Jin et al. *EPJ Data Science* (2024) 13:44
https://doi.org/10.1140/epjds/s13688-024-00465-2

EPJ Data Science
a SpringerOpen Journal

EPJ.org

RESEARCH Open Access

Analyzing user reactions using relevance between location information of tweets and news articles

Yun-Tae Jin[1], JaeBeom You[2], Shoko Wakamiya[3] and Hyuk-Yoon Kwon[4*]

*Correspondence:
hyukyoon.kwon@seoultech.ac.kr
[4]Department of Industrial Engineering / Graduate School of Data Science, Seoul National University of Science and Technology, 01811, Seoul, South Korea
Full list of author information is available at the end of the article

Abstract

In this study, we analyze the extent of user reactions based on user's tweets to news articles, demonstrating the potential for home location prediction. To achieve this, we quantify users' reactions to specific news articles based on the textual similarity between tweets and news articles, showcasing that users' reactions to news articles about their cities are significantly higher than those about other cities. To maximize the difference in reactions, we introduce the concept of *News Distinctness*, which highlights the news articles that affect a specific location. By incorporating News Distinctness with users' reactions to the news, we magnify its effects. Through experiments conducted with tweets collected from users whose home locations are in five representative cities within the United States and news articles describing events occurring in those cities, we observed a 6.75% to 40% improvement in the reaction score when compared to the average reactions towards news for outside of home location, clearly predicting the home location. Furthermore, News Distinctness increases the difference in reaction score between news in the home location and the average of the news outside of the home location by 12% to 194%. These results demonstrate that our proposed idea can be utilized to predict the users' location, potentially recommending meaningful information based on the users' areas of interest.

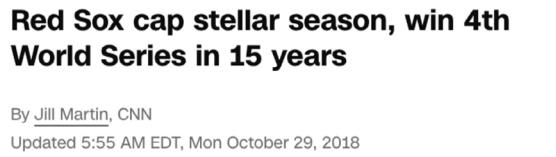
Keywords: Textual similarity; Location Prediction; SNS analysis; News distinctness

1 Introduction

Twitter is an online social networking service (SNS) where users communicate with each other by posting messages known as "tweets", which consist of various data types such as short texts, images, videos, audio, links, and more, all within a limit of 140 characters. Twitter generates around 250 billion tweets annually, and due to its real-time nature, big data potential, and open characteristics, the tweets have been widely utilized in diverse fields such as marketing, advertising, and research [1]. A tweet typically includes text content, media attachments, hashtags, mentions, links, retweets, timestamps, and geo-location. Among these, geo-location data pertains to the home location where the user actually resides and the location from which the tweet was sent.

Springer

(a) Tweet.



(b) News.

Figure 1 An example of a tweet and a news article describing the same event on the same day. Tweet and news content with textual information discuss the victory of the Boston Red Sox in the 2018 World Series. The tweet has a high textual similarity to news about the Red Sox winning the World Series. This implies the possibility of identifying tweets that react to news articles based on textual similarity

Tweets are characterized by their ease of sharing information and interconnection among users, leading to the rapid propagation of news and information. Figure 1 illustrates an example of a news article and a tweet related, which are uploaded on the same date, to a specific event. Since both tweets and news articles consist of textual information, this implies the possibility of identifying tweets that react to news articles based on their textual similarity.

The user location makes great opportunities because it can be utilized for purposes such as targeted advertising, crime tracking, understanding trends and patterns related to consumer preferences, and more. Given that only 16% of all users are presently openly sharing locations due to privacy and data leakage concerns, the effective prediction of this data has become exceedingly important [2].

In this study, we propose a method of measuring the *reaction scores* of tweets to news articles based on their similarity and demonstrating that the reaction score of the user to the news articles for their location is higher than that for other locations. Our main idea is based on the claims that the overall reaction scores of users to events covered in the news articles for a specific location align closely with the home location. This study used the city level as the location level as most previous work for the following reasons:

easy and clear identification and appropriate privacy level. For further improvement, we introduce the concept of the *News Distinctness* to differentiate the news articles according to their distinctness to a specific location. In other words, because some important news events can attract interest from users residing in various locations regardless of the home location, we penalize them and highlight the news articles related to a specific location.

Through experiments involving tweets collected from users whose home locations at the city level are in five major cities within the United States and news articles specifying events that occurred in those cities, we validate the proposed method. By introducing a variety of word embedding techniques for reaction score measurement, we consistently observe substantial improvements in comparison to average reactions towards news articles from outside of the user home location: Word2vec (6.75% $\sim$ 15.75% improvement), Glove (7.5% $\sim$ 23.75% improvement), FastText (10% $\sim$ 40% improvement), and Doc2Vec (9.25% $\sim$ 21% improvement). Additionally, the introduction of the results of News Distinctness in more enhancements, increasing the difference in reaction score between news in home location and the average of the news in outside of home location by 65% $\sim$ 194% for Word2vec, 23% $\sim$ 159% for Glove, 15% $\sim$ 129% for FastText, and 12% $\sim$ 84% for Doc2Vec. Through these findings, we affirm the potential applicability of the proposed method to real-world problems of predicting SNS home location.

The structure of this paper is as follows. In Sect. 2, we introduce related studies. In Sect. 3, we describe the method of our study. In Sect. 4, we present the results of the study and conclude it. Finally, In Sect. 5, we discuss the results.

2 Related work

2.1 Textual similarity

Tajbakhsh et al. [3] focused on the impact of semantic similarity on the problem of the recommendation of hashtags. They proposed a TF-IDF-based weighting vector, which redefines semantic weights and tweet similarity. By recommending hashtags for the top N similar tweets, they demonstrated significant improvements compared to traditional TF-IDF. Kirikae et al. [4] incorporated the semantics of tweets and user subjectivity to evaluate tweet similarity. They estimated quality of experience (QoE) and improved performance through classification based on the number of inclusions, surpassing previous methods. Peng et al. [5] proposed a knowledge enhancement-based multigranularity semantic embedding model structure. This structure involved multiple levels of models such as character embedding, word embedding, and Bi-LSTM, addressing the semantic similarity problem caused by text length differences and inconsistency of subnetworks. Giannaris et al. [6] conducted research to measure the similarity of tweets related to the Russia-Ukraine conflict, aiming to prove the hypothesis that tweets related to the same conflict from the same newsroom are very similar.

