Hindawi
Journal of Sensors
Volume 2022, Article ID 1222692, 14 pages
https://doi.org/10.1155/2022/1222692

Research Article

Effective Preprocessing and Normalization Techniques for COVID-19 Twitter Streams with POS Tagging via Lightweight Hidden Markov Model

Senthil Kumar Narayanasamy,[1] **Yuh-Chung Hu**,[2] **Saeed Mian Qaisar**,[3] **and Kathiravan Srinivasan**[4]

[1]*School of Information Technology and Engineering, Vellore Institute of Technology, Vellore 632014, India*
[2]*Department of Mechanical and Electromechanical Engineering, National Ilan University, Yilan 26047, Taiwan*
[3]*Electrical and Computer Engineering Department, Effat University, Jeddah, Saudi Arabia*
[4]*School of Computer Science and Engineering, Vellore Institute of Technology, Vellore 632014, India*

Correspondence should be addressed to Kathiravan Srinivasan; kathiravan.srinivasan@vit.ac.in

Received 4 July 2022; Accepted 20 July 2022; Published 2 August 2022

Academic Editor: Mohit Mittal

The major focus of this research work is to refine the basic preprocessing steps for the unstructured text content and retrieve the potential conceptual features for further enhancement processes such as semantic enrichment and named entity recognition. Although some of the preprocessing techniques such as text tokenization, normalization, and Part-of-Speech (POS) tagging work exceedingly well on formal text, it has not performed well when it is applied into informal text such as tweets and short messages. Hence, we have given the enhanced text normalization techniques to reduce the complexity persist over the twitter streams and eliminate the overfitting issues such as text anomalies and irregular boundaries while fixing the grammar of the text. The hidden Markov model (HMM) has been pervasively used to extract the core lexical features from the Twitter dataset and suitably adapt the external documents to supplement the extraction techniques to complement the tweet context. Using this Markov process, the POS tags are identified as states of the Markov process, and words are the desired results of the model. As this process is very crucial for the next stage of entity extraction and classification, the effective handling of informal text is considered to be important and therefore proposed the most effective hybrid approach to deal with the issues appropriately.

1. Introduction

In recent years, the impact of social media sites is rampant and gaining huge popularity among social media users consistently. Particularly, Twitter has gained huge momentum among the users and providing an open platform for information exchange in a variety of events and situations. The events can be classified as political crisis, natural calamities, disasters, celebrations, etc. Recently, the tweets related to COVID-19 have been very pervasive and made a prominent impact on the government agencies to take immediate actions. Also, the information pertaining to coronavirus has been used for travelers and business people to take preliminary actions for their proposed plans. The information posted in Twitter are needs to be organized and classified according to its credibility score and further pave way for segregating them into primary and secondary information. Normally, the secondary information in a tweet is a retweet [1]. Nowadays, users prefers to use social media platforms such as Twitter for getting the latest news, and there are high chances of drawing wrong conclusions by reading false news. Hence, the demand for implementing the credible system that is capable of identifying the correct news and classifying them into the right emotions provides the right information for the decision-making processes.

Therefore, automatic detection of events such as people, organizations, locations, and other entities from unstructured content is challenging and has shown very poor performance due to its unorthodox content [2]. Similarly, many named

entity recognition (NER) research works have been carried out recently with respect to twitter streams such as [3, 4]. These research works have largely been aimed at augmenting the capabilities to extract the potential named entities from the tweets and focused on improving the state-of-the-art methods in detecting the Out-Of-Vocabulary (OOV) words. But due to the lack of contexts and noisy structure of the tweets, detecting the potential named entities from tweets poses a great challenge and gives huge difficulties to annotate the tweets with necessary POS tags. Figure 1 illustrates the open challenges in handling COVID-19 twitter streams. Table 1 provides the detailed information about the publically available named entity annotated tweets.

Besides, tweet tokenization has been a great challenge for many NER systems, and the existing methods such as Penn-Tree Bank (PTB), TweetMotif, TwokenizerTool [5], and TwitIE tokenizer [6] failed to address these issues effectively. Therefore, we provide the mechanism to solve the fundamental preprocessing techniques such as tokenization, normalization, and POS tagging of tweets. These normalization processes have reduced the complexity persist over the given datasets and addressed the overfitting issues on the informal text categorically. As this process is very crucial for the next stage of entity extraction and classification, the effective handling of informal text is considered to be important and therefore proposed the most effective hybrid approach to deal with the issues appropriately. In addition to that, informal text would have certain common open challenges that has been listed in Figure 1. This research work has profusely handled these open challenges in the effective way and able to outperform the results with good precision.

The major contributions of this paper are given below:

(i) A detailed discussion on text normalization techniques also delineated the difficulties of converting Out-Of-Vocabulary (OOV) words into In Vocabulary (IV) words

(ii) Proposed an enhanced tweet tokenization technique to the alternate of Stanford tokenizer and Penn Tree Bank Tokenizer

(iii) In order to extract the appropriate named entities from unstructured text, we have utilized the lightweight hidden Markov model (HMM) to filter the correct lexical features that has been generated through the POS tagging

1.1. Paper Structure. The rest of the paper is organized as follows: Section 2 discusses the collaborative work rendered by the researchers in the fields of tweet normalization and preprocessing. Also, the prominent tools and methods used for tweet tokenization and normalization have been discussed briefly. Section 3 gives a comprehensive idea about our proposed methods and techniques followed for tweet normalization. Besides, we have briefed about the Out-Of-Vocabulary conversion methods and some evaluation metrics to detect the OOV methods. Section 4 highlights the procedures for tweet tokenization and segmentation. This section has delineated the procedures to tokenize the tweets with some standard nomenclatures. In Section 5, the tweet preprocessing and normalization approach has been dealt with a novel algorithm and analyzed the existing dictionaries to detect the potential OOV words. In Section 6, POS tagging for tweets has been discussed and also introduced the lightweight hidden Markov model for extracting the lexical features generated from the standard POS tagging. Section 7 gives the empirical shreds of evidence of tweet normalization and effective preprocessing results. Also, we have given some error analyses of our proposed methods.

Figure 1: Open challenges in handling COVID-19 twitter streams.

2. Related Works

As most people use social media sites to post their messages daily, the amount of information stored on these websites get increased exponentially, and the messages are informal in nature due to its limited space constraints. Text normalization plays the seminal role in the process of detecting and removing the noisy text (i.e., tweet in this research) into standard words. Therefore, it has gained huge research attention in recent years and increasingly attracted many researchers to carry forward their research work in this domain. Besides, many academic conferences and workshops [7–9] have been conducted to gather the data related to informal texts. The Association for Computational Linguistics (ACL) [10] and North American Association for Computational Linguistics (NAACL) [11] have been encouraging the researchers and students to actively participate in their conferences and workshops to gain knowledge on both formal and informal text. Recently, the Text Retrieval Conference (TREC) has created a new web page related to informal languages used in social media sites and also conducted workshops [12] relevant to the field. In this

Table 1: Openly available social media annotated corpus.

Annotated corpus	Number of tokens	Entity schema
Finin et al. [58]	7 K	Person, location, and organization
Ritter et al. [59]	46 K	Freebase
Liu et al. [3]	12 K	Person, location, product, and organization
Rowe et al. [60]	29 K	Person, location, misc, and organization
Derczynski et al. [61]	165 K	Person, location, and organization

research work, we have analyzed the works demonstrated by various researchers and their research findings and shortcomings in detail.

Earlier, the researchers [13, 14] have taken their research work only on normalizing the spelling errors produced from the web sources. They have used the *n*-gram model to assess the probability of each word within its context and estimate the relative frequency of the word in the given sequence. Their *n*-gram model has mapped the words using Many-to-One (*N*-to-1) cardinality, and the real word substitutions such as word usage with its context and grammatical structure of the sentence were detected and converted successfully, but their research work faced some serious lapses in dealing with unstructured content and failed to retain the accuracy rate attributed by the many prominent researchers [15–17].

