Hidden Sentiments: The Impact of Low-Level Adversarial Perturbations on Arabic Sentiment Analysis Services

Abdelrahman H. Hefny
Carleton University
Ottawa, Canada
abdelrahmanhefny@cmail.carleton.ca

Abstract

Sentiment analysis is one of the most popular applications of supervised machine learning for natural language processing. A common approach for obtaining a dataset to train sentiment analysis models is to extract user posts and comments from social media and other online platforms. However, this content is subject to various types of perturbations that go beyond the target of common preprocessing techniques and may impact the models' performance. In this paper, a set of six popular corpora used in Arabic sentiment analysis research is analyzed to identify common patterns of character-level perturbations. The samples of three selected corpora were then used to test the performance of the online sentiment analysis services offered by three public cloud providers. This test is done using a clean version of each dataset and four other versions, each perturbed using a different technique. Empirical results indicate that no single sentiment analysis service is superior to others in all cases, and all three services are vulnerable to low-level adversarial attacks which may cause up to a 51% relative drop in macro average F1 score, while maintaining readability.

Keywords: Arabic NLP, Sentiment Analysis, Adversarial Attacks, Cloud Services

1. Introduction

Since the emergence of social media platforms, the amount of user-generated content on the web has been increasing dramatically. This content in the form of text, images, audio, video, and user interactions provides a great source of insights for businesses and decision makers, allowing them to better understand their audiences and make informed decisions. However, with the huge amount of data generated every second, it's not feasible anymore to monitor and analyze this content manually, which makes the use of automated Machine Learning (ML) and data mining techniques a necessity. These techniques can be utilized in many areas. One such area is sentiment analysis of textual content. Sentiment Analysis (SA) is the use of Natural Language Processing (NLP) techniques to analyze sentiments in a given input text, and classify the text as either positive, negative, or neutral. Applications of sentiment analysis include, for example, aggregating user feedback on new products or services, polling public opinions toward a specific subject, predicting stock prices, or forecasting election outcomes (Medhat et al., 2014).

Currently, the state of the art in sentiment analysis uses ML by training a model on a large dataset containing examples of all possible classes (Jim et al., 2024). The better the quality of training data, the better the performance of the trained model on unseen examples. Researchers working on sentiment analysis are concerned with the choice of preprocessing steps applied to input data, which often involves cleansing and normalization to improve the model's performance. For Arabic, this might include, for example, removing punctuation marks, diacritics, and stop words, stemming words, or normalizing each letter to a single form.

However, as sentiment analysis is often applied to textual content obtained from social media platforms and online stores, inputs might be subject to varying levels of perturbation, as in Figure 1, that alter the text in ways that go beyond the target of simple normalization techniques, resulting in degraded performance of NLP and text mining algorithms including sentiment analysis. Such perturbations require a more advanced level of preprocessing to mitigate their effects which is often neglected in many production ML systems. Furthermore, since research on Arabic sentiment analysis utilizes sample datasets collected from the same sources, these samples are also subject to the same perturbations, and, depending on the dataset size, they might affect the model's performance on the test sample.

These perturbations could be intentional to evade content moderation, work around character limits of some platforms such as X (formerly Twitter), or just create an artistic style, but they could also be accidental. They could be perceptible to humans, such as spelling mistakes, omitted spaces between words, redundant diacritics, or words obfuscated with punctuation symbols. Yet, they could also be imperceptible, such as invisible control characters or characters replaced with others that have similar shapes (homoglyphs).

Text perturbations intended to degrade the performance of NLP models are a form of adversarial attacks against ML systems, which are a major

Proceedings of The 7th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT7) with 5 Shared Tasks, pages 1–13
May 11, 2026.

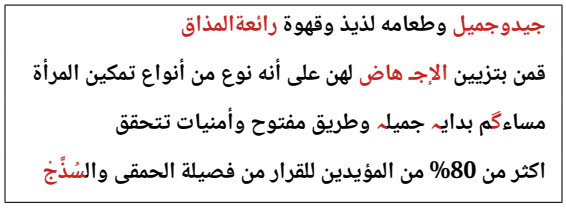

Figure 1: Examples of perturbations in SA corpora.

concern for ML security specialists and an active research area (Zhou et al., 2022). Several recent studies investigated adversarial attacks on computer vision and NLP models (Ren et al., 2020; Xu et al., 2020; Goyal et al., 2023; Costa et al., 2024). However, most adversarial NLP research has focused on English and Latin script languages, while relatively few studies were concerned with Arabic. Nevertheless, the differences between Arabic and Latin script languages make some techniques that are intended for English inapplicable to Arabic and vise versa.

The goal of this research is to investigate the effects of adversarial text perturbations applied at the low (character) level on the performance of SA models. Instead of testing open-source NLP models which are open to retraining, finetuning, and customization, the focus of this work is on production-ready SA services offered by major cloud service providers as part of their Artificial Intelligence (AI) services catalog. These services are often offered as a black box with little or no customizability. However, they provide software developers with easy to access Application Programming Interfaces (APIs) backed by pretrained ML models to utilize in their software applications without having to build or maintain such models by themselves. A client application sends a secure request over the Internet to an API endpoint with one or more attached documents, and the service processes the request and replies with the predicted sentiment of each document. Some services also can predict the sentiment of each sentence in the document, and link sentiments to named entities in the text.

To achieve this goal, a set of corpora commonly used in Arabic SA research is first analyzed to identify common patterns of low-level text perturbations that might affect NLP models performance. Then a subset of these corpora is sampled and used for evaluating the performance of Arabic SA services on three popular cloud platforms. This evaluation is done once on the clean datasets, and once on four versions of the datasets with different types of perturbations.

This paper makes the following contributions:

1. A list of identified low-level text perturbations found in a sample of Arabic SA corpora.
2. A comparison of Arabic sentiment analysis services offered by three popular cloud providers: Amazon Web Services, Microsoft Azure, and Google Cloud Platform, on three different sample corpora covering dialectical and modern standard Arabic.
3. An evaluation of the selected services on different versions of the test samples perturbed using four techniques of low-level adversarial attacks.

