IEICE TRANS. INF. & SYST., VOL.E101–D, NO.7 JULY 2018

PAPER

Error Correction for Search Engine by Mining Bad Case

Jianyong DUAN[†,††a)], ***Member***, **Tianxiao JI**[†,††b)], ***and*** **Hao WANG**[†,††c)], ***Nonmembers***

SUMMARY Automatic error correction of users' search terms for search engines is an important aspect of improving search engine retrieval efficiency, accuracy and user experience. In the era of big data, we can analyze and mine massive search engine logs to release the hidden mind with big data ideas. It can obtain better results through statistical modeling of query errors in search engine log data. But when we cannot find the error query in the log, we can't make good use of the information in the log to correct the query result. These undiscovered error queries are called Bad Case. This paper combines the error correction algorithm model and search engine query log mining analysis. First, we explored Bad Cases in the query error correction process through the search engine query logs. Then we quantified the characteristics of these Bad Cases and built a model to allow search engines to automatically mine Bad Cases with these features. Finally, we applied Bad Cases to the N-gram error correction algorithm model to check the impact of Bad Case mining on error correction. The experimental results show that the error correction based on Bad Case mining makes the precision rate and recall rate of the automatic error correction improved obviously. Users experience is improved and the interaction becomes more friendly.

key words: *query correction, Bad Case mining, N-gram model*

1. Introduction

With the explosive growth of Internet information, search engine as an information query tool plays a more and more important role. Especially in the background of big data and cloud computing, how to use the massive data to provide users with a better search and retrieval experience has become an important aspect of the research on search information retrieval technology.

At June 2014, the latest CNNIC statistical report shows that there are 507.49 million search engine user in China. These searching behaviors which produced by hundreds of millions of users will produce massive search engine logs. The big data age recognizes the hybridity of data, explores the correlation between data, analyzes and forecasts the huge value hid behind data. It can improve the accuracy, convenience and efficiency of the application to provide users with better search services.

Bad Case is a search engine term. It is analyzing obviously incorrect records in search results to see what strategy leads to and amend the relevant parameters. After analyzing a lot of abnormal cases, search engine will collect a lot of case information. When encountering unreasonable search results, it will confirm the characteristics of these cases. If there are similar cases, it will be adjusted.

Shaoping Ma et al used methods based on log mining [1]. Their system will give out the error correcting word and 11.9% of users will click on the error correcting word to query when a user enters the wrong search terms. If the error correction word is incorrect, it not only cannot achieve the correct role of guiding the user, but also will affect the user's search experience. So, we need an effective analytical mining strategy which can analyze search engine logs to comprehensively and efficiently discovery Bad Cases. Then the method will improve the error correction function of search engines, enhance the user's search experience and improve the accuracy of search engine search results.

In this paper, we apply the big data thought. Starting with the relevance of the data and a large number of search engine query logs, we analyzed the correlation between offline logs and established mathematical models to quantify these associated relationships. And then we can automatically dig out a large number of Bad Cases from the massive logs. Using these Bad Cases combined with efficient query error correction method, the recall rate and precision rate of online query error correction are greatly improved.

Query error correction process can be summarized as two stages: error detecting phase and error correction phase. After a user inputs a search term, system goes into error detecting phase and search terms are checked by a certain algorithm model. If no errors are found, the system retrieves the search term, sorts search results and returns them to the user. If errors are found, it will enter the stage of the error correcting phase. System will correct search terms according to the error correcting algorithm and give correct words. Then system will retrieve them according to correct words and sort the search results and return to the users. There may be such a situation in the error detecting phase, that user type wrong search terms in the search process, but system does not check the existence of the search error and give out correct word. It is due to the error detecting phase is defective. This is a Bad Case, it will greatly affect the accuracy of error correction and user experience. In this paper, we excavate, analyze and establish mathematical models to automatically find and collect these Bad Cases, so that it can automatically adjust in the process of automatic error correction to

Manuscript received September 6, 2017.
Manuscript revised December 25, 2017.
Manuscript publicized March 26, 2018.

