Microblog User Geolocation by Extracting Local Words Based on Word Clustering and Wrapper Feature Selection

Hechan Tian[1, 2], Fenlin Liu[1,2*], Xiangyang Luo[1, 2], Fan Zhang[1, 2], and Yaqiong Qiao[1, 2]
[1] State Key Laboratory of Mathematical Engineering and Advanced Computing
Zhengzhou, 450001 - China
[e-mail: hechantian@aliyun.com]
[2] PLA Strategic Support Force Information Engineering University
Zhengzhou, 450001 - China
[e-mail: liufenlin@vip.sina.com]
*Corresponding author: Fenlin Liu

Received March 23, 2020; revised July 10, 2020; accepted July 25, 2020; published October 31, 2020

Abstract

Existing methods always rely on statistical features to extract local words for microblog user geolocation. There are many non-local words in extracted words, which makes geolocation accuracy lower. Considering the statistical and semantic features of local words, this paper proposes a microblog user geolocation method by extracting local words based on word clustering and wrapper feature selection. First, ordinary words without positional indications are initially filtered based on statistical features. Second, a word clustering algorithm based on word vectors is proposed. The remaining semantically similar words are clustered together based on the distance of word vectors with semantic meanings. Next, a wrapper feature selection algorithm based on sequential backward subset search is proposed. The cluster subset with the best geolocation effect is selected. Words in selected cluster subset are extracted as local words. Finally, the Naive Bayes classifier is trained based on local words to geolocate the microblog user. The proposed method is validated based on two different types of microblog data - Twitter and Weibo. The results show that the proposed method outperforms existing two typical methods based on statistical features in terms of accuracy, precision, recall, and F1-score.

Keywords: Location Prediction, Word Clustering, Feature Selection

1. Introduction

Microblog, such as Twitter and Weibo, is developing very fast and has hundreds of millions of users. Microblog users can interact with friends, share real-time updates, and send tweets with geo-tags. Users, online friendships, and generated tweets jointly form a virtual world. As a linkage between the virtual world and the real society, the location of microblog users could be utilized for many location-based applications, such as targeted advertising, regional communities discovering, news popularity predicting, and public opinion monitoring [1-4]. Due to the restriction of privacy protection [5], the locations of microblog users can only be obtained from the public profiles and geo-tags. However, related statistics show that about 21% of Twitter users declare their locations in profiles [6], and only 0.42% of all tweets contain a geo-tag [7]. Therefore, microblog user geolocation is worth studying. Existing geolocation approaches can be divided into two categories: text-based and network-based approaches.

Network-based methods mainly treat interactive users as friends and infer users' location based on the location of users' friends, the relationship between friends and regions, and the closeness between users' friendships. The first kind of approaches [8, 9] based on the location of users' friends add up the corresponding location total number of users' direct and 2-hop friends, and infer the location with the largest number as the inferred location. Different from the former two approaches, Backstrom et al. [10] observe that the probability of two users becoming friends is inversely proportional to the distance between them, and infer the most likely location using these probabilities. McGee et al. [11] further reveal the relationship between the distance of users and related factors (the information of users' followers, the number of interactions, etc.) and use these factors to measure the relationship between users for location inference. Rout et al. [12] also observe that users interact more frequently with people who are closer to them, and infer user location based on the strength of users' friendships. These approaches [10-12] infer users' location based on the relationship between friends and regions. Different from them, Kong et al. [13] firstly compute the cosine similarity of two users' friends as their social tightness coefficient, and then weight which of user's friends are likely to be most predictive of their location based on the social tightness coefficient.

Additionally, some approaches [14, 15] extend the label propagation algorithm to geolocate user based on the mention network. Location labels are passed from the labeled users to the unlabeled users based on adjacency relationship between users. Some following approaches [16, 17] consider the adverse effect of the "celebrity" on location inference, and remove related celebrity users. Then users are geolocated based on refined mention network through label propagation. The limitation of existing network-based approaches is that it is impossible to geolocate isolated users who do not have any interaction with other users.

The main assumption of text-based approaches is that language is geographically biased. Thus, different regionally characteristic words, such as dialects and places, will appear in the texts generated by users in different regions. Eisenstein et al. [18] analyze the relationship between latent topics and geographic regions, and propose a geolocation method for the user-generated raw tweet. Unlike Eisenstein et al. [18], Cheng et al. [19] select words related to cities from texts, and infer the probability of a given user from one city based on selected words. This method can infer the city-level location of Twitter users. Based on the idea of selecting local words, Hecht et al. [20] propose a user geolocation method based on word

frequency statistics and Bayes model, which can infer the state-level location of American Twitter users.

To select local words better for Microblog user geolocaiton, Han et al. [21] consider that the distribution of local words should be more biased than ordinary words, and extract local words based on information gain rate (IGR) or maximum entropy to train a Naive Bayes classifier for location inference. On this basis, Han et al. [22] compared several methods of local words extraction. The experimental results show that local words extraction based on IGR are more helpful for location inference. Chi et al. [23] extract local words based on IGR and various textual features (country/city names, #hashtags, and @mentions) based on frequency statistics to predict the locations of Twitter users and tweets. The idea of above methods is very good at geolocating users using extracted local words. It is possible to geolocate user well, solely based on text. And text-based methods can compensate for the inadequacy of network-based methods that can not geolocate isolated users.

Existing text-based methods only rely on statistical features to extract local words. There are still some non-local words in the extracted words, which make geolocation accuracy lower. To this end, this paper proposes a microblog user geolocation method based on word clustering and wrapper feature selection to extract local words. The proposed method combines word embedding method with the existing method based on statistical features. The proposed method could extract local words more effectively by considering statistical and semantic features of words. On one hand, ordinary words without positional indications are initially filtered based on IGR. On the other hand, local words are extracted based on word clustering and wrapper feature selection. Considering the statistical and semantic features of local words, the method extracts local words more accurately. The Naive Bayes classifier is trained based on extracted local words to geolocate users. Experiments on two different types of microblog data show that the proposed method outperforms existing two typical methods based on statistical features in terms of accuracy, precision, recall, and F1-score.

