spam decreases the communication quality provided by social media platforms. It pollutes social networks and distorts people's perceptions of internet information. The user experience will drastically decrease if someone is exposed to highly undesirable information, which would result in customer losses for the social service provider [1]. Therefore, social platforms must develop algorithms to detect spam content.

According to Masood, Faiza, et al. [1], the spam protection field in social networks has been approached from two perspectives: The detection of spam and the detection of spammers. Research on spam detection approaches can be classified into user-based spam detection methods, content-based spam detection methods, trend topic methods, graph-based spam detection methods, and hybrid spam detection methods [1]. To protect legitimate users from being affected, spam detection aims to develop effective and efficient methods for automatically identifying spam content and its originators. Social spam refers to unwanted or malicious content posted on the social network platform. It includes scam malware, phishing website, commercial advertisement, and promotion information, disguised posts that appear legitimate but contain malicious URLs, reply spams, scam software, information about promotions, and reposting spam that attaches junk information behind original posts [2,3].

Due to the subjective nature of spam detection and the domain used, which is dependent on the researcher's definition of spam, finding a consistent definition is challenging. In this paper, we focus more on detecting the social content spam category on Twitter. Twitter is one of the most popular online social platforms. Due to its widely used, online spammers use it as a target to spread illegal content, fake news, and online advertisements [4]. Spam detection on a noisy site such as Twitter remains challenging due to the short text and the wide range of language used on social networks [5].

Machine learning is widely used in different fields for automatic learning and detection, including spam detection on Twitter [6]. Previous research has extracted and utilized a wide range of features, from simple to complex, as well as a wide range of learning and classification algorithms, from traditional machine learning techniques to deep learning [6,7]. There are not enough data to train a high-quality classifier in many machine learning scenarios. Data augmentation can be utilized to resolve this problem [8]. In the spam detection field, the problem of insufficient data is often encountered due to the difficulty and the weak ability of data collection and labeling. The model will be overfitted if the quantity of samples obtained is insufficient to meet the needs of model training. We can overcome the problem of having a limited number of training instances using data augmentation [9]. Furthermore, a common issue in the spam classification task is data imbalance, resulting in poor classification results. If data augmentation is performed, imbalanced data can be transformed into balanced data, or sample imbalance can be minimized, and model performance can be improved [10]. The model's performance is highly dependent on the data's quality. Therefore, this paper proposes applying Data Augmentation for the Arabic Spam Detection (DASD) domain which would lead to significant improvements in overall accuracy. The Embedding Replacement as a data augmentation substitution technique is used in the proposed framework. It replaces a word in the sentence according to its similarities using a particular pre-trained word embedding vectors. The main contributions of this paper are the following:

- Handling different categories of Arabic spam content such as advertising, spreading malware, adult content, hate speech, and meaningless content.
- Creation of a public and available Arabic spam dataset containing the original tweets and the corresponding annotations.
- Applying replacement embedding data augmentation technique for Arabic text.
- Evaluation of the overall framework by comparing its accuracy to the original (non-augmented) dataset, which leads to improvement of the overall Macro F1 up to ∼32% and recalls improvement of the spam class of up to ∼38%.

The rest of this paper is organized as follows: Section 2 presents background and related work; Section 3 introduces the proposed framework; experimental results are in Section 4; discussion are in Section 5; conclusions and future work are finally presented in Section 6.

2. Background and Related Work

2.1. Data Augmentation

Data augmentation is a technique to increase the diversity and quality of data without directly collecting more data. The first application of data augmentation was in computer vision [11]. Because the image is made up of individual pixels, it can be easily modified by adding noise, cropping, padding, or flipping it without losing any of the original information. Although the idea of augmenting a dataset with perturbed replicas of its samples has been shown to be very successful in other domains (image classification [12–14] and sound classification [15]), it has been underexplored in Natural Language Processing (NLP) field.

This challenge is still present today, but many scientists and researchers are attempting to solve it to achieve a variety of objectives, such as balancing imbalanced dataset classes, generating more data for low-data domains, or protecting against adversarial examples. While flipping and rotating image techniques produce new, valid images with similar semantic information, they cannot be used for text, since they would affect syntax and grammar and even change the meaning of the original sentence. Furthermore, while noise injection is commonly used to enhance audio signal data [16–18], it is not directly suitable for text since word and character tokens are categorical.

Researchers are focusing on introducing text augmentation to address text data imbalance problems and improve the generalization of the model [19]. Data augmentation is a strategy to create more training data by modifying existing training data, either by making minor changes to the original instances or by using the existing data as a template [8]. Text data augmentation can be classified into two approaches: Feature space and data space [20]. In the data space, augmentation refers to the transformation of raw data into readable textual data. The data space in [20] was categorized into four categories: Character Level, word Level, phrase sentence Level, and document Level. In this paper, word-level data space techniques are described for manipulating parts of training examples to generate new samples, such as:

1. Noise Induction: Randomly insert, delete, swap, and substitute words.
2. Synonym Replacement: The paraphrasing transformation of text instances by replacing certain words with synonyms (WordNet) is the popular form of data augmentation.
3. Embedding Replacement: Comparable to synonym replacement, embedding replacement strategies look for words that fit well into the textual context and do not change the text's basic substance. To achieve this, close words from similar contexts are translated into a latent representation space [20]. Word embeddings can be static (word vectors), pre-trained classic word embeddings such as self-trained W2V, Glove, and Google-News W2V, to perform similarity searches. On the contrary, the other type applies contextual data augmentation, such as BERT, which replaces words with words predicted from a bidirectional language model according to the context of the original word.

In text augmentation tasks, sentences are organized and words are always closely connected. That is why a minor change in a sentence could cause the meaning to be flipped. For example, if we use the random swap in the sentence "I like an apple." to "an apple like I ", the meaning is entirely different. Therefore, when augmenting the sentence, the data augmentation methods must follow the grammar of the language and keep the original meaning. For this purpose, we decide to use the word embedding replacement technique to augment the data [20].

It should be mentioned that, despite the progress in Arabic spam detection, which will be explained in the next Section 2.2, data augmentation was not applied as part of these works. According to the literature, a limited number of recent Arabic research works have been conducted using data augmentation on other topics, such as

- Sentiment analysis, Mohammed et al. [21] used shuffling by randomly changing the order of words inside the small size context window and Duwairi et al. [22] used a set of rules to alter or swap branches of the parse trees as per Arabic syntax to generate new sentences with the same labels, and others to insert negation particles into the sentences and, thus, generate new sentences with opposite labels.
- Named entity recognition, Sabty et al. [23] used a set of data augmentation techniques, namely: Word random insertion, swap and deletion in the sentence, sentence back-translation, and word embedding substitution.

According to Feng et al. [24], there are various data augmentation types of research for popular NLP tasks in non-Arabic languages such as text classification [25,26], translation [27,28], summarization [29], question-answering [30], sequence tagging [31], parsing [32], grammatical-error-correction [33], mitigating class imbalance [34], and automated augmentation [35].