In this study, we use an embedding model to compute the similarity between news and tweets, which is used to measure the reaction score of tweets to news articles. Because our model does not depend on a specific word embedding, we use relatively simple yet effective models, such as Word2Vec, Glove, FastText, and Doc2Vec, showing their effectiveness. Notably, more advanced embedding models can be extended to our method.

2.2 Event detection with user reactions

There have been research efforts to detect general events from SNS media. Weng et al. [7] introduced event detection with the clustering of wavelet-based signals based on the content posted by users on Twitter. They constructed individual signals for each word based on user-generated content and then filtered and clustered these signals using their correlations, exhibiting high performance in event detection. Nguyen et al. [8] proposed a model for detecting events from SNS text data generated in real time. They introduced a novel approach to distributed computation and data aggregation, aiming to detect abnormal events by monitoring the number of participating users and the rate of message interactions related to specific topics. They developed a model incorporating this technique to extract and track real-world social events.

Moreover, there have been research efforts to detect specific types of events, such as climate change, cybersecurity events, and hazard events. Dahal et al. [9] analyzed climate change discussions on Twitter using topic modeling and sentiment analysis from both geospatial and temporal perspectives. They extracted various events and topics related to climate change and analyzed users' reactions to climate change events. Shin et al. [10] proposed a contrastive word embedding model to detect cybersecurity-related tweets by analyzing the tweet users who usually write relevant tweets. The proposed model was based on two embedding models based on two contrastive data corpus according to the positiveness or negativeness of cybersecurity. They demonstrated the proposed model significantly improved the existing model for classifying cybersecurity-related tweets. Peng et al. [11] addressed the significant increase in social media usage during hazard events, such as natural disasters. They proposed a novel indicator considering various factors such as tweet occurrence, population, internet usage rates, and natural hazard characteristics per geo-location. Using this indicator, they conducted spatio-temporal analysis on millions of tweet contents, providing valuable analysis results for crisis response and management. Park et al. [12] measured the relevance of tweets to cyberattack-related events and identified the most influential community through community detection to predict cyberattacks. They showed the effectiveness of the proposed method compared to various baseline methods. Kim et al. [13] proposed a streaming event detection related to cybersecurity by monitoring the tweets written by users in a distributed environment. In particular, they focused on the efficient module update to respond to the event changes by proposing the partial model update strategy for the deep learning classification model.

In contrast to detecting events by utilizing the texts written by SNS users as in the previous studies, our work focuses on measuring and analyzing the reaction score of users to news based on the text similarity and news distinctness to demonstrate users are more likely to react to news relevant to their home location.

2.3 Location prediction with SNS contents

Numerous studies have proposed methods for predicting user locations using SNS content such as tweets. Mamud et al. [14] introduced a hierarchical ensemble algorithm combining statistical and heuristic classification to predict users' city-level locations. Zhang et al. [15] extracted distance and address information, spatial relationships between buildings and cities, and toponyms from tweet text. They used heuristic methods, open-sourced named entity recognition (NER) software, and machine learning techniques to predict lo-

cation at the level of buildings and toponyms. Malmasi et al. [16] improved performance for detecting SNS messages mentioning locations from noisy SNS text data using an approach based on noun phrase extraction and n-gram matching, outperforming methods such as NER or conditional random field (CRF).

Recently, research efforts using deep learning or machine learning-based approaches have been actively proposed. Kumar et al. [17] proposed a convolutional neural network-based model to extract various levels of geolocation information, such as building, city, district, and country names, from tweet text. Tang et al. [18] estimated user locations with higher performance using a multilayer recognition model to filter noisy tweet data generated by users. Mahajan et al. [19] combined CNN and LSTM layers to predict tweet locations, achieving a high accuracy of 92.6% in city-level predictions. They focused on extracting useful features associated with tweets. Simanjuntak et al. [20] found that IndoBERT outperformed existing machine learning and deep learning algorithms in predicting user location from tweets generated in Indonesia, using user names, user introductions, and tweet text attributes. Mostafa et al. [21] used machine learning models based on sentiment analysis to extract user locations from tweets, even in cases without geolocation clues. Among the nine models used in the experiment, the decision tree model showed the best effectiveness.

Such as many previous studies cited above, we also used the city level as the location level due to its clear definition of the location. The aforementioned studies mainly focused on only SNS contents for effective text similarity, event detection, and location prediction. We note that, however, no studies have performed user home location prediction based on the relevance analysis of SNS content with external data sources. This study is the first research effort to conduct the relevance analysis of SNS content with news articles to predict home locations at the city level.

3 Methods

3.1 Overall architecture

Figure 2 illustrates the architecture proposed in this study for analyzing reaction scores of the tweet to the news article.

1 *Data Collection*: We collect tweets from Twitter users who designate their home locations at the city level. That is, those tweets are regarded as generated from the user's home locations. At the same time, we collect news articles that mention specific locations at the city level and regard those news articles as generated from the mentioned locations. We classify tweets and news articles by location, respectively. To analyze the same locations, we define them in advance.
2 *Data Pre-processing*: We conduct typical pre-processing of texts for both tweets and news articles to be required for further analysis.
3 *Text Similarity Measurement*: For each combination of the tweet's location and news article's location, we generate pairs of tweets and news articles where tweets are generated within a certain period after news articles. In this context, to measure the relevancy between tweets and news articles, we calculate Text Similarity between the tweets and news articles for each combination of the tweet and news article locations.
4 *News Distinctness*: We define the concept of *News Distinctness* to consider the importance of the news articles reflecting their impacts on a specific location. Then,