Meanwhile, [18, 19] has demonstrated their research work on microtexts such as Twitter and SMS for detecting phonetic misspellings, standard acronyms, and contractions. Generally, misspelled words were detected by Natural Language Processing (NLP) systems using the mult-channel models which effectively find the lexical variance on some factors such as contextual wounding of the word, phonetic similarity, orthographic factors, and expansion of acronym using the standard dictionary. As suggested by previous researchers [20–23], they have utilized the Aspell spell corrector to detect the misspelling on Twitter as well as on SMS datasets.

Later, [24] has developed a spelling corrector which uses Google Style-based spell corrector, and it just find the proximity of a word and recognizes the correct spelling for the given word. The algorithms work on the conditional probability of the word based on edit distance measure and choose the word which has less edit distance (i.e., less number of deletion, replacement, insertion, or transposition used to convert the word into correct form). They have set the threshold limit of edit distance is less than or equal to 2. But due to textual sparseness in informal text, most of the misspelled words in the informal text require more edits and demand more comparison concerning the context of the words. Again, the Norvig system [25] works exceptionally well on standard orthodox text but failed to get the precision on informal text.

Finding the Out-Of-Vocabulary word is very challenging in social media sites; particularly, it has been prevalently used in twitter streams. The OOV word is defined as unorthodox words, and it has not been presented in the standard dictionary for reference. To tackle this issue, many research works had been carried out [26–30] and attained some considerable accuracy rates with respect to the BLEU score. The researchers [31] have used the classifier to detect the OOV words as ill-formed words based on the similarity measures such as phoneme and grapheme score and converted the ill-formed words into standard English words. In their approach, they had used the dictionary, context support for the word, and similarity measure to predict the correct form of the OOV words and finally attained the *F*-score of 68.30%. Even though the result is considerably good at some aspects, it had not performed well on noisy tweets and yields poor results if there was no context support for the ill-formed words.

Later, the author [32–34] has used a hybrid approach to deal with OOV words present in social media sites and prepared heuristic approach such as string similarity measure, edit-distance function, and subsequence overlap function to detect the OOV words and converted them into its appropriate In-Vocabulary words. The correct candidate word was selected based on the *n*-gram model and used the confusion matrix to find the proximity score which is likely to be the correct English form. This approach has reduced the burden of previous researchers and yields a good accuracy rate of up to 72.15%.

3. Proposed Method

In this research work, we have downloaded the Twitter datasets related to COVID-19 from 6th Workshop on Noisy User-generated Text (W-NUT) [35] for our analysis. It has manually annotated almost 10,000 tweets related to COVID-19 and built a corpus called *COVIDKB* that is a well-structured knowledge base to support the SPARQL queries. To extract the structured knowledge from the tweets, our primary task in this preprocessing step is to remove the usernames, special symbols, retweets, hashtags, and emoticons from the tweets and take only the original tweets for the next level of processing.

3.1. Problem Definition. "Given the tweet corpus T, eliminate the tweets which do not convey much information regarding the event E and remove or replace noisy tokens in the tweets with normalized tweets."

The basic regular expression followed to remove the special symbols, retweets, and other emoticons is given below:

```
def process_text():
""" " Remove emoticons, usernames, retweets etc. and returns list of cleaned tweets. """
data=pull_tweets()
regex_remove="(@[A-Za-z0-9]+)|([^0-9A-Za-z  t])|(\w+:\/\/\S+)|^RT|http.+?"
```

```
    stripped_text=[re.sub(regex_remove, “, tweets).strip()
for tweets in data]
    return ". ".join(stripped_text)
```

Once the tweets are cleaned using the above regular expression snippet, we need to perform the tokenization of tweets to fix the proper tagging of words and identify the proper nouns and pronouns for effective entity extraction and disambiguation. Each of the following methods helps to solve the ambiguity that persists over the tweets and identifies the entities with proper references in the external document sources. The three basic components of this research, i.e., tokenization, normalization, and POS tagging are considered to be noncore components but they are very crucial in this research because the informal nature of tweets has condensed the words and give space for ambiguity. Hence, we have proposed a novel method to deal with the issues and remove the ambiguity with the support of external document.

Before we go into the next phase of the preprocessing pipeline as given in Figure 2, we have taken two types of dictionaries to correct the misspelled words in the tweets and fix the correct word form to it. In this case, at first, we have benchmarked some of the standard online spell correction dictionaries for our analysis such as Norvig's Spell Corrector (https://norvig.com/spell-correct.html), BK-Tree (https://issues.apache.org/jira/browse/LUCENE-2230) (Burkhard–Keller Tree), SymSpell (https://symspellpy.readthedocs.io/en/latest/) (Symmetric Delete Spell correction algorithm), LinSpell (https://github.com/wolfgarbe/LinSpell) (Linear Search Spell Correction), and PyEnchant Dictionary(https://pyenchant.github.io/pyenchant/tutorial.html) as given in Table 2. From the analysis, we have observed that the PyEnchant Dictionary is suitable for informal text processing and compatible with all the programming environments. Besides, PyEnchant Dictionary is faster than the above four algorithms, and the indexing method is found very effective for searching the words. The PyEnchant dictionary has been getting updated frequently on every year and enriched its gazetteer words. Hence, we have used the PyEnchant Dictionary for the misspelled words and identified the OOV words in the tweets if any.

Second, we have created our own slang dictionary for converting the slang word into its correct English word form and fixing the correct meaning for the slang word. In this context, we have searched the online slang dictionary application sites such as NoSlang Dictionary (https://www.noslang.com/), Urban Dictionary (https://www.urbandictionary.com/), Translit (https://www.translit.ie/), and a few more web sources. Further we manually extracted the slang words such as contractions, abbreviations, slangs, unorthodox word forms, and canonical words from the above listed online sources and gathered their equivalent English meanings appropriately. Then, we have listed all the slang words with their corresponding English meanings into separate files such as abbreviation, slang words, contractions, and emoticons. Later, we formatted each of the above files and removed the duplicate entry if present in each file (Table 3). Eventually, we checked the absolute meaning of each token in the file and ordered them alphabetically for easy processing of search operation. The first column in each file contains the slang words or abbreviations or contractions, and the second column gives the corresponding meaning or abbreviation of that token.

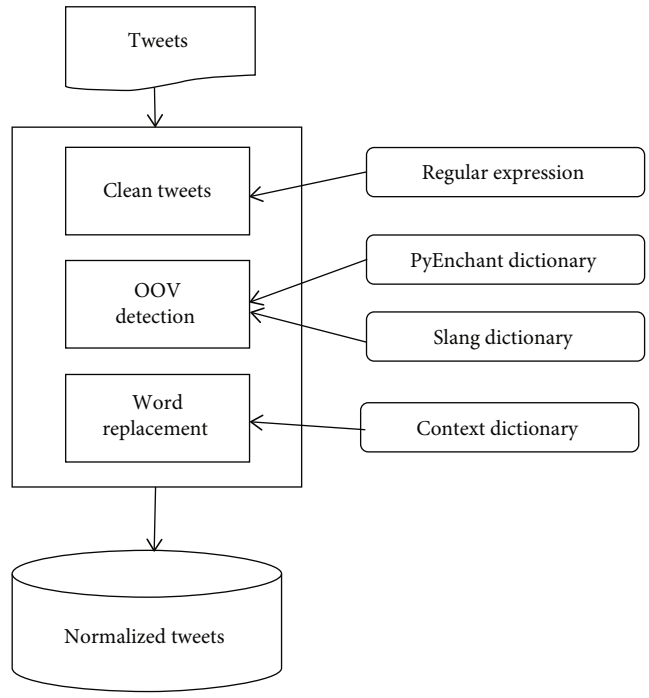

Figure 2: Tweet normalization and word replacement techniques.