The main contributions of this paper are as follows. Firstly, a word clustering algorithm based on word vectors is proposed. The clustering algorithm combines the existing word embedding method to convert words into word vectors with semantic meanings. The algorithm can gather semantically similar words into the same cluster, which is helpful for efficient extraction of local words. Secondly, a wrapper feature selection algorithm based on sequential backward subset search is presented. The algorithm can select the best cluster subset from all word clusters based on geolocation effect. Words in selected cluster subset are extracted as local words to train Naive Bayes classifier, which can improve geolocation accuracy.

The rest of this paper is organized as follows. Section 2 describes the proposed microblog user geolocation method based on word clustering and wrapper feature selection to extract local words. Section 3 analyses the reason why the proposed method can improve geolocation accuracy in principle. Section 4 conducts experiments based on two different types of microblog data, and analyzes the results of the proposed method and two existing typical methods. Finally, Section 5 concludes this paper.

2. Proposed Method

To select local words better, the proposed method considers statistical and semantic features of words. Combining word embedding method with the existing method based on statistical features, this paper proposes a text-based microblog user geolocation method. This method extracts local words based on word clustering and wrapper feature selection, and trains Naive

Bayes classifier to geolocate microblog user. The overall framework of the proposed method is shown in **Fig. 1**. To describe the proposed method more clearly, we define the following terms, as shown in **Table 1**.

Table 1. Terms used in the proposed method

Terms	Definition
$\mathbf{L}$	the set of all users' locations
$\mathbf{U_{tr}}$	the training set of users whose locations are known
$\mathbf{U_{te}}$	the test set of users whose locations are unknown
$\mathbf{U_{tr,j}}$	the set of users in $\mathbf{U_{tr}}$ locating at l_j
$\mathbf{W}$	the set of raw words after word segmention for texts generated by users in $\mathbf{U_{tr}}$
$\mathbf{W}'$	the set of remaining words after removing low-frequency words and stop-words
$\mathbf{W}''$	the set of remaining words after filtering words based on IGR
$\mathbf{W}^*$	the set of the extracted local words
$\mathbf{V}$	The set of word vectors of words in $\mathbf{W}$
$\mathbf{D}$	The set of word vectors of words in $\mathbf{W}''$

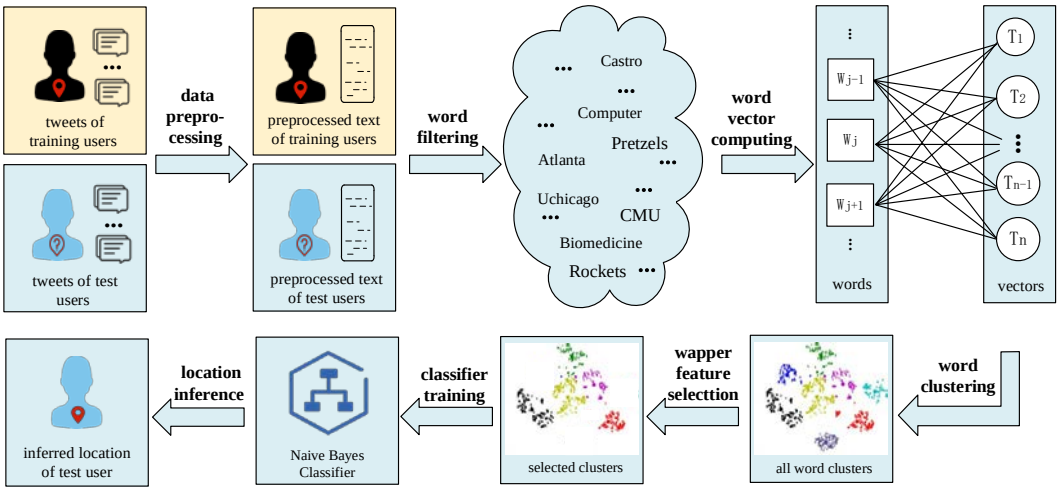


Fig. 1. The overall framework of the proposed method

The main phases are as follows:

(1) data preprocessing. This phase mainly includes four steps: merging tweets, word segmentation, removing stop-words , and removing words whose frequency are less than N_1.

(2) word filtering. Following Han et al. [21], IGR of remianing words after data preprocessing are computed. And words whose IGR are smaller than N_2 are filtered.

(3) word vector computing. All texts in trainging set are used as the corpus of word vectors computing. The existing word embedding method is used to convert word into word vectors based on words' context.

(4) word clustering. The proposed word clustering algorithm based on word vectors is used to divide the remaining words after word filtering into k clusters.

(5) wrapper feature selection. The proposed wrapper feature selection algorithm based on sequential backward subset search is used to select the best cluster subset based on geolocation effect. The words in the selected cluster subset are extracted as local words.

(6) classifier training. The training process of Naive Bayes classifier is the calculation process of probabilities. The training set is used to calculate the prior probability of each location and the conditional probability of each local word appearing in each location.

(7) location inference. Naive Bayes classifier is used to geolocate user. Specifically, based on local words in each test user text, the probability of each test user locating at each location is calculated. The location with the highest probability is inferred as the location of the test user.

In the above phases, data preprocessing, word clustering, wrapper feature selection, classifier training and location inference are the most critical phases. The following four parts are discussed in detail.

2.1 Data Preprocessing

The specific process of data preprocessing is shown in **Fig. 2**. First, all tweets generated by each user are merged into one text, namely, one user corresponds to one text. Then, the text content is segmented sentence by sentence. For texts in different language types, the operation of word segmentation is always different. For English texts, firstly use the existing English Entity Name Recognition (NER) tool to identify the entity names, and the identified entity name composed of several words will be merged into one word. And then the remaining English words are separated by spaces. For Chinese texts, use the existing Chinese word segmentation tool to segment words. **Table 2** show the examples of word segmentation for Twitter user text. Next, stop-words are removed based on corresponding stop-word vocabulary. For text of different language types, it is necessary to construct corresponding stop-word vocabulary. Finally, words whose frequency are less than N_1 are also removed.