This research uses publicly available datasets, and all the code used for analysis and experimental evaluation is also available through the project's code repository.[1] The remainder of this paper is organized as follows. Section 2 provides a brief review of related work. Section 3 describes the research methodology, including the selected cloud services, datasets, and experiment design. Section 4 discusses the main research findings. Finally, Sections 5 highlights the research conclusions.

2. Related Work

The following subsections provide a brief review of the literature on related work under three main topics: Arabic sentiment analysis, adversarial attacks on NLP, and evaluation of cloud-based NLP services, respectively.

2.1. Arabic Sentiment Analysis

Sentiment analysis is one of the most studied applications of NLP and text classification. Traditionally, there have been three main approaches for sentiment analysis: a lexicon-based approach, a supervised ML approach, or a hybrid approach. (Wankhade et al., 2022; Mao et al., 2024) However, with the availability of large training datasets and the recent advances in NLP, Deep Learning (DL), and language models, using ML/DL models has become the state-of-the-art in sentiment analysis for many languages, including Arabic (Jim et al., 2024). Research on Arabic SA has covered many domains, such as movies, hotels, books, mobile applications, vaccines, and political views. As the number of publications on Arabic SA is relatively large, covering them in detail is outside the scope of this paper. However, several recent surveys are available for reference (Oueslati et al., 2020; Obiedat et al., 2021; Matrane et al., 2023).

[1] https://github.com/abdelrahman0101/HiddenSentiments

2.2. Adversarial Attacks on NLP Models

Adversarial attacks against ML models are perturbations applied to the models' inputs in order to hinder their performance. (Xu et al., 2020) These attacks are categorized based on knowledge of the target system into white-box, where the attacker has full knowledge of the model's architecture and access to internal system parameters, and black-box, where they can only get the system's output in response to a given input (Ren et al., 2020). In NLP, they can be further categorized according to the level of perturbation as sentence, word, or character level. These perturbations introduce minor insertions, deletions, or substitutions such that humans are able to perceive the original information, while ML models fail to do so (Alshemali and Kalita, 2020; Qiu et al., 2022; Goyal et al., 2023). Such attacks are known to cause performance degradation in sentiment analysis (Radman and Duwairi, 2025), machine translation (Zhang et al., 2021; Sadrizadeh et al., 2023), offensive language detection (Cooper et al., 2023; Abdellaoui et al., 2024; Berezin et al., 2025), plagiarism detection (Alvi et al., 2017; Creo and Pudasaini, 2025), spam detection (Alajmi et al., 2025), and other tasks (Xie et al., 2022).

Studies on adversarial attacks on NLP models were mostly focused on English and Latin script languages, while a relatively small portion targeted Arabic. For Latin script languages, examples of possible perturbations include intentional typos (Gan et al., 2024), insertion of punctuations (Formento et al., 2023), homo-glyph substitutions (Eger et al., 2019; Cooper et al., 2023), and insertion of invisible formatting characters (Boucher et al., 2022). As for Arabic, examples include misspellings that mimic the mistakes of non-native speakers (Alshemali and Kalita, 2021), substituting dotted characters with their dotless counterparts (Al-Shaibani and Ahmad, 2023), insertion of dots or spaces (Abdellaoui et al., 2024), or diacritical manipulations (Salman et al., 2024; Alshemali, 2025a). Some studies also covered word and sentence level attacks (Alshalan and Rekabdar, 2023; Salman et al., 2024; Alshemali, 2025b).

2.3. Evaluation of cloud-based NLP services

Cloud-based AI services have been the target of several recent studies aiming to compare and evaluate their performance on specific benchmarks. Examples are content moderation services used for filtering undesirable content (AlDahoul et al., 2023; Zheng et al., 2024), and natural language understanding (NLU) services used for building chatbots and conversational agents (Braun et al., 2017; Liu et al., 2021). Sentiment analysis services were also the subject of some recent publications (White and Rege, 2020; Ermakova et al., 2023; Ivan et al., 2024; Pawlik, 2025). However, they were also mostly focused on English.

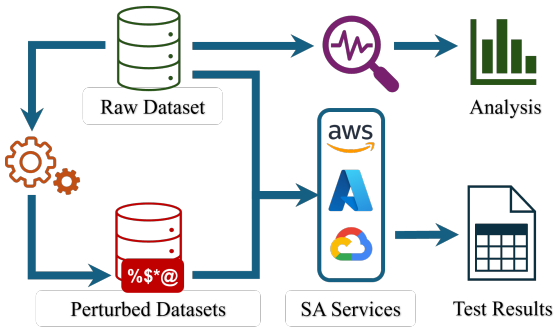

Figure 2: Research process

3. Research Methodology

As depicted in Figure 2, this research involves three parts. The first is an exploratory analysis of six Arabic SA corpora, for the purpose of identifying any existing text perturbations. The second part provides an experimental evaluation of three cloud-based services on samples extracted from three selected corpora containing examples in dialectical and modern standard Arabic. In the final part, the same tests are repeated, but this time on versions of the samples perturbed using four different techniques, and the results are compared to the baseline obtained from the previous experiment.

3.1. Exploratory Analysis of Existing Corpora

The author surveyed existing research on Arabic SA published in the last 10 years and selected a set of six well-cited corpora in the academic literature with varying domains, sources, and dialects, which are listed in Table 1. These corpora were mainly obtained from social media platforms and each record was annotated to indicate its sentiment using either a nominal label or a numeric rating. For nominal labels, at least two labels covering positive and negative sentiments are used. Some labeling schemes also add a neutral/objective label for texts not expressing any sentiments and a mixed label for texts with both sentiments.