2.2. Spam Detection Techniques

Most research in the area of spam detection is focused on English text, while little research has been done on Arabic text. This is mainly due to its morphological complexity and the limited availability of compatible Arabic language tools and a dataset. The English Twitter-sphere has a lot of spam detection research (e.g., [36–47]). However, there has been work on detecting undesirable Arabic content such as advertisement [48–50], adult-content [51,52], offensive, hate, and vulgar speech [53–56], spam reviews [57–59], and detect spam accounts [60–62]. Table 1 summarizes existing Arabic spam-detection studies.

However, concerns of class imbalance for the Arabic spam detection dataset, as well as spam class recall, are not addressed. In this research, we replicate the state-of-the-art data augmentation techniques on a non-text domain and show that there is still a lot of opportunity for improvement by implementing data augmentation techniques in the text domain. Furthermore, Arabic studies did not cover all categories of spam content at once, or the dataset labeling step is unclear or depends on the English translation.

Table 1. Arabic spam detection studies.

Detecting Level	Features	Classifiers	Dataset	Results (Accuracy-F1)
[48] content-level	content-based, user-based	AraBERT, SVM	133,500 tweets.	F1 98%
[49] content-level	content-based	NB, SMO, LR	2224 Facebook comments	91.73%
[51] user-level	content-based, user -based	NB, SVM	500 accounts	90%
[52] content-level	content-based	SVM	-	79%
[53] content-level	content-based	LR, SVM, RNN	6000 Arabic tweets	79%
[50] content-level	content-based	NB, SVM, DT	3503 Arabic tweets	87%
[57] content-level	content-based	LR, SVM, NB, RF	1600 hotels reviews	95.25%
[58] content-level	content-based, user -based	NB, KNN, SVM	2848 hotels reviews	F1 99.59%
[59] content-level	content-based	KNN, SVM, NB	3000 Facebook comments	92.63%
[60] user-level	content-based, user -based	RF	12,486 accounts	90%
[61] user-level	content-based, user -based	NB, RF, SVM	5000 tweets	92.59%

3. Proposed Framework for Enhancing Arabic Spam Detection

As shown in Figure 1, the proposed spam detection framework consists of three main components: Data preparation, data augmentation, and the application of machine learning. In the following subsections, each phase will be described in detail.

3.1. Data Preparation

3.1.1. Data Extraction

During this phase, a dataset of around 1.6 M tweets was collected using the Twitter Streaming API from December 2019 to April 2020. The tweets were collected using the query “lang Ar” (language is Arabic) and the trending hashtag (#YEMEN). Next, each tweet was cleaned to obtain the distinct tweets by removing diacritics, removing repeating char, removing punctuations, normalizing Arabic, removing stop words, removing non-Arabic alphabets, Arabic light stemming, etc. Out of the 1.6 M tweets, there are 313 k distinct tweets. For experimental purposes, a set of 5k tweets was randomly selected for the labeling process, which is explained next.

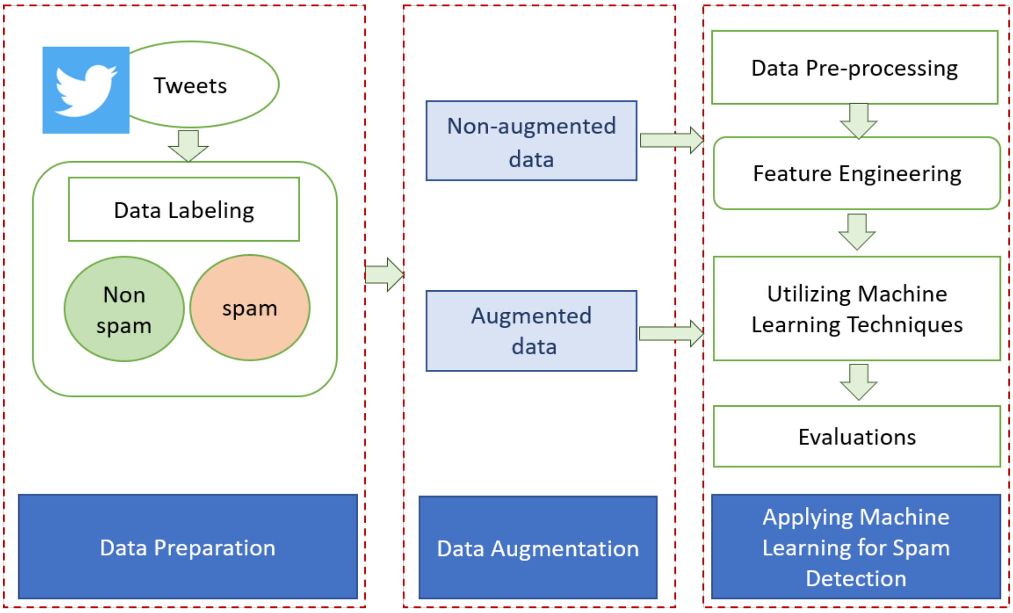


Figure 1. Proposed Framework for Enhancing Arabic Spam Detection (DASD).

3.1.2. Data Labeling

To construct our spam/non-spam dataset, the extracted tweets were manually annotated as spam/non-spam. Fifteen Arabic annotators helped in this process. To accelerate the process, a website (https://tweettag.000webhostapp.com/login.php, accessed on 7 January 2022) has been created to allow multiple experts to annotate the selected tweets as follows:

1. Before starting the annotation process, each annotator was given our definition of spam text as well as some examples of spam content. We asked annotators to carefully read a tweet before evaluating whether it contained spam content (spam class); or did not contain spam content (non-spam class).
2. The 5 k tweets have been split into five chunks, each chunk containing 1000 tweets. After that, every three annotators have worked on the same 1000 tweets separately so that the annotation score (spam/non-spam) is calculated based on the three annotators, as illustrated in the following step
3. Finally, the annotation results for the 5k tweets from the 15 annotators are grouped, such that each tweet has three annotated values, each from a different expert. We marked the tweet as spam or non-spam according to the mode (most occurring value) result. After labeling, we found that the data are imbalanced with 420 spam tweets and 4580 non-spam tweets.

3.2. Applying Data Augmentation to Collected Data

In this section, the method for dataset augmentation will be explained. The goal of this phase is to avoid overfitting on the data level, increase the model's generalization, and address the class imbalance problem [19]. By increasing the diversity of training samples, the model will be able to learn more fundamental features of the data, leading to a high-quality classifier.

Therefore, this phase aims to increase the total number of instances that represent spam training data to improve the accuracy of the model. Word embedding [20] was used as a substitution method, which changes a word by its similarity to its synonyms. Furthermore, AraVec word vectors (Soliman et al. [63]) were employed, which is a pre-trained model available for two Arabic content domains: Wikipedia and Twitter and it is based on the Mikolov et al. [64]. Skip-gram and CBOW versions with 100- and 300-dimensional vector sizes were provided for each domain. We employed the 300-dimensional CBOW vectors

(AraVec-TWI), which have a wide coverage of different dialect terminology used in random Arabic geolocations.

