iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 **Issue:** Issue I **Month of publication:** May 2015

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

An Efficient Model for String Transformation

Christymol Augustine[1], J. Thomas[2]
Department of Computer Science and Engineering, Christ University, Bangalore, India

Abstract— String Transformation is the transformation of string from one form to another. It is an essential problem in many applications like natural language processing, data mining, information retrieval and Bioinformatics. In real word, string transformations are widely used in many computer applications. It converts the input string into multiple output strings by applying a set of operators. The strings can be strings of words, characters or any type of tokens whereas the operator is a transformation rule which defines the replacement of a substring with another substring. This approach employs a log linear model for string transformation, a method for the training of the model and an algorithm for generating the top k candidates. The log linear model includes a conditional probability distribution of an output string and a rule set for the transformation given an input string. The learning method is based on maximum likelihood estimation. Thus, the model is trained towards the objective of generating strings for a given input string. The generation algorithm efficiently performs the top k candidates generation using top k pruning. It is guaranteed to find the best k candidates, which is frequently being searched. This method can be applied for spelling error correction and query reformulation. The proposed approach is very accurate and efficient.

Keywords— String Transformation, Data mining, Log Linear model, Spelling Error Correction, Query Reformulation

I. INTRODUCTION

This paper employs an efficient model for string transformation. It makes the string transformation more accurate in terms of spelling error correction and query reformulation. String transformations are widely used in many computer applications in the real world. The applications include stemming, spelling error correction, pronunciation generation, biometrics, search engine and other applications where string manipulations play an important role. String transformation in data mining includes mining of the synonyms and the database record matching.

String transformation can be defined as the transformation of an input string to the k most likely output strings by applying a set of operators. Here, the strings can be strings of words or characters or any type of tokens [1] and the operators are transformation rules. The transformation rule defines the replacement of a substring in the query string with another substring. The likelihood of the transformation represents the similarity, relevance and association between the two strings.

In spelling error correction, a string consists of characters, whereas in query reformulation, a string is comprised of words. A dictionary can be exploited for spelling error correction, whereas it is not required for query reformulation. Spelling errors can be of two types: typing or writing error and word error. Typing or writing errors are those that are not present in the dictionary. They are majorly caused while typing or writing a word. Word errors are those that are present in the dictionary.

The spelling mistake is a common phenomenon among search engine queries. In order to help users effectively express their information needs, mechanisms for automatically correcting misspelled queries are required [2]. Spelling error correction in the queries mainly consists of two steps: candidate generation and candidate selection. The concept of candidate generation is nothing but the string transformation. i.e, it is used to find the words with similar spelling. Here, the input can be a string of characters and the operators are insertion, substitution and deletion of characters. It is concerned with a single word. The candidate generator keeps a record of the set of transformations that were applied for each mapping, which is essential for learning the transformation weights, and also calculates a set of similarity scores, necessary for learning the mapping rules. When comparing objects, the alignment of the attributes is determined by the user. A rule-based approach can be used for single-word candidate generation. A typical approach is the use of edit distance, which exploits operations of character deletion, insertion and substitution. After this, the words in the query can be further leveraged to make the final candidate selection. The final candidate selection process produces strings which are really useful in real world applications.

Query reformulation in web search deals with mismatch problem. Suppose a user has given a query and the document does not contain it, then the document does not match well and it will not be ranked high. For example, the query is “NY Times” and the document contains “New York Times”, then the document and the query will not match well. So the query reformulation attempts to transform “NY Times” to “New York Times” and makes a matching between the query and the document. In the case of query reformulation, given a query, one needs to generate all the similar queries from the original query. i.e, it involves rewriting the original query with its similar queries and enhancing the effectiveness of search. It manages to mine transformation rules from pairs of queries in the search logs.

This paper aims to learn an efficient model for the string transformation. Basically, there are three fundamental problems with the string transformation: (1) how to define a model, (2) how to train a model efficiently and accurately and (3) how to generate

the top k output strings from the input string. In this paper, a probabilistic approach has been proposed. It employs (1) a log linear model for the string transformation, (2) an effective algorithm for model learning and (3) an efficient algorithm for string generation.

