information

MDPI

Article

Fine-Grained Arabic Post (Tweet) Geolocation Prediction Using Deep Learning Techniques

Marwa K. Elteir

Informatics Research Institute (IRI), City of Scientific Research and Technological Applications (SRTA-City), Alexandria 5220211, Egypt; maelteir@vt.edu

Abstract: Leveraging Twitter data for crisis management necessitates the accurate, fine-grained geolocation of tweets, which unfortunately is often lacking, with only 1–3% of tweets being geolocated. This work addresses the understudied problem of fine-grained geolocation prediction for Arabic tweets, focusing on the Kingdom of Saudi Arabia. The goal is to accurately assign tweets to one of thirteen provinces. Existing approaches for Arabic geolocation are limited in accuracy and often rely on basic machine learning techniques. Additionally, advancements in tweet geolocation for other languages often rely on distinct datasets, hindering direct comparisons and assessments of their relative performance on Arabic datasets. To bridge this gap, we investigate eight advanced deep learning techniques, including two Arabic pretrained language models (PLMs) on one constructed dataset. Through a comprehensive analysis, we assess the strengths and weaknesses of each technique for fine-grained Arabic tweet geolocation. Despite the success of PLMs in various tasks, our results demonstrate that a combination of Convolution Neural Network (CNN) and Long Short-Term Memory (LSTM) layers yields the best performance, achieving a test accuracy of 93.85%.

Keywords: Twitter; geolocation; Arabic; deep learning; CNN; LSTM; attention; transformer; PLM; BERT

Academic Editor: Arkaitz Zubiaga

Received: 9 December 2024
Revised: 30 December 2024
Accepted: 8 January 2025
Published: 18 January 2025

Citation: Elteir, M.K. Fine-Grained Arabic Post (Tweet) Geolocation Prediction Using Deep Learning Techniques. *Information* **2025**, *16*, 65. https://doi.org/10.3390/info16010065

1. Introduction

Social media platforms provide extensive data sources that enable a wide range of applications. For Twitter, it is estimated that 500 million tweets are sent per day [1]. Properly analyzing this vast volume of data can serve as a powerful tool for crisis management [2–5].

A study conducted by [6], involving experts from various countries, revealed that for emergency responders, the most important feature of a software tool for effectively utilizing social media is the ability to categorize social media posts on a map by geographical location. Unfortunately, due to privacy concerns, most Twitter users do not attach location information to their tweets, with only 1% to 3% [7,8] of tweets being geotagged.

The problem of tweet geolocation prediction can be addressed in two primary ways: predicting a user's home location or inferring the location where the tweet was posted. In this study, we focus on the latter. Fine-grained geolocation prediction of tweets has significant applications in various critical domains, including disaster management, epidemiology, outbreak tracking, and crime mapping. For instance, during an epidemic, accurate fine-grained geolocation prediction can assist authorities in effectively managing resources, such as optimizing the distribution of medical supplies and deploying healthcare teams to hospitals. Additionally, it enables the implementation of tailored regulations for specific cities or provinces based on localized infection rates.

Several solutions have been proposed for the fine-grained geolocation prediction of tweets that rely on extracting location information from fields like user-defined locations. However, the accuracy of these solutions is limited, as they depend solely on user inputs, which are not always accurate. More accurate solutions employ machine learning techniques to geolocate tweets by extracting features from various fields, such as tweet text, user name, user location, and so on. We adopt the latter approach in this work.

In this study, we focus on the Kingdom of Saudi Arabia (KSA) as our case study. According to statistics published in April 2024 [9], KSA ranks ninth globally and first among Arab countries in the number of Twitter users, with 16.28 million users, representing 43.44% of the population [10]. Arabic is the official language in KSA, so we consider only Arabic tweets.

Geolocation extraction for low-resource languages like Arabic remains significantly understudied [11]. Progress in Arabic text geolocation prediction, especially for informal text such as tweets, is limited. State-of-the-art models achieve an F1 score of 88.1% at the country level [12] and an accuracy of 67.41% at a 160 km error distance [13]. Additionally, the application of deep learning models, particularly BERT-based models, to Arabic tweet geolocation, especially at a fine-grained level, is still in its early stages. Furthermore, advancements in tweet geolocation prediction for other languages, at both coarse-grained and fine-grained levels, often rely on distinct datasets, making direct comparisons and assessments of their relative performance on specific Arabic datasets challenging.

To address this gap, this work identifies and employs eight advanced deep learning techniques, including two Arabic pretrained language models (PLMs), for the task of fine-grained Arabic tweet geolocation prediction. To the best of our knowledge, Arabic PLMs have not been previously applied to this task [14]. This study represents the first systematic analysis of advanced deep learning techniques for fine-grained Arabic tweet geolocation prediction, conducted through comprehensive experiments. Our approach provides a fair assessment of the respective strengths and weaknesses of these techniques, offering valuable insights and guidance for the research community. (For the remainder of this article, the term 'tweet' will be used interchangeably with 'post' to refer to any message or content shared on the social media platform).

It is noteworthy that while this study primarily focuses on solving tweet geolocation prediction for Arabic tweets, we believe the proposed framework has broader applicability. Specifically, it can potentially benefit geolocation prediction for other languages and extend to other social media platforms, such as Instagram and Facebook.

Our contribution can be summarized as follows:

- Constructing a dataset containing tweets from the thirteen provinces of KSA, ensuring an even distribution across the provinces (The dataset is available at https://www.kaggle.com/datasets/marwaelteir/ksageolocatedtweets, accessed on 8 December 2024).
- Identifying and applying eight deep learning models, including two PLMs for fine-grained geolocation prediction of Arabic tweets across the thirteen provinces of KSA (To ensure the reproducibility of the research findings, the source code for the models is publicly accessible at https://github.com/maelteir/Tweets-Geolocation-Using-Deep-Learning/tree/main, accessed on 8 December 2024).
- Conducting an extensive set of experiments to assess the effectiveness and efficiency of these deep learning models.

The remainder of this paper is organized as follows. Section 2 reviews related work on Arabic tweet geolocation prediction. Section 3 details the methodology employed to construct a balanced, geolocated tweet dataset and describes the architecture of the deep

learning models used. Section 4 presents the experimental setup and results. Finally, Section 5 concludes the paper and outlines future research directions.

2. Related Work

Section 2.1 summarizes the work performed on Arabic Named Entity Recognition, as advances in this area provide means to address the Arabic geolocation prediction problem. Section 2.2 discusses the work performed on geolocation prediction, especially for Arabic tweets.

2.1. Named Entity Recognition

The first Arabic Named Entity Recognition (NER) dataset, ANERCorp, was released in 2007 by Benajiba et al. [15]. Since then, a lot of research efforts have been performed to improve the NER performance on this dataset. Based on a recent survey [16] that studied more than ninety research articles, the state-of-the-art technique for NER on ANERCorp dataset is reported by El Moussaoui et al. [17]. This technique involves using CNN-based character embeddings, BERT-based features, and FastText word embeddings for input encoding, feed-forward neural networks as context layer, and biaffine classifier as the final prediction/decoder layer. This technique achieves an F1 score of 95.77.

ANERCorp represents only Modern Standard Arabic (MSA). Therefore, the best performing technique on this dataset is not performance portable on a Twitter dataset. In 2013, Darwish et al. [18] showed that the performance in terms of F1 score of a Conditional Random Field (CRF)-based NER model degrades from 79.9 on the ANERCorp dataset to 33.1 on a Twitter dataset. This dataset is composed of 1423 manually annotated tweets spanning the period from 23 November 2011 to 27 November 2011. It will be hereafter referred to as TWEETS.

In 2014, Darwish et al. [19] trained a CRF-based model using a manually annotated Twitter dataset of 3646 tweets, in addition to ANERCorp. We will refer to this Twitter dataset as TWEETS_TRAIN. To further improve the NER effectiveness, the authors built a large Wikipedia gazetteer, applied the domain adaption technique, and performed semi-supervised two-pass training. Using the TWEETS dataset [18] as the test dataset, they reported F1 scores of 65.2 and 76.7 for general NER and location NER, respectively.

Khalifa et al. [20] proposed a deep learning model composed of CNN-based character embeddings and pretrained word embeddings as the input layers, BiLSTM as the context layer, and CRF as the output layer. The achieved F1 scores on the TWEETS dataset [18] were 65.34 and 72 for general NER and location NER, respectively.

In 2019, Google released BERT (Bidirectional Encoder Representations from Transformers) [21], a large language model (LLM) that significantly improves the performance of various language understanding tasks. In 2020, Antoun et al. [22] released the first Arabic large language model, AraBERT, that is based on the BERT architecture and pretrained on a large corpus of MSA Arabic documents. Despite not being pretrained on the Twitter dataset or dialectal Arabic, the AraBERT sentiment analysis performance on three benchmark tweets datasets is significantly better than that of the previous state of the art at the time of releasing AraBERT.

In 2021, Abdul-Mageed et al. [23] released MARBERT, another Arabic LLM based on the BERT architecture and pretrained on a large corpus of Arabic Tweets. Its F1 score for a general NER task on the TWEETS dataset [18] is 66.67.

In 2022, Benali et al. [24] compared the performance of MARBERT to several BERT-based LLMs, including AraBERTv02 when these models are used as an embedding layer in a BiLSTM CRF model. This is known as a feature extraction-based approach of employing LLM rather than a fine-tuning approach. The NER results on the TWEETS dataset [18]

show that MARBERT achieved new state-of-the-art results. Specifically, it achieved F1 scores of 67.4 and 77.9 for general NER and location NER, respectively.