To apply a similarity replacement on a given text, two steps are required for the Arabic text data augmentation process, as shown in Figure 2. The first step is to identify the word in the tweet to be replaced, and the second step is to select the appropriate replacement based on AraVec.

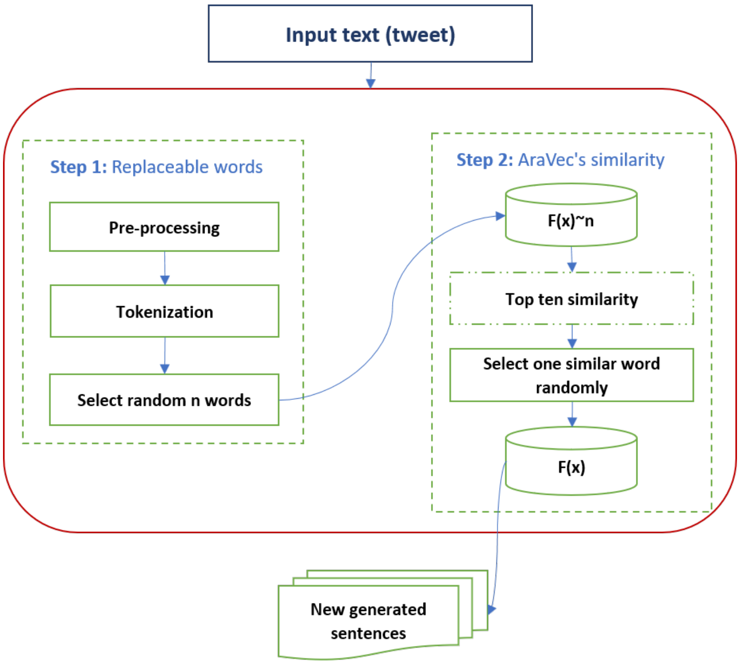


Figure 2. Arabic text data augmentation process.

Step1: The word to be replaced is one of the tokens that have been used in the tweet. Therefore, in this step, all tokens for each tweet are extracted. Next, it is important to decide the number of tokens to be replaced that are not stop words. Therefore, it is significant to quantify the chunk of the sentence to be replaced, i.e., ratio (R) of tokens that should be replaced [65]. Therefore, different experiments have been applied and it was found that the best result was obtained when a ratio between 50% and 70% of the overall tokens were replaced,

$$Number\ of\ replaced\ tokens = R * N. \tag{1}$$

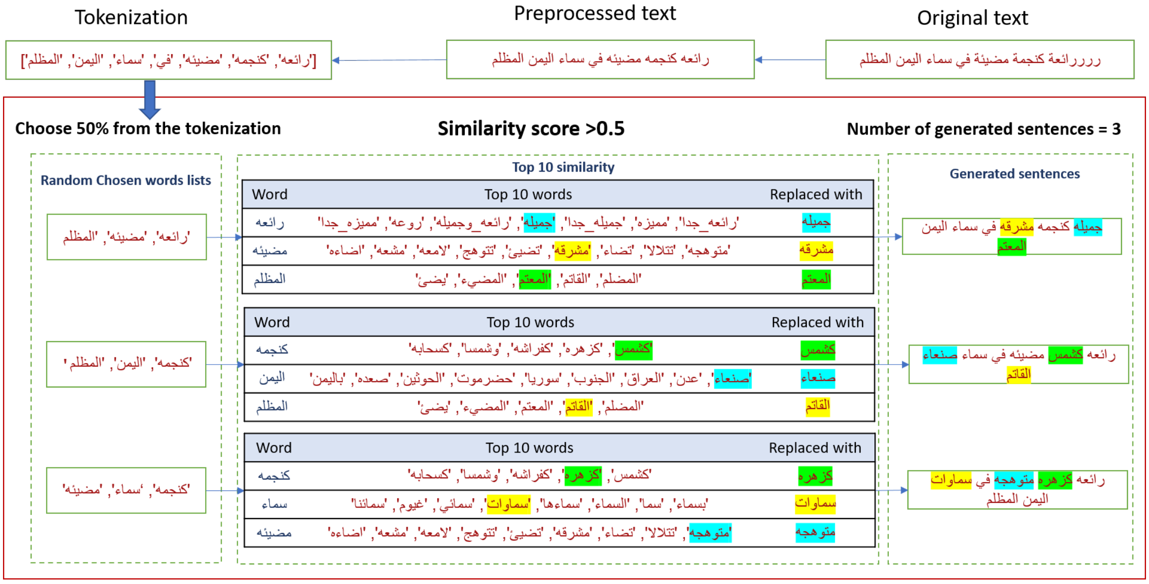
where:

R is a random ratio (between 50–70)

N is the total number of tokens per tweet without stop words.

Step2: The output of step1 is a set of vectors. Each vector contains a list of different combinations of original tokens that are candidates for replacement. To replace each of the selected tokens with one of AraVec's, a similarity scoring mechanism was applied. A score is set for the similar words selected by identifying the surrounding relatively similar terms for each center word in the replaced tokens. Furthermore, relatively minor adjustments have been made by applying sorting criteria regarding word embeddings source (AraVec's), word similarity thresholds, and fixing the number of generated sentences to the original embedding replacement process applied by [20]. For example, to avoid redundancy in the returned similar word, which will be used in replacement, we have returned the top 10 similar words per token with a cutoff similarity score of 0.5. This was applied to ensure that retrieved similar words are the closest to the selected tokens. Furthermore, we fixed the number of generated sentences per tweet to four. This led to double the number of spam tweets four times, resulting in 1648 spam tweets.

An example that illustrates the overall process is shown in Figure 3.


Tokenization
Preprocessed text
Original text
['رائعه', 'كنجمه', 'مضيئه', 'في', 'سماء', 'اليمن', 'المظلم']
رائعه كنجمه مضيئه في سماء اليمن المظلم
رررررائعة كنجمة مضيئة في سماء اليمن المظلم
Choose 50% from the tokenization
Similarity score >0.5
Number of generated sentences = 3
Random Chosen words lists
'رائعه', 'مضيئه', 'المظلم'
'كنجمه', 'اليمن', 'المظلم'
'كنجمه', 'سماء', 'مضيئه'
Top 10 similarity
Generated sentences
جميله كنجمه مشرقه في سماء اليمن المعتم
رائعه كشمس مضيئه في سماء صنعاء القاتم
رائعه كزهره متوهجه في سماوات اليمن المظلم

Word	Top 10 words	Replaced with
رائعه	'رائعه_جدا', 'مميزه', 'جميله_جدا', 'جميله', 'رائعه_وجميله', 'روعه', 'مميزه_جدا'	جميله
مضيئه	'متوهجه', 'تتلالا', 'تضاء', 'مشرقه', 'تضيئ', 'تتوهج', 'لامعه', 'مشعه', 'اضاءه'	مشرقه
المظلم	'المضلم', 'القاتم', 'المعتم', 'المضيء', 'يضئ'	المعتم

Word	Top 10 words	Replaced with
كنجمه	'كشمس', 'كزهره', 'كفراشه', 'وشمسا', 'كسحابه'	كشمس
اليمن	'صنعاء', 'عدن', 'العراق', 'الجنوب', 'سوريا', 'حضرموت', 'الحوثين', 'صعده', 'باليمن'	صنعاء
المظلم	'المضلم', 'القاتم', 'المعتم', 'المضيء', 'يضئ'	القاتم

Word	Top 10 words	Replaced with
كنجمه	'كشمس', 'كزهره', 'كفراشه', 'وشمسا', 'كسحابه'	كزهره
سماء	'بسماء', 'سما', 'السماء', 'سماءها', 'سماوات', 'سمائي', 'غيوم', 'سمائنا'	سماوات
مضيئه	'متوهجه', 'تتلالا', 'تضاء', 'مشرقه', 'تضيئ', 'تتوهج', 'لامعه', 'مشعه', 'اضاءه'	متوهجه

Figure 3. Arabic text data augmentation example. (The colors represent the selected words and each word is distinguished by a different color.)