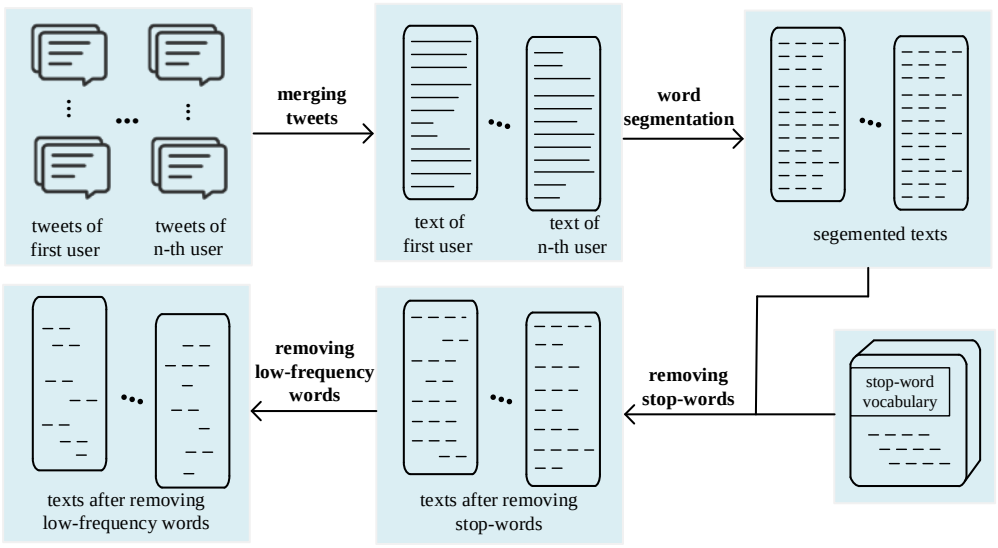


Fig. 2. Diagram of data preprocessing process

Table 2. Word Segmentation Example for Twitter User Text

Text Status	Text Content
Before word segmentation	Amazing how many nutrition podcasts you can listen to on the way from New York to South Carolina
After word segmentation	Amazing/how/many/nutrition/podcasts/you/can/listen/to/on/the/way/ from/NewYork/ to/SouthCarolina

2.2 Word Clustering

In order to divide differently semantical words into different clusters, a word clustering algorithm based on word vectors is proposed, as shown in Algorithm 1.

Algorithm 1: Word clustering based on word vectors

Input: $\mathbf{W}^{"}$, $\mathbf{V}$, k

(1) initialize the set of word vectors to be clustered: $D = \varnothing$
(2) **for** $i = 1, 2, ..., |W^{"}|$ **do**
(3) find word vector v_i of word w_i from $\mathbf{V}$: $D = D \cup \{w_i\}$
(4) **end for**
(5) initialize k center points of clusters: $\{m_1, m_2, ..., m_k\}$
(6) **repeat**
(7) let $C_i = \varnothing, A_i = \varnothing (1 \le i \le k)$
(8) **for** $j = 1, 2, ..., |W^{"}|$ **do**
(9) **for** $i = 1, 2, ..., k$ **do**
(10) calculate the distance d_{ji} between v_j and m_i
(11) **end for**
(12) find the cluster center point m_{τ_j} nearest to v_j
(13) delegate v_j into the corresponding cluster set: $C_{\tau_j} = C_{\tau_j} \cup \{v_j\}$
(14) delegate w_j into the corresponding word cluster set: $A_{\tau_j} = A_{\tau_j} \cup \{w_j\}$
(15) **end for**
(16) **for** $i = 1, 2, ..., k$ **do**
(17) calculate the mean value of all word vectors: $m_i^{'} = \frac{1}{|C_i|} \sum_{v \in C_i} v$
(18) **end for**
(19) **if** $m_i^{'} = m_i$ **then**
(20) $m_i = m_i^{'}$
(21) **else**
(22) keep the current center point of the i th cluster unchanged
(23) **end if**
(24) **until** the center points of all clusters are not changing

Output: word clustering result $\{A_1, A_2, ..., A_k\}$ of words in $\mathbf{W}^{"}$

In Algorithm 1, there are three key points when clustering words. Firstly, corresponding word vectors of words in $\mathbf{W}^{''}$ should be found from $\mathbf{V}$. Secondly, k center points of clusters are initialized based on the principle of the farthest distance. To be specific, a word vector in $\mathbf{D}$ is randomly selected as the center point of the first initial class cluster. The center point of the second initial class cluster is the word vector, which is farthest from the center point of the first initial class cluster. And the word vector which is the farthest from the center points of the first two initial class clusters, is selected as the center point of the third initial class cluster. The rest center points of class clusters are selected by analogy. Finally, words are clustered based on the distance of word vectors. The distance between each word vector and each center point of each initial class cluster is calculated. Each word vector is divided into the cluster where the nearest center point of class cluster is located. At the same time, the word vector corresponding to word is divided into the corresponding word cluster. The mean value of all word vectors in each cluster is calculated to be the new center point of each class cluster. Repeating the above process of dividing word clusters based on the distance of word vectors until the center points of all clusters are not changing.

2.3 Wrapper Feature Selection

In order to select the best cluster subset with the best geolocation effect from k word clusters for location inference, a wrapper feature selection algorithm based on sequential backward subset search is proposed, as shown in Algorithm 2.

Algorithm 2: Wrapper feature selection based on sequential backward subset search

Input: $\{A_1, A_2, ..., A_k\}$, training set

(1) initialize the cluster subset: $A^* = \{A_1, A_2, ..., A_k\}$

(2) initialize the number of clusters in A^*: $d = k$

(3) words in A^* are extracted as local words, the average geolocation error rate e^* of the classifier

(4) trained using 5-fold cross-validation on training set is estimated

(5) **for** $i = 1, 2, ..., k$ **do**

(6) **for** $j = 1, 2, ..., d$ **do**

(7) delete the jth cluster from the current subset A^* to form new subset A^j

(8) words in A^j are extracted as local words**,** the average geolocation error rate e^j of the

(9) classifier trained using 5-fold cross-validation on training set is estimated

(10) **end for**

(11) find the subset $A^{'}$ with the smallest geolocation error rate from all new subsets

(12) update $e^{'}$ to the average geolocation error rate of $A^{'}$

(13) **if** $e^* \le e^{'}$ **then**

(14) stop subset search, and jump out of the loop

(15) **else**

(16) update A^* to the best cluster subset $A^{'}$ found in this round, update e^* to $e^{'}$, update d