In 2023, Suwaileh et al. [11] tackled the problem of location mention recognition (LMR) in informal Arabic text, uniquely. They released a human-labeled dataset, IDRISI-RA (gold version), of 4593 disaster tweets spanning diverse Arab countries. They also released a silver version of the dataset containing 1.2M automatically labeled tweets. The labeling was performed at the coarse- and fine-grained levels. Additionally, the authors benchmarked the dataset using different machine learning models. The best performing model was a MARBERT-based model [24], achieving F1 scores of 75 and 88 for the typeless and type-based LMR, respectively.

Table 1 provides an overview of the major contributions to NER for informal Arabic text. Due to the inherent complexity of informal Arabic, the current state-of-the-art location NER technique on the TWEETS dataset achieves a limited F1 score of 77.9. Furthermore, the existing literature lacks a definitive evaluation of the effectiveness of this approach in developing a fine-grained geolocation solution for Arabic tweets [25]. To address this gap, we frame the geolocation problem as a classification task. The related work on geolocation prediction approached as a classification problem is discussed in the next section.

Table 1. Major NER contributions for informal Arabic text.

Work	Year	Model	Training Dataset	F1 Score	
				NER	Location NER
[18]	2013	Gazetteer-based CRF	ANERCorp	39.90	47.90
[19]	2014	Gazetteer-based CRF	ANERCorp and TWEETS_TRAIN [19]	65.20	76.70
[20]	2019	CNN-based character embeddings, pretrained word embeddings, BiLSTM, and CRF	ANERCorp, Several news datasets, and TWEETS_TRAIN [19]	65.34	72.00
[22]	2020	Fine-tuned AraBERT	24 GB of Arabic news articles	41.26	-
[23]	2021	Fine-tuned MARBERT	a Large in-house dataset of 1B Arabic tweets	66.67	-
[24]	2022	AraBERT, BiLSTM, and CRF	TWEETS_TRAIN [19]	65.70	-
[24]	2022	MARBERT, BiLSTM, and CRF	TWEETS_TRAIN [19]	**67.40**	**77.90**
[11]	2023	MARBERT, BiLSTM, and CRF	IDIRIS-RA	-	**88.00**

The test dataset is the TWEETS dataset [18], except for [11], where the test dataset involves tweets from IDRISI-RA [11]. Bold text is the best.

2.2. Geolocation Prediction

Geolocating social media posts/users has been an active research area, especially for the English language. Several studies address the tweet geolocation prediction problem [8,26–31], while others focus on geolocating the users [32–37].

On the other side, the work performed to address this problem for Arabic tweets is very limited. Geolocating Arabic tweets presents unique challenges due to the complexity of the language and the diversity of the dialects. Mourad et al. [38] studied the influence of the language on the geolocation accuracy and concluded that Arabic tweets are more challenging to geolocate accurately compared to tweets in other languages.

In 2013, Khanwalkar et al. [39] proposed a solution to geolocate Twitter users using only tweet content. Initially, a window of tweets for a specific user is collected. Then named entity recognition technique is applied to recognize location toponym in the document. Several gazetteers are then employed to obtain the corresponding location record. A scoring

mechanism is proposed to rank the user locations, where the top-ranked location is chosen as the geolocation of the user. The solution accepts both English and Arabic tweets; however, the Arabic tweets are translated to English before processing. The best achieved accuracy at 100 miles was 37.7%.

Izbicki et al. [29] proposed the content-only-based tweet geolocation prediction model, Unicode-CNN, using convolution neural network. The model supports 100 languages, including Arabic, by employing the Unicode of the characters as input features. Depending on characters instead of words eliminates the need for a special tokenizer for each supported language. One of their key contribution is revealing the exact GPS coordinates of the tweet, taking into account the non-Euclidean nature of the Earth's surface. The accuracy of geolocating Arabic tweets at the country level was 56.2%. For geolocating the tweet at the city level, they only reported the results for English tweets, and the best value was 13.3%. The best accuracy at 100 km was for the huge version of Unicode-CNN, which was 26.7%

Mubarak et al. [12] developed the UL2C model, which maps Twitter Arabic user location to countries. First, a manually annotated dataset is built that maps a user location to a country. They trained a support vector machine model to automatically geolocate the user location. Additionally, the authors explored using word n-gram and character n-gram as input features. Their results show that the best model uses character n-gram and achieves an F1 score of 88.1.

Recently, Alsaqer et al. [13] proposed a fine-grained tweets geolocating solution. They collected a dataset of 35K unique geotagged tweets from KSA. These tweets belong to 30 cities. The dataset suffers from imbalance. They trained traditional machine learning models, including linear regression, support vector machine, random forest, and Multinomial Naive Bayes. They employed three features—tweet text, user location, and named entities—in the tweet text. The best accuracy was achieved from random forest, reaching 67.41% at 160 km distance.

The geolocation of Arabic text, particularly informal text-like tweets, remains a challenging task. Existing solutions are often not specifically designed for Arabic [29,39] or rely on traditional machine learning techniques [12,13], leading to limited accuracy. Moreover, existing Arabic text classification surveys [40–42] primarily focus on other downstream tasks, such as sentiment analysis. To our knowledge, no standardized technique exists for geolocating Arabic tweets.

In this study, we aim to address this gap by investigating the effectiveness and cost of advanced machine learning techniques, including deep learning and pretrained large language models, for the fine-grained geolocation prediction of Arabic tweets.

3. Materials and Methods

We conceptualize the tweet geolocation prediction problem as a classification task. Formally, given a set of tweets T = $\{t_1, t_2, t_3, \ldots, t_n\}$, where each tweet t_i comprises the tweet text along with metadata, and their respective fine-grained geolocations L= $\{l_1, l_2, l_3, \ldots, l_n\}$, where $l_i \in \{c_1, c_2, c_3, \ldots, c_m\}$ represent m classes corresponding to fine-grained geolocations, we aim to develop a classification model. This model will classify any unseen tweet t_j into the corresponding geolocation l_j with an acceptable level of accuracy.

The granularity of geolocation prediction is determined by both the number of classes and their geographical coverage. When the classes correspond to countries, the geolocation prediction is coarse grained. Conversely, when the classes represent cities or small provinces spanning tens of thousands of square kilometers, the prediction becomes fine grained.

Figure 1 represents the workflow of our study, including the dataset collection, text preprocessing, feature extraction, and machine learning models building. The workflow is applied to KSA as a case study. The following sections illustrate each step in detail.

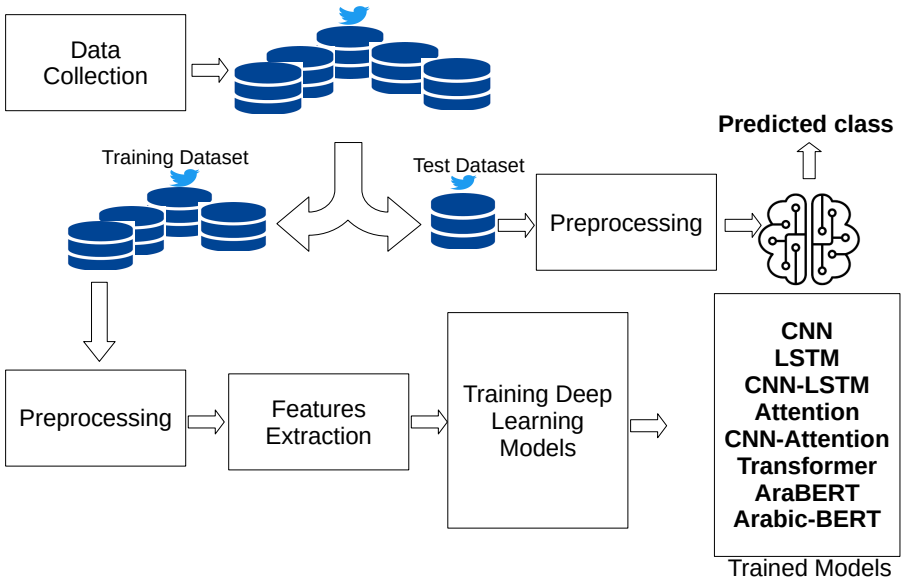

Figure 1. Study workflow.

3.1. Dataset Collection

We collected a dataset of Arabic geotagged tweets from the thirteen provinces of KSA using Twitter's v2 full-archive search endpoint. To access this endpoint, we employed the searchtweets library [43] through an academic research account.

To ensure comprehensive data collection, we defined each province as a group of non-overlapping circular regions. These regions were delineated using the Leaflet v1.0.3 plugin [44] and incorporated into our search queries. By utilizing the `point_radius` operator, we specified the radius of each circular region. Additionally, the `lang` operator was employed to restrict the search to Arabic-language tweets. This methodology, inspired by [45], allowed us to efficiently gather tweets from all thirteen provinces of KSA. By utilizing only two operators in the search query, we were able to generate a diverse dataset encompassing various dialects and topics (For a detailed description of this methodology, please refer to [46]).

We collected the tweets over a two-month period, from 1 May 2022, to 30 June 2022. Due to the non-uniform population distribution across provinces, the volume of retrieved tweets was higher in more densely populated regions. Mourad et al. (2019) [38] demonstrated that data imbalance has a more pronounced impact on model accuracy than geographical coverage. Similarly, Alruily et al. (2023) [47] reported that BERT-based classifiers achieve significantly better performance on balanced datasets compared to imbalanced ones. To mitigate the effects of data imbalance, we constructed a balanced dataset by limiting the number of tweets per province to the first 5000 retrieved tweets. The characteristics of the resulting dataset are summarized in Table 2.