3.3. Applying Machine Learning for Spam Detection

This section describes the spam detection process by applying different machine learning techniques. This process includes the following steps: Data pre-processing, feature engineering, and finally, applying various machine learning classifiers.

3.3.1. Data Pre-Processing

Several techniques were applied to the dataset to ensure data cleansing and the removal of noise that could affect the system's accuracy. Tokenization, normalization, removing diacritics, removing repeated chars, removing punctuations, removing stop words, removing non-Arabic alphabets, and light stemming are some of the techniques used.

3.3.2. Feature Engineering

The first step in applying machine learning classification is to extract suitable features that help identify spam tweets and to represent them as a feature vector (transform them into numerical feature vectors). In this step, three categories of features are extracted as shown in Table 2: Content features that represent the text included in the tweet, interaction features that describe the tweet spreading between users, and finally, user features that represent the users that propagate the spam text.

- Content Features:
  - TF-IDF: Some language features are used, such as Term-Frequency-based (TF-IDF), which is the most popular feature extraction method. This method typically represents each sentence as a vector of term frequencies (TF) and assigns a score for each word in the text based on the number of times its occurrence and how likely it can be found in texts. (TF-IDF) would show the relative importance of a term in a document compared to other words in the corpus.
  - Hashtag count and URLs count: These features can help us detect spam profiles because they tend to share a lot of spam tweets with URLs and hashtags as an advertising strategy. These values should be higher for the spammer than for the regular users [66].
  - Spam words: The number of spam words in tweets should be higher in spam profiles than in regular profiles. As a result, this feature is important in detecting spam tweets, and is calculated based on the spam words dictionary. This dictionary is constructed based on spam tweets in our labeled dataset, and only the

most common and relevant words were selected from the resulting list, which was manually filtered [66].

- Interaction Features:
  - Tweets duplication count: As noticed, the spammers frequently repeat the same tweet to gain the attention of users to its content. Therefore, we believe that this feature is critical to know whether content is spam or not [66].
  - Retweet count: This feature can be used to distinguish between spam and non-spam tweets because spam tweets are not usually retweeted [66].
  - Mention count: This feature can help us detect spam profiles because they tend to share a lot of spam tweets with mentions as an advertisement strategy. These values should be greater for the spammer than for the regular users [66].
  - Favorite count: Spam tweets are usually not marked as favorites. This value should be higher for the regular than the spammer users [66].
- User Features:
  - User followers and user friends count: These features can help us detect spam profiles. If the result of the ratio between followers and friends is too small, then the probability of being a spam user will increase [66].
  - User description length: Number of characters in the user description. If the result is zero, then the probability of being a spam user will increase.

Table 2. Spam Detection features.

	Feature Name	Description
Content Features	TF-IDF	Multiplication of TF and IDF scores whereas TF is a scoring of the frequency of the word in the current document and IDF is a scoring of how rare the word is across documents. *TF = (Frequency of a word in the document) / (Total words in the document)* *IDF = Log ((Total number of docs) / (Number of docs containing the word)).*
	Spam words	Number of spam words in the tweet
	Hashtag counts	Number of hashtags in the tweet
	URLs counts	Number of URLs in the tweet
Interaction features	Retweet counts	Number of retweets for the tweet
	Mention counts	Number of mentions in the tweet
	Duplication counts	The count of tweet text duplication
	Favorite count	Number of favorites for the tweet
Account features	Followers count	Number of followers of this Twitter user
	Friends count	Number of *followings/friends* of this Twitter user
	Description length	Number of characters in the user description

3.3.3. Utilizing Machine Learning Techniques

To detect spam content, a variety of machine learning classifiers have been used. Three classifiers are trained based on the extracted features to evaluate the effectiveness of these features in identifying spam content. Naive Bayes, Logistic Regression, and LinearSVC are the used classifiers. These classifiers were selected because they are widely used as a baseline in the previous works for Arabic spam detection. The outcomes of the experiment are discussed in Section 4.

4. Experiments

Three experiments have been conducted to evaluate the proposed framework's accuracy. The experiments test the effect of the features mentioned in Table 2 as well as the data augmentation on the performance of the resulting models.

Using our datasets described in the data preparation Section 3.1, we ran a series of classification experiments to identify spam tweets. For the test data sets, the classification results were evaluated using accuracy, precision, recall, and F1-measure for the spam (0)/non-spam (1) classes, as well as AUC (Area Under the receiver operating characteristic Curve (ROC)), and macro-averaged F1, which averages F1-measure for the spam and non-spam classes to account for the class imbalance.

Three different classifiers were used in our experiments: Naive Bayes, Logistic Regression Classifier, and LinearSVC with 10-fold cross-validation.

4.1. Experiment 1: Non-Augmented Dataset with TF-IDF Feature

Table 3 and Figure 4 show the obtained results for the non-augmented dataset using only the TF-IDF feature. These findings showed that the Macro F1 score for NB, LR, and LinearSVC is about 58%. The difference was small, only about 0.01%. As for accuracy, the LinearSVC surpasses the other tested classifiers with 91.8% of correctly classified instances. LR Accuracy is 89.3%. The lowest classifier is NB, with an accuracy of 89.1%. The accuracy range was between 92% and 89%. In terms of AUC, LR performs 75% better than the other tested classifiers, and SVC has an AUC of 74%, while NB has the lowest AUC of 59%.

Table 3. Non-augmented dataset–Experiment 1 confusion matrix.