Based on the author's knowledge and after reviewing a sample of these corpora, a small Python script was implemented to detect potential signs of text perturbation. The script searches through a given dataset and identifies any instances of the

Corpus	Source	Size	Label Scheme	Main Dialect
ASTD (Nabil et al., 2015)	Twitter	10K	4 classes	Egyptian
Elsahar2015 (ElSahar and El-Beltagy, 2015)	Multiple Websites	34K	3 classes	Multiple
AraSenTi-Tweet (Al-Twairesh et al., 2017)	Twitter	18K	4 classes	Saudi
OCLAR (Al Omari et al., 2019)	Google Maps, Zomato	4K	5-points rating	Lebanese
NADI 2022 (Abdul-Mageed et al., 2022)	Twitter	5K	3 classes	Multiple
Mawqif (Alturayeif et al., 2022)	Twitter	4K	3 classes	Multiple

Table 1: Selected Arabic sentiment analysis corpora

following potential perturbation and tags them for further review.

Non-standard characters. Including any Unicode characters not available on standard Arabic, English, or French keyboards, which are commonly used by Arab users, while excluding Unicode blocks for emoticons, pictographs, and common symbols.

Overly long words. This is to detect posts containing two or more consecutive words concatenated together. The longest known word in classic Arabic lexicons is 15 letters long, but for the purposes of this research a limit of 10 letters was sufficient to identify potential instances.

Arabic words split using spaces, punctuation, emoticons, or Latin letters. This is a common obfuscation technique that usually relies on Kashida (elongation) characters to maintain the readability of the word, which could be mostly detected using regular expressions.

No dotted letters. Most Arabic letters are written with dots or hamza above or below. Another obfuscation technique used in social media is to replace Arabic letters with their undotted forms.

The posts identified through this search were then manually reviewed to remove false positives and identify existing patterns of text perturbation.

3.2. Baseline performance

For practical reasons, only three of the surveyed corpora were selected for the experimental evaluation of cloud-based Arabic SA services. These are ASTD, AraSenTi-Tweet, and Mawqif. The three corpora contain samples in Modern Standard Arabic (MSA) in addition to one or more Arabic dialects. However, to work within the free trial limits of the selected services, and to avoid throttling the APIs, only a random sample containing 100 instances of each class was selected for testing.

The selected cloud services are Amazon Comprehend from Amazon Web Services (AWS), Azure AI Language from Microsoft Azure, and Cloud Natural Language from Google Cloud Platform (GCP). These platforms are the most popular public cloud providers, respectively, offering a wide range of services. While Azure and AWS APIs use a 4-class labeling scheme, GCP's uses a numeric score in the continuous range [-1, 1]. Thus, to facilitate comparisons, this range is split evenly into 3 subranges of equal length. Scores between –0.3 and 0.3 inclusive are considered neutral, while lower scores are considered negative and higher scores positive. In addition, for tests involving GCP or the Mawqif corpus, the mixed and neutral classes are considered equivalent. Other sentiment analysis tasks such as stance detection are not covered by this research due limited Arabic support of the tested services at the time of this writing.

3.3. Performance on Adversarial Inputs

After setting a baseline for each service on each dataset. Four different versions of the datasets are created by applying the following low-level perturbations:

- **Homoglyphs**: Arabic letters are replaced with Unicode homoglyphs from other Arabic script languages. To maintain the same level of readability, only homoglyphs without extra dots or sub-glyphs are used.
- **Split words**: Every word longer than 3 letters is split using spaces or punctuation symbols.
- **Positional variants**: Every Arabic letter is replaced with its hard-coded positional variant according to its position in the word: initial, middle, final, or isolated.
- **Diacritics**: Arabic diacritics denoting short vowels are added after every consonant using a simple rule-based algorithm based on word syllables.

This experiment is conducted to test the hypothesis that online SA services are vulnerable to adversarial character-level perturbations, and these perturbations can significantly reduce the services' performance, particularly by causing subjective (positive or negative) documents to be classified as objective (neutral).

4. Results and Discussions

The following subsections list and discuss the findings of each of the three research parts.

4.1. Text Perturbations in Existing Corpora

After examining the sample corpora, several cases of low-level text perturbations were identified. Example posts are listed in Table 6 in the appendix. These perturbations can be categorized into the following seven patterns. Nonetheless, it's possible that other types of perturbations exist in the selected corpora, but they were not identified by the search script or during manual review. Also, contrary to the initial expectations, no posts written in undotted letters were found in the selected corpora.

4.1.1. Positional variants of Arabic letters

The most frequently used non-standard characters in the examined corpora were hard-coded positional/contextual variants of Arabic letters from the Unicode block Arabic Presentation Forms B. These Unicode codepoints are provided for backward compatibility with legacy systems where a different codepoint is used for every form of an Arabic letter depending on its position in the word: initial, middle, final, or isolated. Eight of the top ten non-standard characters in the corpora fall under this category, with an average frequency of 117 posts. Despite being completely different on the binary level, it's easy to map these characters to their canonical forms using normalization tools readily available in standard libraries of many programming languages.[2]

4.1.2. Homoglyphs from other alphabets

Homoglyphs from Arabic script languages such as Persian, or Urdu, were also used in many cases to replace standard Arabic letters. In most cases, they were used for creating a decorative style, but in some cases, they were also used to represent specific sounds used in dialectical Arabic. Examples are Heh Goal (ہ) which was used in 96 posts, Gaf (گ) in 94 posts, and Peh (پ) in 34 posts. Homoglyphs from other writing systems were also found. Examples are Hebrew Vav ( ) and Greek Iota (ι) used to mimic Arabic Alef. Mitigating this kind of perturbation is easier when the text language is known in advance. In such a case, it's a matter of character mapping. However, this mapping is not always one to one as the same homoglyph can be used for different Arabic letters in different positions.

4.1.3. English sounds and Arabizi

Instead of using homoglyphs from other languages, an Arabic letter can be replaced with an English/Latin letter that has a similar sound, or with a digit using the Arabizi mapping. The resulting word contains Arabic letters intermixed with digits or Latin letters. This is different from writing an entire word in English letters or Arabizi, which is outside the scope of this paper. This pattern was found in very few posts where words were perturbed mostly to evade content moderation.