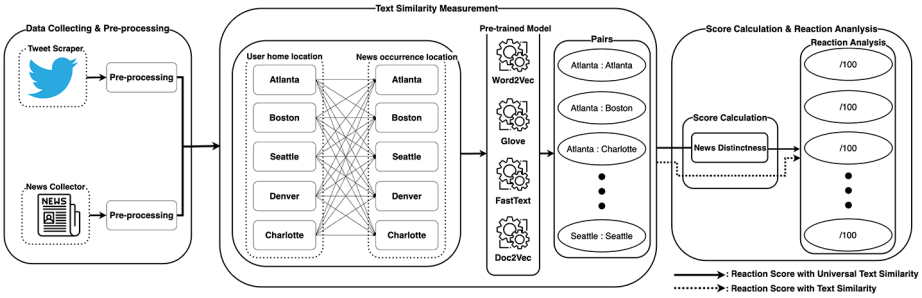


Figure 2 Proposed architecture. The process starts with collecting tweets from users in a home location and news articles mentioning the location, followed by text pre-processing. Next, Text Similarity between tweets and news articles is measured. Next, News Distinctness is reflected to consider the impact of news on specific locations. Next, the relevance of the top-ranked tweets to the news articles is evaluated, determining a reaction score based on this relevancy. Finally, these scores are compared across different location pairs to assess the impact of location on news reactions

```
near:Denver within:5mi
since:2022-09-01 until:2022-09-30
lang:en
```

Listing 1 Tweet search query

we incorporate the new distinctness with the original Text Similarity to improve the accuracy of measuring the reaction of tweets to news articles.

5 *Reaction Analysis*: In each combination of the tweet and news article locations, the top-N tweets are identified in the order of the highest Text Similarity. We manually verify if each tweet is actually relevant to the news article and define the relevant ratio of the entire tweet as the reaction score. Finally, we compare the reaction scores between different pairs of the tweet and news article locations, confirming that the reaction score from the same location pair is clearly higher than that from other location pairs.

3.2 Data collection

In this study, we collect tweets, including the username, tweet text, and posting date of tweets written by users whose home locations at the city level are within a 5-mile radius of the target geo-location (i.e., city) [22, 23]. Listing 1 provides an example of a query, representing "tweets within a 5-mile radius of Denver, occurring in September 2022." Subsequently, to identify users actively engaged in using SNS, we choose users who have written a certain number of tweets per month and year.

For collecting news data, we utilize Selenium to directly search for the target geo-location name on Google News. To select geo-location-related news occurring within the chosen timeframe, we select only news articles where the number of a geo-location name loc mentioned in a given news document $news$, $n(news, loc)$, is equal to or above a certain threshold $threshold_{news,loc}$. To determine an appropriate $threshold_{news,loc}$, we sample the geo-location-related news articles and manually measure the average number of geo-location names mentioned in the news article. Each of the collected news articles includes the news title, content, publication date, and the URL.

3.3 Data pre-processing

We first remove unnecessary keywords from the collected text data by defining a set of stopwords. This stopword set is constructed by combining stopwords from the Python nltk library, those used in Microsoft's Bot Builder, and those utilized in MySQL databases.

For further pre-processing of tweets, each tweet's text undergoes the removal of redundant hashtags and emojis. Subsequently, to ensure the result of Text Similarity, tweets with a word count below a certain threshold, $threshold_{preprocessing}$, are eliminated.

Regarding the pre-processing of news articles, we eliminate the duplicated news articles based on the news article title while they are generated on the same date. Furthermore, to minimize fluctuations of documents' vector values due to differences in document lengths, we extract the first $n_{sentences}$ sentences of all news articles, commonly considering that the main content of news articles is generally presented in the initial part of the article.

3.4 Text similarity measurement

3.4.1 Text embedding

To measure the textual similarity between two documents A and B, we utilize a pre-trained embedding model to derive vectors for the words in each document. When using a word embedding model, after pre-processing, embeddings for all words in the documents are extracted and averaged to obtain the document's representation. In the case of document embedding, the entire text of the document is provided as a single input to capture the semantic representation of the entire document through an embedding vector.

3.4.2 Cosine similarity

For respective documents, A and B, the text embedding process is performed, and then *Cosine Similarity* between A and B is calculated using Eq. (1) as a measure of Text Similarity between documents. The resulting ranges of Text Similarity between 0 and 1, where a score of 1 indicates that the two documents are identical.

$$Cosine\ Similarity = \frac{A \cdot B}{\|A\| \cdot \|B\|} \tag{1}$$

3.4.3 Implementation to tweet and news

For each combination of a tweet location t_{loc} and a news location n_{loc}, (t_{loc}, n_{loc}), a pair of a written tweet t and a news article n, (t, n) is considered for the target analysis where t is written within specified days, $interval(t, n)$, after n is generated. Text Similarity is then calculated for each pair. To derive embedding vectors for tweets and news articles, pre-trained embedding models are utilized. Once the embedding vectors A and B are obtained for each text, Text Similarity is determined using Cosine Similarity according to Eq. (1). We sort (t, n) pairs in descending order based on their values of Text Similarity to focus on relevant pairs for each combination of locations.

3.5 Reaction analysis

3.5.1 User reactions to news articles

We conduct reaction analysis on the results of Text Similarity between the tweets and news articles to measure the users' reaction degree to the locations associated with news articles. Table 1 shows the used notations. The reaction score is calculated as shown in Eq. (2). Given T_{loc} and N_{loc}, $TN(T_{loc}, N_{loc})$ is defined as the top-N pairs sorted by Text

Table 1 Notations

Symbols	Description
LOC	a set of the total locations
$USER_{loc}$	a set of the users residing in a location *loc*
T_{loc}	a set of the tweets written by $USER_{loc}$
N_{loc}	a set of the news articles that occurred in a location *loc*
N	$\sum_{loc \in LOC} N_{loc}$
TN_{T_i,N_j}	a set of top-m pairs of tweets in T_i and news articles in N_j in the order of their Text Similarity.
$TN_{T_i,N_j}(n)$	a pair of a tweet and a news article where the news article is n out of TN_{T_i,N_j}.

Similarity for pairs of t_{loc} and n_{loc}. Among $TN(T_{loc}, N_{loc})$, the actual relevance between them is manually validated and the validated pairs are defined as $Rel(T_{loc}, N_{loc})$.

$$Reaction\ Score(T_{loc}, N_{loc}) = \frac{|Rel(T_{loc}, N_{loc})|}{|TN(T_{loc}, N_{loc})|} \tag{2}$$

For example, a pair of the tweet and news article locations (Atlanta, Boston) represents a group of pairs of tweets and news articles representing 'tweets written by users who reside in Atlanta and news articles related to Boston'. The top 100 pairs of tweets and news articles are obtained in the order of Text Similarity and the reaction scores are calculated according to Eq. (2).

