Hindawi
Scientific Programming
Volume 2021, Article ID 9946729, 13 pages
https://doi.org/10.1155/2021/9946729

Research Article

An Efficient Minimal Text Segmentation Method for URL Domain Names

Yiqian Li,[1] **Tao Du**,[1,2] **Lianjiang Zhu**,[1] **and Shouning Qu**[1,2]

[1]*School of Information Science and Engineering, University of Jinan, Jinan, China*
[2]*Shandong Provincial Key Laboratory of Network Based Intelligent Computing, University of Jinan, Jinan, China*

Correspondence should be addressed to Tao Du; ise_dut@ujn.edu.cn

Received 25 March 2021; Revised 31 May 2021; Accepted 22 June 2021; Published 2 July 2021

Academic Editor: Antonio J. Peña

Text segmentation of the URL domain name is a straightforward and convenient method to analyze users' online behaviors and is crucial to determine their areas of interest. However, the performance of popular word segmentation tools is relatively low due to the unique structure of the website domain name (such as extremely short lengths, irregular names, and no contextual relationship). To address this issue, this paper proposes an efficient minimal text segmentation (EMTS) method for URL domain names to achieve efficient adaptive text mining. We first designed a targeted hierarchical task model to reduce noise interference in minimal texts. We then presented a novel method of integrating conflict game into the two-directional maximum matching algorithm, which can make the words with higher weight and greater probability to be selected, thereby enhancing the accuracy of recognition. Next, Chinese Pinyin and English mapping were embedded in the word segmentation rules. Besides, we incorporated a correction factor that considers the text length into the F_1-score to optimize the performance evaluation of text segmentation. The experimental results show that the EMTS yielded around 20 percentage points improvement with other word segmentation tools in terms of accuracy and topic extraction, providing high-quality data for the subsequent text analysis.

1. Introduction

In recent years, the internet has become one of the most important infrastructures in human society, having an increasingly broad and deep impact on people's economic and social activities [1]. Accessing different URLs can be regarded as the user's behavior trajectory. Website domain names are the most representative of massive online behavior data [2]. These data include the name and type of web pages browsed by users, reflecting the user's preferences in websites and the correlation between the chosen websites [3].

Since the URL domain name text has no context semantic relationship, word segmentation is the first step in the rapid extraction of web page attribute information and provides fast and accurate data support for analyzing the users network behaviors [4, 5]. Many new word segmentation techniques with high accuracy have been proposed due to continuous development. However, these large-scale labeled corpora for training are mainly from news websites, are generally highly targeted and nonrobust, and have low resource utilization. The use of these word segmentation techniques to analyze the domain names of websites often results in low performance [6].

Because of the characteristics of the definition and to distinguish them from standard English text, URL domain names consist mostly of English letters, Arabic numerals, and some special characters, such as ".," "@," and "/." The purpose is to facilitate memorization and connections to server addresses (website, e-mail, FTP, etc.) [4]. Standard English text uses a space character as a separator, but URL domain names have not valid separators to split the text into recognizable words; thus, they include many special symbols and numbers [7, 8]. Therefore, dividing URLs is challenging, and in many cases, the terms in the URL domain names are ambiguous [9]. It is difficult to determine the meaning of the words in the URL by the context, causing ambiguity. Therefore, the use of current popular word segmentation

techniques for analyzing domain names is generally unsatisfactory.

In order to cope with the above challenge and promote the research on the behavioral emotion contained in URL, this paper proposes an efficient minimal text segmentation (EMTS) method for URL domain names. Its primary goal is to perform text parsing on random website domain names and extract keywords with high accuracy because the prerequisite for quickly and accurately analyzing users' online behavior preferences is to extract the emotional effects contained between samples [10]. Therefore, the algorithm based on URL analysis needs to mine the implicit interaction relationship [11] in the sample to achieve more accurate extraction of more meaningful words as the basic unit of the sample [12]. Unlike most previous normalized word segmentation methods, EMTS analyzes on the semantic level and can be flexibly learned from data. In addition, the evaluation criteria need to be optimized. We incorporate the possibility of selecting the text based on the URL into evaluation indicators to improve the correlation with human preferences.

In general, the main contributions of this paper are as follows:

(1) An adaptive text mining method for URL domain names is proposed. Experiments have verified that the method has higher accuracy than current mainstream methods.

(2) A semantic encoder scheme is proposed, and it is demonstrated in experiments that mapping between English and Chinese Pinyin can be achieved.

(3) An improved F_1-score evaluation system based on the structure and text length of websites is designed.

The rest of the paper is organized as follows. In Section 2, we describe the related work and refine the problem. The algorithm with details of each technique is presented in Section 3. Section 4 introduces the improvement to the evaluation standard and presents detailed experimental analysis. We summarize the research in Section 5.

2. Related Work

Word segmentation is required to analyze text data. Text is unstructured data [13] and needs to be converted to structured data before it can be analyzed. The structured data can be transformed into mathematical problems, and word segmentation is the first step in this transformation [14]. The basic unit of English text is a word [15]; therefore, standard word segmentation techniques are divided into three steps:

(1) Split the words according to the spaces

(2) Delete stop words

(3) Perform stemming and lemmatization

But it is slightly different in the focus of word segmentation.

The maximum matching (MM) algorithm [16] is a typical dictionary-based word segmentation method. The concept is to construct a dictionary tree based on the dictionary [17]. Since each node of the tree is composed of the common prefix of the word, the storage space is low, and the search efficiency is high [18]. However, this method is susceptible to the length of the initial words. If the word length is very long, the efficiency is low, and the time complexity of the algorithm is high. If the word length is very short, the words with a length exceeding the preset value cannot be segmented, resulting in low accuracy of the word segmentation.

The maximum probability n-gram grammar [19] uses dynamic programming to find the path of maximum probability to obtain the best word segmentation [20]. A dictionary and a sufficient corpus are needed to calculate this probability. Therefore, the task of word segmentation has changed from the use of algorithms to modeling [21]. However, this change results in significant disadvantages, namely [22],

(1) A parameter space too large to be practical

(2) A sparse data matrix

The averaged perceptron (AP) [23] refers to a perceptron that records the cumulative value of the feature weights and uses averaging to obtain the final model [24]. Although the model segmentation performance of the model trained with the original training set is not very high, the improvement is significant after incremental training. Noise interference is significantly reduced by using a penalty. However, the AP has some disadvantages. The segmentation accuracy is substantially affected by irregular text, high frequency of new words, and network words.

Neural network word segmentation is an understanding-based word segmentation method [22]. It saves the word segmentation rules in the middle layer of the neural network in a decentralized and implicit manner [25], learning a large amount of data by training to obtain the correct word segmentation. In recent years, this method has been continuously improved in natural language processing [26, 27] and has achieved good results [28]. However, several problems remain to be solved to make it suitable for real-world applications. For example, few resources are available to provide adequate training support in minimal texts. In addition, when the text has no language dependency in any direction [29], the model cannot determine appropriate word segmentation rules through context semantics.

There are four reasons [30, 31] for the problems of word segmentation tools in minimal text segmentation. (1) The URL domain name is relatively short; thus, existing word segmentation methods cannot accurately extract keywords. (2) The URL domain name is unstructured text, and there is no contextual logical relationship between the words in the URL. It is impossible to evaluate the attribute information of the words using grammar rules, making it difficult to understand text topics and vectorize knowledge at a later stage. (3) Companies, organizations, or individuals tend to determine their domain names based on personal preferences. Thus, domain abbreviations, misspellings, and language inconsistencies often occur. (4) Web mining of existing URL domain names provides few suitable corpora, resulting in extensive training time and space complexity, low efficiency, and high dimensionality.