[†]The authors are with College of Computer Science & Technology, North China University of Technology, Beijing, China.
[††]The authors are with Beijing Key Laboratory on Integration and Analysis of Large-scale Stream Data, Beijing, 100144 China.
a) E-mail: duanjy@hotmail.com
b) E-mail: jitianxiao@outlook.com
c) E-mail: wanghaomails@gmail.com

DOI: 10.1587/transinf.2017EDP7284

improve the precision of error correction.

The work of this paper is shown as follows: Sect. 2 summarizes related work about of Bad Case mining in search engine error correction process; Sect. 3 establishes Bad Case mining model through analyzing the logs' characteristics; Sect. 4 introduces the N-gram algorithm for query error correction methods used in this paper; Bad Cases in logs is mined in Sect. 5. Then we combine the mining results and the error correction algorithm to verify the precision rate and recall rate. At last conclusions and future work are summarized.

2. Related Work

At present, research about search engine query error correction has given some results. These methods mainly use the language model of the query word to calculate the matching degree of word in context, and then determine whether the query word is wrong, to make the correct word and statistical information based on search engine query logs combined with the language model of error correction methods. But error correction research based on data mining about search engine query logs is relatively small.

According to Sullivan's statistics [2], Google as the world's largest search volume and the highest frequency of search engines. Its indexed pages are more than 8 billion while the daily processing ups to 250 million users at the end of 2004. According to the latest CNNIC statistical report shows, as of June 2014, China's search engine user scale reached 507.49 million. Google, Baidu, Sogou and other online search engines have become important network tools to daily access information.

N-gram language model is a common statistical language model, which is widely used in search engine query error correction. Mays [3] used the Trigram model in English auto-proofing, Yangsen Zhang [4] used the Bigram model in the Chinese text proofreading. They all achieved good results. Mining and analyzing the search engine query logs have entered a popular stage with large data, cloud computing and other ideas into the deep. The goal of error correction based on context information includes not only the misspelled words outside the dictionary, but also the inappropriate words in the context. This method mainly uses language model to evaluate each keyword in the query and selects the optimal combination form. The main method is noise channel model [5]. Gao et al [6] combined the noise channel model in a more general sorting scheme, allowing more flexibility to join the sorting feature. Duan et al [7] studied the real-time error correction problem in the online query input and output process. They proposed a model for this, through an unsupervised way to train a Markov model to capture the user's input behavior, and achieved some results. Craig Silverstein [8] et al analyzed the large-scale English search log, and concluded that 85% of the users only looked at the first page of the query results. Broder et al [9] pointed out that the user's query tasks, including navigation, information and things three categories. The starting point for the task of dividing the query tasks is that the different search models, parameters, and even evaluation methods for the three types of searches are also differentiated as the search category changes. Therefore, it is very important to realize the automatic classification of the retrieval categories to improve the retrieval performance and increase the credibility of the retrieval evaluation. Soyeon Park [10] analyzes the user's search behavior and shows that the query entered by the user in the same session tends to completely replace the query, rather than adding or subtracting the search term or modifying the search term. Li et al [11] counted the number of search terms in the same session and found that only one query term in the same session accounts for 70.866% of all sessions.

The research on the error correction algorithm model and the analysis of mining the search engine query logs have entered a mature stage, but the combination of the two is rare, that is applying results of mining the search engine logs to the process of query error correction. It is necessary to enhance query error correction recall rate and precision rate of the search engine in the process and provide better service to users through mining the massive search engine logs. Conditions based on above analysis are very mature.

3. Bad Case

Bad Case mining refers to the search engine did not find the error when a user's input is wrong. In this case, most of the search results cannot meet the needs of users. This kind of Bad Case mining is very difficult, because the error correction algorithm does not judge out the user's search words are incorrect, which did not give out correction words. So there is no way to compare two situations which is Before and after error correction to find them. This type of Bad Case can be found by mining useful information and analyzing the association between the user's query log records [12].

