Combining Statistical and Feature-based Approaches for Query Error Correction (Postprint)

Authors: Duan Jianyong, Guan Xiaolong

Date: 2017-10-11T00:00:00+00:00

Abstract

[Objective] To improve the accuracy and recall rate in search engine query error correction and enhance user retrieval experience. [Method] We propose a query error correction model that combines statistical and feature-based approaches. A confusion set generation model is established to generate corresponding confusion sets for user-input query keywords; a confusion set ranking model is then built to rank entries in the confusion set and select the optimal entry for comparison with the user-input query keyword, thereby achieving error detection and correction. [Results] Experimental results demonstrate that the model achieves favorable performance in search engine queries, with accuracy and recall rates reaching 92.2% and 95% respectively on a 110k test set, which represents improvements of 13.6% and 8.3% respectively over the N-gram error correction model. [Limitations] The confusion set generation rules in this model are limited, and training the model requires substantial computational resources. [Conclusion] This model can improve the accuracy and efficiency of search engine queries, thereby enhancing user retrieval experience.

Full Text

A Query Correction Method Based on the Combination of Statistics and Features

Duan Jianyong, Guan Xiaolong
(College of Computer Science, North China University of Technology, Beijing 100144)

Abstract

[Objective] This study aims to improve the precision and recall of query correction in search engines, thereby enhancing user retrieval experience. **[Methods]** We propose a query correction model that combines statistical and linguistic

features. The model first establishes a confusion set generation model to create candidate sets for user input queries. Then, a confusion set ranking model evaluates and ranks these candidates, selecting the optimal term to compare against the original query, thereby achieving error detection and correction. [**Results**] Experimental results demonstrate that the model performs effectively for search engine query correction, achieving precision and recall rates of 92.2% and 95% respectively on a test set of 110k queries. This represents improvements of 13.6% in precision and 8.3% in recall over the N-gram correction model. [**Limitations**] The model' s confusion set generation rules are limited, and training the model requires substantial computational resources. [**Conclusions**] The proposed model can improve the precision and efficiency of search engine queries and enhance user search experience.

Keywords: Query correction; Confusion sets; N-gram model; N-gram similarity; Edit distance; Click frequency
Classification Number: TP391; G35

1. Introduction

With continuous advancements in Internet technology, users demand higher accuracy and convenience from search engines, posing greater challenges for query correction technologies. Research on user query intent recognition has revealed that users often do not have very clear or precise targets when using search engines. For computer systems, correctly identifying user queries, automatically detecting and correcting erroneous keywords, and providing satisfactory results have become important aspects of search engine technology research.

This paper investigates the process and methods of search engine query correction, proposing a correction approach that combines statistical and linguistic features. We establish a model and validate its effectiveness through experiments, demonstrating improved fault tolerance and usability of search engines while enhancing user search experience.

Foreign research on spelling correction began earlier than domestic efforts. For English text proofreading, word segmentation is unnecessary as words are separated by spaces. Only individual words require spelling checks, typically using edit distance to calculate similarity between words, combined with statistical information from the text to identify misspellings. For example, Senger et al. analyzed spelling errors and their characteristics in drug information system queries to correct them.

Chinese expression uses Chinese characters and has unique linguistic properties. Issues such as synonyms, homophones, and polyphonic characters in Chinese information processing often complicate Chinese query correction. Currently, there are two common approaches for Chinese query correction: dictionary-based methods and text statistics-based methods. Dictionary-based methods require building a massive dictionary and using string matching to query it. While these methods achieve high accuracy, they require constant dictionary

maintenance. With the rapid development of the Internet and natural language, new words and online popular terms emerge constantly, making it difficult to meet current query correction efficiency by merely expanding dictionary coverage. In contrast, text statistics-based methods leverage large-scale corpora to mine and analyze inherent linguistic relationships and features from numerous examples, incorporating them into statistical models without relying on dictionaries, yet still achieving good correction efficiency.

Current research on query correction focuses on using web data and query logs to obtain patterns of user queries and errors, applying them to correction tasks. For instance, Strohmaier et al. used search engine query logs to acquire user intent, while Roy et al. discovered and understood user intentions through rigorous analysis of large query logs, achieving good results. Subramaniam et al. combined edit distance with language statistical models based on query logs for query correction. Using search engine query logs combined with text statistical features for user query correction has become an important research direction.

