Arabic Location Named Entity Recognition for Tweets using a Deep Learning Approach

Bedour Swayelh Alzaidi, Yoosef Abushark, Asif Irshad Khan
Department of Computer Science, Faculty of Computing and Information Technology
King Abdulaziz University, Jeddah, Saudi Arabia

***Abstract*—Social media sites like Twitter have emerged in recent years as a major data source utilized in a variety of disciplines, including economics, politics, and scientific study. To extract pertinent data for decision-making and behavioral analysis, one can use Twitter data. To extract event location names and entities from colloquial Arabic texts using deep learning techniques, this study proposed Named Entity Recognition (NER) and Linking (NEL) models. Google Maps was also used to obtain up-to-date details for each extracted site and link them to the geographical coordination. Our method was able to predict 40% and 48% of the locations of tweets at the regional and city levels, respectively, while the F-measure was able to reliably identify and detect 63% of the locations of tweets at a single Point of Interest.**

Keywords—NER; Named Entity Recognition; NEL; named entity linking; event; location; deep learning; Arabic

I. Introduction

In today's world, information gathering plays an important role in reducing the effects of urgent events, so business managers or emergency agents always focus on gathering up-to-date information to help them make better decisions. So, to assure more precise dealing with situations, different resources of information must be used such as the data posted on social networking sites. The research introduces Twitter as an additional information source, but there are some difficulties and restrictions when using this Information such as data reliability and quality, as tweets may be in different languages and may contain hash tags, internet links, images, or even videos. Also, the geospatial feature of the information is important for disaster response. As for the techniques used to gather information such as Keyword monitoring which is subject and biased and more likely to contain false data. Many models have been developed to extract location-named entities from an informal text [1],[5], however, those models are applied to English text only, And the Arabic language recognition models are not linked to its geographical coordinates and are low accuracy compared to English or other languages that is because there are no functions or converters available to convert the dataset to a format that a model can use it for training. In this paper, we propose a method to extract event location names from unstructured Arabic text documents via NLP methods using the Deep Learning technique [6], first by collecting text data from Twitter and pre-processing it to fit inside the deep learning model and utilizing an Arabic location named-recognition model. In addition, each extracted location will be linked with geographical coordination by using existing knowledge such as Google Maps.

There are several ways to acquire Twitter data which can be divided into two main groups: First way: Purchasing the data from third-party or commercial data vendors [7], which provides a user-friendly GUI and visual environment for better use of the gathered data. Such an option is not recommended for the academic context due to the expenses. Also, users cannot modify the algorithms used for different operations which is a major drawback. Second way: Direct data collection through Twitter API which provides a straightforward way to query and retrieve publicly available tweets.

There are three main types of APIs provided by Twitter which are: REST API [6], Streaming API [9], and Ads API. Both REST API and Streaming API are very similar and provide access to Twitter data, but the author chose to use REST API as it is more useful for this case.

One of the most used REST endpoints is the search API which returns a collection of relevant Tweets matching a specific query. It should be noted that the REST API search service is focused on relevance and not completeness, which means that not all the tweets will be indexed. All interactions with the Twitter API require authentication using the OAuth protocol [10], which provides access to protected API services and server-side resources on behalf of the API owner. For data preparation [11], Twitter API returns tweets in JSON format [12] which is a human-readable data interchange format. Working with JSON tweets requires validation, data cleaning, and data transformation procedures before sorting them in a database. For data cleaning, it is mostly removing unwanted information or redundancy in the raw tweet which can be a group of several tasks that are: (Filtering based on the language "Arabic", Removing usernames, Removing URLs, punctuations, multiple dots, and extra spaces, and Removing words containing non-Arabic characters). Mostly, a tweet in JSON format is composed of several objects which may be more than 70 objects per tweet, some of them are single objects and some are embedded within other objects. Not all the objects will be used but selected objects during different phases for developing and implementing the framework.

Analyzing the Arabic language is challenging due to its complex linguistic structure, also the informal Arabic used in tweets is noisy and poorly structured [13] as in the Arabic region there are more than 25 dialects of Arabic. Text mining is the technique of analyzing mass amounts of text to identify insights and outcomes of new information and it is purely based on Artificial Intelligence (AI). This technique is widely involved with NLP (Natural Language Processing) which is considered an exciting field that enables computers to analyze human language. Previous research proposed a supervised algorithm that produces a weighted average of word embedding [14]. The purpose of the research is to implement text mining and NLP techniques to investigate NER for Arabic tweets using ML (Machine Learning). Recently DL (Deep Learning) has been strongly implemented in NER systems. Named Entity Recognition (NER) is a sub-task of Information Extraction (IE) that aims to look at and classify named entities presented in unstructured tasks and identify useful information such as person name, location, etc. Named Entity Linking (NEL) is the task of assigning a unique identity to the entity. Python supports NER through the library NLTK. NER and NEL can automatically scan entire articles and reveal organizations, locations, and people's names. Google Map API is used for entity analysis as it provides services related to map information.