3.5.2 News distinctness

As a novel feature of the proposed method, we consider the impact of the case where a certain news article has a universal impact across diverse geo-locations. For instance, a news article about '2018 Boston RedSox World Series victory', which widely affects the global US states, showed consistently a high reaction score across diverse locations. Therefore, to hinder these effects and to focus more on news articles for specific cities, we need to mitigate the universal impact of news articles.

Inspired by the document frequency (DF) concept in TF-IDF [24], we define *News Universality* to penalize the news that has a universal impact by quantifying the extent to which a specific news article affects not only a certain location but the overall locations. Therefore, we design *Inverse News Universality* for a news article n as described in Eq. (3). We measure the frequency of a specific news article n out of all the tweet and news article pairs over the diverse cities in the denominator while we normalize it by the total number of pairs in the numerator. A higher value of Inverse News Universality indicates lower universality, meaning the news article's impact is more localized, whereas a lower score indicates higher universality, implying that the news article's impact is more widespread.

$$\begin{aligned} &Inverse\ News\ Universality(n) \\ &= log\left(\frac{\sum_{loc \in LOC} \sum_{u \in u_{loc}} |TN_{T_u,N}|}{1 + \sum_{loc \in LOC} \sum_{u \in u_{loc}} |TN_{T_u,N}(n)|}\right) \end{aligned} \tag{3}$$

We define *News Frequency* for a news article n, given a tweet location, T_{loc}, and a news article location, N_{loc}, as described in Eq. (4) to represent the importance of n. We measure the frequency of a specific news article n out of the tweet and news article pairs on T_{loc} and N_{loc} in the numerator while we normalize it by the total number of pairs of tweets and

news articles on T_{loc} and N_{loc} in the denominator.

$$News\ Frequency(n, T_{loc}) = \frac{\sum_{u \in T_{loc}} |TN_{T_u,N}(n)|}{\sum_{u \in T_{loc}} |TN_{T_u,N}|} \tag{4}$$

Finally, we define News Distinctness by combining Inverse News Universality and News Frequency as shown in Eq. (5).

$$\begin{aligned} &News\ Distinctness(n, T_{loc}) \\ &= News\ Frequency(n, T_{loc}) \times Inverse\ News\ Universality(n) \end{aligned} \tag{5}$$

By combining News Distinctness with Text Similarity, we obtain *Universal Text Similarity* as shown in Eq. (6). By adopting News Distinctness, we can refine the Text Similarity score by focusing on the news articles related to specific locations. k is used to adjust the weight of News Distinctness.

$$\begin{aligned} &Universal\ Text\ Similarity(t, n, T_{loc}) \\ &= Text\ Similarity(t, n) + k \times News\ Distinctness(n, T_{loc}) \end{aligned} \tag{6}$$

The final *Universal Text Similarity* is calculated according to Eq. (6). Then, we order the (t, n) pairs for each location pair (T_{loc}, N_{loc}) based on Universal Text Similarity. This score is used to obtain $TN(T_{loc}, N_{loc})$ in Eq. (2). This method aims to provide a more accurate gauge of user reactions to news articles by considering both textual semantic similarity and the news article's distinctness across various locations.

4 Results

4.1 Experimental dataset and environments

Figure 3 displays Tweetmap, which visualizes all tweets with geo-location information on a map. Using Tweetmap as a reference, 5 representative cities in the US were selected to compare the relative reaction of tweets to those cities. The selected cities are Atlanta,

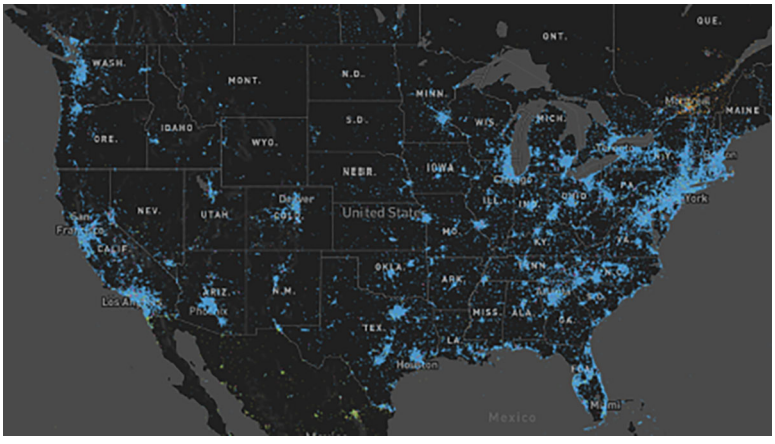


Fig. 3 Tweetmap [25]. Tweetmap visualizes the occurrence of tweets based on geo-location on a map. We selected 5 representative cities that are not close to each other based on the visualization result

Table 2 Collected Dataset

Location	# of News Articles	# of Tweets (# of Tweet Users)
Atlanta	103	456,868 (589)
Boston	157	367,470 (466)
Seattle	170	415,568 (568)
Charlotte	100	364,194 (462)
Denver	140	210,490 (311)

Table 3 Pre-trained Embedding Model List

Embedding Method	Pre-trained Model
Word2Vec	GoogleNews-vectors-negative 300 [30]
Glove	Glove-Twitter2B [31]
FastText	crawl-300d-2M-subword [32]
Doc2Vec	English Wikipedia DBOW [33]

Boston, Charlotte, Denver, and Seattle. Tweet data written by users who designated these 5 cities as their home location in the profile from January 2018 to December 2018, as well as news data associated with these 5 cities, were collected. The target news articles to collect are determined by $threshold_{news,loc}$ of 5. Data pre-processing was carried out with $threshold_{preprocessing}$ set to 5, and $n_{sentences}$ set to 10. To focus on a relevant pair of a tweet t and a news article n, we set $interval(t, n)$ to 2. We randomly select 50 users for $USER_{loc}$ of each loc. We set top-m for $|TN_{T_{loc},N}|$ as 100. The collected sets of tweets and news data are summarized in Table 2.

The list of pre-trained models used for extracting word embedding vector values in this study is provided in Table 3.