3.1 Log Feature Analysis

After a user enters the query keyword, if the search engine does not detect the error, it will not give the error correction proposal. Generally, the results of this query cannot meet the user's retrieval needs. The user will choose to re-enter or modify the query word to retrieve again, so as to get the most appropriate results. Based on analyzing user's behavior by mining logs, Shaopin Ma et al [1] concluded: if the results are not satisfied, 56.1% of users choose to modify the query terms and interact with search engine after users submitted the query. Soyeon et al [10] research on search engine users' behaviors based on mobile internet logs found: in a query session, about 25% of users reduced the number of search terms, 61% of users chosen to modify the original search terms to re-search, rather than completely replace the search terms. We found the sequence of characteristics shown in Table 1 by analyzing users' query logs. We got the Bad Case characteristics through the analysis of this sequence. Through analysis of these Bad Case features, you can build

Table 1 The example of search engine log record

Userid	Retrieve time	Search terms	Url ranking	Url order	Url
9547788631522974	18:06:38	[携程机票预定]	2	14	big5.ctrip.com/
9547788631522974	18:08:51	[携程机票预订]	1	15	flights.ctrip.com/
006886013319249151	12:07:27	[车臣战争]	2	1	v.youku.com/. . . NzgzMg==.html
006886013319249151	12:35:43	[车臣战争电影视频]	1	5	www.ku6.com/ ...GTH4iL.html
006886013319249151	12:38:47	[车臣战争电影]	26	13	video.pplive.com . . . 5793883.html
006886013319249151	12:41:57	[车臣战争视频]	1	29	v.youku.com/ . . . 0NzgzMg==.html
006886013319249151	12:44:23	[俄罗斯电影炼狱视频]	1	1	boardID=12& ID=592&page=1

a model to automatically mine such Bad Cases from logs, so as to automatically adjust the online query to make the search engine system can automatically discover and correct it when this type of Bad Case occurs, thus improving search efficiency and user experience.

The analysis of sequence of users' search logs is mainly from the following aspects:

1. The relationship between preceding words and succeeding words

(1) Similarity based on edit distance: edit distance is often used for the calculation of string similarity problems [13], [14]. The edit distance refers to the minimum number of editing operations between two strings, from one to another. Editing operations here include replacing, inserting, and deleting characters. The edit distance is smaller and the similarity is higher between the two search terms preceding and succeeding, indicating the greater the probability of the error in the preceding word. If the edit distance between the two words is larger, the similarity of the two search terms is lower, indicating that the relationship between the preceding search term and the succeeding search term is small, the probability of error of the previous word is small. For example, in Table 1, the distance between the search terms [车臣战争电影视频] and the search term [车臣战争的电影] is 3, the similarity is 50.8%, indicating that the relationship between the search terms is close. When the user entered the [车臣战争电影视频] to query for the first time, the user may not be satisfied with the result. Then he re-entered [车臣战争的电影] to search. The edit distance of the term [车臣战争视频] and the term [俄罗斯电影炼狱视频] is 7, the similarity is 20.3%, indicating that the relationship between the search terms is not big.

(2) The including relationship of the preceding word and the succeeding word: There are three cases in including relations: the preceding including the succeeding, the succeeding including the preceding and not including. Such as the term [车臣战争] and [车臣战争电影视频] is the succeeding including the preceding, the probability of error of this type is very small. [车臣战争打] and [车臣战争] is the preceding including the succeeding. This type is to delete some words from preceding word. If the difference is very small, then the probability of the word that is wrong is large.

(3) The quantitative relationship of the preceding word and the succeeding word: The number of word segments is larger than the number of word segments which is correct when users entered the wrong search term.

2. The situation of search results clicked on between the preceding word and the succeeding word:
Shaoping Ma et al [1] analyzed the users' clicked behaviors based on mining logs of search engine. They do the analysis from the query click rate, the first time clicked, the first/last click location distribution, the number of clicks within query and so on, summed up for some users tend to more clicks to get more information. So, the search engine will search according to the keyword after users input search terms. If the keyword entered by the user is wrong, the results of the search may be less or the correlation between the search result and the search term is poor, and the probability that the user clicks on search results is very small. While users modify search terms to retrieve again, the probability that users click result is large because of the greater relevance between the result and the search word.

3.2 Quantitative Model

Through the intuitive analysis of the search engine query logs, we need to establish a model to quantify the association of the sequence [12], to dig out Bad Case in the case of users' incorrect inputs.