We propose an approach that is like [15] in that it handles two binary classification tasks. We evaluate our models to classify Arabic tweets about event detectors and identify the heights performance for the training model's accuracy. Although their approach successfully detects the target messages utilizing complex and skill-required machine learning, the results might be further improved by adopting less time-consuming and labor-intensive approaches using both automated and human data labeling methods. The next part provides a summary of the current methodologies and relevant research on location inference of Twitter data after discussing the methods currently used to identify Twitter events.

The objectives of this study are as follows:

- Collect Arabic text data from the Twitter platform using Twitter API.
- Preprocess the collected data in such a way that it is well visualized and can fit inside the deep learning model using data mining, text mining, rescaling, and reshaping techniques.
- Utilize the deep learning algorithms to build an Arabic location named recognition model.
- Design a named entity link method that links the extracted location entities with their geographical coordinates.
- Optimize and retrain the developed models by using several data mining, data preprocessing, scaling data, and many other data cleaning techniques.
- Evaluate the developed models' performance in terms of classification matrix, f1 scores, precision, recall, and accuracy.
- Evaluate performance in real-time to identify issues and the accuracy of the model.

The contents of this paper are organized as follows. In Section II, we give a brief description of the existing related works. In Section III, we describe the proposed framework. In Section IV, the Implementation details of the proposed framework are discussed. In Section V, Result and Evaluation of the proposed model is discussed. Finally, the conclusion is shown in Section VI.

II. Related Work

A. Event Detection on Twitter

An event is often thought of as anything that takes place at a certain time and location [16],[17]. A qualitatively significant change in anything might also be referred to as an event. In the context of this study, the word "event" can refer to any condition in the real world that causes an increase in the frequency and quantity of tweets that are likely to include pertinent information about the circumstance [18]. A critical first step in incorporating Twitter as an extra source of information for any application is the identification of the tweets that were generated in reaction to a real-world occurrence. Event detection methods are often divided into two categories: open events and events that are specific to a certain domain (supervised or unsupervised). As a result, the bulk of research done on Twitter event detection may be grouped according to these criteria. According to the kind of event, the method of detection, and the intended use, (Table I) shows a taxonomy of research on Twitter event detection based on event type and approach.

TABLE I. RESEARCH ON TWITTER EVENT DETECTION

Reference	Event Type	Approach	Application
Hagras et al. (2017)[21]	Specific	Supervised	Natural Disaster tweets detection
Ragini et al. (2018) [22]	Specific	Supervised	Crisis Mitigation using Twitter
AL-Smadi et al.(2016)[19]	Open	Unsupervised	General Event Detection
Alabbas et al.(2017) [20]	Specific	Supervised	High-risk floods Detect
Wang et al. (2015) [23]	Open	Unsupervised	General Event Detection
Alomari et al. (2020)[24]	Specific	Unsupervised	Traffic-related event detection
Toujani et al. (2108)[25]	Specific	Unsupervised	Natural Disaster tweets detection
Fu et al. (2020) [26]	Open	Unsupervised	General Event Detection
Rezaei et al. (2022) [27]	Open	semi-Supervised	General Event Detection

Hagras et al. (2017) [21] classified and evaluated tweets related to the Japan Tsunami using the Latent Dircherilet Allocation (LDA) topic analysis approach. Selected 196 tweets for the test set and 6700 tweets for the training set, resulting in 76% accuracy with successful detection. However, this approach may be improved by adding more datasets to speed up processing in real-time and increase accuracy. Emotion analysis and ML algorithms were used in a crisis governance technique that [22] presented. They classified the information depending on its requirements. After using a Support Vector Machine (SVM) to assess the data acquired from Twitter via human attitudes, both positive and negative. Although the suggested approach makes it easier for emergency crews to identify disastrous situations and take appropriate action immediately, it has some limitations when using social network data for crisis mitigation, specifically the ambiguity in obtaining crisis data from various sources and the absence of the proper criterion. However, these issues may be resolved by collecting data from several sources to properly classify the data and improve precision.

AL-Smadi et al. (2016) [19] describe a knowledge-based approach for fostering event extraction out of Arabic tweets. They used an unsupervised rule-based technique for event extraction. Results show that the approach has an accuracy of, 75.9% for event trigger extraction, 87.5% for event time extraction, and 97.7% for event type identification.

Alabbas et al. (2017) [20] suggested a classification model trained on 3700 Arabic tweets to identify high-risk floods. Several machine learning algorithms were employed, which include k-NN, J48, NNET, SVM, and C5.0. With the training matrix containing the chosen terms and their related values, the classification algorithms were trained. TF-IDF (Term Frequency-Inverse Document Frequency) weights. The outcomes demonstrated that SVM performed with better accuracy.