Table 2. The characteristics of the retrieved dataset.

No. of Unique Tweets	No. of Unique Users	Country	Provinces	Time Zone	Time Period
64,833	12,085	KSA	13	One (GMT + 3:00)	1 May 2022 to 30 June 2022

3.2. Machine Learning Models

We based our selection of the studied machine learning models on techniques for English tweet/user geolocation prediction as described in the existing literature.

3.2.1. Basic Deep Learning Techniques

Deep Neural Networks (DNNs) [48] were among the first deep learning methods applied to the tweet/user geolocation prediction problem. Liu et al. [49] pioneered a DNN architecture comprising three hidden layers, each with 5000 neurons, for predicting Twitter user locations. In a subsequent study, Lourentzou et al. [50] explored the impact of various factors—such as activation functions, batch normalization, dropout, and network architecture—on DNN performance for geolocation prediction. Their findings indicated that the Rectified Linear Unit (ReLU) activation function outperforms others, and they recommended using shallow, wide architectures over deep, compact ones for this classification task.

Convolutional Neural Networks (CNNs) were first introduced for tweet geolocation prediction by Huang et al. [26]. Their base CNN architecture, adapted from Kim [51] for sentence classification, consists of a convolutional layer, a max pooling layer, and a fully connected layer. Using tweet text and several metadata fields as input, they achieved country-level and city-level prediction accuracies of 92.1% and 52.8%, respectively. A recent study by Lu et al. [52] on medical text classification demonstrated that CNNs achieved performance comparable to Transformer encoders and significantly outperformed BERT for balanced datasets. While CNNs are a relatively simple architecture, their superior performance in both country-level geolocation prediction and other text classification tasks motivates their inclusion in our study.

We employed a fundamental CNN architecture consisting of a single convolutional layer, as this design is commonly utilized in the literature for text classification [26,51,52]. Additionally, we explored various modifications of this architecture, which are summarized in Table 3. We also considered a multi-channel architecture [51], wherein channels adopt different kernel sizes to enable the model to capture a wide range of patterns and contexts. Figure 2 illustrates this approach.

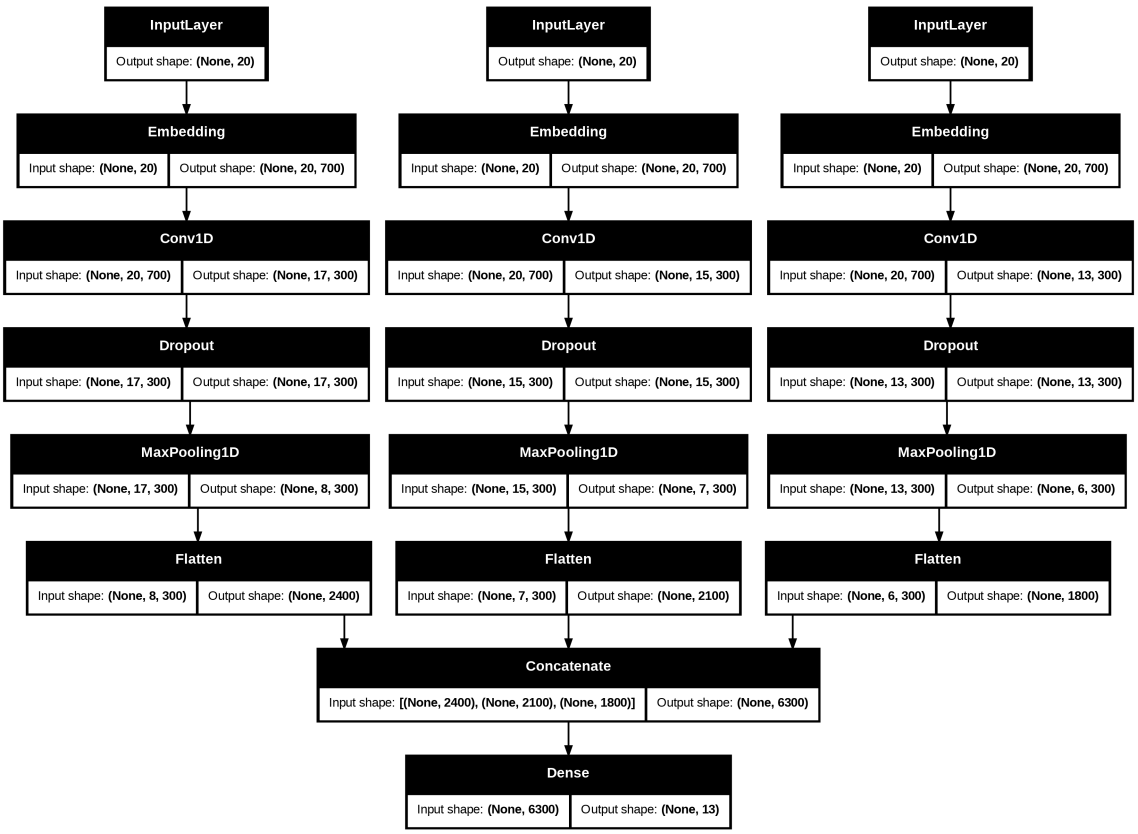

Figure 2. Multi-channel CNN architecture.

Table 3. The CNN architecture.

Layers	Details
Input	Word embedding
Convolution	ReLU activation with 64, 128, 256, 512, or 1024 filters
Dropout	0.2–0.5
Max pooling	-
Flatten	-
Dense	Softmax activation

While CNNs are effective at extracting local patterns from text, particularly in shallow architectures, Long Short-Term Memory (LSTM) networks and Bidirectional LSTMs (BiLSTMs) excel at capturing long-range dependencies and global context. Thomas et al. [53] introduced an LSTM-based model for geolocating tweets using both textual content and metadata. Mahajan et al. [30] proposed a hybrid approach that combines CNN and BiLSTM architectures to leverage the strengths of both models. Their method achieved an impressive city-level tweet geolocation accuracy of 92.6%.

We included LSTM, BiLSTM, and combined CNN-LSTM models, shown in Figure 3, in our study. The specific architecture of the LSTM-based model is illustrated in Table 4. We also explored the use of multiple stacked LSTM and BiLSTM layers, as stacking can significantly improve text classification accuracy.

Table 4. LSTM-based architectures.

Layers	Details
Input	Word embedding
Stacked LSTM or BiLSTM	Sigmoid activation with 64, 128, 256, or 512 filters
Dropout	0.2–0.5
Dense	Softmax activation

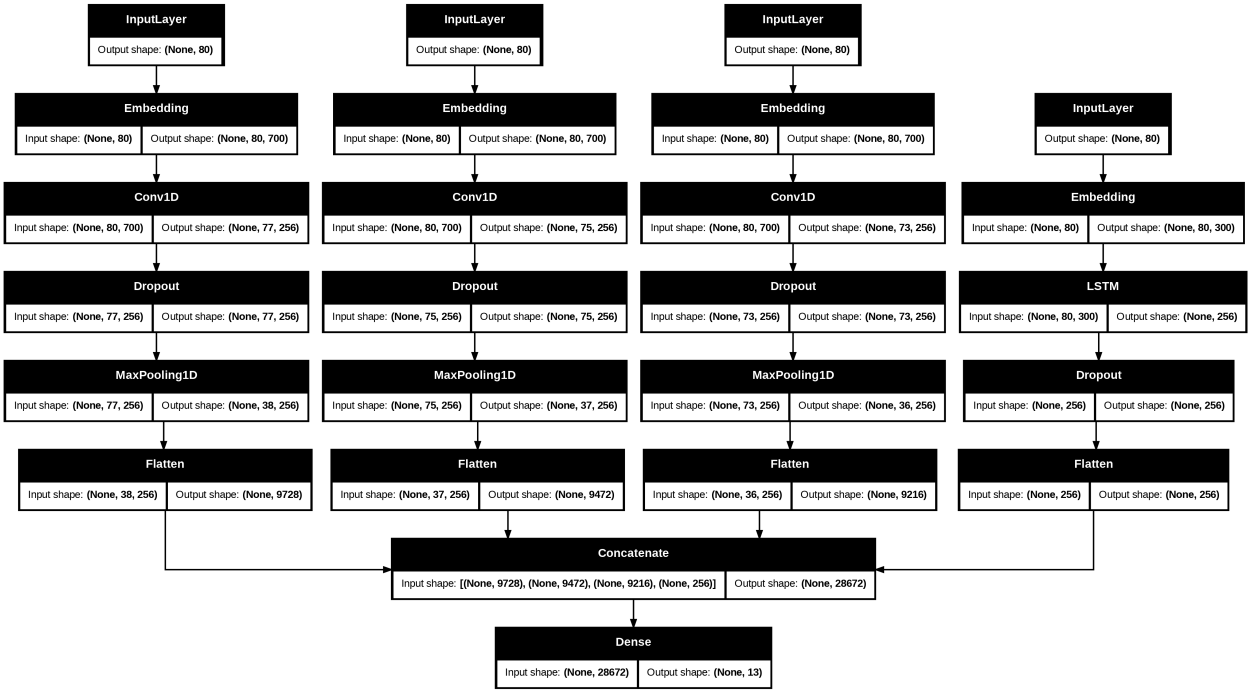

Figure 3. CNN-LSTM architecture.