This paper proposes a query correction model based on the combination of statistics and features. We establish a confusion set generation model to model user input queries and generate corresponding confusion sets. The model treats all user input as unreliable but not useless. The target query string (the user' s true intended query) can be viewed as being generated from the input string through the confusion set generation model, meaning the confusion set generated from the input string contains the target query string. We then establish a confusion set ranking model to sort candidate strings in the confusion set, filter out the best candidate, and compare it with the original string to obtain the correction result. This process combines error detection and correction into a single stage, and the selected optimal candidate term has high accuracy. The entire correction process is shown in Figure 1 [Figure 1: see original paper].

2. Model Structure

The model structure involves two key steps: generating confusion sets based on user input queries, and comprehensively evaluating and ranking candidates within these sets. The generated confusion sets must ensure that the user' s target query string is included, while the set size should not be too large—avoiding impossible error terms to prevent excessive computation. Implementing confusion set ranking is a critical and important step, as it evaluates and selects candidate strings. After ranking, the highest-scoring candidate is selected as the optimal result and compared with the user input string to produce the correction result. This process requires knowledge of linguistics, statistics, and extensive data mining and analysis to ensure optimal candidate selection.

3. Confusion Set Generation Model

The confusion set generation model is crucial to the entire correction process and must satisfy two conditions: include as many possible error terms as possible, and exclude as many impossible error terms as possible. Meeting the first condition ensures correct correction results from the confusion set and improves correction accuracy; meeting the second condition prevents interference from impossible error terms, avoids massive computation, and improves correction efficiency.

User input query keywords are unreliable, so confusion sets cannot be generated directly from query strings as units. Instead, we first segment each input query string, generate candidate word sets for each segment, and then cross-combine these candidates based on the original segmentation to form confusion sets. Analysis of search engine query logs reveals that 93.15% of queries contain no more than 3 segments, keeping the candidate phrase matrix within an acceptable processing range.

User input keywords have two important characteristics: errors are local at the character/word level, and different input methods have corresponding error patterns. For a given user input term, its possible error set (confusion set) can be effectively generated through predefined rules.

Wang Siyu et al. adopted a confusion set and context feature-based approach for text automatic proofreading in their CSSCI-based system, establishing confusion sets for characters and words based on Chinese input methods. This paper' s confusion sets primarily generate segment candidates based on phonetic similarity, then cross-combine them to form candidate sets (confusion sets). The specific process is as follows: assume input , where q_i represents the i-th segment. For each q_i, candidates are generated according to certain rules (referencing literature [9]), denoted as nkw representing the k-th candidate of the n-th segment. These are then cross-combined to form the confusion set.

The generation rules mainly include:

(1) Polyphonic and Homophonic Cases
When using search engines, users primarily rely on input methods to manually select appropriate characters. This process lacks syllable information, and pinyin input methods have high code overlap rates, leading to polyphonic and homophonic character selection errors. For example, polyphonic cases include "大夫 (doctor)" vs. "大夫 (ancient official title)" ; homophonic cases include inputting "jishu" which may produce "技术 (technology)" , "级数 (series)" , or "奇数 (odd number)" . Such selection-induced errors are common.

(2) Abbreviation Cases
Modern input methods facilitate users by allowing pinyin abbreviations. For example, user input "mdl" may represent "麦当劳 (McDonald' s)" , "没电了 (out of power)" , or "矛盾论 (On Contradiction)" .

(3) Syllable Ambiguity Cases
When inputting query keywords, some words have syllable ambiguity. For example, "xianren" can be segmented as "xian ren" producing "仙人 (immortal)" or "线人 (informant)" , or as "xi an ren" producing "西安人 (Xi' an people)" .

(4) Near-phonetic Cases
Near-phonetic cases include similar initials and finals. Commonly confused initials include "s" and "sh" , "r" and "l" , "l" and "n" , "f" and "h" . Commonly confused finals include "ui" and "ei" , "an" and "ang" , "on" and "ong" .