4.1.4. Concatenated words

Although the search was focused on overly long words with a minimum length of 11 letters, it was able to identify several instances of concatenated Arabic words. In some cases, this was likely the result of typos, but in many cases, the first word ended with a terminal letter that does not connect to its following neighbors. This suggests that spaces were deliberately omitted either as a typing habit or to confine with the length limitation imposed by the platform. Mitigating this kind of perturbation requires the use of a spellchecker that supports dialectical Arabic, but some terminal letters that only occur at the end of words such as Taa Marbuta (ة) can be used to determine where a space should be inserted.

4.1.5. Split words

Instead of merging consecutive words, some words were split into two or more parts using extra spaces or punctuation marks, along with Kashida characters to maintain readability. In most detected cases, it separated a word prefix from its stem, which might in fact facilitate keyword matching, but also might hinder morphological or syntactical parsing. Nonetheless, in some cases, it was also used to split a word stem, which is commonly used with words related to sensitive topics that might be flagged by content moderation. Regular expressions can be used to detect many potential cases of this perturbation, but a dialectical spell checker is still needed to mitigate them.

[2] Examples are the *unicodedata* module for Python, and the *java.text.Normalizer* class for Java.

4.1.6. Invisible formatting characters

Formatting and control characters are non-printing characters that have no visible glyphs but can significantly affect how text is displayed and formatted. They can be used to delete characters, reverse the display order, or split a word. Invisible characters found in the corpora include Zero-Width Joiner (43 posts), Right-to-Left Mark (14 posts), and Zero-Width Non-Joiner (11 posts). They were mainly used to enhance the readability of surrounding text, especially when homoglyphs or positional variants are used, or when the text contains emoticons or non-Arabic left-to-right words. However, any adversarial perturbations involving these characters can be easily mitigated using a rule-based approach.

4.1.7. Diacritics

Diacritical marks are an essential part of Arabic script. A subset of these marks used for denoting short vowels in Arabic are available on standard keyboard schemes. These were used in over 3200 posts in the sample corpora. However, other diacritics were also used for decorating text, including ones from other writing systems and the Combining Diacritical Marks block. For normalizing inputs to Arabic NLP models, it's mostly considered safe to simply remove diacritics from Arabic text.

4.2. Baseline Performance

The first experiment was conducted on samples extracted from the three selected corpora. Evaluation includes two sentiment analysis tasks with different numbers of target classes. The first task uses a four-classes scheme to classify an input as Positive, Negative, Neutral, or Mixed. This, however, is only possible with the platforms and datasets that support this scheme. The second task uses a three-classes scheme, by treating the Neutral and Mixed classes as equivalent. Before taking the samples, cleansing and normalization steps were performed to ensure they are mostly free from identified perturbations. First, all posts were normalized to Unicode's Normalization Form KC (NFKC) to map all positional variants to their canonical forms. Then, diacritics and invisible formatting characters were removed. Then, every Taa Marbuta (ة) followed by another letter was separated from it by a space. Finally, detected posts with split words, or unusual characters were manually reviewed to determine whether to drop or keep them, and the samples were taken from the remaining posts. Dropped posts are those containing split word stems or decorative homoglyphs. The final test samples contain 100 examples of each class selected at random.

Detailed results of testing the three services on each sample dataset are provided in Table 2 for the first task, and Table 3 for the second. The highest score in each row is displayed in bold.

For the first task, AWS achieved a significantly better performance on average than Azure on the AraSenti-Tweet dataset, and slightly better on ASTD. For positive sentiments, Azure consistently achieved better recall while AWS had better precision. The opposite was true for the neutral class where AWS achieved better recall and Azure higher precision. However, both services performed equally well on detecting negative sentiments, but also failed to correctly classify most posts labeled as mixed, with AWS showing a relatively better performance on the AraSenti-Tweet dataset. This indicates that either the classification models used by AWS and Azure were not trained well to detect mixed sentiments, or the posts labeled as mixed in the test sets were mislabeled or ambiguous. In either case, it signals the difficulty of classifying mixed sentiments. A good example of this ambiguity is the use of the religious phrase "لا حول ولا قوة إلا بالله", meaning: *"There is no might or power but by Allah"*, which is often used to express sadness or shock, but could also be seen as an expression of faith or resilience.

For the second task, where the mixed and neutral classes are considered equivalent, none of the three services showed superior performance on all datasets. However, GCP performed better on average on the ASTD and Mawqif datasets, and equally well to AWS on the AraSenti-Tweet dataset. Although Azure performed worse on average than both AWS and GCP, it consistently achieved better recall for the positive class on all three datasets. It's worth noting that since the mixed and neutral classes were merged, the resulting class now has a larger size. Thus, the macro-average F1 score becomes a better indicator of performance.

The relatively low baseline scores could be attributed to several factors: (1) low-quality or incompatible annotations of the tested samples especially for the *mixed* and *neutral* classes, (2) the difficulty of handling various colloquial Arabic dialects included in the corpora, and (3) for practical reasons, some instances of perturbations, particularly concatenated words, were still included in the test samples, which could have impacted the models' performance.

4.3. The Impact of Adversarial Perturbations

In the final set of experiments, the three services are tested on a three-class sentiment analysis task using perturbed versions of the test sets, with the goal of determining how much performance degra-

Corpus	Class	AWS			Azure		
		Precision	Recall	F1	Precision	Recall	F1
ASTD	Positive	0.75	0.52	**0.62**	0.46	0.77	0.58
	Negative	0.49	0.66	0.56	0.48	0.71	**0.57**
	Neutral	0.39	0.75	**0.51**	0.46	0.36	0.40
	Mixed	0.25	0.01	0.02	0.57	0.04	**0.07**
	Macro avg	0.47	0.49	**0.43**	0.49	0.47	0.41
	Accuracy			**0.49**			0.47
AraSenTi-Tweet	Positive	0.65	0.68	**0.66**	0.54	0.85	**0.66**
	Negative	0.68	0.86	**0.76**	0.51	0.78	0.62
	Neutral	0.70	0.83	**0.76**	0.75	0.50	0.60
	Mixed	0.70	0.35	**0.47**	0.44	0.11	0.18
	Macro avg	0.68	0.68	**0.66**	0.56	0.56	0.51
	Accuracy			**0.68**			0.56

Table 2: Baseline performance on 4-class sentiment analysis.