Algorithm	Non-Augmented Dataset						
	Class	Precision	Recall	F1	Macro-F1	Accuracy	AUC
NB	0 1	0.22 0.96	0.29 0.93	0.20 0.94	0.57	0.891	0.59
LR	0 1	0.24 0.96	0.31 0.93	0.22 0.94	0.58	0.893	0.75
SVC	0 1	0.12 1.00	0.55 0.92	0.18 0.96	0.57	0.918	0.74

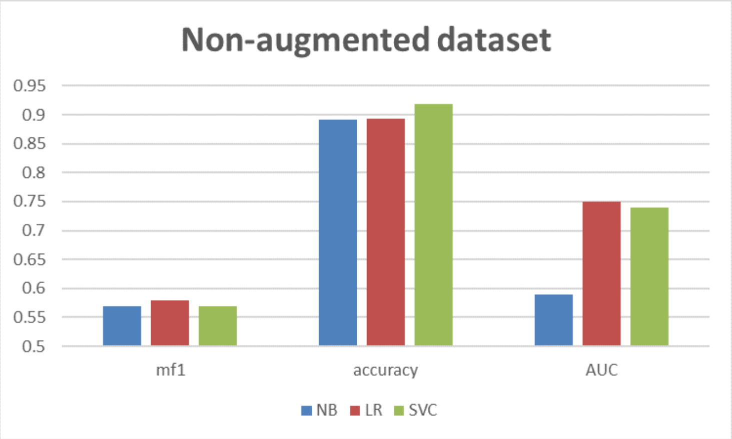


Figure 4. Plot of non-augmented dataset results (Experiment 1).

4.2. Experiment 2: Non-Augmented Dataset with All Features

This experiment aims to identify the effect of using all features in the classification process. Therefore, all the features mentioned in Table 2 are utilized with all three classifiers. After several experiments, the following features gave the best results: TF-IDF, spam words, hashtag counts, URL counts, retweet counts, mentioned counts, and Tweet duplication counts.

Table 4 and Figure 5 show the obtained results for the non-augmented dataset with all features. These findings indicated that the LR classifier had the highest Macro F1 score of 65%. LinearSVC Macro F1 is 61% and the lowest classifier is NB, with Macro F1 of 54%. In addition, the LinearSVC surpasses the other tested classifiers with 92.3% of correctly classified instances. While LR's Accuracy is 92.2%, and the lowest classifier is NB, with an accuracy of 74%. The accuracy range was between 92.3% and 74%. In terms of AUC, LR performs 79% better than the other tested classifiers, and SVC has an AUC of 77%, whereas NB has the lowest AUC of 63%.

Table 4. Non-augmented dataset with all features—Experiment 2 confusion matrix.

Algorithm	**Non-Augmented Dataset-All Features**						
	Class	**Precision**	**Recall**	**F1**	**Macro-F1**	**Accuracy**	**AUC**
NB	0 1	0.52 0.76	0.16 0.95	0.25 0.84	0.54	0.74	0.63
LR	0 1	0.24 0.98	0.59 0.93	0.33 0.96	0.65	0.922	0.79
SVC	0 1	0.16 0.99	0.65 0.93	0.26 0.96	0.61	0.923	0.77

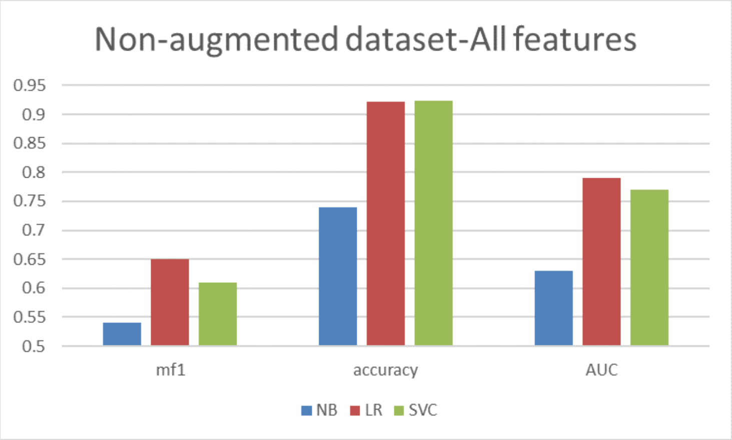


Figure 5. Plot of the non-augmented dataset with all feature results (Experiment 2).

4.3. Experiment 3: Augmented Dataset

In our dataset, 5 k tweets were randomly selected for the labeling process, leading to 420 spam tweets and 4580 non-spam tweets. Using the data augmentation process in Section 3.2, the number of spam tweets were doubled four times, resulting in 1648 spam tweets.

Table 5 and Figure 6 show the obtained results for the augmented dataset. These findings showed that experimenting with data augmentation achieved the maximum macro F1 score value of 89% for the LinearSVC classifier. While the LR classifier Macro F1 is 87%, and the lowest classifier is NB, with Macro F1 of 85%. In addition, the LinearSVC surpasses the other tested classifiers with 96% of the AUC score. While LR's AUC score is 95%, and the lowest classifier is NB, with an AUC score of 85%. In terms of accuracy, the LinearSVC surpasses the other tested classifiers with 92.1% of correctly classified instances. While LR's Accuracy is 91.3%, and the lowest classifier is NB, with an accuracy of 88.5%. The accuracy range was between 92.1% and 88.5%.

Table 5. Augmented dataset—Experiment 3 confusion matrix.

Algorithm	Augmented Dataset						
	Class	Precision	Recall	F1	Macro-F1	Accuracy	AUC
NB	0 1	0.78 0.92	0.78 0.92	0.78 0.92	0.85	0.885	0.85
LR	0 1	0.81 0.95	0.85 0.93	0.83 0.94	0.87	0.913	0.95
SVC	0 1	0.76 0.98	0.93 0.92	0.84 0.95	0.89	0.921	0.96

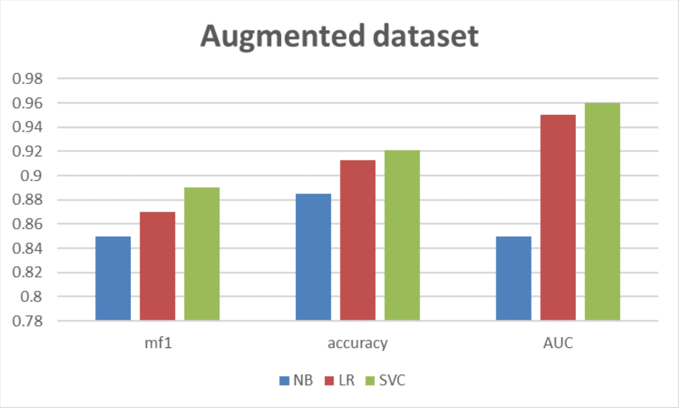


Figure 6. Plot of augmented dataset results (Experiment 3).

5. Discussion

The research framework presented in this paper applied different methods to improve Arabic spam detection on social networks, considering Twitter as a case study. As shown in the first experiment Section 4.1 which follows the current state of the art of research work in the field of Arabic spam detection, only one of the content features (TF/IDF) was used with different classifiers on the extracted dataset (this is used as a baseline method to be compared with the next experiments). The best classifier tends to predict true non-spam rather than true spam. With recall rates of 29%, 31%, and 55% for NB, LR, and SVC, respectively, which is not good compared to the current research work. Therefore, in the second experiment Section 4.2, more features are considered for different categories, such as content, interaction, and user features. This led to predicting more true spam tweets. The recall rates for the correct classification of the spam class increased in LR and SVC, with rates of 59% and 65%, respectively, and the macro F1 score of the LR and SVM classifiers improved by approximately 4% to 7%, while the AUC of the three classifiers improved by approximately 3% to 4 %.