Based on these candidate generation rules, segment candidate sets are generated and cross-combined to form query string confusion sets. The confusion set generation process is shown in Figure 2 [Figure 2: see original paper]. The cross-combination process of segment candidate terms is shown in Figure 3 [Figure 3: see original paper], where the i-th segment of the string is represented, W_i represents the original input segment, and iw represents the i-th candidate of the first segment.

3.2 Confusion Set Ranking Model

Selecting the optimal candidate term from the confusion set generated from user input is essentially an evaluation and selection process. To achieve optimal correction results, effective scoring and ranking are required. By mining query string features to describe the confusion set ranking model, effective ranking results can be obtained to select the optimal candidate and produce correction results. Therefore, constructing an effective ranking model is crucial, and model parameters must be determined through training on large-scale corpora.

To effectively reflect candidate string context features, we introduce the widely used N-gram language model from natural language processing. Current research on user behavior analysis in large-scale search engine logs has yielded valuable insights. Yu Huijia et al. analyzed Sogou search engine query logs over one month, examining query length, frequency, session count, and user click behavior to predict user intent. This demonstrates that click features in query logs contain valuable information about user search purposes, which we also incorporate. Additionally, considering morphological similarity between Chinese character strings through word shape and edit distance comparison, we establish a candidate ranking model combining N-gram model, query term click-through rate, word shape similarity, and edit distance to rank candidate terms and obtain optimal results from the candidate set.

(1) N-gram Model
The N-gram model is a commonly used algorithm in natural language processing. For Chinese, it is also called the Chinese Language Model (CLM). Zhang Yangsen et al. used Bigram models for Chinese text proofreading, and also utilized trigram and context dependency analysis for automatic Chinese error detection, achieving certain results.

From a statistical perspective, a sentence s in natural language is composed of a specific sequence of words $q_1 q_2$. According to the chain rule, the probability of sentence s appearing is:

$$p(q_1)p(q_2|q_1)p(q_3|q_1q_2) \qquad p(q_i|q_{i-1} \qquad p(q_n|q_{n-1}$$

One could assume that each word' s probability depends on all preceding words. However, computationally this is infeasible due to excessive calculation. The N-gram model assumes that a word' s appearance probability depends only on the preceding $n-1$ words:

$$p(q_i|q_{i-n+1}$$

Formula (2) is derived from extensive corpus statistics and calculations. Larger corpora yield frequency values closer to true probabilities. Therefore, with large-scale corpora, the N-gram model can be expressed as:

$$p(q_i|q_{i-n+1} \qquad = \frac{feq(q_{i-n+1} \qquad q_i)}{feq(q_{i-n+1} \qquad)}$$

where $feq(q_{i-n+1} \qquad q_i)$ represents the co-occurrence frequency of $q_{i-n+1} \qquad q_i$ in the corpus, and $feq(q_{i-n+1} \qquad)$ represents the frequency of q_{i-n+1} in the corpus.

Due to limited corpus size, many reasonable collocations may not appear, causing data sparsity ("zero probability" problem). Without expanding the corpus, data smoothing techniques can adjust the model to eliminate zero-probability parameters and make the probability distribution more uniform, improving overall accuracy.

Many smoothing techniques exist, such as Additive Smoothing, Add-one Smoothing, Add-delta Smoothing, Witten-Bell Smoothing, Good-Turing Smoothing, Jelinek-Mercer Smoothing, Church-Gale Smoothing, and Katz Smoothing. This paper applies Additive Smoothing technology, calculated as:

$$additive(q_i|q_{i-n+1} \qquad = \frac{feq(q_{i-n+1} \qquad q_i) + \delta}{feq(q_{i-n+1} \qquad) + \delta|V|}$$

where $\delta \leq\leq$, and V represents the total number of distinct words in the corpus.

For the bigram model, we set $\delta = 1$, yielding the final bigram model formula:

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

(2) Query Term Click-Through Rate
Click record features can measure a candidate string' s query frequency, defined as the total number of times the query term is submitted within a period. Query frequency serves as an important heuristic for understanding user search behavior. Chen et al. utilized log click records to analyze user preferences, improving query correction efficiency. Wan Fei et al. used click-through rates from log records to study user search behavior and predict potential needs. This paper uses query term frequencies from the log database. Since input strings are segmented, for candidate strings composed of multiple words, we use the average frequency of constituent words.