3.2.2. Attention-Based Techniques

The attention mechanism [54,55] is a revolutionary mechanism that combines the benefits of CNN and LSTM. It assigns a score to each token in the text, indicating its contribution to the classification, thus successfully capturing relationships between tokens in long sequences. Huang et al. [56] proposed using the multi-head self-attention model [55] for tweet geolocation along with subword features extracted using CNN and joint training. They achieved state-of-the-art performance in the geolocation prediction shared task W-NUT2016. Fornaciari et al. [28] investigated the effectiveness of integrating CNNs with an attention mechanism to capture relationships between the local patterns extracted by the CNN, rather than focusing directly on the tokens of the text. They proposed a multi-channel CNN architecture, with its outputs processed through an attention mechanism. Their results demonstrated that incorporating the attention mechanism into the architecture improved geolocation accuracy by 10%.

We included the attention-based model in our study, with the architecture illustrated in Table 5. Note that attention, global average pooling, and global max pooling layers operate in parallel, and their outputs are concatenated before being fed to the softmax layer. We also included a combined CNN and attention model in our study, though we employed a slightly different architecture based on preliminary experiments as illustrated in Figures 4 and 5.

Yang et al. [57] proposed a classification model based solely on the attention mechanism for document classification. Their approach incorporated the attention mechanism introduced by Bahdanau et al. [54] to develop a hierarchical attention-based model operating at both the word and sentence levels. Later, Vaswani et al. [55] introduced the Transformer, a transduction model entirely reliant on attention mechanisms. The Transformer architecture employs stacked encoder and decoder layers for tasks such as machine translation, while only the encoder is used for classification tasks. Given the demonstrated success of the Transformer in fine-grained geolocation [58], we incorporated it into our study. The details of the Transformer architecture used in this work are presented in Table 6.

Table 5. Attention-based architectures.

Layers	Details
Input	Word embedding
SpatialDropout1D	0.2–0.5
BiLSTM	Sigmoid activation with 64, 128, 256, or 1024 filters
Attention *	Bahdanau et al. [54]
GlobalAveragePooling1D *	-
GlobalMaxPooling1D *	-
concatenate	-
Dense	Softmax activation

* These layers run in parallel.

Table 6. Transformer-based architectures.

Layers	Details
Input	Token embedding and position embedding
TransformerBlock *	Self-attention layer Dropout layer (0.1) Normalization layer (epsilon = 1×10^{-6}) Feed-forward layer Dropout layer (0.1) Normalization layer (epsilon = 1×10^{-6})
GlobalAveragePooling1D	-
Dropout	0.1
Dense	RelU activation
Dropout	0.1
Dense	Softmax activation

* This block is repeated 1, 2, 4, or 6 times.

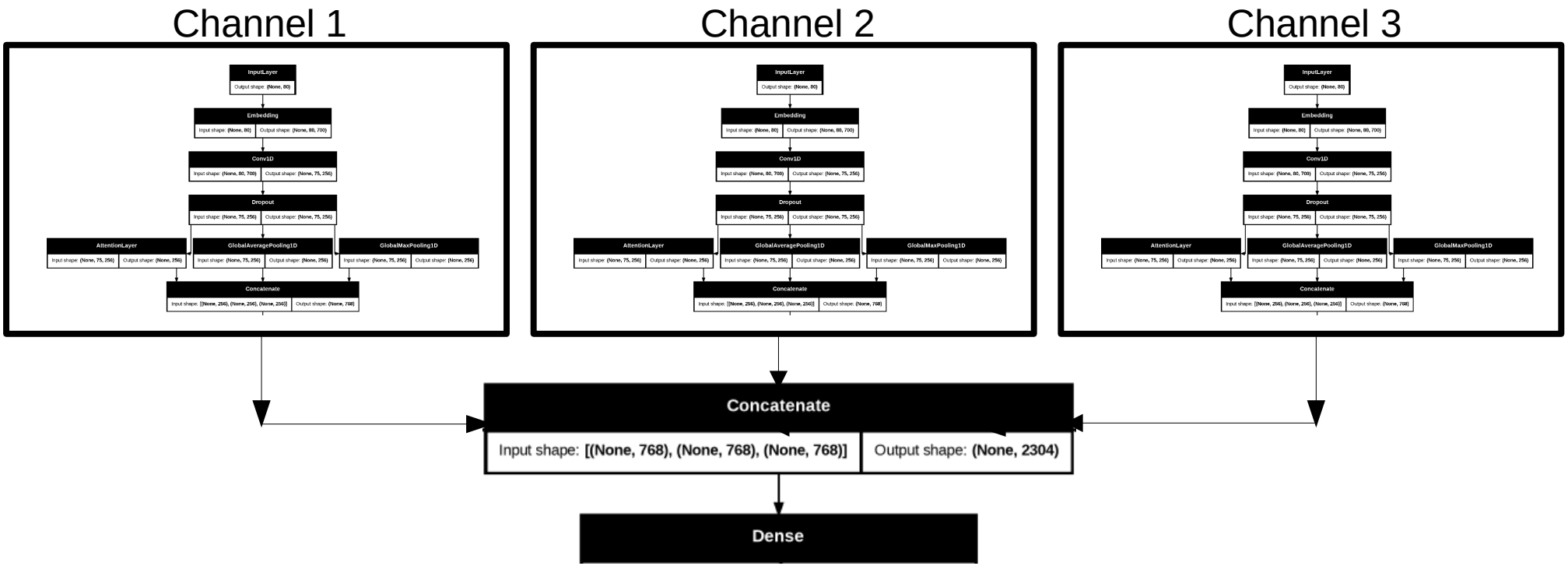

Figure 4. CNN-Attention architecture (all channels have the same architecture; however, they adopt different kernel sizes).

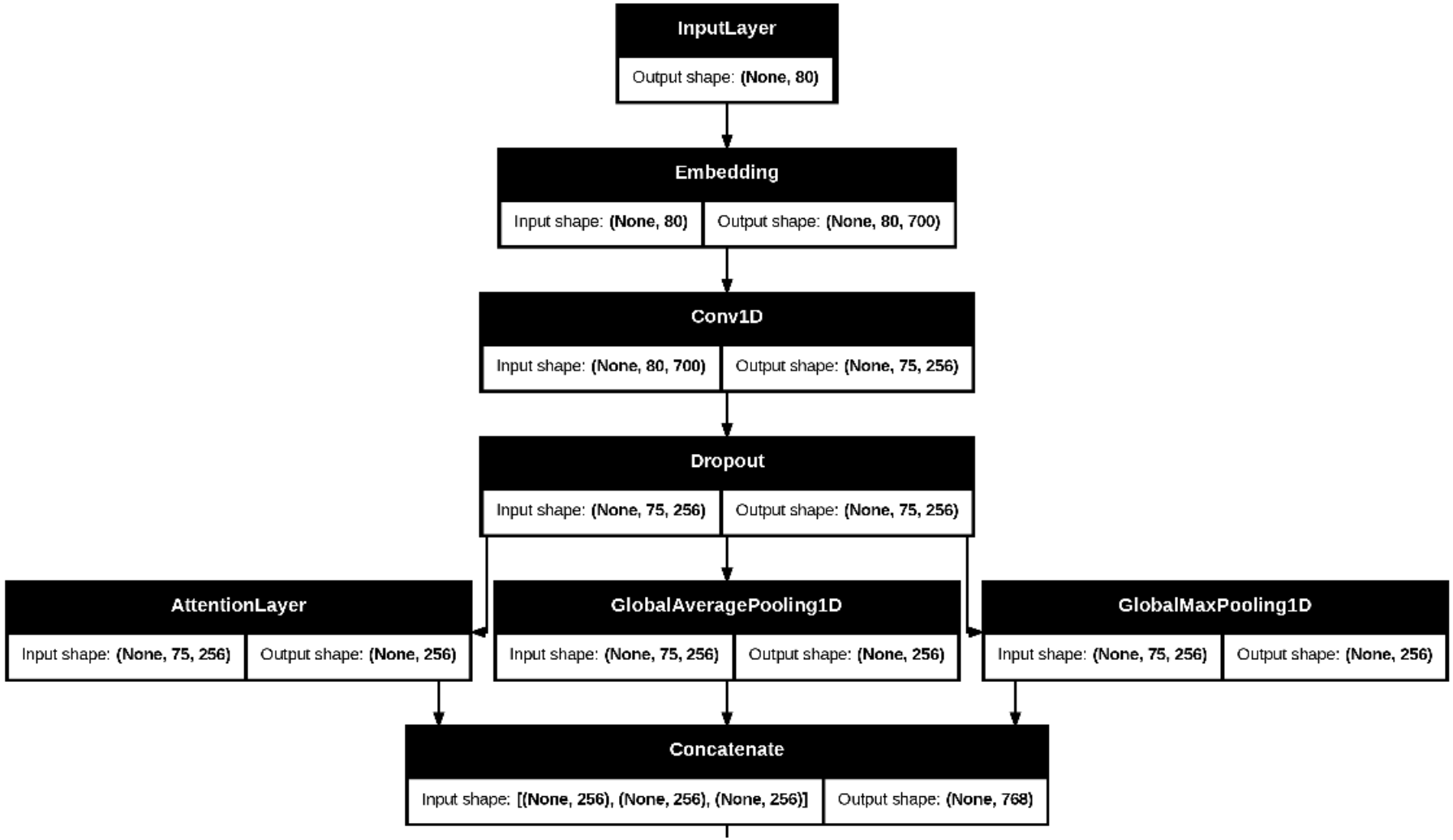

Figure 5. One channel of a CNN-attention architecture.