(17) **end if**

(18) **end for**

Output: the selected subset A^*

The algorithm considers each word cluster as a whole feature, and the process of word cluster subset selection from all word clusters is similar to that of feature subset selection. Input of Algorithm 2 is k word clusters and training set. Output is the selected cluster subset with best geolocation effect. All words in the selected cluster subset are extracted as local words. This algorithm adopts the heuristic strategy of sequential backward subset search for searching the cluster subset from the complete set which included k word clusters. In each round of cluster subset search, removing each cluster from the current subset forms a new subset. The classifier is trained based on words in each new subset on the training set. Using 5-fold cross-validation, the average geolocation error rate of each new trained classifier is estimated. A new subset with the smallest geolocation error rate is selected and compared with the geolocation error rate of the current cluster subset. If the geolocation error rate of the selected new subset is lower than that of the current cluster subset, the current subset is updated to the selected new subset. Repeat this process until a arbitrary cluster is deleted, the geolocation error rate of the selected new subset is higher than that of the current cluster subset. Then, stop searching.

2.4 Classifier Training and Location Inference

The process of training the Naive Bayes classifier is the calculation process of probabilities.

Firstly, the prior probability of l_j is calculated using Equation 1.

$$P\left(l_j\right)=\frac{|U_{tr,j}|}{|U_{tr}|} \tag{1}$$

Secondly, the conditional probability that w_i appears at l_j is calculated using Equation 2.

$$P\left(w_i \mid l_j\right)=\frac{f_{w_i,l_j}}{\sum_{j=1}^{|L|} f_{w_i,l_j}} \tag{2}$$

It's worth noting that some local words may not appear at some locations, which will make the conditional probabilities that the local words appear at the locations are computed as zero using Equation 2, resulting in common Zero-Probabilities of language model. Following Chi et al. [23], Laplace smoothing can be used to correct Equation 2, as Equation 3:

$$P\left(w_i \mid l_j\right)=\frac{f_{w_i,l_j}+1}{\sum_{j=1}^{|L|} f_{w_i,l_j}+|L|} \tag{3}$$

Location inference is to calculate the probability that the user locating at each location. And the location with the highest probability is inferred as user geolocation result. Firstly, the probability that the test user u is from l_j is calculated using Equation 4:

$$P\left(l_j \mid u\right)=P\left(l_j\right)\prod_{f_{w_i,l_j}>0} P\left(w_i \mid l_j\right) \tag{4}$$

Secondly, as Equation 5, take the location with the highest probability as the inference result of the user u.

$$l(u)=\arg\max_{l_j \in L} P\left(l_j \mid u\right) \tag{5}$$

3. Principle Analysis

This section analyses the reason why the proposed method can improve geolocation accuracy in principle.

About data preprocessing. Word segmentation is one of the standard preprocessing operations for text mining. Existing methods often segment English words by spaces, which will split some named entities consisting of words into multiple separate words, such as 'New York', 'Times Square', and 'Statue Of Liberty', etc. Different from the existing methods, the proposed method first identifies named entities and combines the named entities consisting of multiple words into one word. It means that 'New York' is treated as 'NewYork'. This word segmentation way is more conducive to extract local words. For example, 396852 original words are obtained from 358,412 tweets of Twitter users after word segmentation. Compared with other words, stop-words are widely used or have no practical meaning. Stop-words cannot indicate location and do not make contributions to geolocate users. Removing stop-words can not only reduce non-local words but also improve processing efficiency without affecting geolocation accuracy. After filtering stop-words in original words based on the English stop-word vocabulary, there are 395961 words left.

The low-frequency words may seldom appear at a certain location and contribute little to location inference. Moreover, in order to get a better word vectors for analyzing the semantic similarity of words, low-frequency words are often not considered. Removing low-frequency words is both reasonable and useful to reduce the computational cost of filtering words. Fig. 3(a) shows the statistical distribution about frequency of 395,961 words. It can be seen that frequency of 41.99% words is 1 and frequency of 19.74% words is 2. If only words whose frequency are less than 2 are removed, the computational cost of all remaining steps will increase sharply. Considering the computational cost of proposed method, removing words whose frequency are less than 3, there are still 151,534 words left.

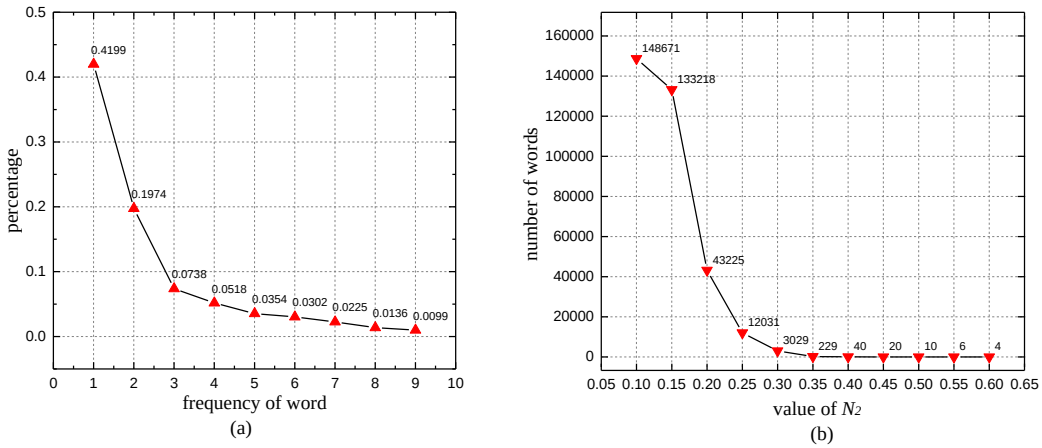


Fig. 3. Statistical information of words. (a) The statistical distribution about frequency of 395,961 words. (b) The changing trend of the number of words as the threshold N_2.

About word filtering. The IGR of a given word describes the ratio of its information gain to its intrinsic entropy. Words with larger IGR may be concentrated in a few locations and have relatively small intrinsic entropy, such as dialects, landmarks, and local proper nouns. At the same time, filtering out words with smaller IGR can reduce the computational cost of

LIWs selection. The IGR of 151,534 words are computed and sorted in descending order. As the threshold N_2 increases, the number of remaining words decreases.