(3) N-gram Similarity
This paper needs to calculate morphological similarity between user query strings and candidate strings. N-gram similarity addresses this effectively. It uses N-gram 思想 to combine word similarities into N-word N-gram similarity, then calculates similarity between different length N-grams to obtain overall string similarity. The classic application is the BLEU method in machine translation evaluation. Referencing this approach, we determine N-gram similarity between query and candidate strings by calculating the proportion of matching N-gram tuples out of total candidate N-gram tuples:

$$sim\_ngram(c,q) = \frac{ngram_c \cap ngram_q}{ngram_c} = \frac{\sum count(ngram)}{\sum count(ngram)}$$

where the numerator represents the number of matching N-gram tuples between query and candidate strings, and the denominator represents the total number of N-gram tuples in the candidate string.

(4) Edit Distance
The Levenshtein Distance (LD) algorithm is frequently used for string similarity calculation, with wide applications in text comparison and information processing. Edit distance refers to the minimum number of editing operations required to transform one string into another, including substitution, insertion, and deletion.

Recent improvements to edit distance algorithms for string similarity have achieved significant progress. Liang et al. treated entire records as strings to judge similarity through edit distance. The basic formula for calculating string similarity based on edit distance is:

$$\frac{max(m,n) - ld}{max(m,n)}$$

where ld is the edit distance between two strings, and m, n are their lengths. However, this formula lacks universality.

Zhao Zuopeng et al. proposed an improved edit distance similarity algorithm incorporating LD, longest common substring length $LCS(s,t)$, and the position δ of the first mismatched character. This paper adopts this improved algorithm:

$$sim(s,q) = \frac{mL - ld + LCS(s,q)}{mL + \delta}$$

where s, q are the compared strings, mL is the length of string s, ld is the edit distance, and δ is the position of the first mismatched character.

(5) Confusion Set Ranking Model
For generated candidate strings, we need to compare the original string with the best candidate from the ranked list to obtain correction results. The ranking model must combine the above factors for comprehensive scoring. The model is:

$$overall\_eval(s,q) = \lambda_1 \cdot \text{N-gram}(s,q) + \lambda_2 \cdot \text{CTR}(s,q) + \lambda_3 \cdot sim\_ngram(s,q) + \lambda_4 \cdot \text{EditDistance}(s,q)$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ represent weights for N-gram model, query term click-through rate, word shape similarity, and edit distance features respectively.

4. Experiments and Analysis

4.1 Experimental Data The query logs used in experiments were obtained from Sogou Labs. After denoising (removing special characters, typos, meaningless characters, and duplicate records), we extracted representative records, assigned IDs and pinyin annotations, forming a query log of 500,000 records. The structure is shown in Table 1 .

The experimental dictionary contains 104,041 phrases with pinyin (simplified for phrases of three or more characters). The dictionary structure is shown in Table 2 . Dictionary entries were organized by word, pinyin, tonal pinyin, pinyin abbreviation (for three+ character words), and query term frequency, then matched with the log database to obtain total occurrence counts. Query term frequency information came from Sogou Labs' word frequency statistics, forming a corpus of 106,246 records. The corpus structure is shown in Table 3 .

The training set was derived from query logs, yielding 110,000 training records from the 500,000 log records. The training set was used to determine the optimal parameters $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ for the confusion set ranking model. The training set structure is shown in Table 4 .

The test set was generated by selecting common queries from log records, randomly choosing words that appear in the corpus, generating candidate terms for these words, replacing correct words to form <error term, correct term> pairs. The test set structure is shown in Table 5 .

4.2 Evaluation Metrics Correction system evaluation typically uses recall and precision as standards to judge model quality. This experiment employs these two metrics:

$$Recall = \frac{\text{Number of non-empty terms returned by correction system}}{\text{Total number of terms in keyword test set}}$$

$$Precision = \frac{\text{Number of correctly identified error terms}}{\text{Total number of terms in keyword test set}}$$

4.3 Experimental Process and Results Using query logs (training set), corpus, and dictionary combined with the proposed correction model from Section 3, we obtained optimal model parameters and validated correction effectiveness on the test set.