The log linear model is defined as a conditional probability distribution of a corrected word and a rule set for the correction given the misspelled word. The learning method is mainly based on the maximum likelihood estimation. It employs a criterion that represents the goal of making both accurate and efficient prediction in the training process. As a result, the model is optimally trained toward its objective. The retrieval algorithm uses special data structures and efficiently performs the top k candidates finding. It is guaranteed to find the best k candidates by means of pruning techniques without enumerating all the possible ones. An Aho-Corasick tree is employed to index the transformation rules in the model. When a dictionary is used in the transformation, a trie is used to efficiently retrieve the strings in the dictionary [1], [4].

II. RELATED WORK

String transformations and string manipulations are widely used in various kinds of computer applications such as information retrieval, natural language processing, data mining, etc. It has been studied in various tasks which includes database record matching, spelling error corrected, query reformulation and synonym mining [1]. Our work mainly focuses on accuracy and efficiency of string transformation.

Approximate string search has been studied by many researchers [3]. In this method, it is assumed that the model (similarity or distance) is fixed and the goal is to efficiently find all the strings in the dictionary whose similarity distances are within a threshold. Most of the existing methods employ n-gram based algorithms [5], [6], [7], [8] or trie based algorithms [9]. Instead of finding all the candidates in a fixed range, methods for finding the top k candidates have also been developed. Efficiency is the major focus for all these methods and the similarity functions in them are predefined.

Spelling error correction usually consists of candidate generation and candidate selection. The important step for string transformation is to generate candidates to which the given string is likely to be transformed. Candidate generation is used to find the most likely corrections of a misspelled word from the dictionary. A rule based approach can be commonly used for a single word candidate generation. The edit distance approach is being used which exploits the operations like character insertion, deletion and substitution. In some methods, candidates are generated within a fixed range of edit distance or different ranges for strings with different lengths [10], [11]. Edit distance does not consider the context information. Some methods have been proposed to address this challenge in such a way that it uses a large number of substitution rules containing context information. Spelling error and correction pairs can be mined by using search log data since the users' behavior of misspelling and correction can be frequently observed in web search log data. These mined pairs can be directly used in spelling error correction. For selecting these pairs, methods like entropy model [13] and similarity functions [12] have been developed. However, only high frequency pairs can be found from the log data.

Query reformulation means rewriting the original query with its similar queries and hence increasing the effectiveness of search. Most of the existing methods manage to mine the transformation rules from query pairs in the search logs. This query pair includes original query and the similar query. The contextual substitution patterns [14] can be mined and it can be replaced with the words in the input query by using the patterns. Here, a set of candidates has been generated that each differ from the input query in one word. The existing method is mainly focused on how to extract the patterns and rank the candidates with the patterns.

III.PROPOSED WORK

We propose an efficient model for string transformation by means of spelling error correction and query reformulation. In this method, a large number of input and output string pairs are given as the training data. A set of operators is also provided. A model can be obtained by applying these operators on this training data. This model can assign scores to the candidates of output string in which the best candidate can have the highest probabilistic score with respect to the training data.

In this method, rules are first extracted from the training data. Then the model is constructed which consists of rules and weights. These rules and weights can be stored in a rule index, which can be further used for the reference purpose. Based on these, the system can generate top k candidates of input string which is frequently being searched. Here, log linear model is being used.

A. Log linear model

Basically, the model consists of rules and weights. The rule indicates the replacement of a substring in the input string with another substring. For different applications, character-level or word-level transformation can be considered and thus we can employ character-level or word-level rules. These rules can be derived from the training data based on their string alignment. i.e, based on the edit distance, the characters in the input and output string must be aligned. Next, the rules can be derived from

the alignment and derived rules can be expanded with the surrounding contexts. If a set of rules can be used to convert the input string s_i to the output string s_o, then the rule is defined as a transformation for the string pairs s_i and s_o. Multiple transformations are possible for a given string pair. Here, we assume that the maximum number of rules applied to a string pair is predefined. Hence, the number of possible transformations for a string pair is also limited. The chance of making more errors must be lower than that of making fewer errors. The log linear model uses the binary features to indicate the rules are applied or not. Fig. 1 shows the overview of the model.

B. Model Learning

In order to learn the log linear model, we consider employing the maximum likelihood estimation [1]. The likelihood of transformation can represent association, relevance, and similarity between two strings in a specific application. The likelihood can be defined on the basis of conditional probability of output string given an input string. It can be marginalized over all the possible transformations. i.e, it is defined over observed data. The transformation that actually generates the correction among all the possible transformations is the one that can give the maximum conditional probability.