Toujani et al. (2018) [25] suggested a novel method that uses social networks as the primary source to identify event information after a natural disaster. Then, based on the period of risk, they group people into tiers. Reporters can benefit from this clustering process by streamlining the way they obtain information in urgent situations. Additionally, they applied fuzzy theory techniques to these incidents to enhance clustering performance and get rid of opacity in the data that was gathered.

Alomari et al. (2020) [24] created a lexicon that makes it easier to identify traffic occurrences in Saudi Arabia using Twitter and the big data methodology. For Arabic and Saudi dialect terms, they also applied sentiment analysis based on the lexicon method. To improve the indexing and categorization of nonspecific events [23] provide unsupervised algorithms that create structured lexicon-based event information. Additionally, other research has focused on early event prediction using the Twitter data stream.

Fu et al. (2020) [26] perform an open-domain event text generation task with an entity chain as its skeleton. To build this dataset, a wiki-augmented generator framework containing an encoder, a retriever, and a decoder is proposed. The encoder encodes the entity chain into hidden representations while the decoder decodes from these hidden representations and generates related stories. The retriever is responsible for collecting reliable information to enhance the readability of the generated text.

Rezaei et al. (2022) [27] proposed a semi-supervised framework for the detection of data events on Twitter. Then, they used the Hierarchical Attention Network (HAN) method to categorize the data events. A virtual backbone was employed so that the stream data could be divided into one or more classes with various grades.

To extract location-named entities from informal texts like those seen on social media sites, many models have been developed [1],[2],[3],[4],[5]. While other languages have gotten comparatively less attention, those models were created to simply apply to the English text. When evaluated in the informal Arabic language, most location-named entity identification models or algorithms underperformed because they were unable to identify the precise pattern of Arabic keywords in the data. Additionally, the retrieved location's geographic coordinates on those systems are not connected to it. To the best of our knowledge, Arabic text recognition models perform poorly when compared to those in English or other languages. This is because any machine learning algorithm requires mathematical parameters to train. Because there are many programming functions for English tweets, we can readily transform them into mathematical form; but, for Arabic, there are no such converters or functions that can quickly transform our dataset into a format that the model can effectively train on.

The globe has been producing data on social media on the terabyte scale, and this unused data might be the key to many future research projects [8]. Obtaining data on behaviors affected by shifting geographies is one promising study area, allowing researchers to make focused, well-informed judgments [1]. These unstructured Arabic text documents are a significant source of such data, and we specifically want to extract geographic information from them [2]. However, most of this data is not provided in an obvious manner and must be retrieved using different NLP techniques [3]. Machine learning methods have traditionally been used to do this [4]. The technique of employing neural networks to learn a task is known as deep learning. Deep learning significantly outperforms all other conventional learning techniques and aims to replicate how the human brain functions. In this area, deep learning has not yet been applied. However, English language research makes up most of these studies. Arabic is one language that has not received enough attention. The few Arabic works that have been created, have problems with generalization. They were completed by concentrating on a tiny, extremely conservative group, leaving a sizable portion unaffected.

The location extraction and linkage from Arabic-language tweets will benefit from the findings of this study. To the best of our knowledge, there is no clear study that has tackled location extraction and linking from Arabic tweets using deep learning approaches, in addition to the scant amount of research on location extraction from Arabic text.

III. Framework

Framework design. which describes the framework's design proposed by the author to achieve the aim of this research which is only limited to tweets in the Arabic language.