Word2Vec [26], developed by Google in 2013, is a methodology for vectorizing sentences at the word level to infer semantic similarity between words, considering their meanings. It maintains contextual meanings and associations of words, enabling semantic deductions. However, it struggles with out-of-vocabulary words and may not fully consider global word co-occurrence, leading to potential omissions in semantic relationships.

Global Vectors for Word Representation(GloVe) [27] is an algorithm similar to Word2Vec but uses a co-occurrence matrix to represent word frequencies. This method combines global word co-occurrence with local context, offering a balanced view of semantic relationships. However, it falls short in explicitly handling subword information and involves complexities in matrix factorization computations.

FastText [28] breaks down words into n-grams and sums these vectors to create word vectors. It effectively handles rare and out-of-vocabulary words, providing multi-level embeddings that capture various semantic aspects of words. However, this approach increases memory requirements and can be slower in inference compared to other models due to additional computations for subword embeddings.

Doc2Vec [29] extends Word2Vec to vectorize text segments like sentences, paragraphs, or documents. It generates continuous, dense vector representations for entire documents, facilitating tasks like document similarity, topic modeling, sentiment analysis, and classification. Despite its flexibility in handling varying input lengths, its training process can be complex, often requiring more data for effective generalization and posing challenges in hyperparameter optimization.

Table 4 Reaction scores with Text Similarity. Let us suppose that T_{loc} is Boston and N_{loc} is Boston. After validating 100 pairs of tweets and news articles, if the number of validated pairs is 35, then the reaction score becomes 35%

Embedding Model	Twitter	News				
		Boston	Atlanta	Seattle	Denver	Charlotte
Word2Vec	Boston	**35**	26	19	14	18
	Atlanta	28	**31**	23	23	19
	Seattle	29	22	**31**	19	13
	Denver	20	17	19	**26**	14
	Charlotte	22	16	17	14	**24**
Glove	Boston	**42**	26	23	13	11
	Atlanta	20	**28**	18	20	17
	Seattle	23	22	**31**	7	10
	Denver	24	23	18	**28**	12
	Charlotte	22	22	17	13	**26**
FastText	Boston	**54**	26	20	3	7
	Atlanta	34	**41**	11	14	5
	Seattle	33	27	**38**	8	18
	Denver	21	17	7	**24**	11
	Charlotte	21	18	18	9	**27**
Doc2Vec	Boston	**35**	23	15	8	10
	Atlanta	26	**37**	10	10	17
	Seattle	24	19	**28**	13	14
	Denver	29	24	13	**31**	18
	Charlotte	24	23	13	15	**28**

4.2 Experimental results

Table 4 presents the results of measuring the reaction score among various pairs of (T_{loc}, N_{loc}) when we use Text Similarity across various embedding methods. According to the results, it is evident that the reaction score is normally the highest when N_{loc} matches T_{loc}. Notably, FastText shows the highest performance, with an average reaction score of 36.8%.

Figure 4 presents a comparison graph based on Table 4. It illustrates the reaction scores when T_{loc} and N_{loc} are the same, denoted by *correct*, the average reaction scores when they are not, denoted by *avg*(*incorrect*), and the highest reaction scores when they are not, denoted by *max*(*incorrect*), by each embedding model. As shown in the graph, we can observe a clear improvement due to the employment of the reaction score. Specifically, the improvement of *correct* over *avg*(*incorrect*) is as follows: 6.75% $\sim$ 15.75% in Word2Vec, 7.50% $\sim$ 23.75% in Glove, 10.00% $\sim$ 40.00% in FastText, and 9.25% $\sim$ 21.00% in Doc2Vec. The improvement of *correct* over *max*(*incorrect*) is as follows: 2.00% $\sim$ 9.00% in Word2Vec, 4.00% $\sim$ 16.00% in Glove, 3.00% $\sim$ 28.00% in FastText, and 2.00% $\sim$ 11.00% in Doc2Vec.

Overall, despite the significant impact of the reaction score, it is important to note a phenomenon of overall high reaction for news articles to a specific location, i.e., Boston. This suggests that news from a specific location may also affect users in different locations. This emphasizes that we need to mitigate this issue by adopting the concept of News Distinctness.

We improve the reaction score by considering News Distinctness defined in Eq. (5). The optimal weight k was obtained while varying k from 1 to 2, 5, and 10, setting k to 5. Table 5 presents the results when we use Universal Text Similarity in Eq. (6) to measure the reaction score. We note that the overall performance clearly improves compared to the reaction score with Text Similarity in Table 4.

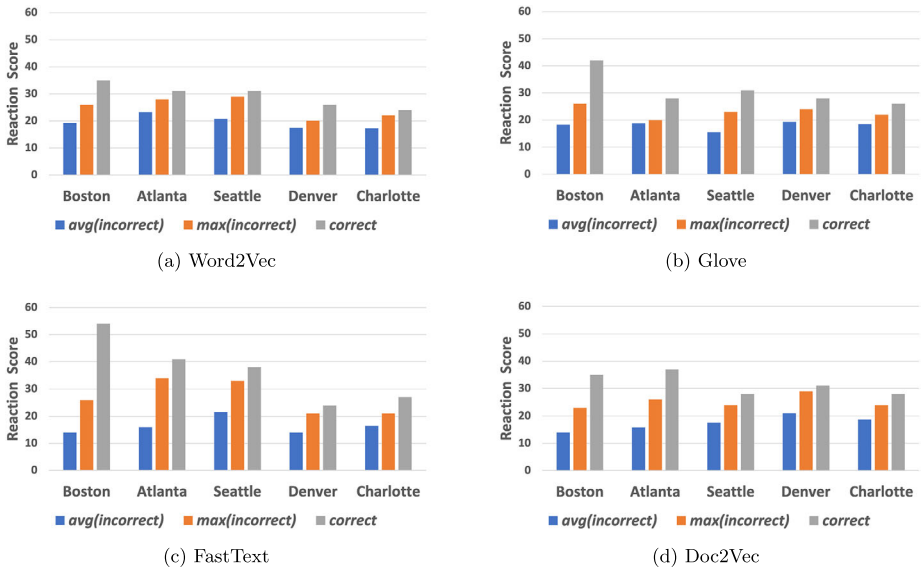


Fig. 4 Reaction score analysis with Text Similarity. The legend in the figure, *correct*, means the reaction score for N_{loc} matches T_{loc}, and *avg*(*incorrect*) and *max*(*incorrect*) mean the average and maximum values of the reaction score for N_{loc} not matches T_{loc}, respectively. Overall, the reaction score is clearly the highest when N_{loc} matches T_{loc}