The corpus and training set were used to determine the specific model expression and optimal values for $\lambda_1, \lambda_2, \lambda_3, \lambda_4$. The test set then validated correction effectiveness under these parameters. Test terms were read, segmented, and candidates were generated for each segment according to the confusion set generation model. These were cross-combined to form test term confusion sets, which were then evaluated and ranked using the confusion set ranking model from N-gram features, click frequency features, word shape similarity features, and edit distance features (see formula (1)). The highest-scoring candidate was compared with the test term; if they matched, correction succeeded, otherwise it failed.

The entire experimental process consists of two steps: (1) obtaining optimal model parameters from the training set (Figure 4 [Figure 4: see original paper]); (2) validating correction effectiveness on the test set (Figure 5 [Figure 5: see original paper]).

The obtained model parameters and correction results are shown in Table 6 . The parameter set maximizing correction precision was selected as optimal: $\lambda_1 = 0.78$, $\lambda_4 = 0.01$, $\lambda_2 = 0.20$, $\lambda_3 = 0.01$. Precision and recall were tested on six test sets of sizes 10k, 30k, 50k, 70k, 90k, and 110k. The results are shown in Figure 6 [Figure 6: see original paper].

The results show that the statistical features considered in our query correction model achieve good precision and recall, demonstrating the model' s feasibility and effectiveness. We compared our model with approaches considering only single features (N-gram, N-gram similarity, edit distance) by setting $\lambda_1 = 1$, $\lambda_3 = 1$, or $\lambda_4 = 1$ individually. The precision results across different test sets are shown in Figure 7 [Figure 7: see original paper], revealing that considering only single statistical features yields lower precision.

Chen Zhipeng et al. proposed a method using N-gram statistical models based on contextual statistical information for automatic query checking and correction

in search engines. Their results are shown in Figure 8 [Figure 8: see original paper].

From the experimental statistics, we conclude: (1) Figures 6 and 7 show that considering only single statistical features (N-gram model, edit distance, N-gram similarity) yields lower correction precision. (2) Comparing Figures 6 and 8 reveals that richer contextual statistical information in the correction model produces better results. The difference between our method (Figure 6) and the N-gram statistical model (Figure 8) is significant: our model improves precision and recall by 13.6% and 8.3% respectively at the maximum test set size of 110k. (3) Comparison across Figures 6, 7, and 8 demonstrates that our model is reasonable and effective. Combining various statistical features achieves ideal and stable precision and recall values, which improve as test set size increases and more statistical features become available. (4) Comparison between Figures 7 and 8 shows experimental error within an acceptable 4%-6% range.

5. Conclusion

This paper proposes a query correction model combining statistics and features. By analyzing statistical features of input strings and integrating N-gram models, click-through rates, N-gram similarity, and edit distance, we form confusion sets for input strings and rank candidates based on these features, comparing the top candidate with the input string to obtain correction results. Experiments show that model precision and recall are affected by corpus size, with larger corpora yielding better performance. However, limitations exist: confusion set generation rules are limited (only four cases considered), and model training requires substantial computation. Future research should address these limitations to further improve correction precision and efficiency.

References

[1] Luo Cheng, Liu Yiqun, Zhang Min, et al. Query Recommendation Based on User Intent Recognition[J]. Journal of Chinese Information Processing, 2014, 28(1): 64-72.

[2] Jiang Hua, Han Anqi, Wang Meijia, et al. Solution Algorithm of String Similarity Based on Improved Levenshtein Distance[J]. Computer Engineering, 2014, 40(1): 222-227.

[3] Senger C, Kaltschmidt J, Schmitt S P W, et al. Misspellings in Drug Information System Queries: Characteristics of Drug Name Spelling Errors and Strategies for Their Prevention[J]. International Journal of Medical Informatics, 2010, 79(12): 832-838.

[4] Hu Xiaoqing. The Examples Analysis of Chinese-Error Correction Function in Search Engines[J]. Library and Information Service Online, 2008(1): 1-6.

[5] Zhang Yangsen, Cao Yuanda, Yu Shiwen. A Hybrid Model of Combining Rule-based and Statistics-based Approaches for Automatic Detecting Errors in

Chinese Text[J]. Journal of Chinese Information Processing, 2006, 20(1): 1-7, 55.