3.2.1 Association Relationship Model

There are three categories of association between the preceding word and the succeeding word in 3.1:

1. The editing distance is quantified as follows: Generally, for calculating the similarity based on the Levenshtein distance, the formula is as follows

$$p_{sim}(s,q) = 1 - \frac{ld}{m+n} \tag{1}$$

$$p_{sim}(s,q) = 1 - \frac{ld}{\max(m,n)} \tag{2}$$

Where, ld is the Levenshtein distance between strings s and q, m and n are the length of two strings. The large the p_{sim}, the higher the similarity between the two strings. But when we assume we have three strings: S_1 = "BC", S_2 = "CD", S_3 = "EF", according to the formula (1) calculated,

$$p_{sim}(S_1,S_2) = 0.5, \quad p_{sim}(S_1,S_3) = 0.5$$

according to the formula (2) calculated,

$$p_{sim}(S_1,S_2) = 0, \quad p_{sim}(S_1,S_3) = 0$$

when only Levenshtein distance is used, the similarity between S_1 and S_2 is the same as S_1 and S_3. This is inconsistent with the actual situation. The similarity between S_1 and S_2 is greater than the similarity between S_1 and S_3 because there is a common substring "C". For strings S_4 = "$ABCD$", S_5 = "$BADC$", S_6 = "$DEFG$", the longest common substring between S_4 and S_5 is the same as S_4 and S_6. Therefore, we combine the Levenshtein distance with the longest common substring. In [13], [15], an algorithm based on improved edit distance similarity is proposed through making Bad Case that is not recalled as event Y. The algorithm contains the LD distance, the longest common substring length $LCS(s,t)$ of the two strings, and taking into account the position of the first occurrence of the unmatched character when the two strings are compared. In this paper, the calculation of edit distance is based on this improved algorithm, the formula is as follows:

$$p_{sim}(s,q) = \frac{lcs}{ld + lcs + \frac{L_m - \delta}{L_m}} \tag{3}$$

s, q is the two strings to be compared. L_m is the length of the s string, ld is the distance of Levenshtein between the two strings, δ is the position that two strings do not match first time.
According formula (3) calculated,

$$p_{sim}(S_1,S_2) \approx 0.29, \quad p_{sim}(S_1,S_3) = 0$$
$$p_{sim}(S_4,S_5) \approx 0.21, \quad p_{sim}(S_4,S_6) \approx 0.17$$

This is obviously more in line with the actual situation than using the two methods respectively.
When the edit distance of front and back search terms is $p_{sim}(s,q)$, the probability that the search term belongs to the unrecalled Bad Case is:

$$P(Y|D) = p_{sim}(s,q) \tag{4}$$

$P(Y|D)$ is the probability that the search term belongs to the unacknowledged Bad Case is when the edit distance of the search terms is D. And the higher the similarity (i.e. $p_{sim}(s,q)$) of the search terms is, the greater probability (i.e. $P(Y|D)$) that the search term belongs to the unrecalled Bad Case.

2. Two attributes that the former word contains the word and the number of before and after words is not independent, so we put them together to quantify. And then we convert the two attributes into a two-tuple to represent $Q = (n,i)$. n and i has the following meanings:

(1) i means that the include relationship of the front word and the back word. The former includes the following word as −1, and the latter contains the first word as 1.
(2) n indicates the number of words before and after changes.

The probability that the search term belongs to an unrecalled Bad Case can be expressed as:

$$P(Y|Q) = \frac{|n+i|}{LD_1 + LD_2} \tag{5}$$

LD_1 denotes the segmentation number of the former search term and LD_2 denotes the segmentation number of the latter search term.

3.2.2 Model of Clicked Record

The result that the user want may not return when he enters the wrong search term. So the number of users clicked very little in this case. On the other hand, if the user has a small number of clicking on a query, the greater probability that the input search term will be wrong. Note the number of users' clicks for C, there are

$$P(Y|C) = 1 - \frac{1}{\log_2(C+2)} \tag{6}$$

In summary, to determine whether a search term belongs to the Bad Case need to synthesize the above factors to give a comprehensive score in order to achieve better results, the integrated model is as follows:

$$I = \alpha_1 \times P(Y|D) + \alpha_2 \times P(Y|Q) + \alpha_3 \times P(Y|C) \tag{7}$$

α_1, α_2, and α_3 denote the corresponding weights of the model, and they can be adjusted according to the mining effect. The sum of these weights is 1,

$$\alpha_1 + \alpha_2 + \alpha_3 = 1 \tag{8}$$

4. N-gram Model

N-gram model is a commonly used algorithm for natural language processing. For Chinese, it is also called Chinese language model.