3. Efficient Minimal Text Segmentation (EMTS) Model

3.1. Adaptive Text Mining of URL Domain Names. The goal of the proposed EMTS framework is to understand the text of website domain names. The users' online records are obtained and collected in privacy and saved in the dataset S. As shown in the flowchart in Figure 1, the dataset S is the sole input of the model, and the output has several core meanings that can represent a sample. Since the dataset is the user's original online record, it contains noise, and factors such as the Internet environment need to be considered [14]. The dataset S must be standardized and cleaned to facilitate the subsequent analysis [32].

The original dataset S is processed to ensure completeness and accuracy. The missing values are addressed on a case-by-case basis. If there are only a few missing attributes, the corresponding input can be deleted. Otherwise, mean value interpolation is used [33]. Subsequently, preprocessing and simple denoising are performed. The raw data (Figure 2) contain information on the server, user terminal, and file type. We need to distinguish the tags between the texts and extract data on the user to determine the users' online behavior [34].

A directed acyclic graph (DAG) is generated based on the tree structure; this graph contains all possible word combinations of the words in the sample [35]. Dynamic programming is used to find the most probable path and the largest segmentation scale based on the word frequency. Since the data obtained from the network usually contains a large number of html entities, such as "<," "&," "." and ":," we need to perform text formatting to convert these symbol entities into standard text data and save them in the dataset S_1.

The text is transformed into a unified symbol by formulating a tag set. Then, according to the word formation rules of word suffixes, regular expressions can be used to infer the part of speech of a word [36]. The child records in the dataset S_1 are matched in order. Since it is not possible to infer the part of speech of a word based on the context due to the uniqueness of the URL domain name, it is likely that a word may not match with all parts of the speech. To address this problem, we assume that the part of speech of the word is a discrete random variable, and $\theta = (\theta_1, \theta_2, \ldots \theta_k)$ is a multidimensional part-of-speech parameter vector, where k is the number of possible parts of speech of the word. A sample with n values $X = (X_1, X_2, \ldots X_n)$ is selected from this distribution. Assuming that the part-of-speech random variables are independent of each other, the probability function is obtained:

$$P(X_1 = x_1, \ldots X_n = x_n) = \prod_{i=1}^{n} p(x_i; \theta_1, \theta_2, \ldots \theta_k). \quad (1)$$

When $\theta = (\theta_1, \theta_2, \ldots \theta_k)$ is fixed, (1) is expressed as the probability of $X_1 = x_1, \ldots X_n = x_n$. The corresponding part-of-speech probability can be calculated based on the word attributes; the following equation is the likelihood function:

$$L(\theta_1, \theta_2, \ldots \theta_k) = \prod_{i=1}^{n} p(x_i; \theta_1, \theta_2, \ldots \theta_k). \quad (2)$$

The maximum value of the sample values $X_1 = x_1, \ldots X_n = x_n$ is the true estimate of the part of speech. Lemmatization and standardization are performed on the original word. The results are linked to the URL domain name and saved in the dataset S_2.

Because network nouns are not part of publicly available dictionaries, existing word segmentation tools typically divide meaningful URL domain names into invalid words, which is one of the reasons that current word segmentation tools result in poor performance [37, 38]. We consider directional compression and redundancy reduction of known dictionaries to generate a dictionary D_1 with improved accuracy. In order to solve the problem that the exact matching technology relies on the original dictionary and cannot be flexibly recognized for newly added words, a personalized dictionary D_2 is created for users. It can be adaptively improved according to two schemes. One is that according to user preference and frequency of use, the area in the dictionary that has been compressed by the direction is reopened, and words with similar emotions are added to the D_2 personalized dictionary. The other is to determine the appearance of new words by calculating word frequency and boundary entropy. When a certain character combination appears very frequently but the characters or word combinations distributed on the left and right are very random, it is considered as a new word and added to the D_2 dictionary. Finally, the union of the personalized dictionary D_2 and the dictionary D_1 is used to generate the dictionary D_3, where $D_3 = D_2 \cup D_1$.

The bidirectional MM algorithm based on the domain name of the website is used to segment the dataset S_2 into meaningful unordered strings. This algorithm is designed to segment URL domain names and can restore words based on abbreviations, delete meaningless words, and perform high-accuracy splitting of combined words. Unlike in traditional text word segmentation, the input of this algorithm is a word in the domain name of the URL without a space delimiter, and the input word does not require any context. Therefore, we propose to improve the traditional MM algorithm and incorporate the unique characteristics of the URL domain names. The model is shown in Figure 3.

First, the dataset S_2 is divided into a number of single characters based on a single URL domain name, which becomes the input into the forward MM algorithm. Then, matching is performed with the dictionary D_3. If the data match, the next round of matching is performed. Otherwise, one letter is used as a unit, and matching continues by increasing the unit length. This process is then repeated. If the match fails, the words are labeled as unregistered. The unregistered words are sent to corpus C_3 for conflict assessment. Unregistered words with a high frequency of occurrence are recorded for adaptive updating of the dictionary. The collated words are aggregated in the dataset R_1, which is the output of the forward MM algorithm based on the URL domain name. S_2 is then input into the URL domain

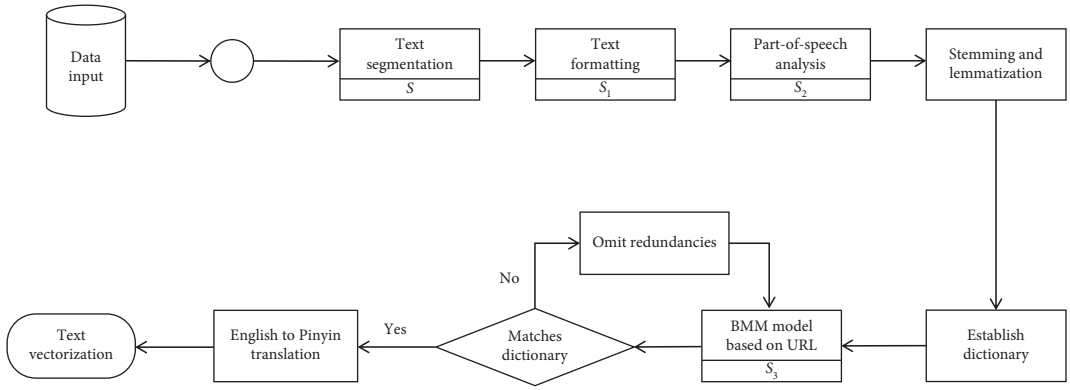

Figure 1: Flow chart of EMTS.

```
<?xml version = "1.0" encoding = "utf-8"?><_d>
<_f n = "private_type">1</_f>
<_f n = "is_webapp">0</_f>
<_f n = "line_no">0</_f>
<_f n = “dealed_line_no”>line 1</_f>
<_f n = "urllog_type"">0</_f>
<_f n = "filename">wrd_template_HEAD_06281609.v80002486fa55fald3a4b43bab792c6
a8ff463f72.zip</_f>
<_f n = "mac">9c-37-f4-0a-bb-25</_f>
<_f n = "termtype">mobile_terminal(Android system)</_f>
<_f n = "nProtocol">6</_f>
<_f n = "filetype">compressed_file</_f>
<_f n = "host">didir1.qq.coms</_t>
<_f n = "trace_t">downfile_https</_f>
<_f n = "urldata">dldirl.qq.com/weixin/checkresupdate/dianying/wrd_template_HEAD_062
81609,v80002486_fa55fald3a4b43bab792c6a8ff463f72.zip</_f>
<_f n = "url">dldir1.qq.com/weixin/checkresupdate/dianying/wrd_template_HEAD_062816
09.v80002416_fa55fald3a4b43bab792c6a8ff463f72.zip</_f>
<_f n = "usr_name">201830431004</_f>
<_f n = "DNS">dldir.tc.qq.com</_f>
```

Figure 2: A sample of the raw data.