The main problem when applying those classifiers was the unbalancing between spam and non-spam tweets. Therefore, we considered data augmentation to increase the number of spammed tweets, resulting in 6228 total tweets. Table 5 shows the results achieved by experiment Section 4.3, which reveals that adding an augmented dataset outperforms the two previous experiments, and achieved the maximum macro F1 score value of 89%, an accuracy value of 92%, and an AUC score value of 96%. The results achieved outperform the work [53], which already targets Arabic religious hate speech classification, which is part of the spam content categories that we use. This work is closest to our work considering that they used the same classifiers and an Arabic dataset of similar size, where they reached 79% accuracy and 84% AUC. Another previous work [50] that already targets Arabic spam tweets detection using the same classifiers and considering content features, the proposed work outperforms that work, and for a fair evaluation, their dataset, which contains 3503 tweets, is less than our dataset, where they achieved an accuracy rate of 87%.

Unlike the non-augmented dataset experiments, the confusion matrix of LinearSVC for the augmented dataset tends to predict true spam and improve the percentage of detected spam tweets. In addition, we have noticed that the data augmentation technique is effective, bringing a consistent improvement to the baseline classifiers. The spam class correct classification has increased and is close to the non-spam class rates, with recall rates of 78%, 85%, and 93% for NB, LR, and SVC, respectively. As a result, the Macro F1 score of all classifiers is improved by approximately 28% to 32% and up to 38% in spam class recall, and the AUC of all classifiers is improved by around 20% to 26%. Figure 7 shows the ROC curve analysis of the three experiments.

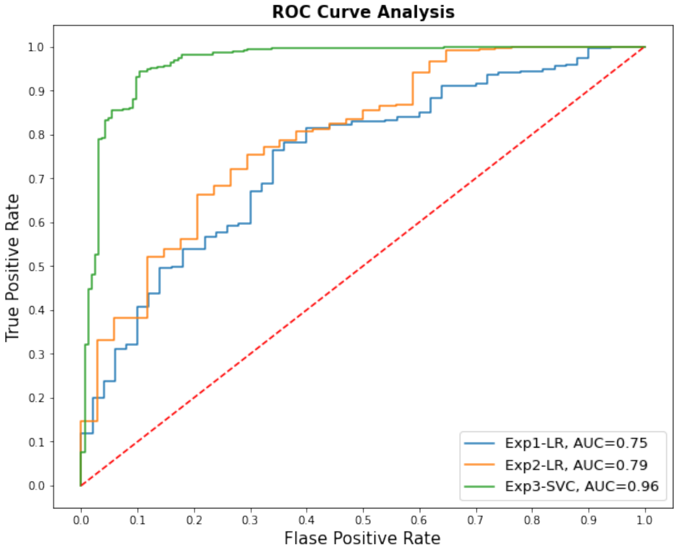


Figure 7. Plot of the three experiments ROC curve analysis.

6. Conclusions

In conclusion, we set out to investigate the problem of online Arabic spam detection on Twitter, motivated by the scarcity of previous studies that address the problem of class imbalance in social media platforms, as well as the scarcity of data augmentation techniques for Arabic text.

We first tried to improve the performance of classifiers as much as possible by adding more features that help build better models for spam detection, paying particular attention to the recall of minority classes. We also showed that utilizing pre-trained neural embedding vectors on large corpora, specifically the Word2Vec (Ara2vec) model pre-trained on Arabic tweets, was fit for the embedding layer, introducing word context that was not present in the limited datasets.

The first improvement presented in experiment Section 4.2, which employed content, interaction, and user features, increased Macro F1 of the best classifier (LR) from 58% to 65%. The added features also improved the recall of the spam class from 55% to 65%, both achieved by SVC, indicating that the selected features were able to improve the performance of our models.

The second improvement presented in experiment Section 4.3 was the proposed augmentation method, which achieved an improvement of 24% (from 65% to 89%) for the Macro F1 score, in addition to an increase of 28% (from 65% to 93%) in the spam class recall compared to the non-augmented with added features baselines. LinearSVC was the best performing classifier, which surpasses the other tested classifiers with 92.1% of correctly classified instances, a Macro F1 score value of 89%, and an AUC score value of 96%.

We found that we can use these pre-trained embeddings in a vector similarity-based word replacement technique to augment the dataset within the text classification framework, reducing class imbalance proportions, and encouraging the model to capture the context of spam tweets rather than depending on the manual labeling of spam tweets.

We believe that various approaches are worth studying for future work on data augmentation for Arabic spam detection tweets. We believe that replacement strategies could be further automated by training a model that predicts which words should be replaced and learning word-specific similarity thresholds rather than a global one. Although the generative method we provided in this research achieved a good improvement over the data augmentation method, we believe that using contextual data augmentation, such as BERT, for data augmentation of Arabic text can be very promising. In addition, it would be interesting to show how the data augmentation strategies we described perform when using deep learning techniques or NLP applications other than the spam detection domain.

Author Contributions: Conceptualization, A.M.A., A.E. and C.A.; data curation, A.M.A.; formal analysis, A.M.A.; investigation, A.M.A., A.E. and C.A.; methodology, A.M.A., A.E. and C.A.; resources, A.M.A.; software, A.M.A.; validation, A.M.A., A.E. and C.A.; writing—original draft preparation, A.M.A.; writing—review and editing, A.E. and C.A.; supervision, A.E. and C.A.; visualization, A.M.A., A.E., and C.A.; work administration, A.E. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Available upon request.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AraBERT	Arabic Bidirectional Encoder Representations from Transformers
AUC	Area Under the receiver operating characteristic Curve (ROC)
BERT	Bidirectional Encoder Representations from Transformers
CBOW	Continuous Bag of Words Model
DASD	Data Augmentation for Arabic Spam Detection
DT	Decision Trees
KNN	k-Nearest Neighbours
LR	Logistic Regression
NB	Naive Bayes
NLP	Natural Language Processing
RF	Random Forest
RNN	Recurrent Neural Network
SMO	Social Media Optimization
SVM	Support Vector Machine
TF-IDF	Term Frequency-Inverse Document Frequency

References

1. Masood, F.; Almogren, A.; Abbas, A.; Khattak, H.A.; Din, I.U.; Guizani, M.; Zuair, M. Spammer detection and fake user identification on social networks. *IEEE Access* **2019**, *7*, 68140–68152. [CrossRef]
2. Liu, N.; Hu, X. Spam Detection on Social Networks. In *Encyclopedia of Social Network Analysis and Mining*; Alhajj, R., Rokne, J., Eds.; Springer: New York, NY, USA, 2018; pp. 2851–2859. [CrossRef]
3. Benevenuto, F.; Magno, G.; Rodrigues, T.; Almeida, V. Detecting spammers on twitter. In Proceedings of the Collaboration, Electronic Messaging, Antiabuse and Spam Conference (CEAS), Redmond, WA, USA, 13–14 July 2010; Volume 6, p. 12.
4. Shen, H.; Liu, X.; Zhang, X. Boosting Social Spam Detection via Attention Mechanisms on Twitter. *Electronics* **2022**, *11*, 1129. [CrossRef]
5. Jain, G.; Sharma, M.; Agarwal, B. Spam detection in social media using convolutional and long short term memory neural network. *Ann. Math. Artif. Intell.* **2019**, *85*, 21–44. [CrossRef]