Fig. 3(b) shows the changing trend of the number of words as the threshold N_2. When N_2 is changed from 0.15 to 0.2, the number of words declines most dramatically. This is a particularly clear demarcation point. If N_2 is set to 0.15, just a small number of words will be filtered, and a large number of words with small IGR will still be retained, causing a sharp increase in the computational cost of all following steps. If N_2 is set to 0.25, a large number of words with high IGR will be filtered out, leading to excessive filtering of words. Therefore, the final threshold N_2 is set to 0.2, and 43225 words are left.

About word vectors computing. The word vectors computed based on words' context can be used to compare the semantic similarities of words and effectively improve the effect of clustering words. Because of the much lower computational complexity, word2vec proposed by Mikolov et al. [24] is possible to compute very accurate high dimensional word vectors from a much larger data set. Therefore, it is suitable for us to compute word vectors using word2vec. Texts generated by the above Twitter users are used to compute word vectors of 151534 words. Based on the distance of word vectors, the semantically similarity of words are calculated. The examples of five words which are the most similar to 'New York', 'howdy', 'phone' and 'headache' are listed in **Table 3**. We can see that the most similar words to 'New York' are also city names, the most similar words to 'howdy' are also dialects, the most similar words to 'phone' are also daily life-related nouns, and the most similar words to 'headache' are also health-related words. This shows that word vectors computed using the word embedding method can be well used to analyze the semantic similarity of words.

Table 3. Five words which are most similar to 'New York', 'howdy', 'phone' and 'headache'

NewYork		howdy		phone		headache	
word	similarity	word	similarity	word	similarity	word	similarity
Chicago	0.60078	phillies	0.65561	computer	0.63314	fever	0.64315
Atlanta	0.56328	trash	0.55560	ipad	0.60835	cold	0.63130
Boston	0.54379	Redneck	0.47348	watch	0.58880	cough	0.61807
Detroit	0.53773	yankee	0.43600	iphone	0.57744	dizziness	0.60023
Houston	0.53757	tawk	0.42688	sumsang	0.57049	sneeze	0.56088

About word clustering. After word filtering, most remaining words can indicate specific locations, but there are still non-local noise words. To further filter non-local words, the proposed method extract local words based on word clustering and wrapper feature selection. The remaining words after word filtering are clustered using the proposed clustering algorithm based on word vectors, which can bring together semantically similar words to help local words selection. For example, 45 words are randomly selected from the remaining 43225 words. These 45 words are clustered using Algorithm 1 and reduce the dimensionality of word vectors using the descending dimension algorithm. The two-dimensional visualization of the clustering result of 45 randomly selected words is shown in **Fig. 4**. The words are grouped into 7 clusters, which are composed of subject terms, gourmet snacks, famous attractions, university names, local teams, representative landmarks, and city names, respectively. This shows that clustering words based on word vectors can effectively bring together semantically similar words.

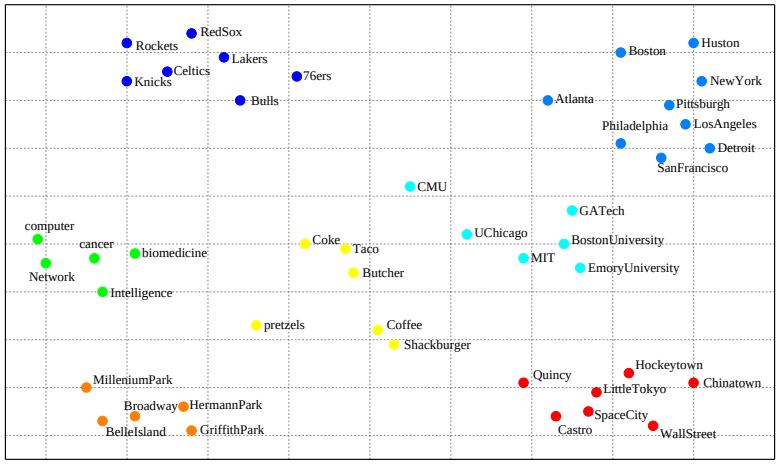


Fig. 4. The two-dimensional visualization of clustering result

About wrapper feature selection. The wrapper feature selection algorithm is one of feature selection methods in machine learning. The wrapper approach directly takes the performance of the classifier as the evaluation criteria of the selected clusters and can select the best subset from all clusters that is most beneficial to its performance. The sequential backward subset search strategy is one of feature subset selection methods. This subset search strategy searches for feature subsets from the complete set. Each time a feature is removed from the current feature subset, and the evaluation function value should be optimized by removing features. The proposed method uses the same idea to select the corresponding location indicative word clusters from all clusters formed by clustering. Using the wrapper feature selection algorithm based on sequential backward subset search, the corresponding location indicative word clusters can be selected quickly and efficiently.

Compared with the existing methods based on statistical features, the proposed method is further based on word clustering and wrapper feature selection, which can extract local words more accurately, thus improving geolocation accuracy.

4. Experiments

In this section, we describe the datasets used in the experiments, highlight our experimental setup and discuss the results of proposed method and comparsion methods.

4.1 Experimental Data

For Weibo, we collected 274,459 users' data. The statistical results of the location information in profiles show that about 48.36% of users declared their city-level locations, about 22.16% of users declared their province-level locations, and the rest users did not explicitly declare their locations. In order to clean and remove the abnormal data, the users who claimed unclear locations or whose tweets were less than 5 were filtered out. In addition, delete cities with fewer than 100 users and take the users' claimed locations as the ground-truth. Finally, the city-level Weibo dataset, called Weibo(City) for short, consists of 102,735 users in 179 cities. According to the administrative division, determine the province-level locations of the users in Weibo(City). The province-level Weibo dataset, called Weibo(Prov) for short, consists of all users in Weibo(City) and other users who accurately claimed their province-level locations.

For Twitter, we collected 594718 users' data. The statistical results of the location information in profiles show that about 41.5% of users declared their city-level locations and about 17.29% of users declared their state-level locations. We process Twitter data in the same way that Weibo data is processed. Finally, the city-level Twitter dataset, called Twitter(City) for short, consists of 235120 users in 378 cities. According to the administrative division, determine the state-level locations of the users in Twitter(City). The state-level Twitter dataset, called Twitter(State) for short, consists of all users in Twitter(City) and other users who accurately claimed their state-level locations.