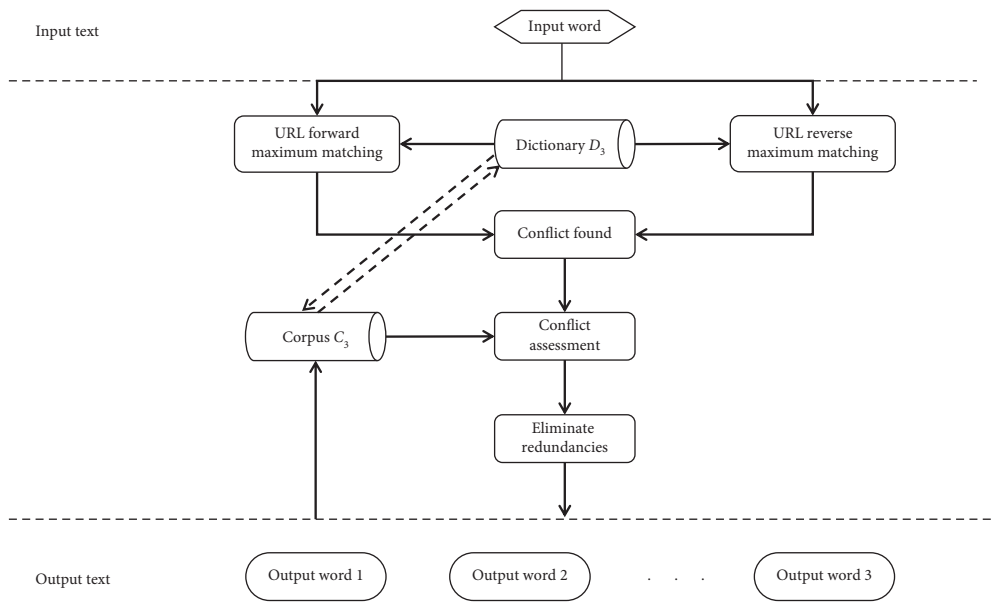

Figure 3: Bidirectional maximum matching algorithm model based on the URL domain name.

reverse MM algorithm, reducing the unit length in the opposite direction to perform similarity matching. Finally, the result of the reverse MM algorithm is obtained (R_2). If $R_1 = R_2$, we obtain the final result $R_3 = R_1 \cap R_2$. If $R_1 \neq R_2$, we need to evaluate the conflicts. R_1 and R_2 are matched with the newly introduced network domain corpus C_3. Finally, for the disputed segments in R_1 and R_2, the long words and the detailed meanings are selected as the final result R_3 of the

bidirectional MM algorithm, and the redundant words are removed. The final result R_3 is saved in the dataset S_3, and the dictionary D_3 and the corpus C_3 are adaptively updated using all records of S_3. An example is shown in Figure 4. We assume that the input word is "movieqqyoutubedownload." After being processed by the bidirectional MM algorithm, the output is "movie/qq/youtube/download."

3.2. Semantic Encoder. In the field of URL domain names, neural network word translation remains the bottleneck of extracting semantics [39]. Traditional word segmentation algorithms do not meet the users readability requirements. The main reason is the website domain name convention. There is no standard for URL domain names, and different regions and countries have different naming preferences, posing a significant challenge to the uniform processing of URL domain names and the extracting of key information from samples.

A typical semantic conflict is that some Chinese URLs are named using Chinese Pinyin, causing loss of semantic information during sample extraction. The reason is that current word segmentation algorithms do not recognize Chinese Pinyin and delete it. In minimal texts, the ratio of words with correct semantics significantly affects the results, and the random deletion of Chinese Pinyin words changes the results substantially.

Therefore, we propose a semantic encoder based on mutual translation between English and Chinese Pinyin; its structure is shown in Figure 5. First, collect the words in batches that were not in the original dictionary at first but were later identified as new words by calculating word frequency and boundary entropy. The Viterbi algorithm is used to calculate the confidence of the appearance of each "word" in the syntactic analysis, then select the more likely Chinese words, and exclude the "words" with less confidence. Here, we set a threshold. The Chinese words with a confidence of less than 1 are regarded as not Pinyin. On the contrary, Chinese words with a confidence of greater than 1 are arranged according to the score. Next, set another threshold n to control the maximum number of Pinyin translated into Chinese, and filter according to the frequency of word occurrence. This double-threshold method greatly reduces the influence of noise on the results. Then, the Chinese-encoded words are adjusted using different weights to create a list linking the vocabulary. The linked list is decoded in English and matched with the original sample. If there is a match, the initial input Pinyin is used to form a closed loop. Finally, in the D_2 personalized dictionary, replace the original Pinyin string with the processed English words to ensure that all recorded language rules are unified. In this way, the semantic encoder realizes the mapping of Pinyin to English.

More experimental results are shown in Table 1.

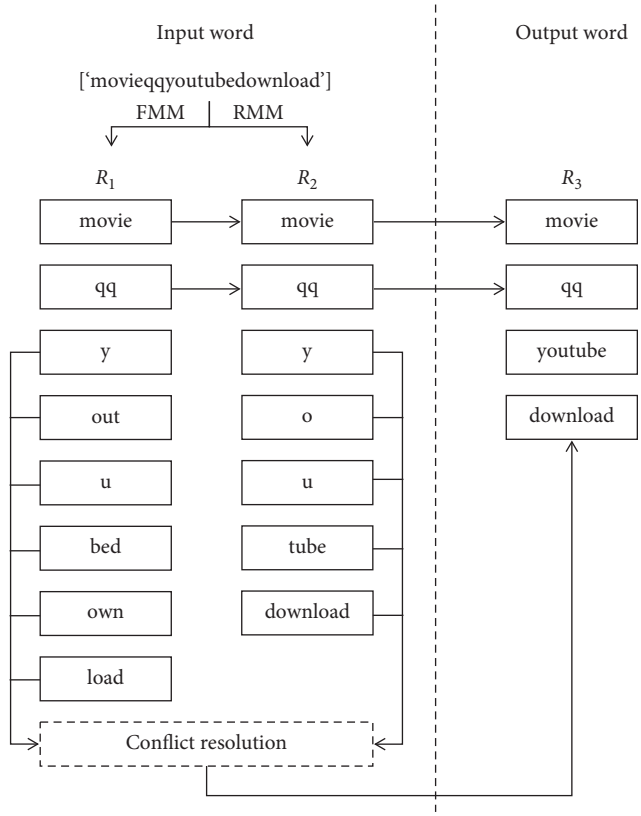

Figure 4: Examples of conflict resolution.

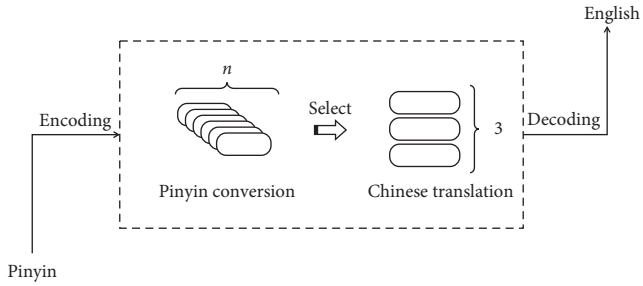

Figure 5: Semantic coding model.

Table 1: Example of Pinyin to English.

Pinyin	English
Yinhang	Bank
Shiwu	Food
Shiyanshi	Laboratory
Nianling	Age
Xinwen	News
Tiyu	Sports
Jinrong	Financial
Qiche	Car
Guangao	Advertising
Sheji	Design

4. Experiments

We propose an improved evaluation standard based on the F_1-score. The performance of the proposed method is compared with that of several popular tools. We perform detailed experimental comparisons of the methods on key data to obtain comprehensive and accurate performance

measurements. In this study, a word in a string containing noise, such as numbers, symbols, and garbled letters, is considered to be segmented correctly when the boundary is correctly placed before the first character and after the last character of the word, and only one word exists between the two boundaries [40, 41].