3.2. Tweet Normalization. The first step in the tweet normalization approach is tokenization. This is the basic preprocessing step followed in all the natural language processing as well as in the information extraction projects. The proposed approach for tweet normalization has been given in the following Figure 2.

Mainly, the process of tweet normalization takes three critical analysis: (i) detect the candidate tokens on the mutual comparison of standard vocabulary sets; (ii) identify the symbols, emoticons, and OOV words from the tweets with respect to word contortions such as spelling mistakes and displacement of the grammatical structure of words; and (iii) discard the OOV words from the tweets using the standard corpus. All the above-described steps are completely language independent and work exceedingly well with language specifies resources (see Figure 3). For the OOV words, it has been largely dependent on standard abbreviations and acronyms to filter out the In Vocabulary (IV) words and produce the candidate list of IV words for POS tagging.

The main challenge in processing informal text such as tweets is that it gives difficulty in dividing the tweets into multiple tokens and categorically identifies the potential named entities from the divided tokens. The major task of classifying the tokens into IV words and OOV words would be a serious implication in the process of tweet normalization. The standard dictionary (i.e., in this case, we have used PyEnchant Dictionary for word comparison and dictionary lookup) is more than enough to identify the IV words from the tweets and appropriately categorize those into one of the predefined

Table 2: Conventional text normalization methods and techniques.

Technique	Abbreviations	Repeated characters	Misspelled words
Regular expression	X	√	X
Replace() function using WordNet	X	√	√
Expanding abbreviations by CSV file replacement	√	X	X
Probability model using edit distance	X	√	√
Spell correction using TextBlob	X	X	√
NLTK library	X	√	√
Phonetic edit distance	X	√	√
PyEnchant library	X	√	√

Table 3: Proposed slang dictionary for unorthodox content.

Type of dictionary	Total entry
Abbreviations	1346
Contractions	131
Slang words	1296
Emoticons	164
Total	2937

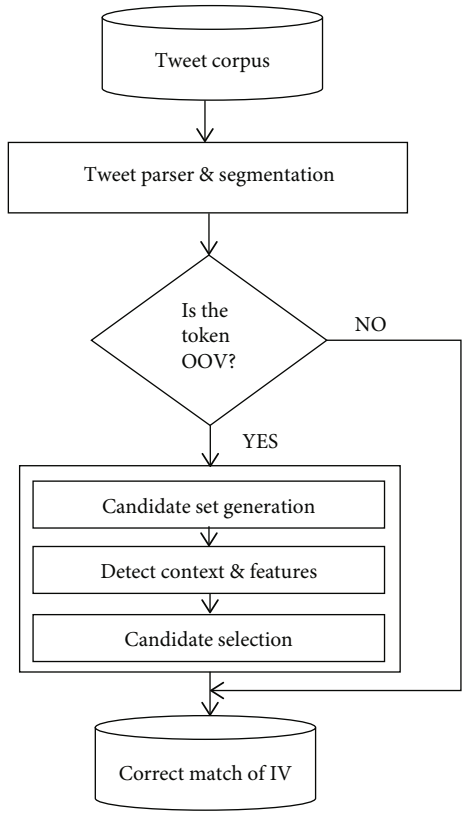

Figure 3: General framework for detecting IV and OOV words.

categories sets based on the POS tags assigned to it, but the remaining nonstandard tokens (i.e., OOV words) need to be compared and find the appropriate candidate list to fit it into the correct word match.

3.2.1. Statistical Rules. So far, we have discussed handling the OOV words and the problem of choosing the appropriate candidate words for the given nonstandard token in the tweets/sentences. We have identified some of the implicit traits of the nonstandard tokens (i.e., OOV words) after running through a huge tweet corpus downloaded from COVIDKB [35] and defined the following basic procedure to tackle the OOV words if any present on the user-generated content.

Here are the following examples:

Type 1: in the OOV words, the social users may have missed the spaces either knowingly or unknowingly and stretched into two or more standard words. Example: stywith (stay with) and cometogether (come together)

Type 2: the OOV words are framed upon the sound of the words rather than the lexical structure of the words. Example: c u agn (see you again)

Type 3: the OOV words are constructed based on the first letters of the standard words or the phonetic positioning of the words. Example: u r (you are) and thx (thanks)

For these types of errors in the OOV words, we have constructed the Slang dictionary (Table 3) as mentioned in the previous Section 3 and detect the possible candidate set of words to disambiguate the OOV words. We also find the exact fit of matching words to the given token based on the context given by the language model. We have reduced the 1-to-*N* mapping of the candidate list for the OOV words into 1-to-1 mapping and increased the efficiency of the tweet normalization.

3.2.2. Multiple Character Reduction. The OOV words have occurred at many places in the tweets and disturb the process of transforming them into standard English words. The major problem faced in handling the OOV words was that it contains many nonword tokens and repetition of multiple characters to express the inherent emotions to the reader. This was also explained in [36]. The characters which occurred more than once will be deduced to single characters, and then PyEnchant Dictionary has been utilized subsequently to prevent the further mistakes caused by the earlier reduction of multiple characters. For example, the words such as speed, speech, and breed have contained the same character that appears more than once in the word and still gives the correct English meanings. If we reduce those multiple occurrences of characters to a single character, then it would cause spelling mistake and leads to bad normalization. To get the correct form of the word, we have proposed the appropriate method where it has taken the

utmost care to tokenize the OOV words and split the OOV words based on some designed patterns. Besides, multiple punctuation symbols posed similar difficulties, and we reduced the multiple punctuation marks into single punctuation using the defined regular expression.

3.3. Evaluation Metrics. To assess the quality of results for the OOV words, the following metrics have been used and have been evaluated the performance of the system.

3.3.1. Miss Rate (MR). It measures that the number of OOV words was missed with respect to OOV words detected.

$$\mathrm{MR} = \frac{\#\text{OOVs references} - \#\text{OOVs detected}}{\#\text{OOVs references}}. \quad (1)$$

3.3.2. False Alarm Rate (FAR). It measures the number of IV words that had been falsely reported as OOV words.

$$\mathrm{FAR} = \frac{\#\text{OOVs reported} - \#\text{OOVs detected}}{\#\text{IVs references}}. \quad (2)$$

3.3.3. Word Error Rate (WER). It measures the number of errors that occurred during the substitution, deletion, or insertion of characters by the proposed system.

$$\mathrm{WER} = \frac{\#\text{Substitution errors} + \#\text{deletion errors} + \#\text{insertion errors}}{\#\text{Words in references}}. \quad (3)$$

3.3.4. Precision. It measures that the number of OOV words has been detected correctly by the proposed system.

$$\mathrm{Precision} = \frac{\#\text{OOVs detected}}{\#\text{OOVs reported}}. \quad (4)$$

3.3.5. Recall. It measures the number of OOV words detected with respect to the OOV references.

$$\mathrm{Recall} = \frac{\#\text{OOVs detected}}{\#\text{OOVs references}}. \quad (5)$$

3.3.6. F1. It measures both the precision and recall of the OOV words detected by the proposed system.

$$\mathrm{F1} = \frac{2.\text{Precision X recall}}{\text{Precision} + \text{recall}}. \quad (6)$$

3.4. Experimental Analysis. We have manually analyzed the tweet normalization for the COVIDKB since there is no gold standard dataset followed to assess this language model and hence assessed the performance of COVIDKB which contains 10,000 tweets. For each tweet, we have considered all the modifications that happened in the process of normalization by the proposed system and validated them. The four major tweet normalization operations such as insertion, deletion, substitution, and tokenization have been monitored manually, and the correctness of the results has been measured through the F1-score produced by the proposed system. Table 4 shows the number of OOV words detected, pronunciation accuracy, and identical pronunciation score of the system.