The N-gram model assumes that the probability of occurrence of any word is only related to words in front of it:

$$P(s) = \prod_{i=1}^{n} p(q_i|q_{i-n+i} \cdots q_{i-1}) \tag{9}$$

The above formula is calculated through a large number of corpus. The capacity of the corpus is larger, the frequency is closer to its probability. Under the premise of large-scale corpus, N-gram model can be expressed as:

$$p(q_i|q_{i-1}) = \frac{feq(q_i q_{i-1})}{feq(q_{i-1})} \tag{10}$$

$feq(q_i q_{i-1})$ indicates the frequency at which $q_i q_{i-1}$ is present in the corpus, and $feq(q_{i-1})$ indicates the frequency at which q_{i-1} is present in the corpus.

Due to the limited size of the corpus, many reasonable relationships do not appear in the corpus, so there will be sparse data ("zero probability" problem). Data smoothing technology is usually used to eliminate data sparse phenomenon without expanding the size of the corpus, So that the probability distribution of the model parameters tends to be uniform. And it can improve the accuracy of the whole model.

A lot of data smoothing techniques have been proposed. Such as Additive smoothing, Add-one smoothing, Add-delta smoothing, Witten-Bell smoothing, Good-Turing smoothing, Jelinek-Mercer smoothing, Church-Gale smoothing, Katz smoothing and so on. This paper applies Additive smoothing smoothing technology [11], [16], which is calculated as follows:

$$P_{additive}(q_n|q_{n-k+1} \cdots q_{n-1}) = \frac{\delta + feq(q_{n-k+1} \cdots q_n)}{\delta|V| + feq(q_{n-k+1} \cdots q_{n-1})} \tag{11}$$

$0 \le \delta \le 1$, V is the total number of different words in the corpus. For the binary grammar model, we take $\delta = 1$, the final binary grammar model is calculated as:

$$P_{additive}(q_i|q_(i-1)) = \frac{1 + feq(q_i, q_{i-1})}{|V| + feq(q_{i-1})} \tag{12}$$

5. Experimental Process and Result Analysis

In order to verify the impact of the Bad Case mining model on the error correction accuracy of the search engine, we need to experiment with the evaluation index of the search engine, such as the recall rate and the precision rate. In this paper, the search engine used the N-gram algorithm model to automatically correct the error, and combined with the Bad Case mining model. Finally, we gave out the analysis of Bad Case mining model on the impact of error correction.

we implemented a search engine based on Sogou Lab's query logs, combined with the open source Java-implemented search engine Nutch and the open-source standalone enterprise search engine server Solr. Nutch mainly acts as a crawler and Solr acts as indexer and retriever. Through this system, we mainly prove that it can really improve the performance of the original error correction model after using Bad Case mining.

5.1 Data Set

The search engine log used in this experiment is the query log file obtained from Sogou's lab. Firstly, we extracted representative records to get the formation of experimental query logs. The number of records is 1.7 million, the structure shown in Table 1. Log records of each field description, as shown in Table 2:

The dictionary used in the experiment contains 104041 phrases with their pinyin (three or more form of abbreviation), the structure of the dictionary file as shown in Table 3:

We matched entries and log library in the dictionary by word, pinyin, with phonetic Pinyin, Pinyin shorthand (three or more form a short note), query word frequency to get the total number of times that words appear in the log library. The size of corpus is 106246 records. Corpus file structure as shown in Table 4:

For corpora used in language model training, we also use the internet corpus provided by Sogou Lab. It is available from Sougou Lab's website. We used about 1GB of text after preprocessing.

5.2 Experimental Process and Results

In this paper, the Bi-Gram model is established by selecting the n value as 2 according to the N-gram model described in 4. Applying Additive smoothing technology to adjust the "data sparse" phenomenon in the model. The model is trained by the above corpus, and finally the error correction effect of the model is tested by different test sets. After analyzing and comparing the experimental data, the recall rate and precision rate of the error correction model are shown in Fig. 1.