6. Sarker, I.H. Machine learning: Algorithms, real-world applications and research directions. *SN Comput. Sci.* **2021**, *2*, 1–60. [CrossRef]
7. Huy, D.T.N.; Le, T.H.; Hang, N.T.; Gwoździewicz, S.; Trung, N.D.; Van Tuan, P. Further researches and discussion on machine learning meanings-and methods of classifying and recognizing users gender on internet. *Adv. Mech.* **2021**, *9*, 1190–1204.
8. Wong, C. Analyzing Easy Data Augmentation Techniques for Text Classification. Ph.D. Thesis, Harvard College, Cambridge, MA, USA, 2021.
9. Anaby-Tavor, A.; Carmeli, B.; Goldbraich, E.; Kantor, A.; Kour, G.; Shlomov, S.; Tepper, N.; Zwerdling, N. Do not have enough data? Deep learning to the rescue! In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 7383–7390.
10. Wang, W.Y.; Yang, D. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 2557–2563.
11. Li, B.; Hou, Y.; Che, W. Data augmentation approaches in natural language processing: A survey. *AI Open* **2022**, *3*, 71–90. [CrossRef]
12. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
13. Tran, T.; Pham, T.; Carneiro, G.; Palmer, L.; Reid, I. A bayesian data augmentation approach for learning deep models. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 2794–2803. .
14. Al-Dhabyani, W.; Gomaa, M.; Khaled, H.; Aly, F. Deep learning approaches for data augmentation and classification of breast masses using ultrasound images. *Int. J. Adv. Comput. Sci. Appl* **2019**, *10*, 1–11. [CrossRef]
15. Salamon, J.; Bello, J.P. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Process. Lett.* **2017**, *24*, 279–283. [CrossRef]
16. Hannun, A.; Case, C.; Casper, J.; Catanzaro, B.; Diamos, G.; Elsen, E.; Prenger, R.; Satheesh, S.; Sengupta, S.; Coates, A.; et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv* **2014**, arXiv:1412.5567.
17. Jaitly, N.; Hinton, G.E. Vocal tract length perturbation (VTLP) improves speech recognition. In Proceedings of the ICML Workshop on Deep Learning for Audio, Speech and Language, Atlanta, GA, USA, 16–21 June 2013; Volume 117, p. 21.
18. Tzirakis, P.; Trigeorgis, G.; Nicolaou, M.A.; Schuller, B.W.; Zafeiriou, S. End-to-end multimodal emotion recognition using deep neural networks. *IEEE J. Sel. Top. Signal Process.* **2017**, *11*, 1301–1309. [CrossRef]
19. Gao, J. Data Augmentation in Solving Data Imbalance Problems, Master's Thesis. KTH, School of Electrical Engineering and Computer Science (EECS), Stockholm, Sweden, 2020. Available online: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-289208.
20. Bayer, M.; Kaufhold, M.A.; Reuter, C. A survey on data augmentation for text classification. *arXiv* **2021**, arXiv:2107.03158.
21. Mohammed, A.; Kora, R. Deep learning approaches for Arabic sentiment analysis. *Soc. Netw. Anal. Min.* **2019**, *9*, 52. [CrossRef]
22. Duwairi, R.; Abushaqra, F. Syntactic-and morphology-based text augmentation framework for Arabic sentiment analysis. *Peerj Comput. Sci.* **2021**, *7*, e469. [CrossRef]
23. Sabty, C.; Omar, I.; Wasfalla, F.; Islam, M.; Abdennadher, S. Data augmentation techniques on arabic data for named entity recognition. *Procedia Comput. Sci.* **2021**, *189*, 292–299. [CrossRef]
24. Feng, S.Y.; Gangal, V.; Wei, J.; Chandar, S.; Vosoughi, S.; Mitamura, T.; Hovy, E. A survey of data augmentation approaches for nlp. *arXiv* **2021**, arXiv:2105.03075.
25. Wei, J.; Huang, C.; Vosoughi, S.; Cheng, Y.; Xu, S. Few-shot text classification with triplet networks, data augmentation, and curriculum learning. *arXiv* **2021**, arXiv:2103.07552.
26. Yoo, K.M.; Park, D.; Kang, J.; Lee, S.W.; Park, W. Gpt3mix: Leveraging large-scale language models for text augmentation. *arXiv* **2021**, arXiv:2104.08826.
27. Peng, W.; Huang, C.; Li, T.; Chen, Y.; Liu, Q. Dictionary-based data augmentation for cross-domain neural machine translation. *arXiv* **2020**, arXiv:2004.02577.
28. Xia, M.; Kong, X.; Anastasopoulos, A.; Neubig, G. Generalized data augmentation for low-resource translation. *arXiv* **2019**, arXiv:1906.03785.
29. Pasunuru, R.; Celikyilmaz, A.; Galley, M.; Xiong, C.; Zhang, Y.; Bansal, M.; Gao, J. Data augmentation for abstractive query-focused multi-document summarization. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2021), Online, 2–9 February 2021; pp. 13666–13674.
30. Asai, A.; Hajishirzi, H. Logic-guided data augmentation and regularization for consistent question answering. *arXiv* **2020**, arXiv:2004.10157.
31. Zhang, R.; Yu, Y.; Zhang, C. Seqmix: Augmenting active sequence labeling via sequence mixup. *arXiv* **2020**, arXiv:2010.02322.
32. Yu, T.; Wu, C.S.; Lin, X.V.; Wang, B.; Tan, Y.C.; Yang, X.; Radev, D.; Socher, R.; Xiong, C. GraPPa: Grammar-augmented pre-training for table semantic parsing. *arXiv* **2020**, arXiv:2009.13845.
33. Wan, Z.; Wan, X.; Wang, W. Improving grammatical error correction with data augmentation by editing latent representation. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 8–13 December 2020; pp. 2202–2212.
34. Fernández, A.; Garcia, S.; Herrera, F.; Chawla, N.V. SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *J. Artif. Intell. Res.* **2018**, *61*, 863–905. [CrossRef]