4.1. F_L-Score. We modified the F_1-score according to the characteristics of the URL domain name and the desired segmentation results. The F_1-score is a weighted combination of precision and recall [42] and is used to evaluate the accuracy of a test. This indicator is commonly used in text mining [43] and is defined as follows:

$$F_1 = \frac{2 \times PR}{P + R}, \tag{3}$$

where P is precision and R is recall.

However, the F_1-score has certain limitations for word segmentation of minimal texts, such as URL domain names [44]. The reason is that the recall is equal to extracting the number of correct words in the sample divided by the total number of samples. Thus, in word segmentation, the more the correct words segmented are, the higher the recall value is. In other words, if the method oversegments the words, the word segmentation tool provides better results as Wimmer and Altmann put forward the hypothesis that the polysemy is inversely proportional to the length of the word. In other words, especially in the field of minimal text. In other words, especially in the field of minimal text, longer words have more accurate word semantics, so their URLs are more interpretable.

Figure 6 shows a real URL in the dataset. We used the EMTS algorithm and the representative Jieba word segmentation tool to compare, in order to verify that F_1-score has a certain degree of limitations in this field. It is worth noting that the original data contains the word "codetermination," which has accurate semantics, that is to say, we understand the user's intentions relatively appropriately through this word. The algorithm not only retains words with high analytical value but also deletes words with ambiguous semantics, which is more helpful for later analysis work, but F_1-score gives lower feedback. Again, Jieba split "codetermination" into three ambiguous words, and the result completely deviated from the meaning of the entire URL, which would have a great negative impact on the later analysis work. However, its F_1-score is higher. Therefore, the data shows that F_1-score cannot effectively reflect such situations. Combined with the F_1-score formula, the recall rate greatly affects the final result. In the field of minimal text, this principle runs counter to the need for results. The F_1-score only considers the number of words that are correctly segmented and does not consider the relationship between the word length and semantics. Minimal texts are highly sensitive to the number, length, and semantics of words, and an incorrect segmentation of a word causes noise interference. Thus, the F_1-score is not applicable to minimal text segmentation.

To address this problem, we propose an improvement of the F_1-score by adding a correction factor that considers the sample length. Suppose the sample set after word segmentation is $(x_1, x_1, \ldots, x_m)$, and the sample set with correct segmentation result is $(x_1, x_1, \ldots, x_n)$. Here, m is the number of characters in the list after word segmentation and n is the number of characters that are correct. The length of the word x is denoted as $\mathrm{len}(x)$. The optimized correction factor L is defined as

$$\alpha = \frac{\sum_{i=1}^{n} \mathrm{len}(x_i)}{m}, \tag{4}$$

$$L = \frac{1 - e^{-\sqrt{\alpha}}}{1 + e^{-\sqrt{\alpha}}}. \tag{5}$$

In word segmentation, the correction factor L can be used as a weight. The larger the value of L, the longer the average word length after segmentation is and the more semantic information it contains. We propose the F_L-score, which combines F_1-score (3) with correction factor L (5), which considers the sample length. The evaluation indices to assess the performance of the algorithm include word precision, word recall, and word length. The F_L-score is defined as

$$\begin{aligned} \frac{3}{F_L} &= \frac{1}{P} + \frac{1}{R} + \frac{1}{L}, \\ \mathbf{F_L} &= \frac{3 \times \mathbf{PRL}}{\mathbf{LP} + \mathbf{LR} + \mathbf{PR}}. \end{aligned} \tag{6}$$

4.2. Dataset. One of the reasons for the relative scarcity of public datasets related to this research is that URLs that reflect user online behavior involve privacy issues. And, since publicly available corpora are processed data that consider the uniqueness of minimal texts, such as URL domain names, these corpora do not meet our experimental needs. In this section, we apply the proposed EMTS algorithm on two datasets to ensure that the experiment was fair and accurate. One is collected from real network behavior data of university users, and the other is a public dataset.

We use real datasets to implement our method. The dataset was collected from a campus public server in 2020 and contained raw online records of real users for the past four years (80 million data points per year). An example of actual collected data is shown in Figure 7. Here, the "user" column represents the user number, which can be used to identify the user information. The number is unique, and the field has no null values. The "Tm_type" column represents the user's Internet access method. "PC" represents a personal computer, and "MT" represents a mobile terminal device. The "serv" column represents the user's online behavior, where "web" represents browsing the web. The "app" column represents the type of user browsing the web page. Its data quality is relatively low, with many null values. The "result" column represents the details of the server-side data of the user's online data record, including the link address, terminal device, and file type. The data format of this column

Raw data	w/search = codetermination&Special%3Afulltext = 1ns0	F_1-score	F_L-score
EMTS	[search, codetermination, title, special, fulltext]	0.7143	0.6944
Jieba	[w, search, code, termi, nation, title, Special, 3A, full, text, 1ns0]	0.7992	0.5638

Figure 6: Comparison of F_1-score and F_L-score experiments.

	User	Tm_type	Serv	App	Result
0	201821200621	/PC/MAC PC	Web		<?xml version = "1.0" encoding = "utf-8"?><_d)\n<_f n = ….
1	201821100167	/PC/MAC PC	Web	Game	<?xml version = "1.0" encoding = "utf-8"?><_d)\n<_f n = ….
2	201821100333	/PC/MAC PC	Web	IT	<?xml version = "1.0" encoding = "utf-8"?><_d)\n<_f n = ….
3	201821100003	/PC/MAC PC	Web		<?xml version = "1.0" encoding = "utf-8"?><_d)\n<_f n = ….
4	201821200775	/PC/MAC PC	Web	News	<?xml version = "1.0" encoding = "utf-8"?><_d)\n<_f n = ….

Figure 7: An example of the data.

is more standardized and contains sufficient information. View the details of the "result" of the example, as shown in Figure 2. Since the data in the other columns is of low value for analysis and there are many missing values, the "result" column of each data is extracted for the following experiment. The specific extraction steps can be found in the section "adaptive text mining of URL domain names," which will not be repeated here.

We use a dataset that is used to evaluate the identification of a suitable URL for a given entity. The website information is collected from four general domains: those are of type company, city, movie, and person. Per domain, 25 entities are selected indiscriminately; hence, 100 entities in total. The actual numbers ranged from 10 to over 3000 connections per entity, so a total of 347,527 records were collected. We extract the URLs in the dataset and conduct experiments. Some data examples are shown in Figure 8. The dataset has been introduced in a paper at the ISWC workshop on Web of Linked Entities 2012.

4.3. Comparison of Experimental Results on Real Datasets. This dataset uses real users' online records. The EMTS is compared with several popular word segmentation tools, such as Jieba, forward MM (FMM) algorithm, reverse MM (RMM) algorithm, Porter Stemmer, Lancaster Stemmer, Snowball Stemmer, and Lemmatizer.

Jieba is a Python-based word segmentation library, which has very powerful word segmentation capabilities for text. The Jieba library supports three words segmentation modes: precise mode, full mode, and search engine mode. We use Jieba's precise mode because it has high accuracy and relatively fast speed, which are highly suitable for text analysis.

The FMM and RMM algorithms are classic word segmentation methods. Because the accuracy of these algorithms depends heavily on the composition of the dictionary, we use the same dictionary in these algorithms as the EMTS for a fair comparison. These algorithms use a partial segmentation method that is widely used in text

ID	URL
0	http://www.utexas.edu/
1	http://www.city-data.com/city/Albany-Texas.html
2	http://www.allacrosstexas.com/albany.htm
3	http://albany.craigslist.org/
4	http://en.wikipedia.org/wiki/Albany,_Texas

Figure 8: Some examples of public datasets.

processing to balance the quality of text segmentation and efficiency.

The Porter Stemmer, Lancaster Stemmer, Snowball Stemmer, and Lemmatizer segmentation tools, which are part of the Natural Language Toolkit (NLTK), are the most commonly used segmentation methods in natural language processing. The text is analyzed comprehensively from two aspects: rule-based stem extraction method and dictionary-based lemmatization strategies.