3.2.3. BERT-Based Techniques

Following the success of the Transformer, Devlin et al. [21] introduced BERT, a pre-trained English language representation model composed of a multi-layer bidirectional Transformer encoder. It has achieved state-of-the-art results on various NLP tasks [59], including geolocation prediction [37,58]. Several BERT-based models have been released for Arabic language representation, including AraBERT [22], Arabic-BERT [60], MARBERT [23], and ARBERT [23].

We included AraBERTv2 [61] and Arabic-BERT [62] in our study. AraBERT is the first pretrained language model for Arabic and is widely utilized in the literature [14]. Arabic-BERT, pretrained on large corpora of Modern Standard Arabic (MSA) and dialectical Arabic, was included to assess the impact of increasing dataset size and diversity on prediction performance. Although MARBERT is reported to be state of the art for Arabic text understanding based on [23], we opted for AraBERT. For the most relevant downstream tasks, namely NER and topic classification, AraBERT outperforms MARBERT in 7 out of 10 individual test datasets, and consistently outperforms MARBERT in all NER and topic classification tests on the combined ARLUE dataset.

Based on the findings of [21,41], we adopted the fine-tuning approach rather than the feature extraction approach, as it yields better performance, particularly for the NER task.

Each of the models discussed in Sections 3.2.1–3.2.3 demonstrates varying performance due to their application on different datasets. In the absence of a standardized baseline for comparison, we aim to evaluate all these models on one Arabic dataset. This approach enables a fair assessment of their respective strengths and weaknesses for the fine-grained tweet geolocation prediction task, offering valuable insights and guidance for the research community.

3.3. Feature Extraction and Preprocessing

To enhance geolocation prediction accuracy, we leverage three features extracted from tweets and their metadata: tweet text, user name, and user location (if available). To maintain contextual information, these features are concatenated as outlined in Figure 6 after undergoing preprocessing steps.

The preprocessing pipeline involves the following steps:

- Tokenization: We tokenize the text data using the PyArabic library [63]. This step breaks down the text into individual words or meaningful units.
- Text Cleaning: To improve model performance, we remove various elements from the text, including emoji, Arabic stop words, special characters (e.g., ”, ’, or :), English characters, spaces, underscores, punctuation, numbers, and repeated letters.
- Normalization and Stripping: We utilize the PyArabic library [63] to remove diacritical marks and elongation symbols that may not be crucial for geolocation prediction. Additionally, the library normalizes specific Arabic letters, such as hamza.
- Stemming: Finally, we employ the Farasa library [64] to perform stemming, which reduces words to their root forms.

For models utilizing the Arabic pretrained BERT model, AraBERT [61] provides dedicated preprocessing procedures that can be integrated into the overall pipeline to ensure compatibility with the model.

أنا أسمى <User name> أسكن فى <User location> و رأى <Tweet text>

Figure 6. Three concatenated features: tweet text, user name, and user location.

3.4. Word Embeddings

For all the aforementioned models, except for Transformer, AraBERT, and Arabic-BERT, the input is represented using an embedding layer. This embedding layer represents each word using a dense vector of user-defined length. We investigated initializing the embedding matrix with two non-contextual word embeddings: the Arabic version of Word2Vec [65] and FastText [66]. Additionally, we examined the performance when there was no initialization. Furthermore, we explored two options for these embeddings: non-trainable representations, where the embedding vectors remain constant during training, and trainable representations where a new embedding matrix is learned during training.

4. Results

4.1. Experimental Setup

All machine learning models discussed in Section 3.2.1 were implemented in Python. Specifically, we utilized the Keras library (version 2.10.0), the Transformers library (version 4.25.1), and the TensorFlow framework (version 2.10.0). Preliminary experiments were conducted on the Grid'5000 infrastructure [67], while the main experiments were performed on the Intel Developer Cloud [68].

The Adam optimizer was employed, as preliminary experiments indicated it achieved superior validation accuracy compared to root-mean-square propagation and stochastic gradient descent. Furthermore, Adam is the standard optimizer for BERT-based models [21]. For the loss function, we adopted sparse categorical cross entropy, which is commonly used in the deep learning literature.

Table 7 presents the range of hyperparameter values considered in this study. Given the average size of our dataset, we selected batch sizes of 16, 32, 64, and 128. To accommodate potential divergence in the studied models, a wide range of learning rates from 1×10^{-2} to 5×10^{-5} was explored. The input sequence length was determined based on the average and maximum numbers of words per input. After dropping unavailable features, preprocessing, and concatenation, we found these values to be 22.09 and 72, respectively. Consequently, we considered input lengths ranging from 20 to 80. Hyperparameters were tuned on the test set. The reported validation and test scores were averaged over five random runs, utilizing the optimal hyperparameter configurations identified in each experiment.

Table 7. Hyperparameters.

Hyperparameter	Range
Batch size	16, 32, 64, and 128
Learning rate	1×10^{-2} to 5×10^{-5}
Input sequence length	20 to 80

To control overfitting, we adopted the early stopping strategy. Specifically, we employed the patience approach, which monitors model performance on the validation set during training. If the model's validation loss does not improve for five consecutive epochs, the training is halted.

We evaluated the performance of the models using the metrics of accuracy, precision, recall, and F1 score as suggested by [41] for evaluating text classification models. The formulas used to calculate these metrics are described as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives.

For the specific task of Arabic text geolocation, we also measured the weighted average as recommended by [38]. During testing, we compared the predicted location classes with the ground truth labels to assess performance metrics.

4.2. Preliminary Experiments

Our initial investigations, shown in Table 8, were applied to a CNN model with a single embedding layer and three channels. The initial experiment, conducted on the entire dataset using only the tweet text as a feature, employed a trainable embedding layer without initialization. This experiment yielded a poor validation accuracy of 10.86%. Subsequent experiments incorporated proper shuffling and stratification of the dataset to ensure balanced distribution of classes across training, validation, and test sets. This resulted in a significant improvement in performance, with validation accuracy increasing to 23.96%. Further enhancement was achieved by including user name and user location as additional features, leading to a substantial boost in validation accuracy to 63.44%.

Removing noise from the dataset by excluding any records containing null features improved the validation accuracy to 84.88%. All experiments used a vector representation of length 50. Increasing the vector length to 300 further enhanced performance, achieving a validation accuracy of 86.42%.

Table 8. Preliminary experiments using CNN architecture with one embedding layer.

Dataset	Features Number	Shuffle and Stratify	Preprocessed	Input Vector Length	Embedding	Validation Loss	Validation Accuracy
W	1	No	No	50	T—NI	2.5297	0.1086
W	1	Yes	No	50	T—NI	2.3252	0.2396
W	1	Yes	Yes	50	T—NI	2.2544	0.2318
W	3	Yes	Yes	50	T—NI	1.2371	0.6344
W	3	Yes	No	50	T—NI	0.8461	0.7760
D	3	Yes	Yes	50	T—NI	0.5466	0.8488
D	**3**	**Yes**	**Yes**	**300**	**T—NI**	**0.5139**	**0.8642**
D	3	Yes	Yes	300	T—AraVec-sg	0.6070	0.8333
D	3	Yes	Yes	300	NT—AraVec-sg	0.8568	0.7599
D	3	Yes	Yes	300	T—AraVec-CBOW	0.8477	0.7859
D	3	Yes	Yes	300	NT—AraVec-CBOW	0.9172	0.7577
D	3	Yes	Yes	300	T—FastText	0.5731	0.8600
D	3	Yes	Yes	300	NT—FastText	0.6902	0.8142

W: whole dataset, D: dataset with empty features dropped, T: trainable, NT: not trainable, and NI: not initialized. All runs consider the default pre-padding and pre-truncation. Bold text is the best.

We examined the impact of initializing the embedding matrix with either AraVec2.0 or FastText embeddings. The results demonstrate that FastText outperforms both versions of AraVec, primarily due to AraVec's limited coverage. Specifically, the number of out-of-vocabulary (OOV) tokens for AraVec is 7185 out of 25,750, whereas FastText has no OOV tokens. Additionally, fine-tuning the weights of the embedding layer consistently yields better performance than using static vector representations for all embeddings, including AraVec, FastText, and uninitialized embeddings.

The results also indicate that, for the clean dataset with null features removed, preprocessing has minimal impact. We attribute this to the fact that the Keras tokenizer inherently applies several cleaning and filtering processes to the text, which may mimic the effects of explicit preprocessing. Moreover, the literature suggests that deep learning models exhibit greater resilience to unpreprocessed text [40].

4.3. Models Performance

Table 9 presents the optimal performance achieved by each model. Building upon the preliminary experiments detailed in Section 4.2, these results were obtained using a dataset with the empty features removed, proper shuffling and stratification, three features, a trainable uninitialized embedding layer, and preprocessed text. The specific configurations and hyperparameters that yielded the best results for each model are provided in the Appendix A.

Surprisingly, the best-performing model is not a large language model. The model that achieves the highest performance across all test metrics is the CNN-LSTM, with a test accuracy of 0.9385. In terms of validation loss and validation accuracy, the CNN-Attention model outperforms the others, achieving a validation accuracy of 0.9482. These models capture local features through the shallow CNN architecture, while long-range dependencies and global context are effectively modeled using the LSTM and attention architectures. The performance of CNN-LSTM per class is shown in Table 10, and the confusion matrix is shown in Figure 7. The figure shows the number of correctly geolocated tweets in the test dataset for each province. For example, for Riyadh, the number of correctly geolocated tweets is 105 out of 122 tweets.