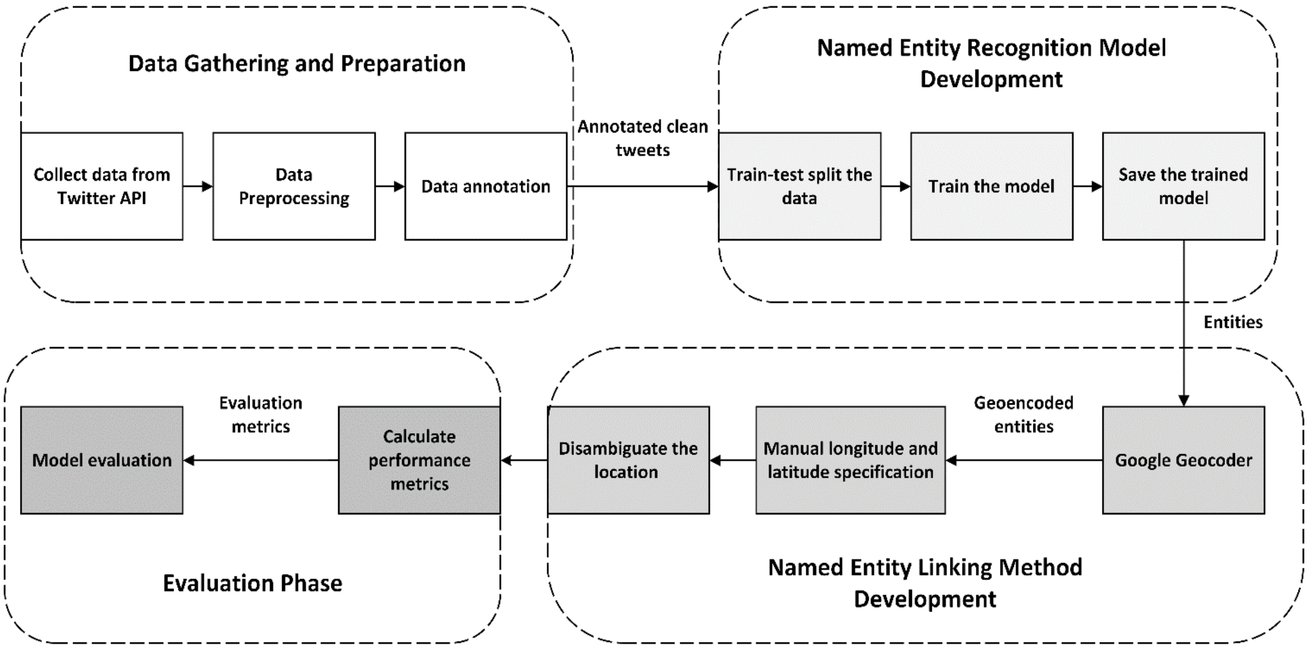

Fig. 1. Framework design.

1) First part: Data collection and Preparation which focus on the collection of Twitter data, the pre-processing, and the labeling of the data, which consists of two main components:

a) The data collection component establishes a persistent connection with the Twitter API and collects public tweets in real time.

b) The Data Preparation component focuses on the cleaning and pre-processing of the collected tweets.

Data preparation includes a few steps:

- Text Processing: a crucial step as the language used by Twitter is informal. Tweets contain noise that must be removed which may be spelling shortcuts, misspellings, new words, URLs, hashtags, tags, emoticons, or HTML characters.
- Data Annotations: an integral part of the ML model training process, because the NN learns from the pattern that exists in the annotated data and uses such patterns in new unseen data. So, we assign various entity tags to the text data.

2) Second part: Location Detection using the NER model which is created by the training set of data obtained in the first part. The NER model analyses the properties of the human Natural Language and the linguistic patterns of the incoming tweet and figures out possible words/phrases that indicate the location. Geocoding service is applied to convert the input location names extracted by the NER to output coordinates and a structured address of the location. The research uses the outputs of the Geo-coding service to hierarchize the location names.

3) Third part: Develop a Named Entity Linking method where the tweet's locations are specified and disambiguated. The location of the non-geotagged tweets can be a challenge in this context, so the Location inference component is designed to predict the location of the tweet in the absence of geo-tagging. The Location inference component uses NEL to perform the task of mapping words of interest from the tweet text to corresponding unique entities in a target knowledge base such as Wikipedia. The location assigned by the location inference component is the inferred location of a tweet.

4) Fourth part: Evaluation Phase in which the results and accuracy level are evaluated using a set of metrics for the assessment of the accuracy and performance of the system. Performance metrics are such as [Recall, precision, and F-measure] to determine the accuracy of the event-relatedness. Also, to evaluate the performance of the location inference component, Mean and Median distance errors are calculated between the inferred and actual locations. The timeliness of the processes done by the prototype is evaluated using several Twitter datasets of different sizes.

IV. Implementation

The system implementation is shown in The implementation architecture, delivers the functionalities of the proposed framework which are:

1) Twitter data collection using Twitter REST API which can cope with a large number of tweets and handle errors automatically.

2) Data preparation which includes tasks of raw data validation, reduction, annotating, and cleaning.

3) Identification of the event-related tweets and inferring their location.

4) Presentation of the results in an appropriate data format.

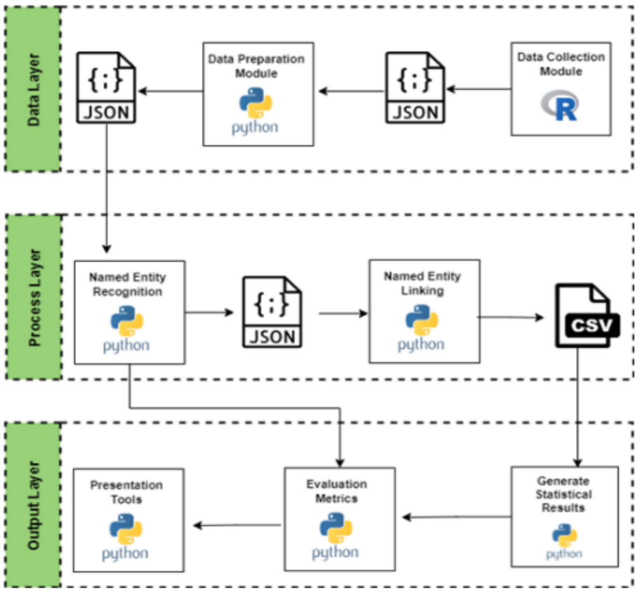

Fig. 2. The implementation architecture.