The experimental results are listed in Table 2. The following observations are made. The proposed EMTS provides significantly better results than the other methods for all indicators using this dataset. The precision of 1 indicates that all words are segmented correctly, and there are no segmentation errors. The recall value shows improvements over the other methods in terms of missing segmentation. The proposed correction factor for text length, "len," accurately quantifies the character length and improves the text segmentation result. Since the URL name is relatively long, the use of the correction factor in the EMTS provides more meaningful and high-quality results. The F_L-score shows the advantage of the EMTS over the other methods. Furthermore, the stability of the theory is proved from the view of the S.D. (Standard Deviation). Finally, since the samples are independent random variables, we can approximate the normal distribution of the samples. Figure 9 shows the normal distribution of the performance of the eight word segmentation methods, denoted as $N(\mu, \sigma^2)$. There are negligible differences in the scores of the other seven words

Table 2: Experimental results on real datasets.

Location		Precision	Recall	Len	F_L-score	S. D.
EMTS		**1.0**	**0.590**	**0.745**	**0.743**	**0.147**
Forward maximum match		0.383	0.549	0.500	0.466	0.207
Reverse maximum match		0.392	0.544	0.516	0.474	0.206
Jieba		0.500	0.417	0.568	0.510	0.196
NLTK	Porter Stemmer	0.461	0.437	0.540	0.475	0.182
	Lancaster Stemmer	0.422	0.402	0.501	0.438	0.183
	Snowball Stemmer	0.464	0.439	0.542	0.478	0.181
	Lemmatizer	0.501	0.473	0.568	0.511	0.196

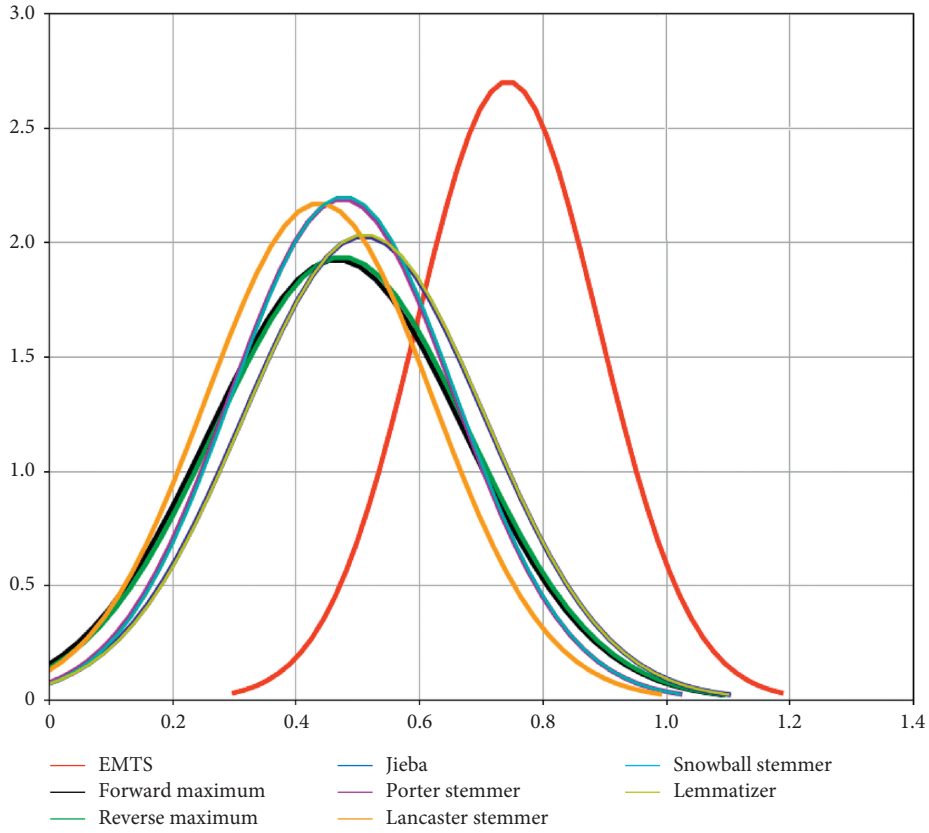

Figure 9: Normal distribution of performance scores.

segmentation tools. The EMTS has the highest μ value (representing the performance score) and the lowest σ value (representing the standard deviation). These results indicate that the proposed method is more stable, the variability in the text segmentation results is lower, and the robustness to the text content is higher than in the other methods.

It is well known that the text length considerably affects the quality of word segmentation results. The reason is that the samples are complete sentences, and each word has a context relationship, which can be used to evaluate the part of speech, tense, and other factors. The more complete the context is and the more words it contains, the richer the meaning of the text is and the easier it is to mine.

However, the content of unstructured data is not standardized, such as URL domain names, which contain words and symbols. Therefore, this type of data does not conform to the assumption the longer the text, the more the information it contains. Figure 10 shows the relationship between the text length and the segmentation quality of different types of segmentation tools used to analyze this nonstandard text. The noise in a long text affects the performance of all segmentation methods to some extent. In the EMTS, we use bidirectional MM, secondary review, and disconnection between the front and back links. Therefore, problems caused by noise interference are minimized in word segmentation. The difference in the segmentation quality between long text and short text is significantly smaller, and the overall performance is higher for the EMTS than the other segmentation methods. This result shows that, in unstructured data with irregular content, the length of the text and the word segmentation performance are inversely proportional.

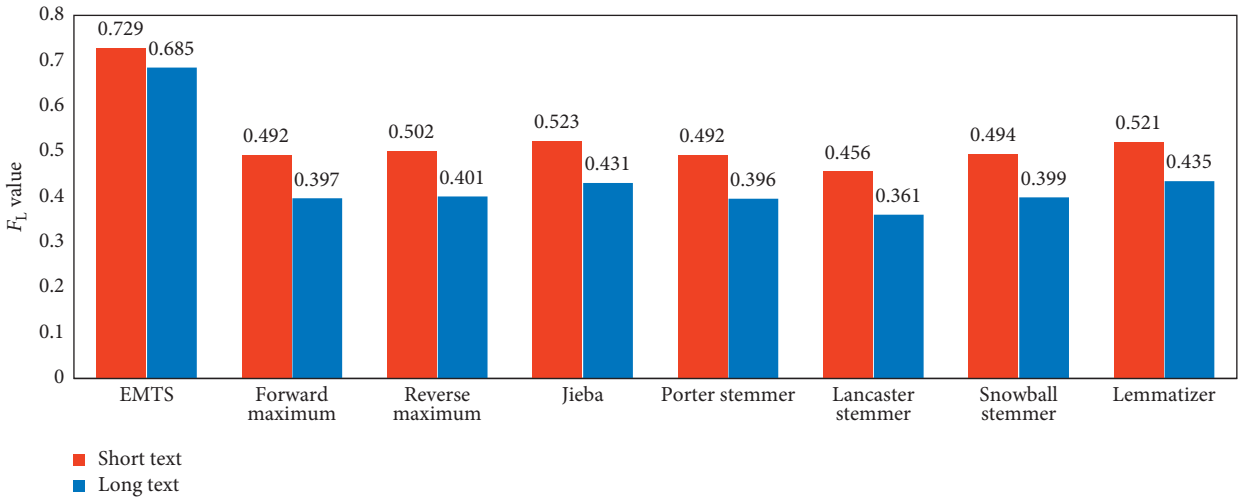

Figure 10: Effect of the text length on performance.