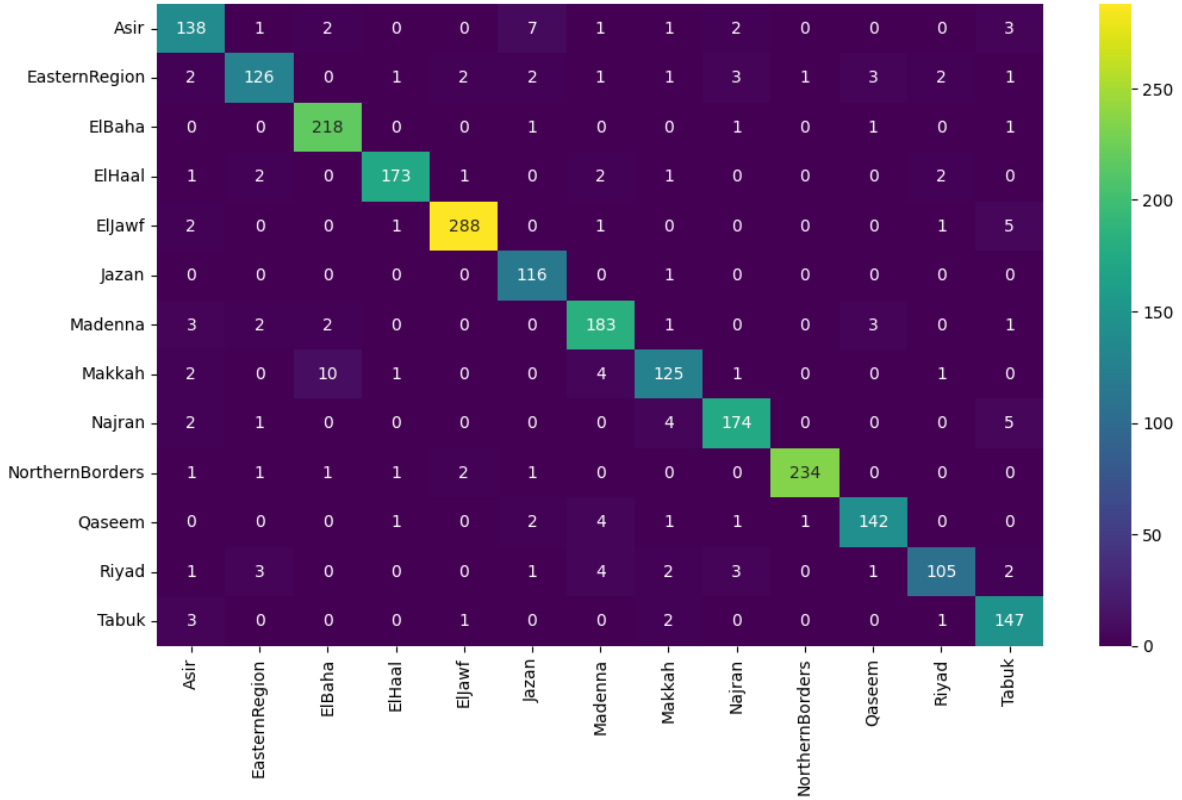

Figure 7. Confusion matrix of the CNN-LSTM architecture (the numbers in the diagonal represent the correctly geolocated tweets; however, the other numbers represent the number of incorrectly geolocated tweets).

Table 9. Performance of the studied machine learning models.

Model	Validation Set		Test Set						
	Loss	**Accuracy**	**Accuracy**	**Precision**	**Weighted Precision**	**F1 Score**	**Weighted F1 Score**	**Recall**	**Weighted Recall**
CNN	0.21382	0.94120	0.93428	0.93428	0.93520	0.93428	0.93429	0.93428	0.93428
LSTM	0.33636	0.92632	0.91388	0.91388	0.91499	0.91388	0.91399	0.91388	0.91387
CNN-LSTM	0.23176	0.93810	**0.93852**	**0.93852**	**0.93929**	**0.93852**	**0.93843**	**0.93852**	**0.93852**
Attention	0.27106	0.92900	0.92719	0.92719	0.92866	0.92719	0.92731	0.92719	0.92719
CNN-Attention	**0.20144**	**0.94822**	0.93722	0.93722	0.93724	0.93722	0.93705	0.93722	0.93722
Transformer	0.34336	0.92640	0.92036	0.92036	0.92131	0.92036	0.92014	0.92036	0.92036
AraBERT	0.28862	0.93230	0.93541	0.93541	0.93736	0.93541	0.93551	0.93541	0.93541
Arabic-BERT	0.32446	0.92208	0.92339	0.92339	0.92522	0.92339	0.92351	0.92339	0.92339

Bold text is the best.

Table 10. Performance metrics for each class for the best CNN-LSTM model. Support determines the total number of tweets per province.

Province	Precision	Recall	F1 Score	Support
Asir	0.89	0.89	0.89	155
EasternRegion	0.93	0.87	0.90	145
ElBaha	0.94	0.98	0.96	222
ElHaal	0.97	0.95	0.96	182
ElJawf	0.98	0.97	0.97	298
Jazan	0.89	0.99	0.94	117
Madenna	0.92	0.94	0.93	195
Makkah	0.90	0.87	0.88	144
Najran	0.94	0.94	0.94	186
NorthernBorders	0.99	0.97	0.98	241
Qaseem	0.95	0.93	0.94	152
Riyad	0.94	0.86	0.90	122
Tabuk	0.89	0.95	0.92	154
Macro avg	0.93	0.93	0.93	2313
Weighted avg	0.94	0.94	0.94	2313

As shown in Table A1, the best performance for CNN is achieved using three embedding layers, with an input sequence length of 20, post-truncation, and post-padding. This CNN architecture achieves a test accuracy of 0.9342844, ranking fourth among the eight studied models, even better than BERT-based models. This result is attributed to the short length of input records, with the average and maximum numbers of words per input being 22.09 and 72, respectively. A shallow CNN architecture is effective for this input length [29].

Furthermore, as shown in Table A1, the optimal performance for the LSTM model is achieved using a single layer of LSTM (not a BiLSTM), an input sequence length of 80, and post-truncation and pre-padding. The LSTM-only architecture exhibits the lowest performance, with a test accuracy of 0.9138782. This can be attributed to the relatively short length of input records. The advantages of LSTM are typically realized for longer sequences. A similar explanation applies to the Attention-only architecture.

CNN, LSTM, and Attention-based architectures are highly sensitive to the padding and truncation techniques employed. The selection of optimal techniques can significantly enhance performance, improving the CNN architecture from 0.79 to 0.93, the LSTM architecture from 0.66 to 0.91, and the attention architecture from 0.78 to 0.92.

The transformer-based model shows the second lowest performance after the LSTM-only architecture, achieving a test accuracy of 0.9203632. This is attributed to the complexity of the architecture, which requires a large dataset to converge; otherwise, it may suffer from overfitting. Additionally, since our dataset has an average input length of less than 100 characters, the complexity of the transformer architecture is unnecessary. Simpler architectures can effectively capture tweet semantics. Transformers demonstrate effectiveness in handling large datasets consisting of hundreds of thousands of records [58], or long inputs [52]. On the other hand, fine-tuning BERT-based models can be effective with smaller datasets, in the range of tens of thousands of samples, which explains the better performance of AraBERT and Arabic-BERT compared to the transformer architecture.

4.4. Ablation Study

This study aims to investigate the contribution of each feature to the performance of the best-performing deep learning model, namely, CNN-LSTM. Table 11 demonstrates that removing the user location reduces the validation accuracy from 93.81% to 87.64%. Furthermore, removing both the user name and the user location significantly reduces the validation accuracy from 87.64% to 33.98%, highlighting the importance of the user name feature for the model's performance.

Table 11. Ablation study.

Features Number	Features Name	Validation Loss	Validation Accuracy
1	Tweet text	2.0386	0.3398
2	Tweet text and user name	0.4632	0.8764
3	Tweet text, user name, and user location	0.2318	0.9381

4.5. Effectiveness vs. Efficiency

Model effectiveness is evaluated based on performance metrics such as accuracy and F1 score. Efficiency encompasses the resource consumption associated with model training, including the training time and the memory footprint of the model parameters. These factors directly influence the energy efficiency of the model deployment.

As shown in Table 12, in terms of space complexity, BERT-based models require the largest number of parameters, approximately 110 million, nearly double that of the best-performing model. This substantial parameter count suggests potential energy inefficiency during inference [23].

Regarding training time, BERT-based models also exhibit the second-longest training duration, exceeding two hours, compared to just one hour for the top-performing model.

The CNN-LSTM and CNN-Attention architectures offer a favorable balance between accuracy, space complexity, and training time, making them the first and second-best models for the fine-grained tweet geolocation task.

Table 12. Space and time analysis of the studied machine learning models.

Model	No. of Model Parameters (Million)	Execution Time per Epoch (s)	Training Time (min)
CNN	61	438	115
LSTM	**20**	284	43
CNN-LSTM	70	563	64
Attention	34	2244	256
CNN-Attention	65	278	102
Transformer	24	**262**	**38**
AraBERT	110	1210	139
Arabic-BERT	110	1334	158

Bold text is the best.

5. Discussions and Conclusions

As discussed in the Related Work section (Section 2), there is limited research on the geolocation prediction of Arabic tweets, particularly at a fine-grained level. Ref. [13] presents work closely related to ours. They address the fine-grained geolocation problem for Arabic tweets from Saudi Arabia but focus on geolocation at the city level. Additionally, they only use traditional machine learning models, resulting in a geolocation accuracy of 67.41% at a 160 km distance. In contrast, we investigate the problem at the provincial level and evaluate eight advanced deep learning techniques, including two pretrained language models (PLMs). To the best of our knowledge, Arabic PLMs have not been previously applied to the Arabic tweet geolocation task. The architecture of our best model combines CNN and LSTM layers, achieving a test accuracy of 93.85%.