Similar to Fig. 4, Fig. 5 presents a comparison graph based on Table 5. Specifically, the improvement of *correct* over *avg*(*incorrect*) is as follows: 14.00% ∼ 30.00% in Word2Vec, 16.75% ∼ 29.25% in Glove, 20.75% ∼ 46.00% in FastText, and 16.00% ∼ 33.50% in Doc2Vec. The improvement of *correct* over *max*(*incorrect*) is as follows: 10.00% ∼ 24.00% in Word2Vec, 7.00% ∼ 22.00% in Glove, 12.00% ∼ 35.00% in FastText, and 8.00% ∼ 31.00% in Doc2Vec.

The average of *correct* is also improved. For Word2Vec, there was an improvement of approximately 26.5% compared to Text Similarity, reaching 37.2% in Universal Text Similarity. For Glove, the improvement was around 18.0%, reaching 36.6% in Universal Text Similarity. For FastText, the improvement was about 19.0%, reaching 43.8% in Universal Text Similarity. Lastly, for Doc2Vec, there was an improvement of approximately 16.9%, reaching 37.2% in Universal Text Similarity.

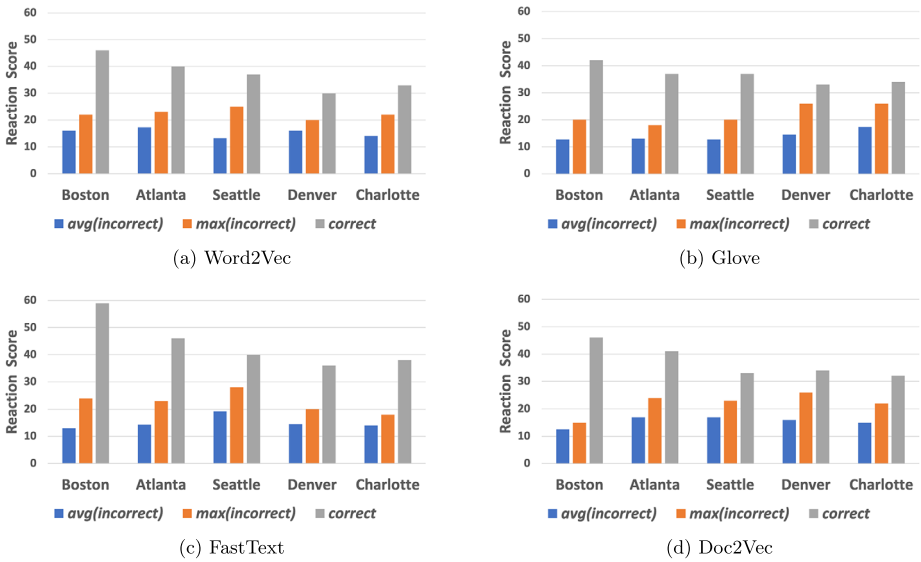
Figure 6 presents a comparison between the results obtained with Universal Text Similarity and the results only with Text Similarity. This comparison is based on the difference between *correct* and *avg*(*incorrect*), which we denote as *diff()* in the graph. The graph clearly shows the effectiveness of both cases, highlighting the effects of adopting the concept of News Distinctness. We note that Universal Text Similarity improves the value of *diff*() compared to Text Similarity. Specifically, Universal Text Similarity improves it by 65 ∼ 194% in Word2Vec, by 23 ∼ 159% in Glove, by 15∼ 129% in FastText, and by 12∼ 84% in Doc2Vec.

4.3 Conclusions

In this study, we proposed a method to measure the reaction scores of tweets written by users to news articles that occurred in specific locations. We observed that the reaction scores based on the measurement of Text Similarity on the text embedding increase

Table 5 Reaction scores optimized with $k = 5$ with Universal Text Similarity. After reflecting on Universal Text Similarity, we confirm that the reaction scores show a stronger tendency when N_{loc} matches T_{loc} compared to the results in Table 4

Embedding Model	Twitter	News				
		Boston	Atlanta	Seattle	Denver	Charlotte
Word2Vec	Boston	**46**	22	15	12	15
	Atlanta	23	**40**	18	18	10
	Seattle	25	14	**37**	8	6
	Denver	20	17	13	**30**	14
	Charlotte	22	14	14	6	**33**
Glove	Boston	**42**	20	10	8	13
	Atlanta	18	**37**	14	10	10
	Seattle	20	18	**37**	7	6
	Denver	26	14	8	**33**	10
	Charlotte	26	16	17	10	**34**
FastText	Boston	**59**	24	10	8	10
	Atlanta	23	**46**	11	18	5
	Seattle	28	27	**40**	8	14
	Denver	20	17	16	**36**	5
	Charlotte	18	17	10	11	**38**
Doc2Vec	Boston	**46**	15	14	9	12
	Atlanta	24	**41**	13	16	15
	Seattle	23	20	**33**	11	14
	Denver	26	12	14	**34**	12
	Charlotte	22	14	11	13	**32**



(a) Word2Vec (b) Glove

(c) FastText (d) Doc2Vec

Fig. 5 Reaction score analysis with Universal Text Similarity. The legend in the figure, *correct*, means the reaction score for N_{loc} matches T_{loc}, and *avg*(*incorrect*) and *max*(*incorrect*) mean the average and maximum values of the reaction score for N_{loc} not matches T_{loc}, respectively. Overall, the reaction score is clearly the highest when N_{loc} matches T_{loc}

when the user's home location at the city level matches with the news occurrence location. Specifically, the reaction scores when the home location matches with the news occurrence locations were higher than the average reaction score when they do not match by