4.4. Comparison of Experimental Results on Public Datasets. We used the same method in the above experimental chapter to conduct experiments on public datasets and use the same word segmentation tools (i.e., Jieba, FMM, and NLTK) for comparison. This helps to verify the effectiveness and accuracy of the algorithm. The experimental results are listed in Table 3. The overall indicator value shows that the EMTS algorithm has higher segmentation accuracy and good mining quality in the field of URL analysis, compared with several other popular universal word segmentation tools. Combining the indicator values (i.e., Table 2) in the above experimental section, this experiment indicates that the algorithm has not obvious fluctuations in the analysis of different datasets, which proves that EMTS has good robustness.

It is worth noting that Jieba has the highest recall rate and is about 20% higher than EMTS. However, combined with the accuracy and processed data (as shown in Figure 11) for comprehensive analysis, this reflects that the number of "words" segmented by the algorithm is more, but in most cases, complete words cannot be obtained, resulting in poor interpretability. And, EMTS deletes many stop words with ambiguous semantics, and its segmentation principle is to obtain words with longer length, more accurate semantics, and more in line with user habits. Therefore, the experimental results show that EMTS has the highest accuracy rate, but the number of processed words (i.e., recall rate) will be relatively inevitable loss. In general, this recall rate loss is far less than the accuracy rate gains.

Figure 12 shows the normal distribution of the performance of different word segmentation methods. According to the experimental data, the word segmentation effects of the 8 word segmentation methods are roughly divided into two categories. The overall effect of EMTS word segmentation is significantly better than other methods. However, compared with the effect diagram of the previous experiment (Figure 9), the standard deviation of the EMTS of this experiment is not significantly different from other methods. Based on the characteristics of the data, the public dataset used in this experiment has less interference from other noises such as Pinyin. In other words, the format of this dataset is relatively standard, so the effect of various word segmentation algorithms is relatively stable, and the standard deviation is reduced to varying degrees.

Figure 13 illustrates typical problems occurring with URL domain names. (1) There are several interference items, such as "dldir1." These characters have no meaning and should be deleted. Otherwise, the result will be affected. (2) Several words are concatenated, such as "checkresupdate." In this example, the EMTS determines the pattern according to the words and obtains the abbreviated and misspelled words with high probability. (3) When the text contains a Chinese Pinyin term, such as "dianying," it is replaced with the most likely English term according to the frequency of the Chinese Pinyin combination. (4) The string contains mixed characters, such as "wrd_template_HEAD_06281609 and v80002486_fa55fa1d3a4b43bab792c6a8ff463f72.zip." After deleting the symbols, the EMTS extracts meaningful words and recognizes changes in the tense and passivity. The text suffix receives a relatively high weight because it has a high degree of recognition. Figure 14 shows the result of the EMTS algorithm. The theme of the sample can be clearly identified.

Table 4 shows the processing results of different algorithms using representative examples of the experimental data. The results demonstrate the superiority of the EMTS; the segmentation results are much more succinct and clear than those of the other methods and provide a high-quality output with clear meaning.

Table 1 shows some examples of actual Pinyin conversion to English, all of which are collected from the real dataset in the above experiment. The results reflect that, among these algorithms, only the EMTS algorithm can process Pinyin in URL analysis, and it has high accuracy.

Table 3: Experimental results on public datasets.

Location		Precision	Recall	Len	F_L-score	S. D.
EMTS		**1.0**	**0.532**	**0.787**	**0.723**	**0.146**
Forward maximum match		0.369	0.407	0.519	0.423	0.163
Reverse maximum match		0.378	0.405	0.521	0.427	0.170
Jieba		0.356	0.713	0.504	0.484	0.148
NLTK	Porter Stemmer	0.407	0.366	0.547	0.428	0.167
	Lancaster Stemmer	0.405	0.365	0.537	0.424	0.170
	Snowball Stemmer	0.406	0.366	0.546	0.427	0.168
	Lemmatizer	0.437	0.396	0.572	0.458	0.153

Raw data	[youtube.com/watch?v = y_ Rzr6FAjHo_utexasrobertoorcisellfemaledriven_drama tocbswith
EMTS	[youtube, com, watch, texas, robert, sell, female, driven, drama]
Jieba	[you, tube, com, watch, v, y, R, zr, 6, FA, j, H, o, ute, x, s, robert, o, or, cis, ell, female, driven, drama, t, o, c, bs, wi, th]

Figure 11: Comparison of experimental results of EMTS and Jieba.

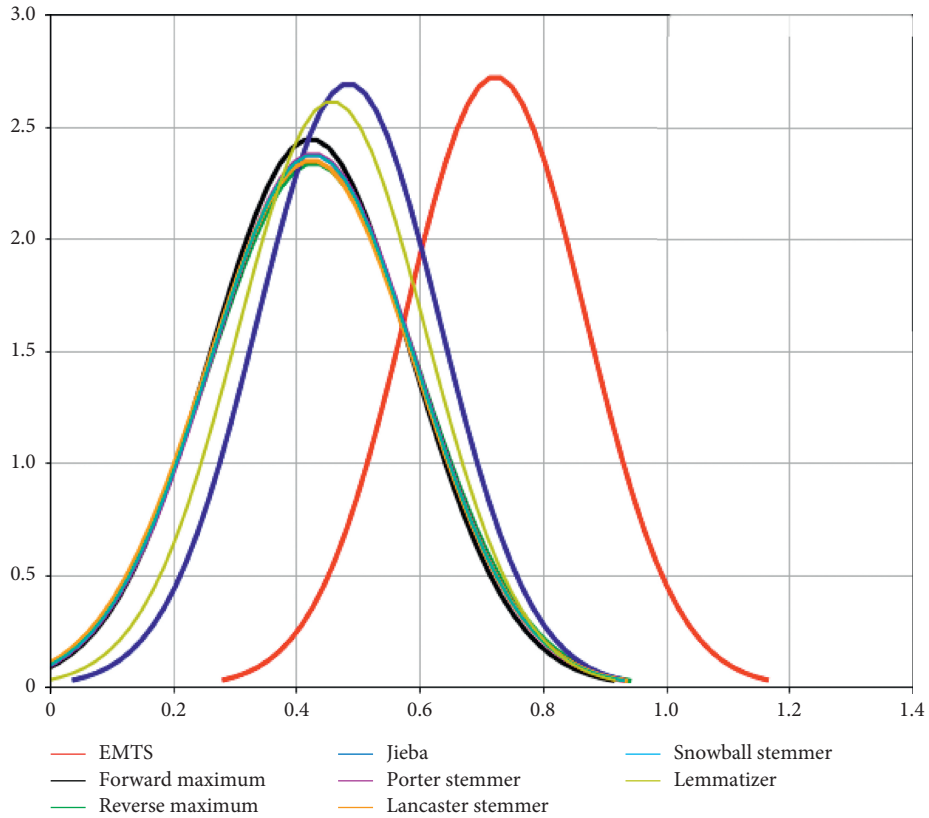

Figure 12: Normal distribution of the public dataset. Example of raw and processed data.

dldir1.qq.com/weixin/checkresupdate/dianying/wrd_template_HEAD_06281609, v80002486_fa55fald3a4b43bab792c6a8ff463f72.zip

Figure 13: An example of a random raw data string.

["qq", "com", "weixin", "check", "update", "movie", "template", "head", "zip"]

Figure 14: A portion of the data after EMTS processing. Comparison of outputs of different models.

Table 4: Experimental result.