On the other hand, the English tweet geolocation problem has been addressed by several studies [26,28,30,37,49,50,53,56,58]. However, these studies have been applied to different datasets, making it challenging to determine the most effective technique, particularly for fine-grained Arabic tweet geolocation. In the absence of a standardized baseline for comparison, we aim to apply the best-performing machine learning techniques from the English tweet geolocation problem to a single dataset of Arabic tweets. This approach will provide a fair assessment of their respective strengths and weaknesses for fine-grained Arabic tweet geolocation prediction, offering valuable insights and guidance for the research community.

This study has a few limitations:

- Inherent Heterogeneity: There is inherent heterogeneity in the areas of the different provinces studied. For instance, the Eastern Region province covers a vast area of 672,522 km^2, making geolocating tweets to this province not very informative. As a mitigation technique, we propose modifying the data collection phase by dividing this province into four sub-provinces, north, south, east, and west, and apply the classification problem to 16 classes instead of 13.
- Model Evaluation: The models' performance is only evaluated on a test dataset. It is preferable to assess the generalizability of the models on external datasets collected at different timeframes.
- Privacy Concerns: Predicting the geolocation of tweets may be viewed as a violation of user privacy. We emphasize that this work aims solely for social good during crises by providing collective summaries about the situation at finer granularity, without exposing individuals' home locations. Additionally, only Tweet IDs are

shared in the publicly released dataset. We also limit the dataset usage to research purposes only, by releasing it under the CC BY 4.0 license.

Our future research will explore several avenues. First, we aim to investigate the integration of complex machine learning pipelines that combine BERT-based features with Transformer models as demonstrated by Li et al. [58]. Additionally, we intend to incorporate recent Arabic PLMs such as the Twitter-specific AraBERTv02 [69] and MARBERTv2 [70] into our framework.

Furthermore, we will evaluate the effectiveness of state-of-the-art Arabic location mention recognition and ambiguity resolution models for this task, such as IDRISI-RA proposed by Suwaileh et al. [11].

Finally, we will assess the generalizability of the proposed techniques by applying them to datasets from other social media platforms, such as Instagram. This will provide valuable insights into the efficacy of these techniques across different domains.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: Experiments presented in this paper were partially carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr). This research was also partially supported by computational resources provided by Intel Developer Cloud.

Conflicts of Interest: The author declares no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MSA	Modern Standard Arabic
NER	Named Entity Recognition
CRF	Conditional Random Field
CNN	Convolution Neural Network
LSTM	Long Short-Term Memory
BiLSTM	A Bidirectional Long Short-Term Memory
LLM	Large Language Model
KSA	Kingdom of Saudi Arabia

Appendix A

Table A1. Best configurations/hyperparameters of the studied machine learning models.

Model	Words per Input	Padding and Truncation	Embedding	No. of Embedding Layers	Stacked Layers	Input Vector Length	Dropout	Filters	Learning Rate	Batch Size
CNN	20	T1P1	T—NI	3	-	700	0.3	256	5×10^{-5}	16
LSTM	80	T1P0	T—NI	1	1	700	0.6	256	5×10^{-4}	64
CNN-LSTM	80	T1P1, T1P0	T—NI	4	1	700, 300	0.6	256, 256	5×10^{-4}	32
Attention	80	T1P1	T—NI	1	-	700	0.4	1024	1×10^{-3}	16
CNN-Attention	80	T1P1	T—NI	3	-	700	0.3	256	5×10^{-5}	32
Transformer	80	T0P1	T—NI	1	2	512	0.1	2048	1×10^{-4}	64
AraBERT	40	-	T—NI	1	-	-	-	-	2×10^{-5}	32
Arabic-BERT	50	-	T—NI	1	-	-	-	-	5×10^{-5}	32

All models run using the dataset after dropping empty features. T1P1: Truncating post and padding post, T1P0: Truncating post and padding pre, T0P1: Truncating pre and padding post, T: trainable, NT: not trainable, and NI: not initialized.

References

1. Whitney, M. 39 Twitter Statistics Marketers Need to Know in 2024. Available online: https://www.wordstream.com/blog/ws/2020/04/14/twitter-statistics (accessed on 1 November 2024).
2. Masri, S.; Jia, J.; Li, C.; Zhou, G.; Lee, M.C.; Yan, G.; Wu, J. Use of Twitter data to improve Zika virus surveillance in the United States during the 2016 epidemic. *BMC Public Health* **2019**, *19*, 761. [CrossRef]
3. Kabir, M.Y.; Madria, S. CoronaVis: A real-time COVID-19 tweets data analyzer and data repository. *arXiv* **2020**, arXiv:2004.13932.
4. Shen, C.; Chen, A.; Luo, C.; Zhang, J.; Feng, B.; Liao, W. Using reports of symptoms and diagnoses on social media to predict COVID-19 case counts in mainland China: Observational infoveillance study. *J. Med. Internet Res.* **2020**, *22*, e19421. [CrossRef]
5. Broniatowski, D.A.; Paul, M.J.; Dredze, M. National and local influenza surveillance through Twitter: An analysis of the 2012–2013 influenza epidemic. *PLoS ONE* **2013**, *8*, e83672. [CrossRef] [PubMed]
6. Hiltz, S.R.; Hughes, A.L.; Imran, M.; Plotnick, L.; Power, R.; Turoff, M. Exploring the usefulness and feasibility of software requirements for social media use in emergency management. *Int. J. Disaster Risk Reduct.* **2020**, *42*, 101367. [CrossRef]
7. Huang, B.; Carley, K.M. A large-scale empirical study of geotagging behavior on Twitter. In Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Vancouver, BC, Canada, 27–30 August 2019; pp. 365–373.
8. Lamsal, R.; Harwood, A.; Read, M.R. Where did you tweet from? Inferring the origin locations of tweets based on contextual information. In Proceedings of the 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan, 17–20 December 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 3935–3944.
9. Dixon, S. Leading Countries Based on Number of X (Formerly Twitter) Users as of April 2024. Available online: https://www.statista.com/statistics/242606/number-of-active-twitter-users-in-selected-countries/ (accessed on 1 November 2024).
10. World Population Review. Saudi Arabia Population 2024 (Live). Available online: https://worldpopulationreview.com/countries/saudi-arabia-population (accessed on 1 November 2024).
11. Suwaileh, R.; Elsayed, T.; Imran, M. IDRISI-RE: A generalizable dataset with benchmarks for location mention recognition on disaster tweets. *Inf. Process. Manag.* **2023**, *60*, 103340. [CrossRef]
12. Mubarak, H.; Hassan, S. UL2C: Mapping user locations to countries on Arabic Twitter. In Proceedings of the Sixth Arabic Natural Language Processing Workshop, Kyiv, Ukraine, 9 April 2021; pp. 145–153.
13. Alsaqer, M.; Alelyani, S.; Mohana, M.; Alreemy, K.; Alqahtani, A. Predicting location of tweets using machine learning approaches. *Appl. Sci.* **2023**, *13*, 3025. [CrossRef]
14. Ghaddar, A.; Wu, Y.; Bagga, S.; Rashid, A.; Bibi, K.; Rezagholizadeh, M.; Xing, C.; Wang, Y.; Xinyu, D.; Wang, Z.; et al. Revisiting pre-trained language models and their evaluation for arabic natural language understanding. *arXiv* **2022**, arXiv:2205.10687.
15. Benajiba, Y.; Rosso, P.; Benedíruiz, J.M. Anersys: An arabic named entity recognition system based on maximum entropy. In *Proceedings of the Computational Linguistics and Intelligent Text Processing: 8th International Conference, CICLing 2007, Mexico City, Mexico, 18–24 February 2007*; Proceedings 8; Springer: Berlin/Heidelberg, Germany, 2007; pp. 143–153.
16. El Moussaoui, T.; Loqman, C. Advancements in Arabic Named Entity Recognition: A Comprehensive Review. *IEEE Access* **2024**, *12*, 180238–180266. [CrossRef]
17. El Moussaoui, T.; Chakir, L.; Boumhidi, J. Preserving privacy in Arabic judgments: Ai-powered anonymization for enhanced legal data privacy. *IEEE Access* **2023**, *11*, 117851–117864. [CrossRef]
18. Darwish, K. Named entity recognition using cross-lingual resources: Arabic as an example. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Sofia, Bulgaria, 4–9 August 2013; pp. 1558–1567.
19. Darwish, K.; Gao, W. Simple Effective Microblog Named Entity Recognition: Arabic as an Example. In Proceedings of the LREC, Reykjavik, Iceland, 26–31 May 2014; pp. 2513–2517.
20. Khalifa, M.; Shaalan, K. Character convolutions for Arabic named entity recognition with long short-term memory networks. *Comput. Speech Lang.* **2019**, *58*, 335–346. [CrossRef]
21. Kenton, J.D.M.W.C.; Toutanova, L.K. Bert: Pre-training of deep bidirectional transformers for language understanding. 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), Minneapolis, MI, USA, 2–7 June 2019; Volume 1, p. 2.
22. Antoun, W.; Baly, F.; Hajj, H. Arabert: Transformer-based model for arabic language understanding. *arXiv* **2020**, arXiv:2003.00104.
23. Abdul-Mageed, M.; Elmadany, A.; Nagoudi, E.M.B. ARBERT & MARBERT: Deep Bidirectional Transformers for Arabic. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Online, 1–6 August 2021; pp. 7088–7105. [CrossRef]
24. Benali, B.A.; Mihi, S.; Laachfoubi, N.; Mlouk, A.A. Arabic named entity recognition in arabic tweets using bert-based models. *Procedia Comput. Sci.* **2022**, *203*, 733–738. [CrossRef]
25. Qu, X.; Gu, Y.; Xia, Q.; Li, Z.; Wang, Z.; Huai, B. A survey on arabic named entity recognition: Past, recent advances, and future trends. *IEEE Trans. Knowl. Data Eng.* **2023**, *36*, 943–959. [CrossRef]