The test set size is X, the value of X is 100K, 300K, 500K, 700K, 900K, 1100K, 1300k, 1500k, 1600k. With the increase of test set scale, it can be seen that recall rate and precision tends to be stable from the experimental data. And search engine performance is also in an acceptable range.

In order to optimize the effect of error correction, the Bad Case mining model proposed in this paper is used to optimize the error correction process of the N-gram model. A user (userid) in logs may correspond to multiple queries or sessions, and the topics within each session may not be the same. Such as the user asked the "哄抢救灾物资" and "汶川地址图片" in a day of two time periods is the two sessions. It is possible in the same session, that is, the browser did not close when the new topic content is queried. Such as

Table 2 Search engine log description

Log records	Record Description
Userid	The user id represents a specific user
time	The search time indicates the time the user has made the search
query	The input query word indicates the search term that the user entered
urlRank	The rank of the url that the user clicked in the search results.
urlSeq	The order number of the url that the user clicked
url	The url that the user clicked

Table 3 Dictionary file structure

Word number	Word	Pinyin	Pinyin with tone	Pinyin abbreviation
945	白面书生	baimianshusheng	bai2mian4shu1sheng1	bmss
978	白手起家	baishouqijia	bai2shou3qi3jia1	bsqj
3986	不可同日而语	buketongrieryu	bu4ke3tong2ri4er2yu3	bktrey

Table 4 Corpus file structure

Word number	Word	Pinyin	Pinyin with tone	The number of occurrences in the log
203	安全理事会	anquanlishihui	an1quan2li3shi4hui4	73040
204	安全门	anquanmen	an1quan2men2	226164
205	报告文学	baogaowenxue	bao4gao4wen2xue2	345656

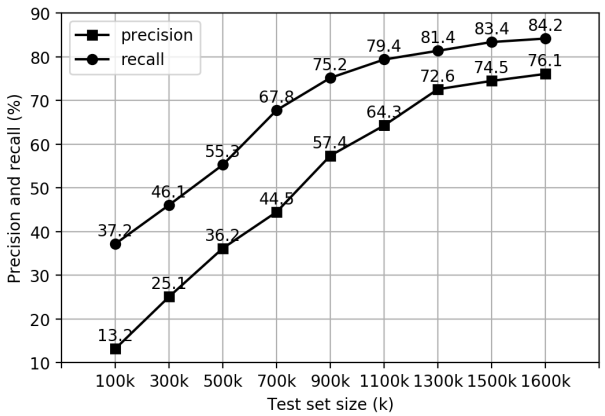


Fig. 1 The recall and precision of N-gram error correction model

a user queried "林青霞的婚礼" after he did not close the browser, then he queried "福利彩票". These problems bring difficulties to the application of the Bad Case mining model. So the first to handle the log records, get a user's set with the same theme, and then use the Bad Case model to mine the set.

We get the best values of α_1, α_2, α_3 through Grid Search. The search step is 0.01. By searching, the optimal parameters we get are $\alpha_1 = 0.52$, $\alpha_2 = 0.10$, $\alpha_3 = 0.38$. The main parameters and results are shown in Table 5.

The processing of the log is summarized as follows: first we handled the logs to get no repeat records of the user uerid. A user may have more than one query topic in a single session, so these topics will be grouped according to its theme group. Use the Bad Case mining model to mine the Bad Case in each group that is the same theme, to find out records belonging to the Bad Case. The specific process is shown in Fig. 2.

Table 5 Model parameters

No.	α_1	α_2	α_3	Precision	Recall
1	0.01	0.01	0.98	82.86%	89.12%
2	0.01	0.02	0.97	82.85%	89.25%
3	0.01	0.03	0.96	82.74%	89.16%
···	0.50	0.01	0.49	82.90%	89.18%
···	0.98	0.01	0.01	82.89%	89.13%

The results of mining are applied to the error correction model to realize the optimization of error correction model. The recall rate and precision rate of the experimental data are as follows: Fig. 3.

In order to compare the precision rate and recall rate of the N-gram error correction model adding Bad Case, we compare Fig. 1 and Fig. 3 to get Fig. 4.