We followed the Ntair technique for implementing the proposed framework, which is a classic technique for dividing a complex development task into manageable parts (Layered Architecture)[28]. The implementation architecture consists of three layers: Data Layer, the Process Layer, and the Output Layer. The prototype implementation is based on open-source tools using R and Python which support rapid development and fast prototyping [30]. Each section will be explained as shown in Fig. 2, The Implementation Architecture.

A. *Data Layer*

The data layer performs the tasks of collecting, preparing, and sorting Twitter data. A combination of R and Python packages and libraries were used for the implementation of this layer.

1) Data collection module: Using the Twitter API, Arabic tweets of certain keywords related to different events were collected. The Module performs the following procedure:

- Perform API authentication using Twitter OAuth credentials.s.
- Establish and maintain a persistent connection with the API.
- Infer the approximate location of the event-related tweets from the potential sources of location information embedded within the Twitter data.

Data retrieval on Twitter is done using the web scraping method which aims to obtain information from a website and turn it into a structure that is easy to understand, store and analyze in a database. Data retrieval was done using RStudio with R programming language with the help of the R package “rtweet” which provides different functions to extract data from Twitter REST APIs. The author extracted tweets with Arabic keywords of different events such as (festivals, accidents, tornados, etc.) and stored the tweet data in feather files which are used to exchange data between Python and R. For the authentication process, The OAuth server generates the OAuth credentials (Consumer Key, Consumer Secret, Access Token and Token Secret) which the user must manually paste into the code fields in the OAuth handler of the rtweet library, as shown in the following The OAuth credentials.

The API Listener writes the raw JSON tweets to a temporary dataset which acts as a data buffer between the data collection module and the data collection module. This data buffer is nothing but a simple text-based JSON file. The collected tweets should be structured in the form of an array by the API Listener where each Twitter object is an element of this array within this file.

```
library(rtweet)
```

```
# create token named "twitter_token"
token <- "********************************************" # From dev.twitter.com
token_secret <- "*****************************************" # From dev.twitter.com

consumerKey <- "**********************************************"
consumerSecret <- "*********************************************"

twitter_token <- create_token(
  consumer_key = consumerKey,
  consumer_secret = consumerSecret,
  access_token = token,
  access_secret = token_secret)
```

Fig. 3. The OAuth credentials.

2) Data preparation module: This module which is implemented in python comprises the following steps of validating and cleaning the collected raw tweets:

- Reads the temporary dataset of the collected tweets.
- Handles the JSON validation for each tweet within the dataset.
- Change the annotation file format to fit the SpaCy requirements.
- Performs data cleaning and pre-processing.
- Writes the cleaned tweets to a structured JSON file.

The temporary dataset is loaded into the python environment, then the module should read JSON tweets and check them for JSON schema conformance. For pre-processing the text as shown in Steps of pre-processing a sample tweet, the first step is iterating over the tweets to remove all numbers, English alphabets, and punctuations such as commas (,), period (.), semi-colons (;), colons (:), question marks (?), Arabic question marks, semicolons and so on, to reduce the size of the feature set. Further, removing the hash (#) and underscore (_) symbols from the hashtags. Also, the author removed Arabic diacritic and vowel marks such as Shaddah, which is a diacritic shaped like a small written "w." After that, the text is divided into words (tokens). The tokens are normalized to replace letters that have different forms into the basic shape. Finally, the Stop Words are filtered using the Arabic stop words list in the Natural Language Toolkit (NLTK). The author modified the list to add the missing word and normalize the words before using them.

Furthermore, checking the result of the pre-processing phase before starting the classification. If the remaining number of tokens is equal to zero, the tweet is excluded from the analysis. The following figure shows the steps of pre-processing applied to a sample tweet:

Fig. 4. Steps of pre-processing a sample tweet.

Once the dataset is cleaned, it is stored in a new JSON file that will be later accessed by the Process Layer for conducting the processes associated with the location extraction using NER.