Table 4 lists the basic statistics of four datasets. In our experiments, each dataset is divided into two parts: 20% of the users are randomly selected as test data from all users in each location, and the remaining 80% are used as training data. We evaluate the performance of location inference using four measures for multi-class classification: accuracy (percentage of the users whose locations are inferred correctly), and precision, recall, and F1-score (the average precision, recall and F1-score of each class).

Table 4. The basic statistics of four datasets

Dataset	Weibo(City)	Weibo(Prov)	Twitter(City)	Twitter(State)
No. of users	102735	154478	235120	337946
No. of tweets	3085972	3862117	5728037	8279695
No. of locations	179	34	378	50

4.2 Experimental Setup

For Weibo data, use the existing Chinese word segmentation tool [25] for word segmentation. For Twitter data, the named entities are identified using the Stanford Named Entity Recognizer published by the Stanford University Natural Languages Research Group [26]. According to the relevant experience of text mining, N_1 is set to 3. The value of N_2 needs to be set according to statistical result of words' IGR. Specifically, set the thresholds at equal intervals and count the corresponding number of words, respectively. Then the threshold corresponding to the maximum variation of the number of words is selected as the final N_2. We make Chinese and English stop-word vocabularies, containing 1598 words and 891 words, respectively. The parameter settings of computing word vectors using word2vec are shown in **Table 5**.

Table 5. The parameter settings of computing word vectors

Parameter	Description of parameter	Value
size	the dimension of the output word vector	200
window	the maximum distance between the current word and the target word in the sentence	5
min_count	the word whose frequency is less than min_count is not calculated	3
sg	sg=1 indicates training using the skip-gram model	1

Set the value of k. Considering the computational cost and estimation of clustering effect, we analyze the changing trend of within-cluster sum of squared errors (SSE) as k using the 'Elbow Rule' and find the elbow point to set the appropriate value of k. In short, the changing trend graph of SSE as k is like an elbow, and the value of k corresponding to the elbow point is probably the true number of clusters.

4.3 Experimental Results

Firstly, the proposed method is used to extract local words from the training set of four datasets. **Table 6** lists the changes in the number of remaining words after different operations.

Table 6. Statistical results of the number of remaining words after different operations

Dataset	Weibo(City)	Weibo(Prov)	Twitter(City)	Twitter(State)
After word segementation	384328	400193	375917	391794
After filtering low-frequency words	162,496	169198	158934	165647
After filtering words based on IGR	46265	48267	45330	47250
After wrapper feature selection	25864	27245	25093	28466

As shown in **Table 6**, after word filtering based on IGR, there are 46 265, 48 267, 45 330 and 47 250 words left for the four datasets, respectively. For the four datasets, different k values are set at intervals of 4, and the remaining words are clustered based on word vectors. According to the 'Elbow Rule', the corresponding values of k are setted as 28, 28, 32 and 32, respectively. From the clusters formed by Algorithm 1, local words are extracted by Algorithm 2. As shown in **Table 6**, 25864, 27245, 25093 and 28466 local words are extracted for four datasets, respectively. Finally, location inference is performed using the extracted local words.

The geolocation accuracy of proposed method, MNB-PART [23] and NB+IGR [21] on four datasets are shown in Table 7. It can be seen that the geolocation accuracy of our proposed method on the four datasets are higher than those of two comparison methods based on statistical features. And the geolocation accuracy of MNB-PART is the lowest. MNB-PART uses the textual features of high frequency as the features for location inference, but not all textual features of high frequency can indicate locations, which affects the geolocation accuracy. Besides, the geolocation accuracy of proposed method are still 4.66%, 3.58%, 5.72%, and 6.08% higher than those of NB+IGR.

NB+IGR [21] only selects local words based on statistical features (IGR), and outperforms MNB-PART [23] which selects local words based on word frequency statistics. The former two methods do not consider the semantic features of local words, there are still many non-local words in the set of selected words. Different from existing methods, the proposed method considers statistical and semantic features of local words. After filtering words based on words' IGR, the proposed method further selects the corresponding location indicative word clusters using the wrapper feature selection algorithm based on sequential backward subset search, which can further enhance the filtering effect of noise words and improve the geolocation accuracy.

Table 7. The geolocation accuracy of three methods on four datasets

Dataset	Weibo(City)	Weibo(Prov)	Twitter(City)	Twitter(State)
MNB-PART[23]	0.4523	0.6315	0.4509	0.6247
NB+IGR[21]	0.4752	0.6596	0.4637	0.6415
Proposed method	**0.5218**	**0.6954**	**0.5209**	**0.7023**

On four datasets, the performance of three geolocation methods is evaluated using the three measures - precision, recall and F1-score, respectively. The comparison results of three geolocation methods are shown in **Fig. 5**. It can be seen clearly that the proposed method also outperforms two comparison methods in terms of precision, recall and F1-score, respectively.

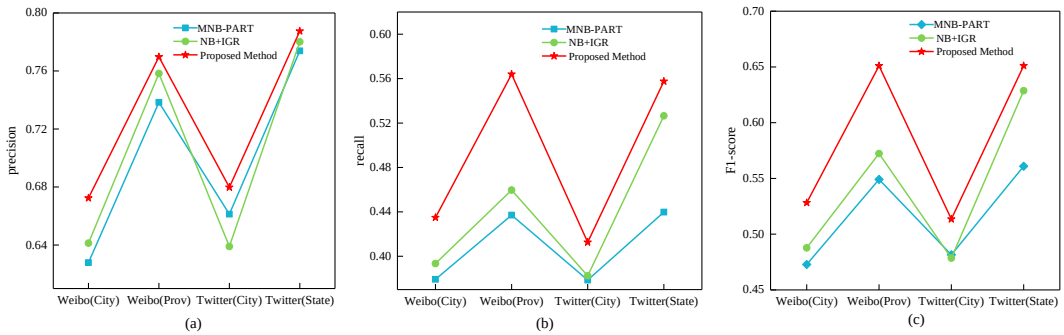


Fig. 5. The comparison results of three geolocation methods. (a) The precision comparison of methods. (b) The recall comparison of methods. (c) The F1-score comparison of methods

Finally, we analyze the impact of threshold setting in the proposed method on the geolocation accuracy. **Fig. 6**(a) shows the geolocation accuracy of the proposed method on Twitter(State) and Weibo(Prov) datasets when varying the threshold N_1 from 3 to 10. The results indicate that the performance is sensitive to the threshold, while a larger N_1 usually leads to slightly worse performance. It is logical for us to set the threshold N_1 to 3 in our experiments. When setting the threshold N_1 to 3, the threshold N_2 are varied from 0.11 to 0.24. **Fig. 6**(b) shows the geolocation accuracy of the proposed method on Twitter(State) and Weibo(Prov) datasets under different thresholds. The results indicate that the geolocation accuracy is not sensitive to the threshold over a range. However, a too large or too small N_2 will hurt the geolocation accuracy.