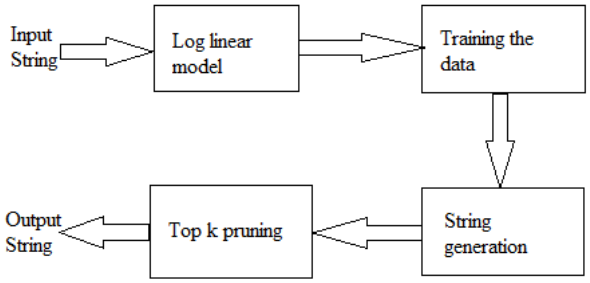


Fig. 1. Overview of the model

C. String Generation

Top k pruning method can be employed for generating optimal k output strings. Two data structures can be utilized to facilitate the efficient generation. One is a trie for storing and matching words in the vocabulary and the other is an Aho-Corasick tree (AC tree) which is used for storing and applying correction rules, referred to as rule index [3].

All the rules and weights can be stored in rule index using AC tree. AC tree is a dictionary matching algorithm which can easily locate the elements of a finite set of strings within an input string. In string generation, we first retrieve all the applicable rules and weights from the AC tree given an input string.

To improve the efficiency of pruning algorithm, the search space and prune unfavorable paths must be limited. For absolute pruning, the number of paths to be explored at each position in the output query is limited. With relative pruning, the paths that have probabilities higher than a certain percentage of the maximum probability at each position is only being explored. The threshold values can be designed to achieve the best efficiency and accuracy.

A dictionary can be utilized in string transformation in which the output strings must exist in the dictionary, such as spelling error correction, database record matching, and synonym mining [1]. The efficiency can be enhanced in this method. The dictionary can be indexed in a trie in such a way that each string in the dictionary corresponds to the path from the root node to a leaf node. While expanding a path (substring) in candidate generation, we match it against the trie, and check whether the expansions from it are legitimate paths. If they are not, the expansions will be discarded and we avoid generating unlikely candidates. i.e, candidate generation is guided by the traversal of the trie. A search session in the web search is comprised of a sequence of queries from the same user within a short time period. Many of the search sessions may consist of misspelled queries and their corrections. In order to mine training pairs automatically from search session data certain heuristics can be employed. Finally, we aggregated the identified word pairs across sessions and users and discarded the pairs with low frequency. In the case of spelling error correction, heuristic is used for mining the word pairs, while in query reformulation, similar query pairs can be used for training the data. Here also, the two queries can be considered similar, if the coefficient is larger than the threshold. If the number of possible rules is more, then the query can be transformed into more candidates.

In our method, we can identify the candidates, which are frequently being searched. So the user would be able to find the most frequently searched items. Thus the efficiency and accuracy of the model can be achieved.

IV. RESULTS

This method can be applied in two applications such as spelling error correction and query reformulation. The major

Volume 3, Special Issue-1, May 2015
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

difference between the two applications is that string transformation can be performed at character level for spelling error correction and at a word level for the query reformulation.

TABLE I: EXAMPLES OF WORD PAIRS

Misspelled word	Correct word
modificatian	modification
randem	random
cantidate	candidate
rezult	result
superindendent	superintendent

A search session is comprised of a sequence of queries from the same user within a time period. There are many spelling errors and their corrections [4] occurring in the same sessions. So we employed heuristics to mine training pairs automatically from search session data.

Table I shows the examples of word pairs. It shows the misspelled word and the corresponding correct word.

In the case of query reformulation, similar query pairs can be used for the training. Table II shows the examples of similar query pairs. Here, one represents an original query and the other one represents a similar query.

TABLE II: EXAMPLES OF SIMILAR QUERY PAIRS

Similar query pairs	
jobs in Australia	full time jobs in australia
define management	what is management
great wall of china	china great wall facts
customer service	rules for good customer service
NY times	New York Times

The number of matches is not largely affected when the size of the rule sets increases in the rule index. It implies that the time for searching applicable rules is close to a constant and does not change much with different numbers of rules.