Corp.	Class	AWS			Azure			GCP		
		Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1
ASTD	Positive	0.75	0.52	0.62	0.46	0.77	0.58	0.66	0.65	0.66
	Negative	0.49	0.66	0.56	0.48	0.71	0.57	0.50	0.84	**0.63**
	Neutral	0.64	0.62	**0.63**	0.63	0.27	0.38	0.73	0.49	0.59
	Macro avg	0.63	0.60	0.60	0.52	0.58	0.51	0.63	0.66	**0.62**
	Accuracy			0.61			0.51			**0.62**
AraSenTi-Tweet	Positive	0.65	0.68	0.66	0.54	0.85	0.66	0.64	0.71	**0.67**
	Negative	0.68	0.86	**0.76**	0.51	0.78	0.62	0.61	0.84	0.71
	Neutral	0.78	0.66	**0.71**	0.75	0.34	0.47	0.74	0.56	0.64
	Macro avg	0.70	0.73	**0.71**	0.60	0.66	0.59	0.66	0.70	0.67
	Accuracy			**0.71**			0.58			0.67
Mawqif	Positive	0.89	0.55	0.68	0.63	0.72	0.67	0.82	0.70	**0.76**
	Negative	0.67	0.62	0.65	0.66	0.61	0.63	0.72	0.84	**0.77**
	Neutral	0.51	0.74	**0.60**	0.57	0.52	0.54	0.57	0.56	0.57
	Macro avg	0.69	0.64	0.64	0.62	0.62	0.61	0.70	0.70	**0.70**
	Accuracy			0.64			0.62			**0.70**

Table 3: Baseline performance on 3-class sentiment analysis. Mixed and Neutral sentiments are considered equivalent.

dation occurs due to each method of perturbation. Table 4 shows macro-F1 scores for the classification results of perturbed test sets compared to the baseline obtained in the previous experiment.

All three sentiment analysis services were found to be vulnerable to low-level adversarial attacks. Notably, the most effective technique in degrading the performance of sentiment analysis is the insertion of Arabic diacritics with a 39% relative drop in macro-F1 score on average. AWS was the most vulnerable to this attack with an average drop of 51%, while GCP was the least vulnerable with only a drop of 29%. Homoglyphs and split-words had a similar impact on the macro-F1 score with 22% average drops. As for positional variants, only GCP was vulnerable to this attack with an average drop of 19%. However, since this attack reencodes every Arabic letter in the text but could be easily mitigated using normalization tools, it would be expected that NLP models either fail to recognize the inputs or be able to completely recover it. The results of GCP indicate that examples of text containing positional variants might have been used in the training. Nonetheless, this technique could still be effective even with the use of normalization tools when, for example, spaces between words are omitted.

Based only on the average drop in macro-F1, we cannot conclude that any specific sentiment analysis service is generally more vulnerable than another. While AWS was the most susceptible to Arabic diacritics, GCP was the most suscepti-

		Baseline	Homoglyphs		Diacritics		Pos-variants		Split-words	
		F1	F1	RE	F1	RE	F1	RE	F1	RE
AWS	**ASTD**	0.60	0.4	33%	0.32	47%	0.60	0%	0.49	18%
	AraSenti	0.71	0.54	24%	0.34	52%	0.71	0%	0.57	20%
	Mawqif	0.64	0.41	36%	0.29	55%	0.64	0%	0.53	17%
Azure	**ASTD**	0.51	0.44	14%	0.35	31%	0.51	0%	0.45	12%
	AraSenti	0.59	0.57	3%	0.33	44%	0.59	0%	0.45	24%
	Mawqif	0.61	0.47	23%	0.38	38%	0.62	-2%	0.46	25%
GCP	**ASTD**	0.62	0.5	19%	0.45	27%	0.47	24%	0.47	24%
	AraSenti	0.67	0.55	18%	0.50	25%	0.64	4%	0.50	25%
	Mawqif	0.70	0.52	26%	0.46	34%	0.50	29%	0.48	31%

Table 4: The effects of adversarial perturbations on macro average F1 score. The relative error in macro F1 score as compared to the baseline is indicated as "RE" and darker shades denote larger errors.

ble to split-word attacks based on these experiments. Nevertheless, measuring the effectiveness of the attacks using only the drop in F1-score is not always sufficient. For example, apart from the cases of positional variants, the lowest drop in F1-score was obtained when testing Azure using the homoglyphs version of AraSenti-Tweet. Although the F1 drop was only 3.4%, investigating the detailed metrics reveals that the effect of using homoglyphs was not that simple. While Azure achieved 0.85 recall for the positive class on the baseline, it dropped to only 0.58 with the use of homoglyphs. Furthermore, the precision for the neutral class decreased from 0.75 to only 0.67, and the recall increased from 0.34 to 0.49. This indicates higher tendency to classify text as neutral/mixed due to perturbations. However, as the baseline F1 was already low, the effect of perturbation was not prominent by only measuring the relative drop in the score. In general, adversarial perturbations caused up to 37% reduction in the neutral class precision. More details are provided in Table 5 and examples of perturbed posts and their predicted labels are provided in Table 7 in the appendix.

5. Conclusions

Several forms of low-level perturbations are found in Arabic corpora extracted from social media. Many of these perturbations are often overlooked by researchers and ML engineers building text mining and NLP pipelines. While some of them can be corrected automatically by simple omission or character mapping, others might require using more advanced toolkits or manual review. This work identified seven patterns of character-level perturbations in highly cited Arabic sentiment analysis corpora. Samples from three selected corpora were used to test the performance of sentiment analysis services offered by the three public cloud providers with the largest market shares. The tests measured their performance on clean samples and on four intentionally perturbed versions of the same samples. The results indicate that adversarial character-level perturbations can significantly degrade the performance of Arabic SA models, even those offered by popular cloud vendors. The relative drop in the macro-F1 score reached 51% in some cases, although the degree of perturbation tested was kept as low as possible to maintain the text readability. Higher degrees of perturbation such as the use of non-Arabic diacritics, non-Arabic homoglyphs, or technique combinations are expected to be even more effective in degrading the classification performance.