5.3 Analysis of Results

Comparing and analyzing experimental data, we can draw the following conclusions:

1. The size of the test set affects the recall rate and precision rate of error correction. The test set is larger, the recall rate and precision rate is higher. But to a certain order of magnitude, the impact of the test set on the precision of error correction will gradually decrease. There are more information in the larger the test set. But to a certain limit, the information has reached saturation.

2. Although the traditional error correction model in the error correction accuracy and recall rate has reached a certain efficiency, but not no other way. You can see in the use of Bad Case mining model, error correction precision rate and recall rate has been improved.

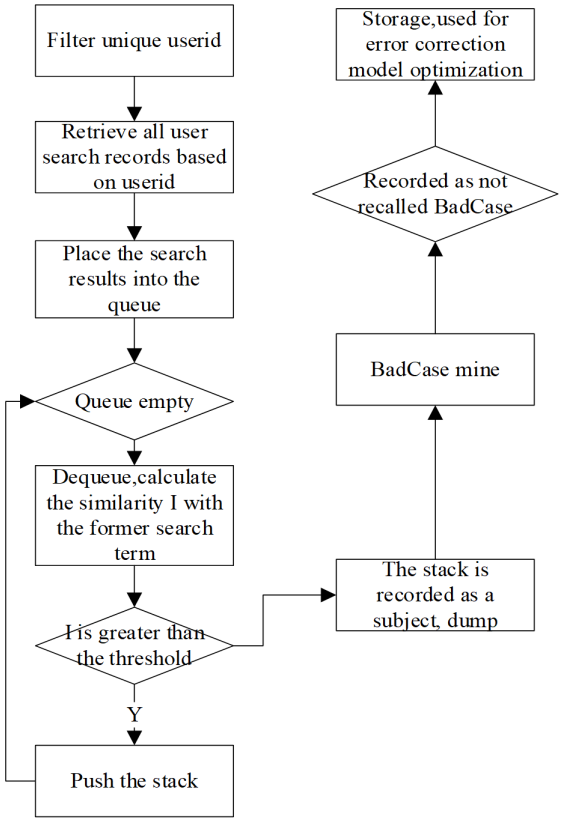


Fig. 2 Processing flowchart

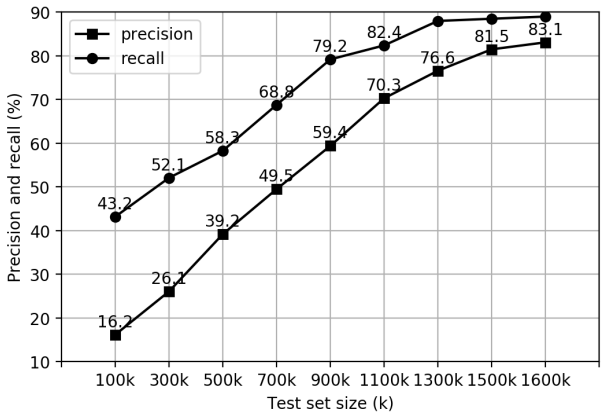


Fig. 3 The recall and precision of N-gram error correction model based on BadCase mining

6. Conclusion and Outlook

In this paper, we presents a query error correction method based on Bad Case mining. Through the analyzing and mining the Bad Case in the process of query error correction, a model is established to optimize the query error correction process. The precision, recall rate of the query error correction and the user search experience are improved. But there are some shortcomings. One is the limitation of the data. This article is based on Sogou search engine logs within a day. Secondly, quantitative factors are still not comprehensive. In our future work, we need to improve the above shortcomings, making accuracy and efficiency of the model of error correction to be further improved.

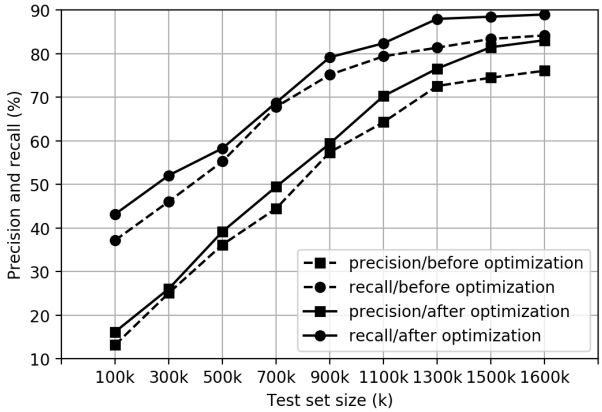


Fig. 4 The recall and precision of N-gram error correction model based on BadCase mining

Acknowledgements

This work was supported by the National Natural Science Foundation of China (61672040) and the North China University of Technology Startup Fund.