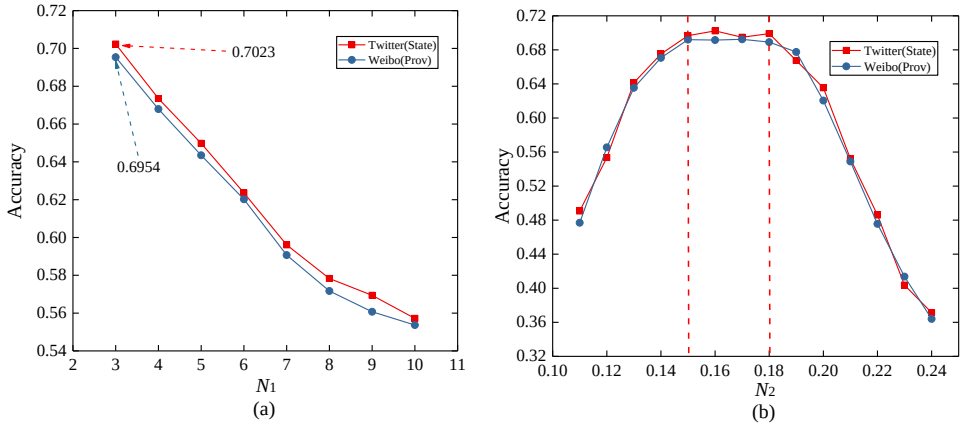


Fig. 6. The impact of threshold setting on the geolocation accuracy. (a) The impact of different N_1 on the geolocation accuracy. (b) The impact of different N_2 on the geolocation accuracy.

5. Conclusion

This paper studied the problem of microblog user geolocation, and proposed a text-based user geolocation method using extracted local words based on word clustering and wrapper feature selection. This method combines the word embedding method with the existing text-based geolocation method. A lot of ordinary words are filtered effectively based on IGR, which reduces the computational cost of word clustering. Taking full advantage of the ability of word embedding method in describing the semantic similarity of words, a word clustering algorithm based on word vectors is proposed. Combining with the existing feature selection methods, a wrapper feature selection algorithm based on sequential backward subset search is presented, which can extract local words with good geolocation effect and improve the geolocation accuracy. The effectiveness of the proposed method are also analyzed in principle.

Predicting the location of microblog users relying solely on text may have limitations. Our future plan is to propose a hybrid method to geolocate microblog user by combining the location indicative informations of text and friendship-based network.

References

[1] O. Ajao, J. Hong and W. Liu, "A survey of location inference techniques on Twitter," *Journal of Information Science*, vol. 41, no. 6, pp. 855-864, December, 2015. Article (CrossRef Link)

[2] G. Jang and S.H. Myaeng, "Predicting event mentions based on a semantic analysis of microblogs for inter-region relationships," *Journal of Information Science*, vol. 44, no. 6, pp. 818-829, March, 2018. Article (CrossRef Link)

[3] K. Akyol and B. Şen, "Modeling and Predicting of News Popularity in Social Media Sources," *Computers, Materials & Continua*, vol. 61, no. 1, pp.69-80, 2019. Article (CrossRef Link)

[4] C. You, D. Zhu, Y. Sun, A. Ye, G. Wu, N. Cao, J. Qiu and H. M. Zhou**,** "SNES: Social-Network-Oriented Public Opinion Monitoring Platform Based on ElasticSearch," *Computers, Materials & Continua*, vol. 61, no. 3, pp.1271-1283, 2019. Article (CrossRef Link)

[5] P. Wang, Z. Wang, T. Chen and Q. Ma, "Personalized Privacy Protecting Model in Mobile Social Network," *Computers, Materials & Continua*, vol. 59, no. 2, pp.533-546, 2019. Article (CrossRef Link)

[6] Z.Y. Cheng, J. Caverlee and K. Lee, "A content-driven framework for geolocating microblog users," *ACM Transactions on Intelligent Systems and Technology*, vol. 4, no. 1, pp. 1-27, February, 2013. Article (CrossRef Link)

[7] K.M. Ryoo and S. Moon, "Inferring Twitter user locations with 10 km accuracy," in *Proc. of the 23rd International Conference on World Wide Web (WWW'14)*, pp. 643-648, April 7-11, 2014. Article (CrossRef Link)

[8] C.A. Davis, G.L. Pappa, Diogo Rennó Rocha De Oliveira, and F.D.L. Arcanjo, "Inferring the location of twitter messages based on user relationships," *Transactions in Gis*, vol. 15, no. 6, pp. 735–751, December, 2011. Article (CrossRef Link)

[9] S. Abrol and L. Khan, "Tweethood: Agglomerative clustering on fuzzy k-closest friends with variable depth for location mining," in *Proc. of the IEEE 2nd International Conference on Social Computing (SocialCom'10)*, pp. 153–160, August 20-22, 2010. Article (CrossRef Link)

[10] L. Backstrom, E. Sun and C. Marlow, "Find me if you can: Improving geographical prediction with social and spatial proximity," in *Proc. of the 19th International Conference on World Wide Web (WWW'10)*, pp. 61-70, April 26-30, 2010. Article (CrossRef Link)

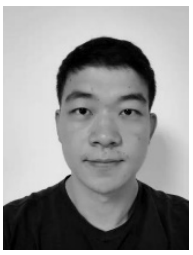
[11] J. McGee, J. Caverlee and Z.Y. Cheng, "Location prediction in social media based on tie strength," in *Proc. of the 22nd ACM International Conference on Conference on Information and Knowledge Management (CIKM'13)*, pp. 459-468, October 27 - November 1, 2013. Article (CrossRef Link)