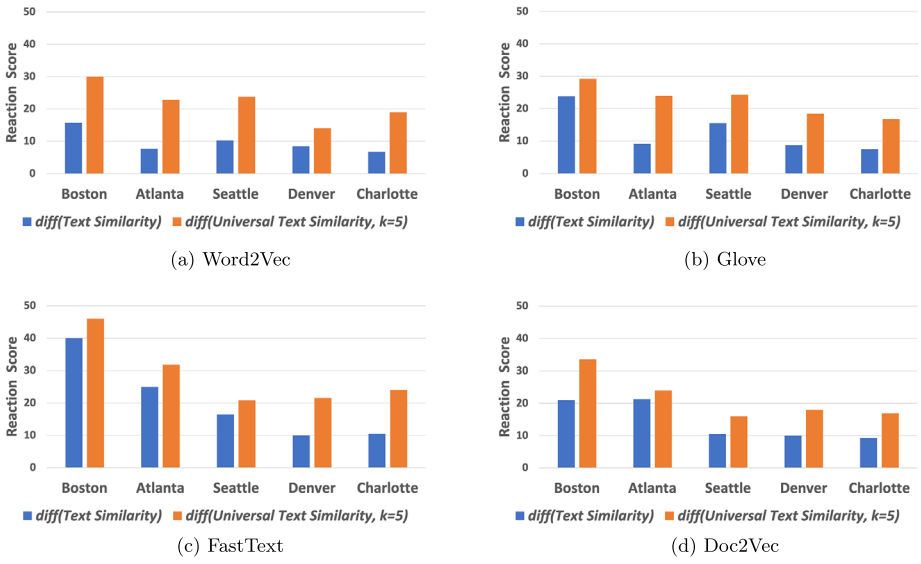


Fig. 6 Difference of the reaction scores between when the location of the tweet and news article is the same and when they are different. In the figure, *diff(Text Similarity)* means the difference between *correct* and *avg*(*incorrect*) of Fig. 4, and *diff(Universal Text Similarity, k = 5)* means the difference between *correct* and *avg*(*incorrect*) through the Table 4. The clear and high difference between *diff(Text Similarity)* and *diff(Universal Text Similarity, k = 5)* shows that the News Distinctness can more accurately measure the reaction score

approximately 6.75 ∼ 15.75% in Word2Vec, 7.5 ∼ 23.75% in Glove, 10 ∼ 40% in FastText, and 9.25 ∼ 21% in Doc2Vec.

Notably, to address the potential impact of the importance of news articles considering their universality spanning across multiple geo-locations, we introduced a novel metric called News Distinctness. By incorporating this metric, we significantly improved the difference in reaction score between news in the user's home location and the average of the news outside the home location. Specifically, it increased by 65 ∼ 194% in Word2Vec, 23 ∼ 159% in Glove, 15 ∼ 129% in FastText, and 12 ∼ 84% in Doc2Vec. This demonstrated that News Distinctness can more accurately measure the reaction score in terms of predicting the home location.

Based on the results of this study, using the proposed framework that utilizes reaction scores along with News Distinctness, we confirmed the potential to measure the reactions of specific news related to a user's home location. This suggests the framework's applicability to real-world problems, such as predicting the home location information of social media users.

5 Discussion

In this section, we discuss some limitations of this work and further study to resolve them. They are summarized as follows:

- In this study, we used the city level as the location level for the following several reasons, even if a lower location level could be more accurate and meaningful. First, collecting and cleaning social media data (SNS) at a lower level is challenging and resource-intensive. Second, privacy concerns increase when a more specific location is handled. Nevertheless, it would be beneficial if we could extend the location level dynamically to broader or narrower, which could be handled in future work.

- Because there is no efficient way to determine that the event describing each tweet is related to a specific city, we need to manually determine each one. Therefore, we selected five representative cities and manually determined the relevance between the tweets and those cities.
- Due to the first reason, the prediction granule of the location is limited to target cities. Therefore, our method cannot be flexibly applied to the location prediction of the user. Finally, our method cannot be directly compared with the existing studies to predict the user location. Nevertheless, we consistently showed the distinct difference between the target location and the others to each user in terms of responsibility. In particular, we note that, regardless of text embedding models, our proposed scheme shows consistent superiority.
- The proposed universality further differentiates the target location and the others. These results clearly show the potential of our method in terms of location prediction. This could be helpful for event detection, fake news detection, appropriate advertisement promotion, and region-based sentiment analysis regarding usability. However, predicting the location of an individual user may incur ethical considerations in the case where there is no intention to disclose the user's home location. Therefore, its cautious usage is required considering both usability and privacy under the complete information security system.
- While depending on the manual labeling in this study, it is beneficial to extend our work to be an automated method by determining if a Twitter user reacts to a specific location. Specifically, to do this, we need to define a clear criterion that determines if each tweet is relevant to each news article. Furthermore, to achieve more effective automation, we need to extract words, sentences, and phrases that have a significant impact on the relevance of tweets with news articles and dynamically adapt to the level of tweet occurrence. Based on this, we can extend our current scheme to wider cities and vary the granules as we want.
- In this study, we do not focus on an advanced text embedding model because our framework is not dependent on a specific text embedding model and shows its effectiveness even with the simple concept of Text Similarity. However, our work can be extended to adopt recent transformer-based text embedding models such as BERT, GPT, or more advanced models.

Abbreviations

SNS, Social Network Service; US, United States; URL, Uniform Resource Locator.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1F1A1067008), and by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2019R1A6A1A03032119).

Author contributions

Y-TJ – Conceptualization; Data curation; Investigation; Methodology; Software; Writing - original draft. J-BY - Data curation; Investigation; Methodology; Software; Validation; Visualization; Writing - review & editing. SW - Conceptualization; Investigation; Methodology; Validation; Writing - review & editing. H-YK – Conceptualization; Funding acquisition; Investigation; Methodology; Project administration; Resources; Supervision; Validation; Writing - review & editing. All authors read and approved the final manuscript.

Funding

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1F1A1067008), and by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2019R1A6A1A03032119).

Data availability
The datasets of Tweet and News Articles collected during the study are available at:
https://drive.google.com/drive/u/0/folders/1ElZiwp3dzVJGInLvnAEY9oVCf-fcs36H

Declarations

Competing interests
The authors declare no competing interests.