26. Huang, B.; Carley, K.M. On predicting geolocation of tweets using convolutional neural networks. In Proceedings of the International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation, Washington, DC, USA, 5–8 July 2017; Springer: Berlin/Heidelberg, Germany, 2017; pp. 281–291.
27. Rahimi, A.; Baldwin, T.; Cohn, T. Continuous representation of location for geolocation and lexical dialectology using mixture density networks. *arXiv* **2017**, arXiv:1708.04358.
28. Fornaciari, T.; Hovy, D. Geolocation with attention-based multitask learning models. In Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019), Hong Kong, China, 4 November 2019; pp. 217–223.
29. Izbicki, M.; Papalexakis, V.; Tsotras, V. Geolocating Tweets in any Language at any Location. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 89–98.
30. Mahajan, R.; Mansotra, V. Predicting geolocation of tweets: Using combination of CNN and BiLSTM. *Data Sci. Eng.* **2021**, *6*, 402–410. [CrossRef]
31. Hu, Y.; Mai, G.; Cundy, C.; Choi, K.; Lao, N.; Liu, W.; Lakhanpal, G.; Zhou, R.Z.; Joseph, K. Geo-knowledge-guided GPT models improve the extraction of location descriptions from disaster-related social media messages. *Int. J. Geogr. Inf. Sci.* **2023**, *37*, 2289–2318. [CrossRef]
32. Cheng, Z.; Caverlee, J.; Lee, K. You are where you tweet: A content-based approach to geo-locating twitter users. In Proceedings of the 19th ACM International Conference on Information and Knowledge Management, Toronto, ON, Canada, 26–30 October 2010; pp. 759–768.
33. Backstrom, L.; Sun, E.; Marlow, C. Find me if you can: Improving geographical prediction with social and spatial proximity. In Proceedings of the 19th International Conference on World Wide Web, Raleigh, NC, USA, 26–30 April 2010; pp. 61–70.
34. Duong-Trung, N.; Schilling, N.; Schmidt-Thieme, L. Near real-time geolocation prediction in twitter streams via matrix factorization based regression. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, Indianapolis, IN, USA, 24–28 October 2016; pp. 1973–1976.
35. Meng, K.; Li, H.; Wang, Z.; Fan, X.; Sun, F.; Luo, Z. A deep multi-modal fusion approach for semantic place prediction in social media. In Proceedings of the Workshop on Multimodal Understanding of Social, Affective and Subjective Attributes, Mountain View, CA, USA, 27 October 2017; pp. 31–37.
36. Miura, Y.; Taniguchi, M.; Taniguchi, T.; Ohkuma, T. Unifying text, metadata, and user network representations with a neural network for geolocation prediction. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1260–1272.
37. Simanjuntak, L.F.; Mahendra, R.; Yulianti, E. We know you are living in bali: Location prediction of twitter users using bert language model. *Big Data Cogn. Comput.* **2022**, *6*, 77. [CrossRef]
38. Mourad, A. Influence of Geographic Biases on Geolocation Prediction in Twitter. Ph.D. Thesis, RMIT University, Melbourne, Australia, 2019.
39. Khanwalkar, S.; Seldin, M.; Srivastava, A.; Kumar, A.; Colbath, S. Content-based geo-location detection for placing tweets pertaining to trending news on map. In Proceedings of the Fourth International Workshop on Mining Ubiquitous and Social Environments, Prague, Czech Republic, 23 September 2013; Citeseer: Princeton, NJ, USA, 2013; Volume 37.
40. Alruily, M. Classification of arabic tweets: A review. *Electronics* **2021**, *10*, 1143. [CrossRef]
41. Alammary, A.S. BERT models for Arabic text classification: A systematic review. *Appl. Sci.* **2022**, *12*, 5720. [CrossRef]
42. Wahdan, A.; Al-Emran, M.; Shaalan, K. A systematic review of Arabic text classification: Areas, applications, and future directions. *Soft Comput.* **2024**, *28*, 1545–1566. [CrossRef]
43. Pigott, F.; Kolb, J.; Montague, J.; Gonzales, A.; Moffitt, J. Searchtweets-v2 1.1.1 API. Available online: https://pypi.org/project/searchtweets-v2/ (accessed on 1 July 2022).
44. Leaflet Documentation. Available online: https://leafletjs.com/reference.html (accessed on 13 June 2021).
45. Schlosser, S.; Toninelli, D.; Cameletti, M. Comparing methods to collect and geolocate tweets in Great Britain. *J. Open Innov. Technol. Mark. Complex.* **2021**, *7*, 44. [CrossRef]
46. Elteir, M. Cost-effective time-efficient subnational-level surveillance using Twitter: Kingdom of Saudi Arabia case study. *Disc. Appl. Sci.* **2025**, *7*, 1. [CrossRef]
47. Alruily, M.; Manaf Fazal, A.; Mostafa, A.M.; Ezz, M. Automated Arabic long-tweet classification using transfer learning with BERT. *Appl. Sci.* **2023**, *13*, 3482. [CrossRef]
48. Aggarwal, C.C. *Neural Networks and Deep Learning*; Springer: Berlin/Heidelberg, Germany, 2018; Volume 10.
49. Liu, J.; Inkpen, D. Estimating user location in social media with stacked denoising auto-encoders. In Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, Denver, CO, USA, 5 June 2015; pp. 201–210.
50. Lourentzou, I.; Morales, A.; Zhai, C. Text-based geolocation prediction of social media users with neural networks. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 696–705.
51. Kim, Y. Convolutional Neural Networks for Sentence Classification. *arXiv* **2014**, arXiv:1408.5882.

52. Lu, H.; Ehwerhemuepha, L.; Rakovski, C. A comparative study on deep learning models for text classification of unstructured medical notes with various levels of class imbalance. *BMC Med. Res. Methodol.* **2022**, *22*, 181. [CrossRef]
53. Thomas, P.; Hennig, L. Twitter geolocation prediction using neural networks. In *Proceedings of the Language Technologies for the Challenges of the Digital Age: 27th International Conference, GSCL 2017, Berlin, Germany, 13–14 September 2017*; Proceedings 27; Springer: Berlin/Heidelberg, Germany, 2018; pp. 248–255.
54. Bahdanau, D. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
55. Vaswani, A. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**.
56. Huang, C.Y.; Tong, H.; He, J.; Maciejewski, R. Location prediction for tweets. *Front. Big Data* **2019**, *2*, 5. [CrossRef] [PubMed]
57. Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; Hovy, E. Hierarchical attention networks for document classification. In Proceedings of the 2016 conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 1480–1489.
58. Li, M.; Lim, K.H.; Guo, T.; Liu, J. A transformer-based framework for poi-level social post geolocation. In Proceedings of the European Conference on Information Retrieval, Dublin, Ireland, 2–6 April 2023; Springer: Berlin/Heidelberg, Germany, 2023; pp. 588–604.
59. González-Carvajal, S.; Garrido-Merchán, E.C. Comparing BERT against traditional machine learning text classification. *arXiv* **2020**, arXiv:2005.13012.
60. Safaya, A.; Abdullatif, M.; Yuret, D. KUISAIL at SemEval-2020 Task 12: BERT-CNN for Offensive Speech Identification in Social Media. In Proceedings of the Fourteenth Workshop on Semantic Evaluation, Barcelona, Spain, 12–13 December 2020; pp. 2054–2059.
61. AraBERT v1 and v2: Pre-Training BERT for Arabic Language Understanding. Available online: https://huggingface.co/aubmindlab/bert-base-arabertv2 (accessed on 10 October 2023).
62. Arabic BERT Model. Available online: https://huggingface.co/asafaya/bert-base-arabic (accessed on 10 October 2023).
63. PyArabic 0.6.15. Available online: https://pypi.org/project/PyArabic/ (accessed on 10 October 2023).
64. Farasa. Available online: https://farasa.qcri.org/ (accessed on 10 October 2023).
65. AraVec 2.0. Available online: https://github.com/bakrianoo/aravec/blob/master/AraVec%202.0/README.md (accessed on 10 October 2023).
66. FastText: Library for Efficient Text Classification and Representation Learning. Available online: https://fasttext.cc/ (accessed on 10 October 2023).
67. Grid5000:Home. Available online: https://www.grid5000.fr (accessed on 1 May 2020).
68. Developer Clouds for Accelerated Computing. Available online: https://www.intel.com/content/www/us/en/developer/tools/devcloud/overview.html (accessed on 1 February 2023).
69. AraBERTv0.2-Twitter. Available online: https://huggingface.co/aubmindlab/bert-large-arabertv02-twitter (accessed on 1 November 2024).
70. MARBERTv2. Available online: https://huggingface.co/UBC-NLP/MARBERTv2 (accessed on 1 November 2024).