3) Data annotation: it is an important part of the ML training process as the NN will be trained from the patterns that exist in the annotated data and tries to identify such patterns for unseen data. Data Annotation is carried out using Label Studio which is an open-source data annotation tool whether locally or on a server using a simple user interface. For NER, we would need to assign various entity tags to the text data we have which are used later by the NN to identify a generalized pattern. The author identified three main entities (Point of interest, City, Region) POI, CTY, REG. Then using word frequency analysis, evaluated each tweet against term classes that are associated with practical locations related to certain events to find the tweets are connected to which location. So, the author considered random location-related tweets which are manually annotated then by applying different entity tags to the existing data to perform NER. The model analyses these collections of tweets with their entity tags to find a generalized pattern. Such a setup is simple for NER tagging, with only the names of tags changed on the customization panel. File imported in the Label Studio is of the supported formats (.csv, .json, .tsv), and after the job is done exporting the annotated data in a format of the same previous supported formats. The time-consuming process of data annotations is made simpler with Label Studio.

B. Process Layer

The core implementation layer deals with tasks of location extraction of the collected tweets.

Model Training:

Data is split into two sets: training and testing sets. The annotated tweets are used for the training set to train the model. Then the model is used to identify the location entities in the testing set. The NER Model development process is carried out using the SpaCy library. The training data is converted in a form of tuples containing text data and a dictionary which is a format supported by SpaCy. The following steps are carried out to train the model:

1) Load the model or create an empty model using spaCy blank with the ID of desired language (Arabic in this case).

2) Add the new entity label to the entity recognizer using the add_label method.

3) Loop over the examples and call the Natural Language Processor (NLP) which steps through the words of the input. Predict each word. Then consult the annotations, to see whether it was right. If it was wrong, it adjusts its weights so that the correct action will score higher next time.

4) Save the trained model.

5) Test the model to make sure the new entity is recognized correctly.

6) Explore the results of the evaluation.

Location Named Entity Linking

Using a Geo-coding service, it takes an input text such as an address or the name of a place and returns a latitude/longitude location on the Earth's surface for that place. The locations are decoded using geocoding APIs offered by Google Places and OpenStreetMap. The geocoded service results in a JSON-formatted list of geocoded addresses representing possible matches, including longitude and latitude coordinate and well-formatted address with all the super regions included. If the extracted location is linked to more than a location, The Linker will disambiguate the locations by estimating the distance between the geospatial location and candidate geospatial locations.

C. Output Layer

This layer uses the structured data of the processed tweets to display the results by generating statistical results and evaluation metrics. Results are output from two main operations:

1) Displacement Computations where the displacement between Twitter's entity location points and the coordinates that result from Google Geocoder API, to assess the Geocoding accuracy.

2) Generating statistical results: the frequency distributions of displacement values for SpaCy are computed. Then various groupings of SpaCy entities are evaluated to find the entity grouping giving the highest accuracy in predicting.

3) Evaluation metrics: Three metrics are commonly used for NER and NEL tasks which are Precision, Recall, and F1.

V. Results and Evaluation

A. Data Collection and Preparation

Twitter REST API was used to collect tweets with a specific keyword, where the author conducted the experiment using "accident" as the initial keyword.

Data collection continued up to weeks, during which 11134 unique tweets were collected and stored in different feather files. Then the feather files were concatenated to construct the initial dataset that is processed by the Data preparation module. After the data preparation module, the initial dataset was reduced to 9894 unique tweets after pre-processing and further reduced to 1982 tweets after annotation. A reference dataset of 320 tweets can be defined as a representation of the correctly classified sample tweets for verifying the model is manually annotated by experts.

B. *Evaluation of the NER Model*

For measuring the performance, each output word was compared to the truth file and assigned a value of True Positive (TP), False Positive (FP), True Negative (TN), or False Negative (FN).TP, FP, TN, and FN counts for all tweets were added and Precision, recall, and F1 scores were calculated per entity (POI, REG, and CTY as the core entity sets for evaluation). The author only relied on F-score to measure the optimal performance of the results as it combines both precision and recall metrics. The F-measure was able to correctly identify and detect 63% of the location of tweets in a certain Point of interest while on the region and city level, the system was only able to predict 40% and 48% of the location of tweets, respectively, as described in Table II.

TABLE II. RESULTS OF THE NER MODEL

Entity	Precision	Recall	F1
POI	77	54	63%
REG	60	30	40%
CTY	57	41	48%

C. *Evaluation of the NEL Model*

Second phase in the proposed framework is NEL, this phase aims to extract the geolocation for the extracted entities. Table III represent the evaluation of the NEL model, where:

1) The (Label) field represents the corresponding location name label (POI, CTY, REG) assigned to each tweet.

2) (Entity), (lat pred) and (lng pred) fields represent the name and the geo-coordinates (latitude and longitude) of the extracted location.

3) (lat act) and (lng act) fields correspond to the geotagged coordinates of the tweets that extracted manually (Ground truth).