Author details
[1]Department of Industrial Engineering, Seoul National University of Science and Technology, 01811, Seoul, South Korea. [2]Graduate School of Data Science, Seoul National University of Science and Technology, 01811, Seoul, South Korea. [3]Division of Information Science, Graduate School of Science and Technology, Nara Institute of Science and Technology, 6300192, Nara, Japan. [4]Department of Industrial Engineering / Graduate School of Data Science, Seoul National University of Science and Technology, 01811, Seoul, South Korea.

Received: 26 December 2023 Accepted: 18 March 2024 Published online: 26 June 2024

References

1. Jurgens D, Finethy T, McCorriston J, Xu Y, Ruths D (2015) Geolocation prediction in Twitter using social networks: a critical analysis and review of current practice. In: Proceedings of the international AAAI conference on web and social media, vol 9, pp 188–197
2. Li G, Hu J, Feng J, Tan K-L (2014) Effective location identification from microblogs. In: 2014 IEEE 30th international conference on data engineering. IEEE, pp 880–891
3. Tajbakhsh MS, Bagherzadeh J (2016) Microblogging hash tag recommendation system based on semantic tf-idf: Twitter use case. In: 2016 IEEE 4th international conference on future Internet of things and cloud workshops (FiCloudW). IEEE, pp 252–257
4. Kirikae T, Zhang C, Yamori K, Tanaka Y (2020) Sns data based tweet similarity evaluation for qoe estimation. In: 2020 35th international technical conference on circuits/systems, computers and communications (ITC-CSCC), pp 269–272
5. Peng D, Hao B, Tang X, Chen Y, Sun J, Wang R (2020) Learning long-text semantic similarity with multi-granularity semantic embedding based on knowledge enhancement. In: Proceedings of the 2020 1st international conference on control, robotics and intelligent system, pp 19–25
6. Giannaris PS, Karamanoli V, Agathocleous A, Ilias I, Doukas N (2022) Text similarity study for Twitter-based news on Russian-Ukraine cyber war. In: 2022 12th international conference on dependable systems, services and technologies (DESSERT). IEEE, pp 1–8
7. Weng J, Lee B-S (2011) Event detection in Twitter. In: Proceedings of the international aaai conference on web and social media, vol 5, pp 401–408
8. Nguyen DT, Jung JJ (2015) Real-time event detection on social data stream. Mob Netw Appl 20:475–486
9. Dahal B, Kumar SA, Li Z (2019) Topic modeling and sentiment analysis of global climate change tweets. Soc Netw Anal Min 9:1–20
10. Shin H-S, Kwon H-Y, Ryu S-J (2020) A new text classification model based on contrastive word embedding for detecting cybersecurity intelligence in Twitter. Electronics 9(9):1527
11. Karimiziarani M, Jafarzadegan K, Abbaszadeh P, Shao W, Moradkhani H (2022) Hazard risk awareness and disaster management: extracting the information content of Twitter data. Sustain Cities Soc 77:103577
12. Park J-H, Kwon H-Y (2022) Cyberattack detection model using community detection and text analysis on social media. ICT Express 8(4):499–506
13. Kim M-S, Lim B-Y, Lee K, Kwon H-Y (2022) Effective model update for adaptive classification of text streams in a distributed learning environment. Sensors 22(23):9298
14. Mahmud J, Nichols J, Drews C (2012) Where is this tweet from? Inferring home locations of Twitter users. In: Proceedings of the international AAAI conference on web and social media, vol 6, pp 511–514
15. Zhang W, Gelernter J (2014) Geocoding location expressions in Twitter messages: a preference learning method. J Spat Inf Sci 9:37–70
16. Malmasi S, Dras M (2016) Location mention detection in tweets and microblogs. In: Computational linguistics: 14th international conference of the Pacific association for computational linguistics, PACLING 2015, Bali, Indonesia, May 19-21, 2015. Revised selected papers, vol 14. Springer, Berlin, pp 123–134
17. Kumar A, Singh JP (2019) Location reference identification from tweets during emergencies: a deep learning approach. Int J Disaster Risk Reduct 33:365–375
18. Tang H, Zhao X, Ren Y (2022) A multilayer recognition model for twitter user geolocation. Wireless Networks, 1–6
19. Mahajan R, Mansotra V (2021) Predicting geolocation of tweets: using combination of cnn and bilstm. Data Sci Eng 6:402–410
20. Simanjuntak LF, Mahendra R, Yulianti E (2022) We know you are living in Bali: location prediction of Twitter users using bert language model. Big Data Cognit Comput 6(3):77
21. Mostafa A, Gad W, Abdelkader T, Badr N (2022) Pre-hlsa: predicting home location for Twitter users based on sentimental analysis. Ain Shams Eng J 13(1):101501
22. You J, Lee J, Kwon H-Y (2021) A complete and fast scraping method for collecting tweets. In: 2021 IEEE international conference on big data and smart computing (BigComp), pp 24–27. https://doi.org/10.1109/BigComp51126.2021.00014
23. You J, Lee K, Kwon H-Y (2024) Deepscraper: a complete and efficient tweet scraping method using authenticated multiprocessing. Data Knowl Eng 149:102260

24. Robertson S (2004) Understanding inverse document frequency: on theoretical arguments for idf. J Doc 60(5):503–520
25. HEAVY.AI | Tweetmap. https://www.heavy.ai/demos/tweetmap
26. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, vol 26
27. Pennington J, Socher R, Manning CD (2014) Glove: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543
28. Bojanowski P, Grave E, Joulin A, Mikolov T (2017) Enriching word vectors with subword information. Trans Assoc Comput Linguist 5:135–146
29. Le Q, Mikolov T (2014) Distributed representations of sentences and documents. In: International conference on machine learning, pp 1188–1196. PMLR
30. Google Code Archive - Word2vec. https://code.google.com/archive/p/word2vec/
31. GloVe: Global Vectors for Word Representation. https://nlp.stanford.edu/projects/glove/
32. Mikolov T, Grave E, Bojanowski P, Puhrsch C, Joulin A (2018) Advances in pre-training distributed word representations. In: Proceedings of the international conference on language resources and evaluation (LREC 2018)
33. Lau JH, Baldwin T (2016) An empirical evaluation of doc2vec with practical insights into document embedding generation. arXiv preprint. arXiv:1607.05368

Publisher's Note