Based on this research, future work could follow several directions. For example, only four perturbation patterns were investigated in this research. More investigation of the other identified patterns or different parameters of these patterns is also needed. Another direction is to expand the scope of the research to other NLP applications. One such application is offensive language or hate speech detection, where adversarial perturbations are expected to be more prominent. Adversarial perturbations might also impact the performance of other downstream tasks such as speech synthesis or Optical Character Recognition (OCR); when perturbed text is embedded within images.

6. Limitations

This work identified several patterns of character-level perturbations in Arabic corpora extracted from social media. For practical reasons, the findings are based only on a sample of five sentiment analysis corpora commonly cited in research. However, the same analysis could be applied to other existing or new corpora to find other potential patterns. It is also possible to provide a better quantification of each of these patterns using larger test samples, but this will require more ef-

		Baseline	Homoglyphs		Diacritics		Pos-variants		Split-words	
		Neu-P	Neu-P	RE	Neu-P	RE	Neu-P	RE	Neu-P	RE
AWS	**ASTD**	0.64	0.53	17%	0.48	25%	0.64	0%	0.56	13%
	AraSenti	0.78	0.66	15%	0.55	29%	0.78	0%	0.68	13%
	Mawqif	0.51	0.38	25%	0.35	31%	0.51	0%	0.48	6%
Azure	**ASTD**	0.63	0.51	19%	0.56	11%	0.63	0%	0.53	16%
	AraSenti	0.75	0.67	11%	0.55	27%	0.75	0%	0.56	25%
	Mawqif	0.57	0.39	32%	0.41	28%	0.57	0%	0.36	37%
GCP	**ASTD**	0.73	0.56	23%	0.53	27%	0.53	27%	0.54	26%
	AraSenti	0.74	0.58	22%	0.59	20%	0.69	7%	0.55	26%
	Mawqif	0.57	0.41	28%	0.40	30%	0.40	30%	0.40	30%

Table 5: The effects of adversarial perturbations on the precision of the neutral class in 3-class sentiment analysis. The relative error in precision as compared to the baseline is indicated as "RE".

forts for manual labeling and analysis. The implementation of tested perturbations also has some limitations. While the diacritics and positional variant perturbations were based on deterministic rules, homoglyphs and split words involved using randomized operations to select substitutes or positions. This randomization may produce slightly different results each time it is applied. Furthermore, evaluating the performance of cloud-based sentiment analysis services is beneficial to software engineers looking to incorporate an off-the-shelf tool into their AI applications. However, it should be noted that these services are in continuous development, and the results presented here might change in the future as the ML models backing these services change or evolve. The reported results are based on experiments conducted in late June 2025.

7. Bibliographical References

Israe Abdellaoui, Anass Ibrahimi, Mohamed Amine El Bouni, Asmaa Mourhir, Saad Driouech, and Mohamed Aghzal. 2024. Investigating offensive language detection in a low-resource setting with a robustness perspective. *Big Data and Cognitive Computing*, 8(12).

Muhammad Abdul-Mageed, Chiyu Zhang, Abdelrahim Elmadany, Houda Bouamor, and Nizar Habash. 2022. NADI 2022: The third nuanced Arabic dialect identification shared task. In *Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 85–97, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Marwan Al Omari, Moustafa Al-Hajj, Nacereddine Hammami, and Amani Sabra. 2019. Sentiment classifier: Logistic regression for arabic services' reviews in lebanon. In *2019 International Conference on Computer and Information Sciences (ICCIS)*, pages 1–5.

Maged S Al-Shaibani and Irfan Ahmad. 2023. Dotless representation of arabic text: Analysis and modeling. *arXiv preprint arXiv:2312.16104*.

Nora Al-Twairesh, Hend Al-Khalifa, AbdulMalik Al-Salman, and Yousef Al-Ohali. 2017. Arasenti-tweet: A corpus for arabic sentiment analysis of saudi tweets. *Procedia Computer Science*, 117:63–72. Arabic Computational Linguistics.

Anwar Alajmi, Imtiaz Ahmad, and Ameer Mohammed. 2025. Evaluating the adversarial robustness of arabic spam classifiers. *Neural Computing and Applications*, 37(6):4323–4343.

Nouar AlDahoul, H Abdul Karim, Mhd Adel Momo, Michael Aaron Sy, and Myles Joshua Toledo Tan. 2023. Evaluation of content moderation software for nudity and pornography detection in various scenarios. In *MECON Multimedia University Engineering Conference*.

Hanin Alshalan and Banafsheh Rekabdar. 2023. Attacking a transformer-based models for arabic language as low resources language (lrl) using word-substitution methods. In *2023 Fifth International Conference on Transdisciplinary AI (TransAI)*, pages 95–101.

Basemah Alshemali. 2025a. Diacritical manipulations as adversarial attacks in arabic nlp systems. *Arabian Journal for Science and Engineering*.

Basemah Alshemali. 2025b. Sentence-level adversarial examples in arabic. In *Computational Science and Computational Intelligence*, pages 228–242, Cham. Springer Nature Switzerland.

Basemah Alshemali and Jugal Kalita. 2020. Improving the reliability of deep neural networks in nlp: A review. *Knowledge-Based Systems*, 191:105210.

Basemah Alshemali and Jugal Kalita. 2021. Character-level adversarial examples in arabic. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 9–14.

Nora Saleh Alturayeif, Hamzah Abdullah Luqman, and Moataz Aly Kamaleldin Ahmed. 2022. Mawqif: A multi-label Arabic dataset for target-specific stance detection. In *Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 174–184, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Faisal Alvi, Mark Stevenson, and Paul Clough. 2017. Plagiarism detection in texts obfuscated with homoglyphs. In *Advances in Information Retrieval*, pages 669–675, Cham. Springer International Publishing.