4) The (Distance_km) field represents the distance error (the distance between the actual location and the inferred location of a tweet) which is calculated using the Haversine formula. This field is used as the evaluation metric to measure the accuracy of the proposed solution results. As for the results, in 191 out of 320 tweets with a percentage of (61.3%), the inferred location was at a distance equal to or smaller than 10 km from their actual location. And in 88 tweets (27.5%) the inferred location was located within 10 to 50 km of their actual location. And among the remaining tweets, the locations of 10 tweets (3.1%) were located within 50 to 100 km. While 26 tweets (8.1%) have a distance error greater than 100 km. Fig. 5 below shows the accuracy of the inferred location based on distance error.

TABLE III. RESULTS OF THE NEL MODEL

	Lat act	Lng act	label	entity	Lat pred	Ing pred	Distance km
1	30.04	31.23	['POI', 'CTY']	['القاهرة','طريق رئيسي']	30.04	31.23	0.17
2	24.66	46.68	['POI']	[' طريق الملك سعود']	21.81	39.08	838.17
3	29.30	48.029	['POI']	[' شوارع ميدان حولي']	29.33	48.02	2.84
4	30.01	31.58	['POI']	['الأوسطي']	30.01	31.58	0.004
5	30.01	31.58	['POI']	['الأوسطي']	30.01	31.58	0.004

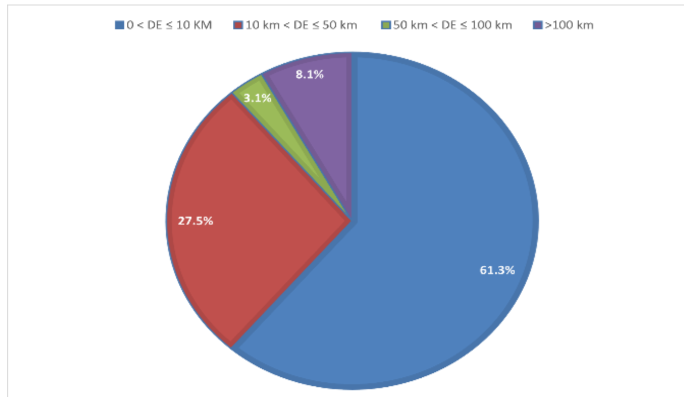

Fig. 5. The accuracy of the inferred location.

VI. CONCLUSION AND FUTURE WORK

The framework proposed can assist emergency/event managers to locate necessary warnings and monitor situations. Also, it can provide information on the type and scope of damage that resulted from a certain event. Twitter data collection and preparation is considered one of the foremost contributions of the research. This study proposes a method to process tweets to determine their degree of relatedness to accident events and infer their approximate location. First of all, getting to know the nature and structure of Twitter data together with the identification of the optimum approaches to its collection, storage, and preparation by the emergency response context, by itself, is an essential knowledge area and can be considered as one of the foremost contributions of this research. However, this assertion, along with the possibility of the adoption of the framework in other types of incidents such as bushfires or terrorist attacks, is subject to further research. Another point is that the framework only focused on Arabic tweets. This could be seen as a major drawback in using the framework on a global scale or in countries that speak a language other than Arabic, especially languages that use non-ASCII characters like in our case (e.g., Turkish and Chinese). Therefore, exploring possible solutions to this issue could form a future research topic. The next section outlines several future research directions.

For future research, the manual annotation process is time-consuming and laborious, so an investigation into automation methods is much recommended. Also, the research only considered the textual part of the tweet excluding any images, videos, or links that may likely provide valuable information, so providing visualization of the tweet's content may be suggested for future work. Also, semantic-based solution for accurate assessment of emoticons, acronyms, or slang. As mentioned earlier, extending the applicability of the proposed framework to other commonly spoken languages can improve the performance of the framework. For location accuracy, integration of other sources of information such as remote sensing data, sensor network data, and crowd-sourced mapping repositories can improve the measurement and minimize the error. For Geo-coding, using other sources for location names and geographical databases other than Google Map API such as GeoNames may enhance the applicability of the proposed framework at larger geographic scales.

REFERENCES

[1] Etaiwi, W., Awajan, A. and Suleiman, D. (2017), Statistical Arabic name entity recognition approaches: A survey, Procedia computer science 113, 57–64.

[2] Feriel, A., & Kholladi, M. K. (2015). Automatic Extraction of Spatio-Temporal Information from Arabic Text Documents. Int. J. Comput. Sci. Inf. Technol, 7, 97-107.

[3] Qiu, Q., Xie, Z., Wu, L. and Tao, L. (2019), Gner: A generative model for geological named entity recognition without labeled data using deep learning, Earth and Space science 6(6), 931–946.

[4] Li, J., Sun, A., Han, J., & Li, C. (2020). A survey on deep learning for named entity recognition. IEEE Transactions on Knowledge and Data Engineering, 34(1), 50-70.

[5] Ali, M. N., Tan, G. and Hussain, A. (2018), Bidirectional recurrent neural network approach for Arabic named entity recognition , Future Internet 10(12), 123.