Case 1			
Raw data	[stdl.qq.com/stdl/qbfilepush/tushu/qqbrowser/cloudctrl/production/15420175063859.txt/ leaves.imtt.x2.sched.dcl oudstc.com]	F_1-score	F_L-score
EMTS	**[qq, push, file, book, browser, qq, cloud, production, txt, leaf, com]**	**0.714**	**0.701**
Jieba	[stdl, qq, com, stdl, qbfilepush, qqbrowser, cloud, ctrl, pro-duction, 15420175063859, txt, leaves, imtt, x2, sched, d- cloudstc, com]	0.424	0.482
Forward maximum	[std, l, qq, com, std, l, q, b, file, push, tush, u, qq, browser, cloud, ct, r, l, production, 15, 42, 0, 175, 0, 6, 38, 59, txt, leave, s, i, mt, t, x, 2, sc, he, d, dc, loud, s, tc, com]	0.387	0.326
Reverse maximum	[s, t, dl, qq, com, s, t, dl, q, b, file, push, tush, u, qq, browser, cloud, ct, r, l, production, 1, 54, 20, 17, 50, 6, 38, 59, txt, l, eaves, i, m, tt, x, 2, sc, h, ed, d, cl, o, u, ds, tc, com]	0.353	0.296
Porter Stemmer	[stdl, qq, com, stdl, qbfilepush, tushu, qqbrowser, cloudc-trl, product, 15420175063859, txt, leav, imtt, x2, sched, dcloudstc, com]	0.258	0.332
Lancaster Stemmer	[stdl, qq, com, stdl, qbfilepush, tushu, qqbrowser, cloud-ctrl, produc, 15420175063859, txt, leav, imtt, x2, sched, dcloudstc, com]	0.210	0.266
Lemmatizer	[stdl, qq, com, stdl, qbfilepush, tushu, qqbrowser, cloudctrl, production, 15420175063859, txt, leaf, imtt, x2, sched, dcloudstc, com]	0.323	0.401
Case 2			
Raw data	[pcs-sdk-server.alibaba.com/l?umid = &csid = c01549d12 f3c447b18bb557a7766f0bb&acnt = &hosttype = 1&a mp; log = RegisterHostPath:[C: \ProgramFiles(x86)\AliWang Wang\ZhiFu]/na61-na62.wagbridge.alibaba.ali]	F_1-score	F_L-score
EMTS	**[server, alibaba, com, mid, type, host, log, path, host, register, program, file, aliwangwang, payment, bridge, wag, Alibaba]**	**0.729**	**0.726**
Jieba	[pcs, dk, server, alibaba, com, l, umid, amp, csid, c01549d12f3c447b18bb557a7766f0bb, amp, acnt, amp, host, type, 1, amp, log, register, host, path, C, program, file, x86, ali, wangwang, Zhifu, na61, na62, wagbridge, alibaba, ali]	0.565	0.439
Forward maximum	[pc, s, sd, k, server, alibaba, com, l, u, mid, amp, cs, id, c, 0, 15, 49, d, 12, f, 3, c, 44, 7, b, 18, bb, 55, 7, a, 77, 66, f, 0, bb, amp, ac, nt, amp, host, type, 1, amp, log, egis, te, r, os, t, at, h, ro, gram, il, es, x, 86, li, ang, ang, hi, u, na, 61, na, 62, wag, bridge, alibaba, ali]	0.275	0.229
Reverse maximum	[p, cs, sd, k, server, alibaba, com, l, u, mid, amp, cs, id, c, 0, 15, 49, d, 12, f, 3, c, 4, 47, b, 18, bb, 5, 57, a, 77, 66, f, 0, bb, amp, ac, nt, amp, host, type, 1, amp, log, e, gist, er, os, t, a, th, ro, gram, il, es, x, 86, li, ang, ang, hi, u, na, 61, na, 62, wag, bridge, alibaba, ali]	0.265	0.223
Porter Stemmer	[pc, sdk, server, alibaba, com, l, umid, amp, csid, c01549d12f3c447b18bb557a7766f0bb, amp, acnt, amp, hosttyp, 1, amp, log, registerhostpath, C, program, file, x86, aliwangwang, zhifu, na61, na62, wagbridge, alibaba, ali]	0.292	0.370
Lancaster Stemmer	[pcs, sdk, serv, alibab, com, l, umid, amp, csid, c01549d12f3c447b18bb557a7766f0bb, amp, acnt, amp, hosttyp, 1, amp, log, registerhostpa, c, program, fil, x86, aliwangwang, zhifu, na61, na62, wagbridge, alibab, al]	0.182	0.217
Lemmatizer	[pc, sdk, server, alibaba, com, l, umid, amp, csid, c01549d12f3c447b18bb557a7766f0bb, amp, acnt, amp, hosttype, 1, amp, log, RegisterHostPath, C, Program, Files, x86, AliWangWang, ZhiFu, na61, na62, wagbridge, alibaba, ali]	0.245	0.318
Case 3			
Raw data	[info.pinyin.sogou.com/aserver push/html/tupian file/6013 86 20190628151003.html]	F_1-score	F_L-score
EMTS	**[pinyin, sogou, com, server, push, image, file, html]**	**0.869**	**0.823**
Jieba	[info, pinyin, sogou, com, aserver, push, html, tupian, file, 601386, 20190628151003, html]	0.653	0.605
Forward maximum	[info, pinyin, sogou, com, as, er, v, er, push, html, tupi, an, file, 60, 13, 86, 20, 190, 62, 81, 51, 0, 0, 3, html]	0.535	0.499
Reverse maximum	[info, pinyin, sogou, com, a, server, push, html, tupi, an, file, 60, 13, 86, 20, 190, 6, 28, 15, 100, 3, html]	0.643	0.626
Porter Stemmer	[info, pinyin, sogou, com, aserv, push, html, tupian, file, 601386, 20190628151003, html]	0.653	0.605
Lancaster Stemmer	[info, pinyin, sogou, com, aserv, push, html, tup, fil, 601386, 20190628151003, html]	0.570	0.500
Lemmatizer	[info, pinyin, sogou, com, aserver, push, html, tupian, file, 601386, 20190628151003, html]	0.653	0.605

5. Conclusion

The demand for the integration of the analysis of users' online behavior and text mining methods is increasing. URL domain names are important source data, but the analysis of minimal texts remains one of the bottlenecks in natural language processing. This paper proposed the EMTS method for ULR domain names to extract valid topics from context-free and very short texts with high quality. We focused on extracting the meaning of the words in URL domain names, considering both Chinese Pinyin and English term and integrating hot topics and social backgrounds to improve the

recognition rate of emerging words. Experiments on real user online datasets showed that the performance of the EMTS method was significantly higher than that of currently popular word segmentation tools and algorithms. Unlike most existing methods, which have complex deep architectures, the proposed EMTS model is concise and efficient and provides straightforward output. This method uses unsupervised word segmentation without relying on any corpus pretraining. It does not need to consider the influence of human factors on the experimental results, improving the stability of the model.

Data Availability

All datasets can be downloaded from https://pan.baidu.com/s/1IHJjIlMcADYJA5aQSTm8VQ.

Conflicts of Interest

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

Acknowledgments

This research was supported by the Natural Science Foundation of China under contract nos. 61873324 and 61673405.

References

[1] D. E. Kouicem, A. Bouabdallah, and H. Lakhlef, "Internet of things security: a top-down survey," *Computer Networks*, vol. 141, pp. 199–221, 2018.

[2] H. Hong, Z. Bo, Z. Hao et al., "A cross-platform consumer behavior analysis of large-scale mobile shopping data," in *Proceedings of the world wide web Conference*, pp. 1785–1794, Lyon, France, April 2018.

[3] M. Tubishat, N. Idris, and M. A. M. Abushariah, "Implicit aspect extraction in sentiment analysis: review, taxonomy, oppportunities, and open challenges," *Information Processing & Management*, vol. 54, no. 4, pp. 545–563, 2018.

[4] F. Farahmand, "The importance of human information processing: a behavioral economics model for predicting domain name choice," *Computer*, vol. 50, no. 9, pp. 67–74, 2017.

[5] Z. Zong and C. Hong, "On application of natural language processing in machine translation," in *Proceedings of the International Conference on Mechanical*, pp. 506–510, Qingdao, China, March 2018.

[6] S. Yang, J. Wei, and X. Zhao, "Study and implementation of scientific research project assessment system on network based on text mining," in *Proceedings of the Advanced Information Technology, Electronic & Automation Control Conference*, pp. 1956–1959, Chongqing, China, March 2017.