Table 4: Accuracy rate of COVIDKB Twitter dataset and OOV detected.

COVIDKB Twitter dataset	Accuracy rate
# of detected OOVs	3728
Pronunciation accuracy	67%
Identical pronunciation	72%

Based on Table 4, the precision, recall, and F1 score for the OOV words have been measured, and it has been given in Table 5. Figure 4 illustrates the ROC curve and statistical analysis of investigating the OOV words in tweets.

This ROC curve depicts the accuracy rate, and sensitivity of OOV words presents in the tweets and was able to identify the missed percentage of Part-of-Speech tags for the tokenized tweets.

In addition to finding the OOV words present in the tweets, there are also some other factors to be considered for effective normalization such as stemming, lemmatization, stop word removal, and emoticons detections [37–39]. Extra supervision is required to handle these preprocessing methods and further, these methods help to provide contextual support for sentiment analysis, word cluster, information extraction, entity detection, and many more (see Table 6). As we discussed finding the potential named entities in the tweets, these features help to solve the ambiguity that persists over the text and largely support the contextual score for the proposed system.

The performance metrics and combinational score of the preprocessing methods have been given in Table 7 for further comprehension and enhancement for accuracy. Figure 5 portrays the ROC curve for text normalization and statistical summary of preprocessing steps. It has become evident that the blend of emotions, lemmatization, and stopwords removal has given the performance marginally high and outperfomed with the other integrated approaches.

4. Tweet Tokenization and Segmentation

As tokenization is a first step in the pipeline, the major aim of tweet tokenization is to split the tweet into some meaningful chunks (i.e., semantic tokens) that can be words, word phrases, or any cardinals. Due to the informal nature of the tweets, tweet tokenization process gives difficulties in handling the informal text and comparatively challenging than formal text processing operations [40]. Hence, it requires some sophisticated techniques to solve the issues and effectively perform the tokenization processes. In this connection, we have analyzed some of the techniques followed by earlier researches for tokenization of normal text content. Since the formal text has been supported with well-structured context and language grammar [41], it had performed well on all the grounds, and the major tokenization approach that was followed by researchers was Stanford tokenization. Normally, the Stanford tokenizer utilized the JFlex lexical analyser [42] for tokenization of sentences and produced the results for the given formal text. In some cases,

Table 5: Evaluation of OOV words and F1 score.

OOV operations	Precision	Recall	F1 score
Insertion	0.872	0.801	0.834993
Deletion	0.857	0.782	0.817784
Substitution	0.839	0.768	0.801932
Tokenization	0.894	0.852	0.872495

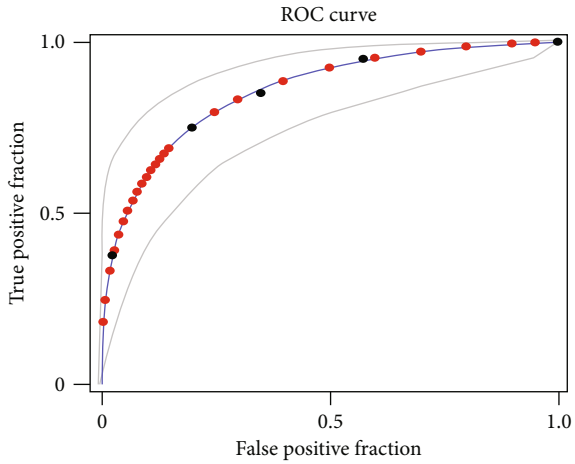

Figure 4: ROC curve and statistical analysis of investigating the OOV words in Tweets.

Table 6: Preprocessing techniques and method used.

Preprocessing techniques	Methods deployed
Stop–word removal	Rainbow list
Stemming	Snowball stemmer
Emoticon	Regular expression
Tokenization	Unigram, bigram, and N-gram
Weighting scheme	TF-IDF

Table 7: Preprocessing impact of text normalization.

Preprocessing methods	Precision	Recall	F1 score
Stop word	0.547	0.521	0.533684
Stemming	0.649	0.592	0.619191
Emoticons	0.723	0.684	0.702959
Stem+stop	0.812	0.769	0.789915
Emo + Lem + stop	0.875	0.837	0.855578
Emo + stem+stop	0.836	0.795	0.814985

the researchers have even used Penn Treebank Tokenizer [43] to tokenize the content which is using the specific regular expression written in SED script. The above tokenizers have been commonly used for most formal text processing and yield a good accuracy rate for all the instances. But when the tweet is informal in nature and mostly unorthodox, then the above tokenizers would have been a bad choice in this regard and produce inappropriate results.

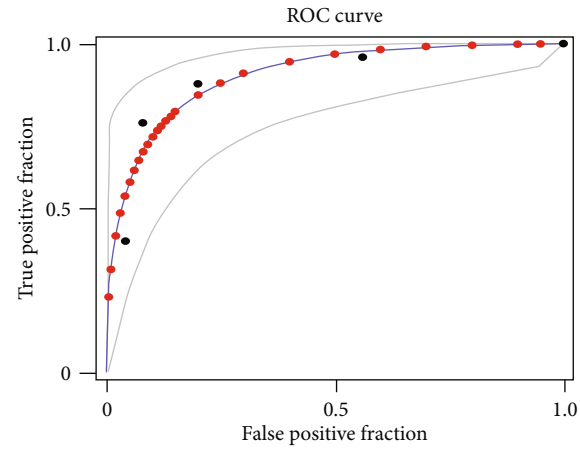

Figure 5: ROC curve for text normalization and statistical summary of preprocessing steps.

4.1. Proposed Approach for Tokenization. Like the procedure followed for formal text tokenization, some researchers followed the same on informal text tokenization [44]. In the formal text tokenization methods, they have divided the token into meaningful chunks if they encountered any spaces or specific delimiters present in the sentence. This method has resulted in poor performance in POS tagging and made the entity detection processes complicated for informal text such as tweets. Therefore, we have considered the key phrases of the token up to the length of 4 (i.e., as almost all the named entities can be restricted within that range) and split the tokens based on the following patterns:

(1) (Noun)+: for the given tweet, the tokenizers find more than one continuous nouns that can be clubbed into one key phrase and considered as one single token. Example: Samsung Galaxy Phone

(2) (Adjective) + (noun)+: if the noun started with one or more adjectives, again, it is considered to be a single token, and division has been made accordingly by the tokenizer. Example: Fantastic Donald Trump and Digital Camera

(3) (Noun) + [CD]: one or more nouns followed with some cardinals. Example: James Bond 007 and iPhone 8i

To support the above patterns and filter the tokens from the tweets, we have emulated the tokenizer called ARKTweetNLP [45] which is an open-source module for download and infused our pattern into the above package to effectively filter out the key phrases for the next phases of POS tagging. The main reason to choose this module over other tokenizers is the fact that this tokenizer has been designed by [46] in considering the Twitter-specific regular expressions, covered a wide range of emoticons, and achieved the good performance on tweet tokenization. We have given some examples in Table 8 that the proposed tokenizer has been able to split the tweets into some meaningful chunks successfully.

Table 8: Proposed method results on tokenized tweets.