[6] Strohmaier M, Kroll M. Acquiring Knowledge About Human Goals from Search Query Logs[J]. Information Processing and Management, 2012, 48(1): 63-82.

[7] Roy R S, Katare R, Ganguly N, et al. Discovering and Understanding Word Level User Intent in Web Search Queries[J]. Web Semantics: Science, Services and Agents on the World Wide Web, 2015, 30: 22-38.

[8] Subramaniam L V, Roy S, Faruquie T A, et al. A Survey of Types of Text Noise and Techniques to Handle Noisy Text[C]. In: Proceedings of the 3rd Workshop on Analytics for Noisy Unstructured Text Data, Barcelona, Spain. New York, NY, USA: ACM, 2009: 115-122.

[9] Wang Yongjing. The Research on the Automatic Proofreading Algorithm of Recognition Flow[D]. Shanghai: Shanghai Jiaotong University, 2008.

[10] Yu Huijia, Liu Yiqun, Zhang Min, et al. Research in Search Engine User Behavior Based on Log Analysis[J]. Journal of Chinese Information Processing, 2007, 21(1): 109-114.

[11] Wang Siyu, Shao Bo. The Construction and Implementation of Text Automatic Proofreading System Based on CSSCI[J]. Library Work in Colleges and Universities, 2014, 34(6): 50-54.

[12] Zhang Yangsen, Ding Bingqing. Automatic Errors Detecting of Chinese Texts Based on the Bi-neighborship[J]. Journal of Chinese Information Processing, 2001, 15(3): 36-43.

[13] Chen Q, Li M, Zhou M. Improving Query Spelling Correction Using Web Search Results[C]. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Prague, Czech Republic. 2007: 181-189.

[14] Wan Fei, Zhao Xi, Liang Xun, et al. Search Behavior Study Based on the Mobile Search Log[J]. Journal of Chinese Information Processing, 2014, 28(2): 144-150.

[15] Wang Jinquan, Liang Maocheng, Yu Hongliang. A Measure of Sentence Similarity Based on N-grams and Vector Space Model[J]. Modern Foreign Languages, 2007, 30(4): 405-413.

[16] Liang J, Chen L, Mehrotra S. Efficient Record Linkage in Large Data Sets[C]. In: Proceedings of the 8th International Conference on Database System for Advanced Application. IEEE Computer Society, 2003: 137-146.

[17] Zhao Zuopeng, Yin Zhimin, Wang Qianping. An Improved Algorithm of Levenshtein Distance and Application in Data Processing[J]. Journal of Computer Applications, 2009, 29(2): 424-426.

[18] Shao Yanqiu. Some Information Retrieval Terms[J]. Terminology Standardization and Information Technology, 2009(4): 9-43.

[19] Chen Zhipeng, Lv Yuqin, Liu Huasheng, et al. Chinese Spelling Correction in Search Engines Based on N-gram Model[J]. Journal of China Academy of Electronics and Information Technology, 2009, 14(3): 323-326.

Author Contributions

Duan Jianyong: Proposed research ideas, designed research plan, and developed methods including confusion set generation and ranking models.
Duan Jianyong, Guan Xiaolong: Conducted experiments, analyzed experimental data, drafted and finalized the manuscript.
Guan Xiaolong: Collected, cleaned, and analyzed data (training set, test set, corpus).

Conflict of Interest Statement

All authors declare no conflict of interest.

Supporting Data

Supporting data is available in the journal's online version at http://www.infotech.ac.cn.

[1] Duan Jianyong, Guan Xiaolong. weblogex.xml. Mini version of user query logs (SougouQ) provided by Sogou Labs.
[2] Duan Jianyong, Guan Xiaolong. dictionary.xml. Chinese dictionary containing pinyin and tonal information.
[3] Duan Jianyong, Guan Xiaolong. corpus.xml. Corpus containing data for model calculations.
[4] Duan Jianyong, Guan Xiaolong. wordclick.xml. Word click-through rates from Sogou Labs.
[5] Duan Jianyong, Guan Xiaolong. trainwords.xml. Test set data.

Note: Figure translations are in progress. See original paper for figures.