Sergey Berezin, Reza Farahbakhsh, and Noel Crespi. 2025. Evading toxicity detection with ASCII-art: A benchmark of spatial attacks on moderation systems. In *Proceedings of the The 9th Workshop on Online Abuse and Harms (WOAH)*, pages 153–162, Vienna, Austria. Association for Computational Linguistics.

Nicholas Boucher, Ilia Shumailov, Ross Anderson, and Nicolas Papernot. 2022. Bad characters: Imperceptible nlp attacks. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1987–2004.

Daniel Braun, Adrian Hernandez Mendez, Florian Matthes, and Manfred Langen. 2017. Evaluating natural language understanding services for conversational question answering systems. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 174–185, Saarbrücken, Germany. Association for Computational Linguistics.

Portia Cooper, Mihai Surdeanu, and Eduardo Blanco. 2023. Hiding in plain sight: Tweets with hate speech masked by homoglyphs. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2922–2929, Singapore. Association for Computational Linguistics.

Joana C. Costa, Tiago Roxo, Hugo Proença, and Pedro Ricardo Morais Inácio. 2024. How deep learning sees the world: A survey on adversarial attacks & defenses. *IEEE Access*, 12:61113–61136.

Aldan Creo and Shushanta Pudasaini. 2025. SilverSpeak: Evading AI-generated text detectors using homoglyphs. In *Proceedings of the 1stWorkshop on GenAI Content Detection (GenAIDetect)*, pages 1–46, Abu Dhabi, UAE. International Conference on Computational Linguistics.

Steffen Eger, Gözde Gül Şahin, Andreas Rücklé, Ji-Ung Lee, Claudia Schulz, Mohsen Mesgar, Krishnkant Swarnkar, Edwin Simpson, and Iryna Gurevych. 2019. Text processing like humans do: Visually attacking and shielding NLP systems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1634–1647, Minneapolis, Minnesota. Association for Computational Linguistics.

Hady ElSahar and Samhaa R. El-Beltagy. 2015. Building large arabic multi-domain resources for sentiment analysis. In *Computational Linguistics and Intelligent Text Processing*, pages 23–34, Cham. Springer International Publishing.

Tatiana Ermakova, Benjamin Fabian, Elena Golimblevskaia, and Max Henke. 2023. A comparison of commercial sentiment analysis services. *SN Computer Science*, 4(5):477.

Brian Formento, Chuan Sheng Foo, Luu Anh Tuan, and See Kiong Ng. 2023. Using punctuation as an adversarial attack on deep learning-based NLP systems: An empirical study. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1–34, Dubrovnik, Croatia. Association for Computational Linguistics.

Esther Gan, Yiran Zhao, Liying Cheng, Mao Yancan, Anirudh Goyal, Kenji Kawaguchi, Min-Yen Kan, and Michael Shieh. 2024. Reasoning robustness of LLMs to adversarial typographical errors. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10449–10459, Miami, Florida, USA. Association for Computational Linguistics.

Shreya Goyal, Sumanth Doddapaneni, Mitesh M. Khapra, and Balaraman Ravindran. 2023. A survey of adversarial defenses and robustness in nlp. *ACM Comput. Surv.*, 55(14s).

Sergiu C. Ivan, Robert Ş. Győrödi, and Cornelia A. Győrödi. 2024. Sentiment analysis using amazon web services and microsoft azure. *Big Data and Cognitive Computing*, 8(12).

Jamin Rahman Jim, Md Apon Riaz Talukder, Partha Malakar, Md Mohsin Kabir, Kamruddin

Nur, and M.F. Mridha. 2024. Recent advancements and challenges of nlp-based sentiment analysis: A state-of-the-art review. *Natural Language Processing Journal*, 6:100059.

Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2021. *Benchmarking Natural Language Understanding Services for Building Conversational Agents*, pages 165–183. Springer Singapore, Singapore.

Yanying Mao, Qun Liu, and Yu Zhang. 2024. Sentiment analysis methods, applications, and challenges: A systematic literature review. *Journal of King Saud University - Computer and Information Sciences*, 36(4):102048.

Yassir Matrane, Faouzia Benabbou, and Nawal Sael. 2023. A systematic literature review of arabic dialect sentiment analysis. *Journal of King Saud University - Computer and Information Sciences*, 35(6):101570.

Walaa Medhat, Ahmed Hassan, and Hoda Korashy. 2014. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4):1093–1113.

Mahmoud Nabil, Mohamed Aly, and Amir Atiya. 2015. ASTD: Arabic sentiment tweets dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2515–2519, Lisbon, Portugal. Association for Computational Linguistics.

Ruba Obiedat, Duha Al-Darras, Esra Alzaghoul, and Osama Harfoushi. 2021. Arabic aspect-based sentiment analysis: A systematic literature review. *IEEE Access*, 9:152628–152645.

Oumaima Oueslati, Erik Cambria, Moez Ben HajHmida, and Habib Ounelli. 2020. A review of sentiment analysis research in arabic language. *Future Generation Computer Systems*, 112:408–430.

Łukasz Pawlik. 2025. Google cloud vs. azure: sentiment analysis accuracy for polish and english across content types. *Journal of Cloud Computing*, 14(1):17.

Shilin Qiu, Qihe Liu, Shijie Zhou, and Wen Huang. 2022. Adversarial attack and defense technologies in natural language processing: A survey. *Neurocomputing*, 492:278–307.

Azzam Radman and Rehab Duwairi. 2025. Towards a robust deep learning framework for arabic sentiment analysis. *Natural Language Processing*, 31(2):500–534.

Kui Ren, Tianhang Zheng, Zhan Qin, and Xue Liu. 2020. Adversarial attacks and defenses in deep learning. *Engineering*, 6(3):346–360.

Sahar Sadrizadeh, AmirHossein Dabiri Aghdam, Ljiljana Dolamic, and Pascal Frossard. 2023. Targeted adversarial attacks against neural machine translation. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Ayed Salman, Anwar Alajmi, and Imtiaz Ahmad. 2024. Evaluation of adversarial robustness in arabic language models. *Available at SSRN 4954707*.