Tweet	Tokenized tweet
Prince Charles met Albert of Monaco just days before he tested positive for coronavirus	Prince Charles/met/Albert of Monaco/just days/before/he/tested/positive/for coronavirus
Just heard Boris Johnson has tested positive for coronavirus. The man is not fit to lead out country.	Just heard/Boris Johnson/has tested/positive/for coronavirus./The man/is not fit/to lead out/country.
Tom Hanks and his wife Rita Wilson have both tested positive for coronavirus, the US actor said Wednesday	Tom Hanks/and/his wife/Rita Wilson/have/both/tested/positive/for coronavirus/, the US actor/said/Wednesday/.
I finally personally know someone that has tested positive for COVID-19. Interesting timing. As they are now getting more tests made, more tests coming back positive.	I/finally/personally/know/someone/that/has tested/positive/for COVID-19/. Interesting timing/. As/they/are now getting/more tests made/, more tests/coming back/positive.
Former movie producer Harvey Weinstein, who is serving a prison sentence for sexual assault and rape, has tested positive for coronavirus	Former/movie producer/Harvey Weinstein/, who/is serving/a prison sentence/for sexual assault/and/rape/, has tested/positive/for coronavirus.
Redmond-based Nintendo of America has confirmed that an employee has tested positive for coronavirus.	Redmond-based Nintendo/of America/has confirmed/that/an employee/has tested/positive/for coronavirus.

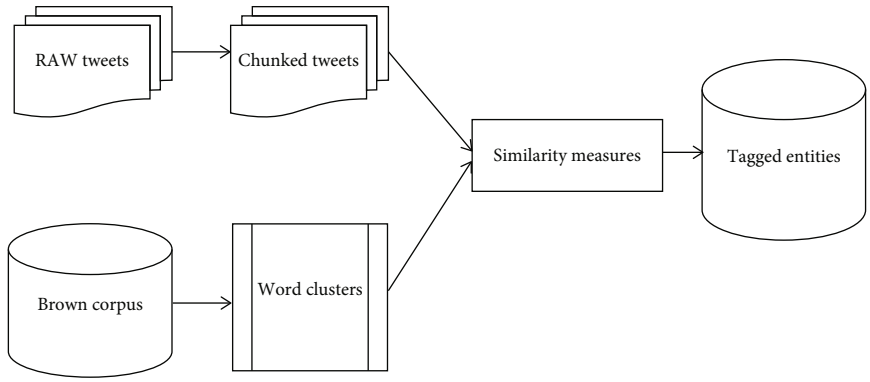

Figure 6: Proposed method for text normalization using word cluster.

5. Tweet Normalization and Processing

The next step in our pipeline is the normalization process that is used to identify the tokens as either Out-Of-Vocabulary or not and convert them into their Standard English word or word phrase. Normalization is very important for text processing because it will help to solve the ambiguity that persists on any token in the text [47, 48]. As explained earlier, many researchers have used statistical machine translations, phrase-based statistical model, character-level edit distance, dependency parsers, and even built-in parallel corpora to train the model to generate the possible candidates for the Out-Of-Vocabulary (OOV) words. Some have even tried to use language models and phonemic edit distance measures to deal with the problems differently, but normalization problem is quite challenging in itself and poses great difficulty to the informal text. For instance, abbreviations and slang words are very difficult to map in the existing spell correctors.

5.1. Proposed Approach for Normalization. We have used the supervised normalization technique for the proposed system and also utilized online resources such as Brown Corpus, PyEnchant Dictionary, and Microsoft Web *N*-gram model. We have used the Brown Corpus for text normalization because of the fact that similar words occur in a similar context. That is, similar words would have the same set of distributions and arrangements of words on either side, i.e., left and right. With that assumption, we have clustered the Brown Corpus and trained it for our supervised classifier to normalize the words if it encounters any OOV words. As the Brown Corpus has already clustered 47 Million tweets and almost produced more than 1200 word clusters (i.e., on each cluster, it has arranged the tweets with similar context), we have effectively utilized these clusters in this normalization process and transform the OOV words successfully as depicted in Figure 6.

Besides, we have used the PyEnchant Dictionary for mapping the OOV words in its dictionary and performed two string similarity measures such as Levenshtein and metaphone edit distance. We used both the string similarity measures for the candidate word selection because some of the words have given the same phonetic sound, for example, know and no and pork and fork. To disambiguate this conundrum, we have taken both the measures and choose the best one using the Microsoft Web *N*-gram model. Out of all the suggestions listed from both the Brown Cluster and PyEnchant Dictionary, the Microsoft Web *N*-gram model will choose the correct word which has the high score based on the conditional probabilities of candidate words given in the context (i.e., word combinations before and after the words as suggested by Brown Cluster and String Similarity Measure). The word

```
Input: Tweet Dataset
Output: Normalized Tweet
BEGIN
        Normalized Tweet ⟵ {}
        FOREACH token IN Tweet DO
                IF token NOT EQUAL TO Noun THEN
                        IF token IN OOV word THEN
                                Token⟵OOV word
                ELSE
                Token ⟵not OOV word
        IF token IN BROWN Cluster THEN
                Cand-Token ⟵ Fetch the candidate words from the BROWN Cluster
                                and perform the Levenshtein and Metaphone Edit distance
                Normalized Tweet ⟵ Append the Cand-token with highest frequency score
        ELSE
                Token⟵ Not in BROWN Cluster
        Sug-token ⟵ Retrieve the suggestion for the token using PyEnchant dictionary
        FOREACH Suggestion from Sug-token DO
                Score=(Prob(Prev-token + Suggestion) + Prob(Suggestion + Next-token))/2
                                                                [Using Microsoft Web N-Gram]
Normalized Tweet ⟵ Append the Suggestion with highest score
RETURN Normalized Tweet
END
```

Algorithm 1: Normalization Algorithm.

with the highest score can be returned as output by the system. The algorithm has been given below for the whole normalization process and how the system has returned the normalized output successfully.

6. Tweet Part-of-Speech Tagging

After normalizing the tweets using the hybrid approach, we need to perform the Part-Of-Speech (POS) tagging, and this process is very crucial for entity extraction and classification. The entity extraction has been performed on the tweets based on the POS tagging and extracts the entities which have been attributed as nouns, proper nouns, pronouns, or any objects as nouns [49]. So, POS tagging of tweets would determine the grammatical structure and category for each token that is segmented on the normalized tweets. Many words in the tweets would have no syntactic features such as hashtags, URLs, emoticons, and @mentions. The dependency parser has taken more time in processing than these nonsyntactic features and consumes time unnecessarily. But without appropriate utilization of standard annotators, with the use of a simple rule-based filter, it can extract and annotate the #hashtags, @mentions, punctuations, and retweet tokens effectively. Next, for the multiword expressions, we have considered two types of approaches. First, the proper nouns have been compounded together for information extraction and assigned the single tag for the compound words. Second, the lexical idioms, such as "stuck in the crowd" and "hay in the stack," have been manipulated with shallow parsing and clubbed into a single token for tagging. The same approach has been followed even for the idiomatic relationships and performs the internal analysis of multiword tokensdependency parsers.

6.1. Proposed Approach on Part-of-Speech (POS) Tagging. To effectively attribute the tags to every token divided on the given tweets, we have implemented the supervised learning approach to train the model and tagged the tokens on the linguistic features followed by the natural language processing toolkits. To assign the tag to the segmented tokens, there are many features considered such as capitalization, surrounding words, tags on the surrounding words, and presence of any cardinal on the word. Upon scrutinizing the model features, appropriate tags have been assigned to the token correctly. Since the tweets lack grammatical structures and a dearth of context around the words, assigning appropriate tags to the segmented tokens has become critical and sometimes failed to attribute the correct tagging. Also, as the tweets contain more slang words, OOV words, spelling errors, and abbreviations, it has become challenging to assign the appropriate POS tagging on the tokens and need extra supervision for picking the features for the proposed model. Particularly, capitalization has not been considered to be a good feature in informal text processing because many social users have not followed the proper capitalization of words. In addition to that, they have used more adjectives to extend their greetings and emphasized more on the thoughts which become a cumbersome task for the POS tagging (see Table 9). Hence, we have used the lightweight hidden Markov model to predict the context and assign the correct tags to the tokens.