References

[1] R. Cen, Y. Liu, M. Zhang, R.U. Liyun, and M.A. Shaoping, "Search engine user behavior analysis based on log mining," Journal of Chinese Information Processing, 2010.

[2] D. Sullivan, "Search engine size wars v erupts," Search Engine Watch, vol.11, 2004.

[3] E. Mays, F.J. Damerau, and R.L. Mercer, "Context based spelling correction," Information Processing & Management, vol.27, no.5, pp.517–522, 1991.

[4] Y. Zhang, "The structuring method of correcting knowledge sets and the producing algorithm of correcting suggestion in the chinese text proofreading system," Journal of Chinese Information Processing, 2001.

[5] X. Sun, J. Gao, D. Micol, and C. Quirk, "Learning phrase-based spelling error models from clickthrough data," Meeting of the Association for Computational Linguistics, pp.266–274, 2010.

[6] J. Gao, X. Li, D. Micol, C. Quirk, and X. Sun, "A large scale ranker-based system for search query spelling correction," International Conference on Computational Linguistics, pp.358–366, 2010.

[7] H. Duan and B.J. Hsu, "Online spelling correction for query completion," International Conference on World Wide Web, pp.117–126, 2011.

[8] C. Silverstein, H. Marais, M. Henzinger, and M. Moricz, "Analysis of a very large web search engine query log," ACm SIGIR Forum, vol.33, no.1, pp.6–12, ACM, 1999.

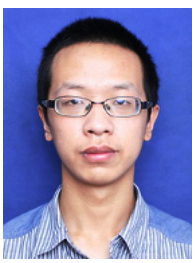
[9] A. Broder, "A taxonomy of web search," Acm Sigir Forum, vol.36, no.2, pp.3–10, 2002.

[10] S. Park, J.H. Lee, and H.J. Bae, "End user searching: A web log analysis of naver, a korean web search engine," Library & Information Science Research, vol.27, no.2, pp.203–221, 2005.

[11] W. Li, Y. Yang, H. Jin, L. Tan, M. Ren, and S. Zhang, "Search engine log based user behavior analysis," Energy Procedia, vol.13, pp.5082–5091, 2011.

[12] M. Delgado, M.D. Ruiz, and D. Sánchez, "Studying interest measures for association rules through a logical model," International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol.18, no.1, pp.87–106, 2010.
[13] Z.-P. Zhao, Z.-M. Yin, Q.-P. Wang, X.-Z. Xu, and H.-F. Jiang, "An improved algorithm of levenshtein distance and its application in data processing," Journal of Computer Applications, vol.29, no.2, pp.424–426, 2009.
[14] T. Akutsu, D. Fukagawa, and A. Takasu, "Approximating tree edit distance through string edit distance," Algorithmica, vol.57, no.2, pp.325–348, 2010.
[15] H. Jiang, A.Q. Han, M.J. Wang, Z. Wang, and W.U. Yun-Ling, "Solution algorithm of string similarity based on improved levenshtein distance," Computer Engineering, vol.40, no.1, pp.222–227, 2014.
[16] Q. Chen, M. Li, and M. Zhou, "Improving query spelling correction using web search results," EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28–30, 2007, Prague, Czech Republic, pp.181–189, 2007.

Jianyong Duan is a professor, born in 1978.10. He graduated from Department of computer science, Shanghai Jiao Tong University by 2007.12. His major research field including natural language processing and information retrieval.



Tianxiao Ji is a master in College of Computer Science and Technology, North China University of Technology. His major research field is information retrieval.

Hao Wang received the Ph.D. degree in Computer Application Technology from Tsinghua University in 2013. He is now an assistant professor in College of Computer Science and Technology, North China University of Technology. He has published more than 20 papers in international conferences and journals. His research interests include machine learning and data analysis.