Mayur Wankhade, Annavarapu Chandra Sekhara Rao, and Chaitanya Kulkarni. 2022. A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review*, 55(7):5731–5780.

Terrence E White and Manjeet Rege. 2020. Sentiment analysis on google cloud platform. *Issues in Information Systems*, 21(2):221–228.

Yong Xie, Dakuo Wang, Pin-Yu Chen, Jinjun Xiong, Sijia Liu, and Sanmi Koyejo. 2022. A word is worth a thousand dollars: Adversarial attack on tweets fools stock predictions. *arXiv preprint arXiv:2205.01094*.

Han Xu, Yao Ma, Hao-Chen Liu, Debayan Deb, Hui Liu, Ji-Liang Tang, and Anil K. Jain. 2020. Adversarial attacks and defenses in images, graphs and text: A review. *International Journal of Automation and Computing*, 17(2):151–178.

Xinze Zhang, Junzhe Zhang, Zhenhua Chen, and Kun He. 2021. Crafting adversarial examples for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1967–1977, Online. Association for Computational Linguistics.

Jiangrui Zheng, Xueqing Liu, Mirazul Haque, Xing Qian, Guanqun Yang, and Wei Yang. 2024. HateModerate: Testing hate speech detectors against content moderation policies. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2691–2710, Mexico City, Mexico. Association for Computational Linguistics.

Shuai Zhou, Chi Liu, Dayong Ye, Tianqing Zhu, Wanlei Zhou, and Philip S. Yu. 2022. Adversarial attacks and defenses in deep learning: From a perspective of cybersecurity. *ACM Comput. Surv.*, 55(8).

A. Extra tables

The following tables provide more examples and details.

Text	Perturbations	Corpus
جيدوجميل وطعامه لذيذ وقهوة رائعةالمذاق Good and beautiful. Their food is delicious, and cafe tastes wonderful	Concatenated words	OCLAR
قمن بتزيين الإجـ هاض لهن على أنه نوع من أنواع تمكين المرأة They misrepresented abortion to them as some form of women empowerment	Split word stem	Mawqif
الشؤون الاجتماعية: آلية جديدة لـ صرف المساعدات المقطوعة للمحتاجين Social Affairs: A new mechanism for disbursing aid allocated to the needy	Split word (separate prefix)	AraSenti-Tweet
غيبتك جرح وشوفة عيونك علاج Your absence is a wound and seeing your eyes is a cure	Positional variants and homoglyphs	NADI2022
مساءگم بدايہ جميلہ وطريق مفتوح وأمنيات تتحقق May your evening be a beautiful start, a clear path, and wishes that come true	Arabic script homoglyphs	NADI2022
كيف نحضّل عالم نظيف واعي والعnصRية والtخلف هذا للحين في الناس How do we get a clean and conscious world, when this racism and backwardness still exists among people?	English sounds and Arabic diacritics	Mawqif
اكثر من 80% من المؤيدين للقرار من فصيلة الحمقى والسُذَّجْ More than 80% of those who support the decision are foolish and gullible	Arabic diacritics	AraSenti-Tweet
بس فيہ #رآجل اقتنع أن #وآحـده بس #مگفيآه but there are men who are convinced that one is enough for them.	Homoglyphs from Arabic and Hebrew scripts	ASTD
وٰ كلما تمنيت الخير لغيرگ . . جآءك الخير من حيث لا تحتسب And whenever you wish good for others ... good things come to you in unexpected ways	Homoglyphs, positional variants, and non-Arabic diacritics	NADI2022

Table 6: Examples of perturbed posts from the examined corpora displayed using the Noto Naskh Arabic font. Perturbed words/syllables are highlighted in red.

Corpus	Platform	Version	Text	Class	Score
AraSenti-Tweet	GCP	original	الاجازه ـ بالنسبالك . راحه سعاده فرح نوم وناسه تواصل رررررررررروقان سهر	positive	0.41
		homoglyphs	الاجازه ـ بالنسبالک . راحه سعادھ فرح نوم وناسہ ٹواصل رررررررررروقان سھر	neutral	0.21
		diacritics	اَلَاجَازَهْ ـَ بَالَنَسَبَالَكْ . رَاحَهْ سَعَادَهْ فَرْحَ نُومَ وَنَاسَهْ تُوَاصَلْ رَرْرَرْرَرْرَرْرَرُوقَانَ سَهْرَ	neutral	0.07
		pos-variants	الاجازه ـ بالنسبالك . راحه سعاده فرح نوم وناسه تواصل رررررررررروقان سهر	neutral	0.15
		split words	الاجـازه ـ بالنسبـالك . راحـ..ه سعا ده فرح نوم وناسـ ـه توا صل ررر ررر ررر روقـان سهر	neutral	0.19
Mawqif	Azure	original	هذا ما نرجوا ونأمل المزيد من التحول الإلكتروني جهودكم مشكوره .	positive	1.00
		homoglyphs	هذا ما نرجوا ونأمل المزيد من التحول الإلكترونې جهودكم مشكورھ .	neutral	0.43
		diacritics	هَذَا مَا نَرْجُوَا وَنَأْمَلَ اَلْمَزِيدَ مَنْ اَلْتَحُولَ اَلْإِلْكَتْرُونِي جَهُودَكُمَ مَشْكُورَهْ .	positive	0.79
		pos-variants	هذا ما نرجوا ونأمل المزيد من التحول الإلكتروني جهودكم مشكوره .	positive	1.00
		split words	هذا ما نرجـوا ونأمـ..ل المزيـ ـد من التحو ل الإلـ ـكتر وني جهو دكم مشكـوره .	negative	0.36

Table 7: Examples of posts and their predicted labels before and after perturbations are applied. For GCP, the score denotes valence in the range [-1, 1], while for Azure, it's a confidence score in the range [0, 1]. Both examples are originally labeled as positive.