6.2. Lightweight Hidden Markov Model–POS Tagging. The lightweight hidden Markov model used the supervised word clusters trained from Brown Corpus and extracted other lexical features generated from Stanford POS tagging. These word clusters have been used to train on the unlabeled tweets and filter out the word clusters again generated from the new labeled

Table 9: General POS tag and its descriptions.

Tag	Description
DT	Determiner
PRP	Person pronoun
VB	Verb, base form
VBP	Verb
IN	Preposition or conjunction
NN	Noun, singular, or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural

Table 10: COVIDKB Twitter dataset and its annotated ground truth.

Tweets	Tokens	Repeated characters	Abbreviations	Misspelled words	Total
1000	13487	672	308	1128	2108

datasets. These tagging features extract only the conventional tagging features such as words, surrounding words, surrounding tags, and use of cardinals, and also, it extracts the Twitter-specific features such as hashtags, usernames, and emoticons. This distributional-based word similarity feature finds useful for twitter streams and outperforms the Stanford tagger on the datasets given. The word-tag probability is considered to be a stochastic model in which the tagger is deemed as a Markov process with unobservable states and yields the observable outputs. In simpler terms, the POS tags are the states of the Markov process, and words are the desired results of the model. The conditional probability of the lightweight hidden Markov model has been given below:

The POS tagger comprises of the following:

(i) $P_s(T_i)$: probability of the sequence beginning in tag T_i

(ii) $P_t(T_j \mid T_i)$: Probability of the sequence changing from tag T_i to tag T_j

(iii) $P_E(W_j \mid T_i)$: Probability of the sequence terminating word W_j on tag T_i

The tagger makes two straightforward suppositions:

(i) The likelihood of a word depends just on its tag, i.e., given its tag, it is autonomous of different words and different labels

(ii) The likelihood of a tag depends just on its past labels, i.e., given the past labels, it is autonomous of next labels and labels before the past labels

In noun phrases, a noun acts like a subject or an object to a verb or an adjective. To create a noun phrase Chunker, we define a noun phrase Chunker that indicates how the tweet to be chunked.

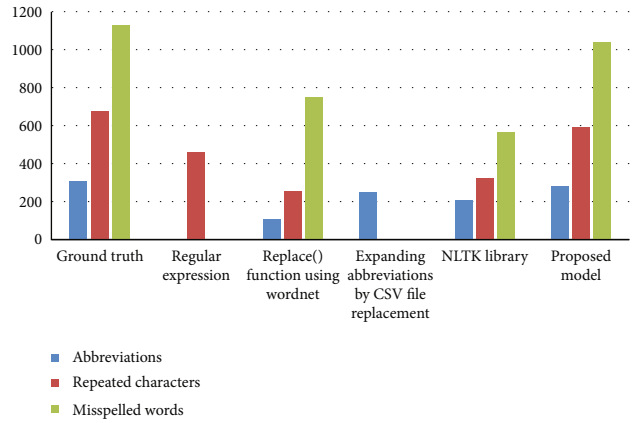

Figure 7: Statistical analysis of text normalization processes.

$$\text{Grammar} = \text{“NP}: \{<\text{DET}>?<\text{ADJ}>*<\text{Noun}>\}\text{”}. \quad (7)$$

Here are the following examples: tweet: “I am at Bicycle Ranch in Scottsdale, AZ”. POS tagging of tweet: (I/PRP, am/VBP, Bicycle/NNP, Ranch/NNP, in/IN, Scottsdale/NNP, AZ/NNP).

7. Experimental Evaluation and Analysis

For our experiment analysis, we have downloaded the COVIDKB Twitter datasets provided by [35]. The COVIDKB Twitter datasets consist of 10,000 annotated tweets with ground truth values. But for our experimental analysis, we have taken 1000 tweets as the test sets due to the condition of the system and to save the computational time (see Table 10). We have considered this COVIDKB Twitter datasets because it has been manually annotated with research students and contains a wide range of entities such as a person, location, travel, and contacts. Therefore, we benchmarked this dataset for our experimental analysis and observed the state of the performance of the proposed system.

7.1. Result Analysis. We have compared our proposed model with existing preprocessing techniques and found that our proposed model has outperformed all other conventional models and yielded a better performance and accuracy rate. For the analysis of the results, we have considered the basic normalizing features such as repeated characters, abbreviations, and misspelled words. As these three components are very crucial in the tweet normalization, it has been compared with regular expression and replace function using WordNet, NLTK Library, and replacement CSV file. Our proposed model has converted the informal words into correct English words with a much higher accuracy rate and almost 20-25% increase in the precision score. As the noisy token has been hindered the performance of POS tagging and further block the entity extraction, the proposed approach has paved the better way for the POS tagging at this level and helps to solve the entity detection and recognition in the next level of processing as it was witnessed in Figure 7. A detailed view of the comparison has been given in the following Table 11.

Table 11: Proposed method on normalizing repeated characters, abbreviations, and misspelled words.

Techniques	Abbreviations	Repeated characters	Misspelled words
Ground truth	308	672	1128
Regular expression	—	462	—
Replace() function using WordNet	108	253	749
Expanding abbreviations by CSV file Replacement	247	—	—
NLTK library	210	319	561
Proposed model	281	590	1036

Table 12: Proposed method accuracy rate on POS tagging of tweets.

Techniques	BLEU (%)	WER (%)
Regular expression	81.26	8.91
Replace() function using WordNet	80.19	10.73
Expanding abbreviations by CSV file Replacement	78.22	12.18
NLTK library	79.14	10.98
Proposed model	83.25	8.23

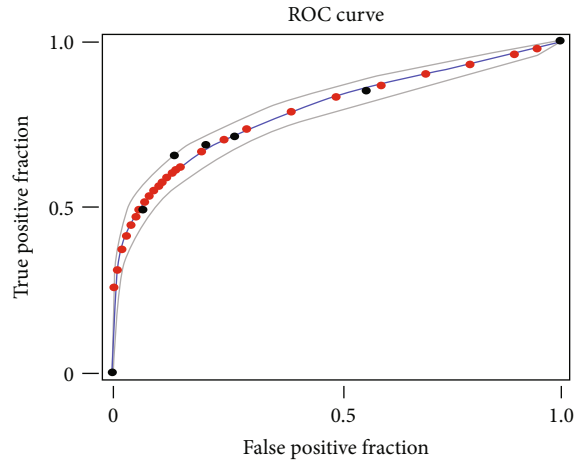

Figure 8: ROC curve for evaluated POS tagging of tweets.

Moreover, Table 12 presents the accuracy rate of the proposed method on POS tagging of tweets.

The experiments are evaluated based on the BLEU score and WER score. While analyzing the output with other comparison models, it has become evident that the proposed method has effectively removed the repeated characters and misspelled words and normalized the appropriate text forms for the given tweets. The calculated p values from the BLEU and WER have been comparatively less than 0.005 at the 95% CI level, 0.0029 and 0.0028, respectively. Therefore, the proposed method has invariably shown the consistent precision with its BLEU score less than 85% or the WER score greater than 8%. Figure 8 depicts the ROC Curve for evaluated POS tagging of tweets.

This ROC curve has been obtained after the comparative analysis over the techniques mentioned in Table 12. It has become very apparent that the proposed model has been yielded with good accuracy rate as it was measured with BLEU and WER techniques.

The overall analysis for the tweet normalization, preprocessing, and assignment of POS tags to the tokenized tweets has been given in Table 13. The accuracy rate and the sensitivity rate of the ROC analysis have been gradually improved after employing the proposed method. Eventually, the fitted ROC accuracy has been increased considerably and proved that the proposed method has outperformed other preprocessing steps carried through for the unstructured datasets.