[12] D. Rout, B. Kalina, PreoţiucPietro Daniel, and C. Trevor, "Where's @wally: A classifcation approach to geolocating users based on their social ties," in *Proc. of the 24th ACM Conference on Hypertext and Social Media (HT'13)*, pp. 11-20, May 1-3, 2013. Article (CrossRef Link)
[13] L. Kong, Z. Liu and Y. Huang, "Spot: Locating social media users based on social network context," *Journal Proceedings of the VLDB Endowment*, vol. 7, no. 13, pp. 1681-1684, January, 2014. Article (CrossRef Link)
[14] D. Jurgens, "That's what friends are for: Inferring location in online social media platforms based on social relationships," in *Proc. of the 7th International AAAI Conference on Weblogs and Social Media (ICWSM'13)*, pp. 273-282, July 8-10, 2013. Article (CrossRef Link)
[15] R. Compton, D. Jurgens and D. Allen, "Geotagging one hundred million twitter accounts with total variation minimization," in *Proc. of the IEEE International Conference on Big Data (Big Data'14)*, pp. 393-401, October 27-30, 2014. Article (CrossRef Link)
[16] A. Rahimi, T. Cohn and T. Baldwin, "Twitter user geolocation using a unified text and network prediction model," in *Proc. of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP'15)*, pp. 630-636, July 26-31, 2015. Article (CrossRef Link)
[17] M. Ebrahimi, E. Shafieibavani, R. Wong, and F. Chen, "Twitter user geolocation by filtering of highly mentioned users," *Journal of the Association for Information Science and Technology*, vol. 69, no. 7, pp. 879-889, February, 2018. Article (CrossRef Link)
[18] J. Eisenstein, B. O'Connor, N.A. Smith, and E.P. Xing, "A latent variable model for geographic lexical variation," in *Proc. of the 7th Conference on Empirical Methods in Natural Language Processing (EMNLP'10)*, pp. 1277-1287, October 9-10, 2010. Article (CrossRef Link)
[19] Z.Y. Cheng, J. Caverlee and K. Lee, "You are where you tweet: a content-based approach to geo-locating twitter users," in *Proc. of the 19th ACM international Conference on Information and Knowledge Management (CIKM'10)*, pp. 759-768, October 26-30, 2010. Article (CrossRef Link)
[20] B. Hecht, L. Hong, B. Suh, and E.H. Chi, "Tweets from Justin Bieber's heart: the dynamics of the location field in user profiles," in *Proc. of the 29th SIGCHI Conference on Human Factors in Computing Systems(CHI'11)*, pp. 237-246, May 7-12, 2011. Article (CrossRef Link)
[21] B. Han, P. Cook and T. Baldwin, "Geolocation prediction in social media data by finding location indicative words," in *Proc. of the 24th International Conference on Computational Linguistics (COLING'12)*, pp. 1045–1062, December 8-15, 2012. Article (CrossRef Link)
[22] B. Han, P. Cook and T. Baldwin, "Text-based twitter user geolocation prediction," *Journal of Artificial Intelligence Research*, vol. 49, no. 1, pp. 451–500, January, 2018. Article (CrossRef Link)
[23] L. Chi, K.H. Lim, N. Alam, and C. Butler, "Geolocation Prediction in Twitter Using Location Indicative Words and Textual Features," in *Proc. of the 2nd Workshop on Noisy User-generated Text(WNUT'16)*, pp. 227-234, December 11, 2016. Article (CrossRef Link)
[24] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," in *Proc. of 1st International Conference on Learning Representations (ICLR'13)*, May 2-4, 2013. Article (CrossRef Link)
[25] W.X. Che, Z.H. Li and T. Liu, "LTP: A Chinese language technology platform," in *Proc. of the 23rd International Conference on Computational Linguistics (COLING'10)*, pp. 13-16, August 23-27, 2010. Article (CrossRef Link)
[26] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling," in *Proc. of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pp. 363-370, June 25-30, 2005. Article (CrossRef Link)

Hechan Tian： She received the B.S. and M.S. degree from the State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, China, in 2017 and in 2020, respectively. Her research interests include machine learning, data mining, and social network analysis.

Fenlin Liu: He received the B.S. degree from the Zhengzhou Science and Technology Institute in 1986, the M.S. degree from the Harbin Institute of Technology in 1992, and the Ph.D. degree from Northeast University in 1998. He is currently a Professor with the Zhengzhou Science and Technology Institute. He has authored or co-authored more than 90 refereed international journal and conference papers. His research interests include network topology and network geolocation. He obtained the support of the National Natural Science Foundation of China and the Found of Innovation Scientists and Technicians Outstanding Talents of Henan Province of China.

Xiangyang Luo: He received his B.S., M.S. and Ph. D. from the State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, China, in 2001, 2004, and 2010, respectively. He has been with the State Key Laboratory of Mathematical Engineering and Advanced Computing since July 2004. From 2011, he is a postdoctoral of Institute of China Electronic System Equipment Engineering Co., Ltd. He is the author or co-author of more than 100 refereed international journal and conference papers. His research interest includes network topology, network security and network geolocation. He obtained the support of the National Natural Science Foundation of China, the National Key R&D Program of China and the Basic and Frontier Technology Research Program of Henan Province.



Fan Zhang: He received the B.S. degree from the Xiangtan University in 2017, the M.S. degree from the State Key Laboratory of Mathematical Engineering and Advanced Computing in 2020. He is currently pursuing the Ph.D. degree with the State Key Laboratory of Mathematical Engineering and Advanced Computing. His research interests include network topology analysis and IP geolocation. He received the support of the National Natural Science Foundation of China and the Basic and Frontier Technology Research Program of Henan Province.

Yaqiong Qiao: She received B.S. and M.S. degrees in control science and engineering from Northwestern Polytechnical University, China, in 2004 and 2007, respectively. She is currently pursuing the Ph.D. degree at the State Key Laboratory of Mathematical Engineering and Advanced Computing, China. Her research interests include machine learning, data mining, and social network analysis.