[7] M. Trevisan and I. Drago, "Robust url classification with generative adversarial networks," *ACM Sigmetrics-Performance Evaluation Review*, vol. 46, no. 3, pp. 143–146, 2019.

[8] Z. M. Arani, A. A. Barforoush, and H. Shirazi, "Representing unstructured text semantics for reasoning purpose," *Journal of Intelligent Information Systems*, vol. 56, no. 3, pp. 1573–7675, 2020.

[9] R. Rajalakshmi and C. Aravindan, "A naive bayes approach for url classification with supervised feature selection and rejection framework," *Computational Intelligence*, vol. 34, no. 1, pp. 363–396, 2018.

[10] X. Xie, Y. Fu, H. Jin, Y. Zhao, and W. Cao, "A novel text mining approach for scholar information extraction from web content in Chinese," *Future Generation Computer Systems*, vol. 111, pp. 859–872, 2020.

[11] L. Chen, M. Han, H. Shi, and X. Liu, "Multi-context embedding based personalized place semantics recognition," *Information Processing & Management*, vol. 58, no. 1, Article ID 102416, 2021.

[12] R. Qu, Y. Fang, W. Bai, and Y. Jiang, "Computing semantic similarity based on novel models of semantic representation using wikipedia," *Information Processing & Management*, vol. 54, no. 6, pp. 1002–1021, 2018.

[13] W. Inoubli, S. Aridhi, H. Mezni, M. Maddouri, and E. Mephu Nguifo, "An experimental survey on big data frameworks," *Future Generation Computer Systems*, vol. 86, pp. 546–564, 2018.

[14] L. Celardo and M. G. Everett, "Network text analysis: a two-way classification approach," *International Journal of Information Management*, vol. 51, Article ID 102009, 2020.

[15] I. Pak and P. L. Teh, *Text Segmentation Techniques: A Critical Review*, Springer International Publishing, New York, NY, USA,, 2018.

[16] M. Mucha, "Maximum matching," in *Encyclopedia of Algorithms*Springer, New York, NY, USA, 2016.

[17] N. Blum, "A new approach to maximum matching in general graphs," in *Proceedings of the 17th International Colloquium on Automata, Languages and Programming*, pp. 586–597, Warwink, England, June 1990.

[18] J. Pei, "A dictionary-based maximum match algorithm via statistical information for Chinese word segmentation," *International Journal of Electronics and Information Engineering*, vol. 12, no. 1, pp. 24–33, 2020.

[19] P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, "Class-based n -gram models of natural language," *Computational Linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[20] B. Ding, H. Luo, Z. Wu, and S. Zhang, "A vectorization approach to language identification of social media short texts," in *Proceedings of the 2020 12th International Conference on Machine Learning and Computing, ICMLC 2020*, February 2020, Article ID 462467.

[21] M. Jimenez, C. Maxime, Y. L. Traon, and M. Papadakis, "On the impact of tokenizer and parameters on n-gram based code analysis," in *Proceedings of the IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pp. 437–448, IEEE Computer Society, Madrid, Spain, September 2018.

[22] S. Takahashi and K. Tanaka-Ishii, "Evaluating computational language models with scaling properties of natural language," *Computational Linguistics*, vol. 45, no. 3, pp. 1–32, 2019.

[23] N. Nguyen and Y. Guo, *Comparisons of Sequence Labeling Algorithms and Extensions*, pp. 681–688, Cornell University, Ithaca, NY, USA, 2007.

[24] S. Dutta and E. M. O'Rourke, "Open-ended questions: the role of natural language processing and text analytics," in *Employee Surveys and Sensing*Oxford University Press, Oxford, England, 2020.

[25] Y. Li and T. Yang, *Word Embedding for Understanding Natural Language: A Survey*, Springer International Publishing, New York, NY, USA, 2018.

[26] Y. Goldberg and G. Hirst, *Neural Network Methods in Natural Language Processing, Synthesis Lectures on Human Language Technologies*, Morgan & Claypool, San Rafael, CA, USA, 2017.

[27] M. Gong, C. Yao, Y. Xie, and M. Xu, "Semi-supervised network embedding with text information," *Pattern Recognition*, vol. 104, Article ID 107347, 2020.

[28] A. Khatua, A. Khatua, and E. Cambria, "A tale of two epidemics: contextual word2vec for classifying twitter streams during outbreaks," *Information Processing & Management*, vol. 56, no. 1, pp. 247–257, 2019.

[29] U. Khandelwal, H. He, P. Qi, and D. Jurafsky, "Sharp nearby, fuzzy far away: how neural language models use context," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 284–294, Melbourne, Australia, July 2018.

[30] H. Balakrishnan, K. Lakshminarayanan, S. Ratnasamy, S. Shenker, I. Stoica, and M. Walfish, "A layered naming architecture for the internet," *ACM SIGCOMM-Computer Communication Review*, vol. 34, no. 4, pp. 343–352, 2004.

[31] M. J. Collins, *Hypertext Markup Language*, Apress, Berkeley, CA, USA, pp. 3–14, 2017.

[32] Y. Taskin, T. Hecking, and H. U. Hoppe, "ESA-T2N: a novel approach to network-text analysis," in *Complex Networks and Their Applications VIII*Springer, New York, NY, USA, 2020.

[33] S. Bruvoll and M. S. Floater, "Transfinite mean value interpolation in general dimension," *Journal of Computational and Applied Mathematics*, vol. 233, no. 7, pp. 1631–1639, 2010.

[34] X. Lin and C. Han, "Chinese text sentiment analysis based on improved convolutional neural networks," in *Proceeding of the IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, pp. 922–926, Beijing, China, November 2018.

[35] W. Wojciech, "An algorithm based on a directed acyclic word graph," *Grammatical Inference*, vol. 673, pp. 77–81, 2017.

[36] F. Zhao, B. Quan, J. Yang, J. Chen, Y. Zhang, and X. Wang, "Document summarization using word and part-of-speech based on attention mechanism," *Journal of Physics: Conference Series*, vol. 1168, Article ID 032008, 2019.

[37] R. Schwarzenberg, L. Raithel, and D. Harbecke, "Neural vector conceptualization for word vector space interpretation," in *Proceedings of the Naacl-hlt Workshop on Evaluating Vector Space Representations for Nlp*, pp. 1–7, Minneapolis, MN, USA, June 2019.

[38] Y. Shao, C. Hardmeier, and J. Nivre, "Universal word segmentation: implementation and interpretation," *Transactions of the Association for Computational Linguistics*, vol. 6, pp. 421–435, 2018.

[39] H. Choi, K. Cho, and Y. Bengio, "Context-dependent word representation for neural machine translation," *Computer Speech & Language*, vol. 45, pp. 149–160, 2017.

[40] Y. Doval and C. GmezRodrguez, "Comparing neural and ngram based language models for word segmentation," *Journal of the Association for Information Science and Technology*, vol. 70, no. 2, Article ID 187197, 2019.

[41] J. Pei, "A dictionary-based maximum match algorithm via statistical information for Chinese word segmentation," *International Journal of Electronics and Information Engineering*, vol. 12, no. 1, pp. 24–33, 2020.

[42] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and *F*-score, with implication for evaluation," in *Advances in Information Retrieval*, D. E. Losada and J. M. Fernandez-Luna, Eds., Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[43] Y. Nan, K. M. A. Chai, W. S. Lee, and H. L. Chieu, *Optimizing F-Measure: A Tale of Two Approaches*, pp. 1555–1562, National University of Singapore, Singapore, 2012.

[44] M. Liu, C. Xu, Y. Luo, C. Xu, Y. Wen, and D. Tao, "Cost-sensitive feature selection by optimizing f-measures," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1323–1335, 2018.

[45] G. Wimmer and G. Altmann, *Two Hypotheses on Synonymy*, Veda, Bratislava, Slovakia, 2001.