Table 13: Overall plotting parameters for ROC utilized for tweet preprocessing.

Plotting parameters	Investigating OOV words in tweets	Tweet normalization and preprocessing	Evaluated POS tagging for tweets
Number of cases	2108	450	950
Number of correct	1907	412	702
Accuracy	77.5%	84%	73.9%
Sensitivity	75%	88%	68.6%
Specificity	80%	80%	79.2%
Positive cases missed	107	77	149
Negative cases missed	94	53	99
Fitted ROC curve	0.85	0.90	0.79

8. Conclusions

Natural language processing (NLP) allows digital gadgets and devices to comprehend the semantics in languages. Usually, the NLP can be broadly characterized into two categories such as data preprocessing and model development [50–57]. There are several text normalization strategies proposed by eminent researchers to solve the impending issues and reduce the error rate considerably [50–53]. However, they have certain confinements and still do not accomplish great outcomes when it has come to informal text processing. Rather than normalizing one kind of ill-formed word, we have considered a wide range of ill-formed words found on the tweets datasets and cleaned them under three primary classifications: incorrectly spelled words, contractions, and repeated letters. The primary motivation behind why we have sorted these unorthodox words is because we might want to guarantee that all subcategories of these three fundamental issues are standardized into their right form by the most appropriate procedures. Hence, the target of this exploration is to locate the best standardization approach with the end goal to productively and precisely clean tweets containing incorrect spellings, shortened forms, and repeated characters. The future scope of this research work would add tremendous advantage if it has been resourcefully utilized for natural disasters and the imminent threat of future disease outbreaks. The major limitation of this work has been restricted only with twitter streams in general and deals particularly on preprocessing techniques of the tweet tokenization and POS tagging. Further, it needs fine-grained datasets for identifying the OOV words if the domain-specific research work has been carried out in future.

Data Availability

The article's original contributions generated for this study are included; further inquiries can be directed to the corresponding author.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] D. Pla Karidi, Y. Stavrakas, and Y. Vassiliou, "Tweet and followee personalized recommendations based on knowledge graphs," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 6, pp. 2035–2049, 2018.

[2] L. Derczynski, A. Ritter, S. Clark, and K. Bontcheva, "Twitter part-of-speech tagging for all: overcoming sparse and noisy data," in *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pp. 198–206, Hissar, Bulgaria, 2013.

[3] F. Liu, F. Weng, B. Wang, and Y. Liu, "Insertion, deletion, or substitution? Normalizing text messages without pre-categorization nor supervision," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 71–76, Portland, Oregon, USA, 2011.

[4] L. Derczynski, D. Maynard, G. Rizzo et al., "Analysis of named entity recognition and linking for tweets," *Information Processing & Management*, vol. 51, no. 2, pp. 32–49, 2015.

[5] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith, "From tweets to polls: linking text sentiment to public opinion time series," in *Fourth international AAAI conference on weblogs and social media*, Washington, DC, USA, 2010.

[6] K. Bontcheva and L. Derczynski, "Extracting information from social media with gate," in *Working with Text*, pp. 133–158, Chandos Publishing, 2016.

[7] J. Tabassum, S. Lee, W. Xu, and A. Ritter, "WNUT-2020 task 1 overview: extracting entities and relations from wet lab protocols," 2020, https://arxiv.org/abs/2010.14576.

[8] Y. Hu, S. Gao, D. Lunga, W. Li, S. Newsam, and B. Bhaduri, "GeoAI at ACM SIGSPATIAL," *Sigspatial Special*, vol. 11, no. 2, pp. 5–15, 2019.

[9] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar, "Semeval-2019 task 6: identifying and categorizing offensive language in social media (offenseval)," 2019, https://arxiv.org/abs/1903.08983.

[10] C. Zong, F. Xia, W. Li, and R. Navigli, *Findings of the association for computational linguistics: ACL-IJCNLP 2021*, Association for Computational Linguistics, Online, edition, 2021.

[11] K. Toutanova, A. Rumshisky, L. Zettlemoyer et al., *Proceedings of the 2021 conference of the north American chapter of the Association for Computational Linguistics: human language technologies*, Association for Computational Linguistics, 2021.

[12] E. M. Voorhees and D. Harman, "Overview of the sixth text retrieval conference (TREC-6)," *Information Processing & Management*, vol. 36, no. 1, pp. 3–35, 2000.

[13] A. L. Berger, V. J. D. Pietra, and S. A. D. Pietra, "A maximum entropy approach to natural language processing," *Computational Linguistics*, vol. 22, no. 1, pp. 39–71, 1996.

[14] D. G. Lee, H. C. Rim, and D. Yook, "Automatic word spacing using probabilistic models based on character n-grams," *IEEE Intelligent Systems*, vol. 22, no. 1, pp. 28–35, 2007.

[15] Z. Xue, D. Yin, and B. D. Davison, "Normalizing microtext," in *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence*, San Francisco, USA, August 2011.

[16] M. Kaufmann and J. Kalita, "Syntactic normalization of twitter messages," in *International conference on natural language processing*, Kharagpur, India, January 2010.

[17] A. Garain, S. K. Mahata, and S. Dutta, "Normalization of numeronyms using nlp techniques," in *2020 IEEE Calcutta Conference (CALCON)*, Kolkata, India, 2020.

[18] E. Mapa, L. Wattaladeniya, C. Chathuranga et al., "Text normalization in social media by using spell correction and dictionary based approach," *Systems learning*, vol. 1, pp. 1–6, 2012.

[19] L. Besacier, E. Barnard, A. Karpov, and T. Schultz, "Automatic speech recognition for under-resourced languages: a survey," *Speech Communication*, vol. 56, pp. 85–100, 2014.

[20] B. Han, P. Cook, and T. Baldwin, "Lexical normalization for social media text," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 4, no. 1, pp. 1–27, 2013.

[21] P. Sosamphan, V. Liesaputra, S. Yongchareon, and M. Mohaghegh, "Evaluation of statistical text normalisation techniques for twitter," in *Proceedings of the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, vol. 1, pp. 413–418, Porto, Portugal, 2016.

[22] A. Sarker, "A customizable pipeline for social media text normalization," *Social Network Analysis and Mining*, vol. 7, no. 1, pp. 1–13, 2017.

[23] J. Kim, E. Lee, T. Hong, and P. Kim, "Correcting Misspelled Words in Twitter Text," in *International Conference on Big Data Technologies and Applications*, pp. 83–90, Springer, Cham, 2017.

[24] C. Napoles and C. Callison-Burch, "Systematically adapting machine translation for grammatical error correction," in *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 345–356, Copenhagen, Denmark, 2017.

[25] R. Beckley, "Bekli: a simple approach to twitter text normalization," in *Proceedings of the Workshop on Noisy User-generated Text*, pp. 82–86, Beijing, China, July 2015.

[26] J. J. Jung, "Online named entity recognition method for microtexts in social networking services: A case study of twitter," *Expert Systems with Applications*, vol. 39, no. 9, pp. 8066–8070, 2012.

[27] Y. Tsvetkov and C. Dyer, "Lexicon stratification for translating out-of-vocabulary words," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pp. 125–131, Beijing, China, July 2015.

[28] N. F. Liu, J. May, M. Pust, and K. Knight, "Augmenting statistical machine translation with subword translation of out-of-vocabulary words," 2018, https://arxiv.org/abs/1808.05700.

[29] M. Savargiv, B. Masoumi, and M. R. Keyvanpour, "A new ensemble learning method based on learning automata," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, pp. 3467–3482, 2020.

[30] V. K. Sharma, N. Mittal, and A. Vidyarthi, "Context-based translation for the out of vocabulary words applied to hindi-english cross-lingual information retrieval," *IETE Technical Review*, vol. 39, no. 2, pp. 276–285, 2020.