[6] Qiu, Q., Xie, Z., Wu, L., & Tao, L. (2019). GNER: A generative model for geological named entity recognition without labeled data using deep learning. Earth and Space science, 6(6), 931-946.

[7] Kim, A. E., Hansen, H. M., Murphy, J., Richards, A. K., Duke, J., & Allen, J. A. (2013). Methodological considerations in analyzing Twitter data. Journal of the National Cancer Institute Monographs, 2013(47), 140-146.

[8] About Twitter | Our company purpose, principles, leadership [n.d.]. URL:https://about.twitter.com/en/who-we-are/our-company.

[9] API reference index | Docs | Twitter Developer Platform [n.d.]. URL: https://developer.twitter.com/en/docs/api-reference-index.

[10] OAuth 2.0 — OAuth [n.d.]. URL: https://oauth.net/2/.

[11] Thakare, S. B. and Gawali, S. Z. (2010), 'An effective and complete preprocessing for web usage mining', International Journal on Computer Science and Engineering 2(03), 848–851.

[12] JSON [n.d.]. URL: http://www.json.org/json-en.html.

[13] AlOtaibi, S. and Khan, M. B. (2017), Sentiment analysis challenges of informal Arabic, IJACSA) International Journal of Advanced Computer Science and Applications 8(2).

[14] Gupta, S., Kanchinadam, T., Conathan, D. and Fung, G. (2020), Task optimized word embeddings for text classification representations, Frontiers in Applied Mathematics and Statistics 5, 67.

[15] Ameen, Y. A., Bahnasy, K., & Elmahdy, A. Classification of Arabic Tweets for Damage Event Detection.

[16] Allan, J., Papka, R. and Lavrenko, V. (1998), On-line new event detection and tracking, in 'Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval', pp. 37–45.

[17] Nallapati, R., Feng, A., Peng, F. and Allan, J. (2004), Event threading within news topics, in 'Proceedings of the thirteenth ACM international conferenceon Information and knowledge management', pp. 446–453.

[18] Dou, W., Wang, X., Skau, D., Ribarsky, W. and Zhou, M. X. (2012), Leadline: Interactive visual analysis of text data through event identification and exploration, in 2012 IEEE Conference on Visual Analytics Science and Technology (VAST), IEEE, pp. 93–102.

[19] Mohammad, A. and Qawasmeh, O. (2016), Knowledge-based approach for event extraction from arabic tweets, International Journal of AdvancedComputer Science & Applications 1(7), 483–490.

[20] Alabbas, W., al Khateeb, H. M., Mansour, A., Epiphaniou, G. and Frommholz,I. (2017), Classification of colloquial arabic tweets in real-time to detect high risk floods, in 2017 International Conference On Social Media,Wearable And Web Analytics (Social Media), IEEE, pp. 1–8.

[21] Hagras, M., Hassan, G. and Farag, N. (2017), Towards natural disasters detection from twitter using topic modelling, in '2017 European Conference on Electrical Engineering and Computer Science (EECS)', IEEE, pp. 272–279.

[22] Ragini, J. R., Anand, P. R. and Bhaskar, V. (2018), Big data analytics for disaster response and recovery through sentiment analysis, International Journal of Information Management 42, 13–24.

[23] Wang, Y., Fink, D. and Agichtein, E. (2015), Seeft: Planned social event discovery and attribute extraction by fusing twitter and web content, in Proceedings of the International AAAI Conference on Web and Social Media, (Vol. 9, pp. 483–492).

[24] Alomari, E., Mehmood, R. and Katib, I. (2020), Sentiment analysis of arabic tweets for road traffic congestion and event detection, in Smart infrastructure and applications, Springer, pp. 37–54.

[25] Toujani, R. and Akaichi, J. (2018), Event news detection and citizens community structure for disaster management in social networks', Online Information Review 43(1), 113–132.

[26] Fu, J., Liu, P. and Neubig, G. (2020), Interpretable multi-dataset evaluation for named entity recognition, arXiv preprint arXiv:2011.06854.

[27] Rezaei, Z., Eslami, B., Amini, M. A. and Eslami, M. (2022), Event detection in twitter by deep learning classification and multi label clustering virtual backbone formation, Evolutionary Intelligence pp. 1–15.

[28] Chengjun, W. (2009), The analyses of operating system structure, in 2009 Second International Symposium on Knowledge Acquisition and Modeling, (Vol. 2, IEEE, pp. 354–357).

[29] Ginige, A. (1998), Web engineering: Methodologies for developing large and maintainable web-based information systems, in 'IEEE International Conference on Networking', pp. 89–92.

[30] Wallace, B. C., Dahabreh, I. J., Trikalinos, T. A., Lau, J., Trow, P. and Schmid, C. H. (2012), Closing the gap between methodologists and end-users: R as a computational back-end, Journal of statistical software 49, 1–15.