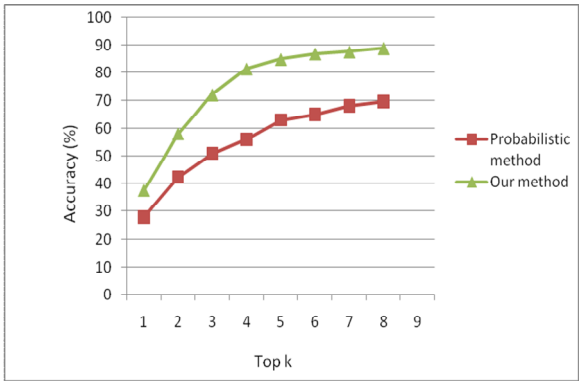


Fig. 2. Performance Evaluation between existing and proposed method

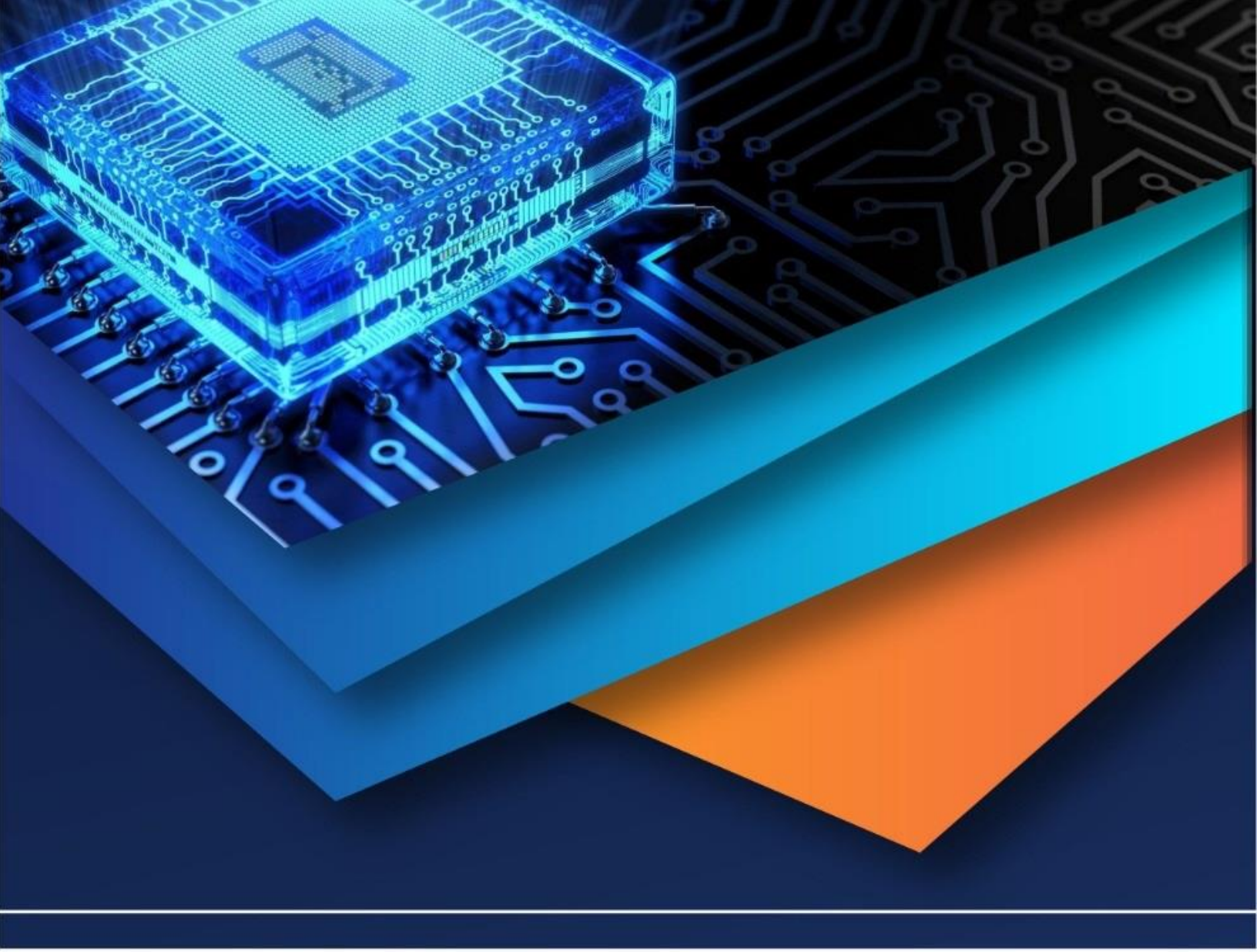
In the case of a probabilistic model, the best k candidates are being identified. But for our method, the frequently searched candidates are being identified. It reduces the execution time and hence the accuracy can be improved. So our model is very efficient compared to the existing models.