[31] E. Egorova and L. Burget, "Out-of-Vocabulary word recovery using fst-based subword unit clustering in a hybrid asr system," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5919–5923, Calgary, AB, Canada, April 2018.

[32] N. K. Singh, D. S. Tomar, and A. K. Sangaiah, "Sentiment analysis: a review and comparative analysis over social media," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 1, pp. 97–117, 2020.

[33] E. Elakiya and N. Rajkumar, "RETRACTED ARTICLE: In text mining: detection of topic and sub-topic using multiple spider hunting model," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 3, pp. 3571–3580, 2021.

[34] K. Liu, *Incorporate out-of-Vocabulary Words for Psycholinguistic Analysis Using Social Media Texts-An OOV-Aware Data Curation Process and a Hybrid Approach*, The Claremont Graduate University, 2021, (Doctoral dissertation.

[35] S. Zong, A. Baheti, W. Xu, and A. Ritter, *Extracting a Knowledge Base of COVID-19 Events from Social Media*, arXiv, 2021.

[36] R. Ferreira, R. D. Lins, S. J. Simske, F. Freitas, and M. Riss, "Assessing sentence similarity through lexical, syntactic and semantic analysis," *Computer Speech & Language*, vol. 39, pp. 1–28, 2016.

[37] D. Maynard, I. Roberts, M. A. Greenwood, D. Rout, and K. Bontcheva, "A framework for real-time semantic social media analysis," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 44, pp. 75–88, 2017.

[38] H. Mulki, C. B. Ali, H. Haddad, and I. Babaoğlu, "Tw-star at semeval-2018 task 1: preprocessing impact on multi-label emotion classification," in *Proceedings of The 12th International Workshop on Semantic Evaluation*, pp. 167–171, New Orleans, Louisiana, USA, June 2018.

[39] S. K. Narayanasamy, K. Srinivasan, S. Mian Qaisar, and C. Y. Chang, "Ontology-enabled emotional sentiment analysis on COVID-19 pandemic related twitter streams," *Frontiers in public health*, vol. 9, 2021.

[40] A. Pinto, H. Gonçalo Oliveira, and A. Oliveira Alves, "Comparing the performance of different NLP toolkits in formal and social media text," in *5th Symposium on Languages, Applications and Technologies (SLATE'16)*, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[41] M. A. Hassonah, R. Al-Sayyed, A. Rodan, A. Z. Ala'M, I. Aljarah, and H. Faris, "An efficient hybrid filter and evolutionary wrapper approach for sentiment analysis of various topics on twitter," *Knowledge-Based Systems*, vol. 192, article 105353, 2020.

[42] D. Bollegala, R. Kiryo, K. Tsujino, and H. Yukawa, "Language-independent tokenisation rivals language-specific tokenisation for word similarity prediction," 2020, https://arxiv.org/abs/2002.11004.

[43] R. Dridan and S. Oepen, "Tokenization: returning to a long solved problem—a survey, contrastive experiment, recommendations, and toolkit—," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 378–382, Jeju Island, Korea, July 2012.

[44] D. Yogish, T. N. Manjunath, and R. S. Hegadi, "Review on Natural Language Processing Trends and Techniques Using

Nltk," in *International Conference on Recent Trends in Image Processing and Pattern Recognition*, pp. 589–606, Springer, Singapore, 2018.

[45] S. Chatterji, R. K. Rahul, and A. Arora, "A hybrid approach for identifying sentiments around aspects," in *2015 IEEE 2nd International Conference on Recent Trends in Information Systems (ReTIS)*, pp. 33–37, Kolkata, India, July 2015.

[46] S. Medeiros, F. Mascarenhas, and R. Ierusalimschy, "From regexes to parsing expression grammars," *Science of Computer Programming*, vol. 93, pp. 3–18, 2014.

[47] N. Hanafiah, A. Kevin, C. Sutanto, Y. Arifin, and J. Hartanto, "Text normalization algorithm on twitter in complaint category," *Procedia computer science*, vol. 116, pp. 20–26, 2017.

[48] V. C. Tran, N. T. Nguyen, H. Fujita, D. T. Hoang, and D. Hwang, "A combination of active learning and self-learning for named entity recognition on twitter using conditional random fields," *Knowledge-Based Systems*, vol. 132, pp. 179–187, 2017.

[49] Y. Jamoussi and A. Y. Nouira, "An extracting model for constructing actions with improved part-of-speech tagging from social networking texts," in *2017 11th international conference on intelligent systems and control (ISCO)*, pp. 77–81, Coimbatore, India, January 2017.

[50] S. K. Narayanasamy and M. Dinakaran, "An algorithmic approach to rank the disambiguous entities in Twitter streams for effective semantic search operations," *Sādhanā*, vol. 45, no. 1, p. 29, 2020.

[51] C. Iwendi, C. G. Huescas, C. Chakraborty, and S. Mohan, "COVID-19 health analysis and prediction using machine learning algorithms for Mexico and Brazil patients," *Journal of Experimental & Theoretical Artificial Intelligence*, pp. 1–21, 2022.

[52] A. Kumar, K. Srinivasan, W.-H. Cheng, and A. Y. Zomaya, "Hybrid context enriched deep learning model for fine-grained sentiment analysis in textual and visual semiotic modality social data," *Information Processing & Management*, vol. 57, no. 1, article 102141, 2020.

[53] S. Chauhan, R. Banerjee, C. Chakraborty, M. Mittal, A. Shiva, and V. Ravi, "A self-congruence and impulse buying effect on user's shopping behaviour over social networking sites: an empirical study," *International Journal of Pervasive Computing and Communications*, vol. 17, no. 4, pp. 404–425, 2021.

[54] A. Shabbir, M. Shabbir, A. R. Javed, M. Rizwan, C. Iwendi, and C. Chakraborty, "Exploratory data analysis, classification, comparative analysis, case severity detection, and internet of things in COVID-19 telemonitoring for smart hospitals," *Journal of Experimental & Theoretical Artificial Intelligence*, pp. 1–28, 2022.

[55] P. A. Pattanaik, M. Mittal, M. Z. Khan, and S. N. Panda, "Malaria detection using deep residual networks with mobile microscopy," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 5, pp. 1700–1705, 2022.

[56] A. R. Javed, M. U. Sarwar, S. Khan, C. Iwendi, M. Mittal, and N. Kumar, "Analyzing the effectiveness and contribution of each axis of tri-axial accelerometer sensor for accurate activity recognition," *Sensors*, vol. 20, no. 8, p. 2216, 2020.

[57] S. Ramaneswaran, S. Vijay, and K. Srinivasan, "TamilATIS: dataset for task-oriented dialog in Tamil," in *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pp. 25–32, Association for Computational Linguistics, Dublin Ireland, 2022.

[58] T. Finin, W. Murnane, A. Karandikar, N. Keller, J. Martineau, and M. Dredze, "Annotating named entities in Twitter data with crowdsourcing," in *Proceedings of the NAACL Workshop on Creating Speech and Text Language Data With Amazon's Mechanical Turk*, pp. 80–88, Los Angeles, 2010.

[59] A. Ritter, C. Cherry, and B. Dolan, "Data-driven response generation in social media," in *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 583–593, Edinburgh, Scotland, UK, 2011.

[60] M. Rowe and H. Alani, "Mining and comparing engagement dynamics across multiple social media platforms," in *Proceedings of the 2014 ACM conference on Web science*, pp. 229–238, New York, NY, USA, June 2014.

[61] L. Derczynski, K. Bontcheva, and I. Roberts, "Broad twitter corpus: A diverse named entity recognition resource," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 1169–1179, Osaka, Japan, December 2016.