35. Cai, H.; Chen, H.; Song, Y.; Zhang, C.; Zhao, X.; Yin, D. Data manipulation: Towards effective instance learning for neural dialogue generation via learning to augment and reweight. *arXiv* **2020**, arXiv:2004.02594.
36. Barushka, A.; Hajek, P. Review spam detection using word embeddings and deep neural networks. In Proceedings of the IFIP International Conference on Artificial Intelligence Applications and Innovations, Crete, Greece, 24–26 May 2019; pp. 340–350.
37. Jain, N.; Kumar, A.; Singh, S.; Singh, C.; Tripathi, S. Deceptive reviews detection using deep learning techniques. In Proceedings of the International Conference on Applications of Natural Language to Information Systems, Salford, UK, 26–28 June 2019; pp. 79–91.
38. Erşahin, B.; Aktaş, Ö.; Kılınç, D.; Akyol, C. Twitter fake account detection. In Proceedings of the 2017 International Conference on Computer Science and Engineering (UBMK), Samsun, Turkey, 11–15 September 2019; pp. 388–392.
39. Gharge, S.; Chavan, M. An integrated approach for malicious tweets detection using NLP. In Proceedings of the 2017 International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, India, 10–11 March 2017; pp. 435–438.
40. Concone, F.; De Paola, A.; Re, G.L.; Morana, M. Twitter analysis for real-time malware discovery. In Proceedings of the 2017 AEIT International Annual Conference, Cagliari, Italy, 20–22 September 2017; pp. 1–6.
41. Chen, C.; Wang, Y.; Zhang, J.; Xiang, Y.; Zhou, W.; Min, G. Statistical features-based real-time detection of drifted Twitter spam. *IEEE Trans. Inf. Forensics Secur.* **2016**, *12*, 914–925. [CrossRef]
42. Buntain, C.; Golbeck, J. Automatically identifying fake news in popular twitter threads. In Proceedings of the 2017 IEEE International Conference on Smart Cloud (SmartCloud), New York, NY, USA, 3–5 November 2017; pp. 208–215.
43. Mateen, M.; Iqbal, M.A.; Aleem, M.; Islam, M.A. A hybrid approach for spam detection for Twitter. In Proceedings of the 2017 14th International Bhurban Conference on Applied Sciences and Technology (IBCAST), Islamabad, Pakistan, 10–14 January 2017; pp. 466–471.
44. Eshraqi, N.; Jalali, M.; Moattar, M.H. Detecting spam tweets in Twitter using a data stream clustering algorithm. In Proceedings of the 2015 International Congress on Technology, Communication and Knowledge (ICTCK), Mashhad, Iran, 11–12 November 2015; pp. 347–351.
45. Gupta, A.; Kaushal, R. Improving spam detection in online social networks. In Proceedings of the 2015 International Conference on Cognitive Computing and Information Processing (CCIP), Noida, India, 3–4 March 2015; pp. 1–6.
46. Chen, C.; Zhang, J.; Xie, Y.; Xiang, Y.; Zhou, W.; Hassan, M.M.; AlElaiwi, A.; Alrubaian, M. A performance evaluation of machine learning-based streaming spam tweets detection. *IEEE Trans. Comput. Soc. Syst.* **2015**, *2*, 65–76. [CrossRef]
47. Stafford, G.; Yu, L.L. An evaluation of the effect of spam on twitter trending topics. In Proceedings of the 2013 International Conference on Social Computing, Washington, DC, USA, 8–14 September 2013; pp. 373–378.
48. Mubarak, H.; Abdelali, A.; Hassan, S.; Darwish, K. Spam detection on arabic twitter. In Proceedings of the International Conference on Social Informatics, Pisa, Italy, 8 October 2020; pp. 237–251.
49. Mataoui, M.; Zelmati, O.; Boughaci, D.; Chaouche, M.; Lagoug, F. A proposed spam detection approach for Arabic social networks content. In Proceedings of the 2017 International Conference on Mathematics and Information Technology (ICMIT), Adrar, Algiers, 4–5 December 2017; pp. 222–226.
50. Al-Azani, S.; El-Alfy, E.S.M. Detection of arabic spam tweets using word embedding and machine learning. In Proceedings of the 2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Sakhier, Bahrain, 18–20 November 2018; pp. 1–5.
51. Abozinadah, E.A.; Mbaziira, A.V.; Jones, J. Detection of abusive accounts with Arabic tweets. *Int. J. Knowl. Eng.-IACSIT* **2015**, *1*, 113–119. [CrossRef]
52. Alshehri, A.; El Moatez Billah Nagoudi, H.A.; Abdul-Mageed, M. Think before your click: Data and models for adult content in arabic twitter. In Proceedings of the TA-COS 2018: 2nd Workshop on Text Analytics for Cybersecurity and Online Safety, Miyazaki, Japan, 7–12 May 2018; Volume 15.
53. Albadi, N.; Kurdi, M.; Mishra, S. Are they our brothers? analysis and detection of religious hate speech in the arabic twittersphere. In Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Barcelona, Spain, 28–31 August 2018; pp. 69–76.
54. Hassan, S.; Samih, Y.; Mubarak, H.; Abdelali, A. ALT at SemEval-2020 task 12: Arabic and English offensive language identification in social media. In Proceedings of the Fourteenth Workshop on Semantic Evaluation, Barcelona, Spain, 12–13 December 2020; pp. 1891–1897.
55. Hassan, S.; Samih, Y.; Mubarak, H.; Abdelali, A.; Rashed, A.; Chowdhury, S.A. ALT Submission for OSACT Shared Task on Offensive Language Detection. In Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, Marseille, France, 12 May 2020; pp. 61–65.
56. Mubarak, H.; Darwish, K. Arabic offensive language classification on twitter. In Proceedings of the International Conference on Social Informatics, Doha, Qatar, 18–21 November 2019; pp. 269–276.
57. Saeed, R.M.; Rady, S.; Gharib, T.F. An ensemble approach for spam detection in Arabic opinion texts. *J. King Saud-Univ.-Comput. Inf. Sci.* **2022**, *34*, 1407–1416. [CrossRef]
58. Abu Hammad, A.S. An Approach for Detecting Spam in Arabic Opinion Reviews. *Int. Arab. J. Inf. Technol.* **2015**, *12*, 9–16.
59. Najadat, H.; Alzubaidi, M.A.; Qarqaz, I. Detecting Arabic spam reviews in social networks based on classification algorithms. *Trans. Asian-Low-Resour. Lang. Inf. Process.* **2021**, *21*, 1–13. [CrossRef]

60. Alharbi, A.R.; Aljaedi, A. Predicting rogue content and Arabic spammers on twitter. *Future Internet* **2019**, *11*, 229. [CrossRef]
61. El-Mawass, N.; Alaboodi, S. Detecting Arabic spammers and content polluters on Twitter. In Proceedings of the 2016 Sixth International Conference on Digital Information Processing and Communications (ICDIPC), Beirut, Lebanon, 21–23 April 2016; pp. 53–58.
62. Al-Khalifa, H.S. On the analysis of twitter spam accounts in Saudi Arabia. *Int. J. Technol. Diffus. (IJTD)* **2015**, *6*, 46–60. [CrossRef]
63. Soliman, A.B.; Eissa, K.; El-Beltagy, S.R. Aravec: A set of arabic word embedding models for use in arabic nlp. *Procedia Comput. Sci.* **2017**, *117*, 256–265. [CrossRef]
64. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
65. Madukwe, K.J.; Gao, X.; Xue, B. Token replacement-based data augmentation methods for hate speech detection. *World Wide Web* **2022**, *25*, 1129–1150. [CrossRef]
66. Herzallah, W.; Faris, H.; Adwan, O. Feature engineering for detecting spammers on Twitter: Modelling and analysis. *J. Inf. Sci.* **2018**, *44*, 230–247. [CrossRef]