V. CONCLUSION

In this paper, we have proposed an efficient model for the string generation. It addresses the problem of spelling error correction as well as query reformulation. In our method, we can identify the candidates, which are frequently being searched. Thus the efficiency and accuracy of the model can be achieved. It is very useful when the dataset is large.

REFERENCES

[1] Ziqi Wang, Gu Xu, Hang Li, and Ming Zhang, "A Probabilistic Approach to String Transformation " , IEEE transactions on knowledge and data engineering, vol. 26, pp. 1063 - 1075 , 2013.
[2] H.Duan and B.J.P.Hsu, "Online spelling correction for query completion," in Proceedings of the 20th international conference on World wide web, ser. WWW '11. New York, NY, USA: ACM, 2011, pp. 117–126.
[3] Ziqi Wang, Gu Xu, Hang Li, and Ming Zhang, " A Fast and Accurate Method for Approximate String Search", in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, Portland, Oregon,vol. 1, pp 52–61,June 19-24, 2011.
[4] Asha Achenkunju, V.R. Bhuma, "An Efficient Reformulated Model for Transformation of String", International Journal of Engineering Research and Applications (IJERA) ISSN: 2248, March 2014
[5] A. Behm, S. Ji, C. Li, and J. Lu, "Space-constrained gram based indexing for efficient approximate string search", in Proceedings of the 2009 IEEE International Conference on Data Engineering, ser. ICDE '09. Washington, DC, USA: Association for Computational Linguistics, 2000, pp.286-293
[6] C. Li, B. Wang, and X. Yang, "Vgram: improving performance of approximate queries on string collections using variable-length grams," in Proceedings of the 33rd international conference on Very large data bases, ser. VLDB '07. VLDB Endowment, 2007, pp. 303–314.
[7] X. Yang, B. Wang, and C. Li, "Cost-based variable-length-gram selection for string collections to support approximate queries efficiently," in Proceedings of the 2008 ACM SIGMD International Conference on Management of data, ser. SIGMOD '08. New York, NY, USA: ACM, 2008, pp. 353–364.
[8] C. Li, J. Lu, and Y. Lu, "Efficient merging and filtering algorithms for approximate string searches," in Proceedings of the 2008 IEEE 24th International Conference on Data Engineering, ser. ICDE '08.Washington, DC, USA: IEEE Computer Society, 2008, pp. 257–266.
[9] S. Ji, G. Li, C. Li, and J. Feng, "Efficient interactive fuzzy keyword search," in Proceedings of the 18th international conference on World wide web, ser. WWW '09. New York, NY, USA: ACM, 2009, pp. 371–380.
[10] M. Li, Y. Zhang, M. Zhu, and M. Zhou, "Exploring distributional similarity based models for query spelling correction," in Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, ser. ACL '06. Morristown, NJ, USA: Association for Computational Linguistics, 2006, pp. 1025–1032.
[11] C. Whitelaw, B. Hutchinson, G. Y. Chung, and G. Ellis, "Using the web for language independent spellchecking and autocorrection," in Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, ser. EMNLP '09. Morristown, NJ, USA: Association for Computational Linguistics, 2009, pp. 890–899.
[12] A. Islam and D. Inkpen, "Real-word spelling correction using google web it 3-grams," in Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, ser. EMNLP '09.Morristown, NJ, USA: Association for Computational Linguistics, 2009, pp. 1241–1249.
[13] Chen Q, M. Li, and M. Zhou, ―Improving query spelling correction using web search results,‖ EMNLP '07, pp. 181–189, 2007.
[14] X. Wang and C. Zhai, "Mining term association patterns from search logs for effective query reformulation," in Proceeding of the 17th ACM conference on Information and knowledge management, ser.CIKM '08. New York, NY, USA: ACM, 2008, pp. 479–488.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY