# String Transformation Approach A Way towards Enhancing the Effectiveness of Web Searching

**Dipika L. Tidke** **Prof. N. R. Wankhade**

*Abstract: -* **End-user interaction with the system recovery information required to run a successful query to the system is not always feasible and if the user is not technical then forming successful query will be difficult. So in this work we are suggesting is that the search for users must be satisfied in minimum time and also this should consume less time and the system must be easy to use and accurate. The main focus of our proposed system is that if any user enters a wrong or incorrect query, also will try to fix it first and have the "n" possible that incorrect query results. Reformulation of queries in search is aimed at addressing the problem of maturity mismatch. For example, if the query is "TOI" and the document only contains "New York Times", the query and the document does not fit well and the document does not classified high. Query Reformulation tries to transform "TOI" the "Times of India" and therefore make a better match between the query and the document. In the task, given a query that is needed to generate all similar queries from the original query. So here we are achieving system usability and nature became our easy to use, since the end user does not insist to give correct query only. The proposed technique is applied to the correction of spelling errors in queries, query reformulation and search the web. Experimental results on a large data sizes show that the proposed approach is very accurate and effective in improving the existing methods in terms of accuracy and reliability in different contexts.**

*Key Words—* **String Transformation, Log Linear Model, Spelling Check, Query Reformulation.**

## I. INTRODUCTION

Main task is to understand natural language given by the end user first, then must be processed to achieve human-computer interaction desired output. And allowing computer systems to get the meaning of the transformations input.

**Fig 1.1 Processing of user's queries**

String human or natural language have been proposed as a solid game mechanism for such variations. However, in many areas, the identification of a suitable set of transformations is difficult that the space of possible transformations is great.

## II. SURVEY OF LITERATURE

### A. *String Transformation*

Can be described as follows; if given one input string is 'S' and a set of operators included in the chain, we can transform the input string to the 'n' most probable output strings [1].strings can be strings of words, characters, or any type chips. Each operator is a transformation rule or semantic definable replace a substring with another substring. The probability of transformation can represent similarity, Relevance, and the association between two strings in a specific application.

**Fig 2.1 strings processed by computer system**

### B. *Approach towards string transformation*

As we know that we can prepare a dictionary to match the given input string against it. When a dictionary is used assumption is that there must be strings from the given dictionary; it may happen that the dictionary size can be large. Another part that needs to be discussed is that we need to study the correction of spelling errors in first giving the string consisting of characters must be entered. Then the next task to use a dictionary to find words or similar characters [11].

### C. *Spell Check*

Correcting spelling errors in queries usually consists of two phases: candidate generation and candidate selection.

**Fig 2.2 Spelling Check**

Candidate generation is used to find the most likely fixes a misspelling of dictionary [6]. In this case, a character string is input and operators represent the insertion, deletion and substitution of characters or no characters around, for example, "a" → "e" and "lly" → "ly" [1]. Obviously candidate generation chain is an example of transformation. Note that the generation of candidate has to do with one word; After candidate generation, the Words in the context.

### *D. Query Reformulation*

Reformulation of queries in search is aimed at addressing the problem of maturity mismatch. For example, if the query is "TOI" and the document only contains "New York Times", the query and the document does not fit well and the document does not classified high. Query Reformulation tries to transform "TOI" the "Times of India" and therefore make a better match between the query and the document. In the task, given a query that is needed to generate all similar queries from the original query. Query Reformulation again involves writing the original query string or its consultations or similar words match the dictionary and improve the search efficiency. Most existing methods for managing transformation rules of the mine from pairs of queries in search logs. Given two words "cat", "FAR" to determine whether you can get from the first to the second through simple transformations of valid words .... for example 1 transformation receives CAT to CAR T change R, then another takes you from the car to change C to F FAR ... all are valid English words.

## III. Proposed System

After studying the various aspects of the formation of the chain or can even say that the reform chain, proposed the following system which we are certainly giving the efficiency and accuracy. This is what can occur with the help of the following model.

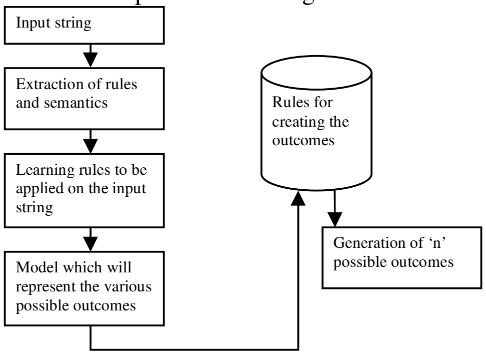

Input string
Extraction of rules and semantics
Learning rules to be applied on the input string
Model which will represent the various possible outcomes
Rules for creating the outcomes
Generation of 'n' possible outcomes

**Fig 3.1 proposed basic model**

### *A. Initial basic steps to be executed on the system*

```
❖ Algorithm : Top 'n' Pruning
❖ Input: rule index which will specify the rules and
        semantics
❖ Ir, input string , candidate number 'n'
❖ Output: top 'n' output strings
❖ begin
❖ apply various  rules applicable to input string
❖ calculate the  minscore
❖ while check various possible paths
❖ do
    ❖ Pickup a possible
    ❖ if score < minscore then
    ❖ continue
    ❖ similarly check with all the possible outcomes
    ❖ if any candidate will have min score
    ❖ then Remove candidate with minimum score
    ❖ Stop
```

### *B. Spell Check as an Obstacle*

We must focus on this question, as it is related to the input string.

- ***Ambiguity:*** Like other languages, the English spelling has never been systematically updated and, therefore, only in part, argues that the alphabetic principle. As a result, the spelling is a need to focus on the weak rules with many exceptions and ambiguities.
- ***Redundant letters:*** Alphabet letters can have multiple feature which are already represented in the alphabet elsewhere. The spelling of some words - such as the tongue and stomach - are so unindicative of it to change the spelling would significantly change the shape of the word. [12].
- Similarly, irregular spellings of common words like is, are, have, do, and that makes it difficult to solve without introducing a noticeable change in the appearance of the English text.

**Table 1.Examples of Word Pairs**

| **Misspelled** | **Correct** | **Misspelled** | **Correct** |
|---|---|---|---|
| Reteive | Retrieve | Tabel | Table |
| Infomation | Information | Chevrole | Chevrolet |
| liyerature | Liyerature | Newpape | newspaper |

### *C. Spell Correction*

We can prepare a set like this set in which all possible outcomes are stored and will look like the following:
tests1 = {'access',' Acess', 'access',' Accesing ',' Accommodations', 'accommodation acomodation lodging','mind','acount",...}

### *D. Pruning on strings*

Pruning is a method in machine learning which decreases the size of the decision trees by removing sections of the tree that provided little power to classify cases[1].

### *E. Predicted Modules in proposed system*

- ➢ First module that handles the input string to be entered by the end user.
- ➢ Second module will check immediately if the string entered is correct with respect to syntax and semantics.
- ➢ Third module suggest spelling corrected for the user's query or chain.
- ➢ Fourth module will recover 'n' possible outcomes of the user's query or chain.
- ➢ Fifth Module retrieve relevant documents from the database that will satisfy the user's request.

These modules handle above the global system smoothly and efficiently.

### *F. Practical Aspects related with our proposed System*

#### *1) Lexical Database*

As WordNet resembling a thesaurus, in which IT groups together words based on their meanings. However, there are some important differences. First, WordNet articulated words not only chains but forms letters-specific meanings of words. As a result, the words that are in close proximity to each other on the network are semantically disambiguated.

#### *2) String Comparison*

As a user entered string as an input so if the query is wrong then the dropdown list will display some strings matching with respect to the user query. therefore pseudo code will look something like the following manner in which the user query and correct query is compared is displayed on the screen [4]. public double getSimilarity (String str1, String str2); For this case, one more to consider as follows: getSimilarity("Professor","Master").

#### *3) Use of Genetic Algorithm*

A standard representation of each candidate solution is as a matrix of bits. The arrays of structures, and other types may be used in substantially the same manner. The main property makes these desirable genetic representations easily aligned parts due to their fixed size, which facilitates the operations of simple crossing.

#### *4) Proposed System Architecture*

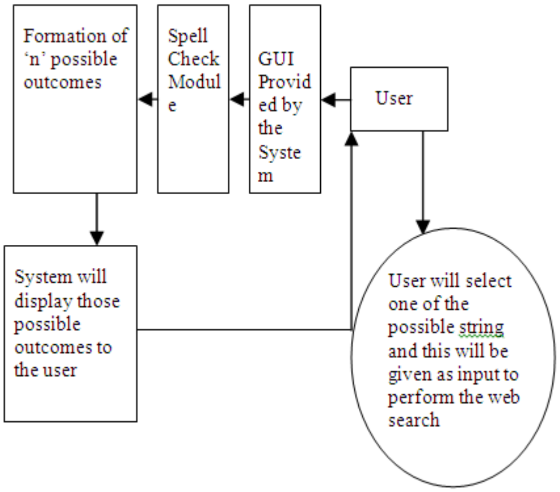

Formation of 'n' possible outcomes
Spell Check Modul e
GUI Provid ed by the Syste m
User
System will display those possible outcomes to the user
User will select one of the possible string and this will be given as input to perform the web search

**Fig 3.1 log linear model**

what we do with this approach is that as soon as the user goes to submit a query that can be in the natural language of our system will check if the query is successful or not. Once this phase again with the help of dictionary concept will try to form the "n" possible outcomes and presented to the user. So as soon as the user that any of them are considered as the input to the next step and Web search is performed is selected. The aim of our system is that in less time user search session must be completed. And while we have to maintain the efficiency and accuracy of the system.

#### *5) Log linear model*

A log-linear model is a mathematical model which takes the form of a function whose logarithm is a first-degree polynomial function of the model parameters, which makes possible the (possibly multivariate) applying linear regression[5].

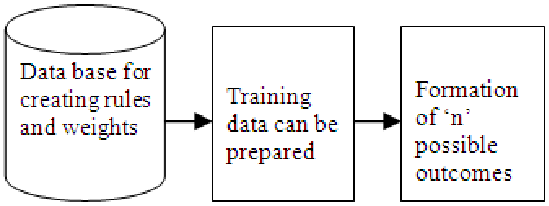

Data base for creating rules and weights
Training data can be prepared
Formation of 'n' possible outcomes

**Fig 3.3 formation of 'n' possible outcomes**

In our method, the model is a linear model that represents the registration rules and weights, learning is driven by maximum likelihood estimation in the training data, and the generation is carried out efficiently with the top k pruning.

#### *6) Word pair mining*

Mining word pair is simply identifying the word pair. Idea is to find the equivalents such peers as after giving a query that is in the natural language system will first check the accuracy and correctness of that word and if it is bad system will automatically assume the correct and present possible chains be formed from the wrong query. Next, the user selects either of them this approach is known as mining pair of words.

#### *7) The use of Aho-Corasick algorithm*

It is a string matching algorithm is a string searching algorithm invented by Alfred V. Aho and Margaret J. Corasick. It is a type of dictionary matching algorithm that locates elements of a finite set of words within a text input. This algorithm matches all patterns simultaneously with the given input by the end user [1]. The complexity of the algorithm is linear in the length of the patterns plus the length of the searched text plus the number of matches output. All matches are found, there can be a number of parties of the second degree if each substring matches (eg dictionary = b, bb, bbb, bbbb bbbb is the input string). With the help of Aho Corasick algorithm system will locate all occurrences of any of a finite number of keywords in a text string. In this approach, the pattern matching will be very useful as we are selecting / guessing 'n' possible strings of the query results provided by the end user.

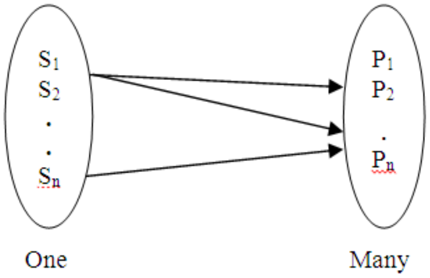
## IV. MATHEMATICAL MODEL

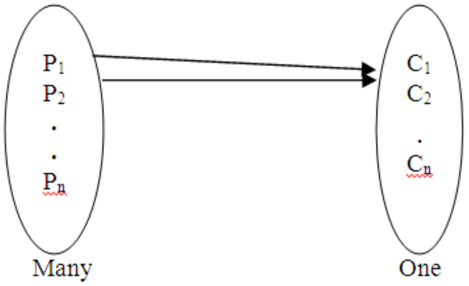
Here we are presenting mathematical model for our proposed system is as follows:

1. U={S, P, C, M, T}

Where S= {$S_1$, S2, S3, Sn, Sn≠0}
where S is set of string
Where P= {$P_1$, P2, P3, Pn, Pn≠0}
where P is set of Patterns
Where C= {$C_1$, C2, C3, Cn}
where C is set of Spell Check.
Where M= {$M_1$, M2, M3 …Mn}
where m is a set of matching string.
Where T={$T_1$,$T_2$,$T_3$,……$T_n$}
where T is set of Transformation.

2. Let $f_p(S) \longrightarrow P$
Where $f_w$ is a function that takes string patterns and provide it.
Let $f_c(P) \longrightarrow C$
Where $f_c$ is a function that checks the spelling..
Let $f_m(C) \longrightarrow M$
Where $f_m$ is a function for String matching.
Let $f_t(M) \longrightarrow T$
Where $f_t$ is a function that Transform String.





## V. Results and Analysis

Here are the following results of existing system and proposed system . They are as follows:

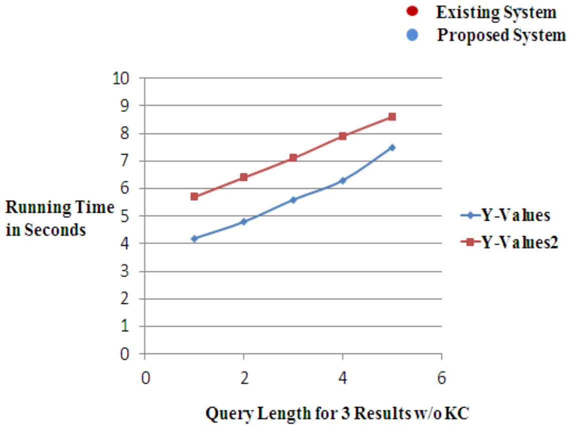


**Fig. 5.1 Query Length for 3 Results w/o KC**

Here in this fig. 5.1: Result is plotted for total of 3 Results with Query Length against project execution time in seconds without keyword count.

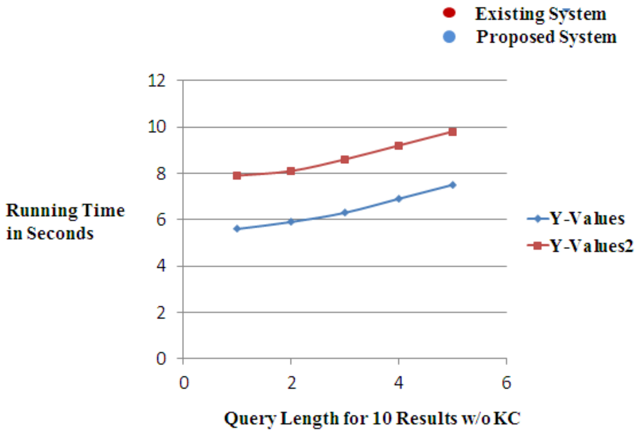


**Fig. 5.2 Query Length for 10 Results w/o KC**

Here in this fig. 5.2: Result is plotted for total of 10 Results with Query Length against project execution time in seconds without keyword count.

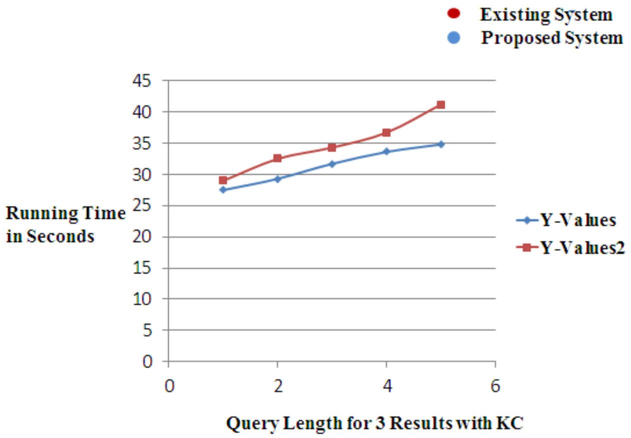


**Fig. 5.3 Query Length for 3 Results with KC**

Here in this fig. 5.3: Result is plotted for total of 3 Results with Query Length against project execution time in seconds with keyword count.

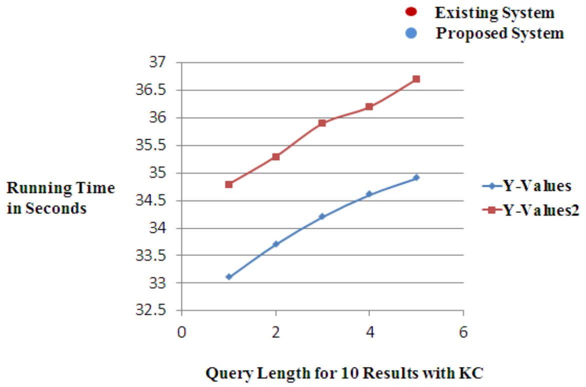


**Fig. 5.4 Query Length for 10 Results with KC**

Here in this fig. 5.4: Result is plotted for total of 10 Results with Query Length against project execution time in seconds with keyword count.

## VI. CONCLUSION & FUTURE SCOPE

After discussing above points now we set our contribution we want to do in our proposed system. As most information retrieval application requires an Internet connection in the state live; What if sometimes the connection is not available? So our only proposed approach is to store these input strings and possible outcomes with respect to the query. It means even in offline mode an end user may find the possible outcomes. If that query string or already running on the system, then our system will prompt a message to user queries or chain is already running. Even reducing the time required to form the possible outcomes that will focus primarily on the string entered by the user. The meaning is that our system in less time to check the spelling of that chain. If done wrong instead of warning the user about the same, the system must give the correct spelling in the drop down list immediately. If the input string is large enough, then we are also thinking of applying algorithms to eliminate frequently occurring word in that chain. So what will be the key words or symbols. Then, using these symbols will form 'n' possible results.

## REFERENCES

[1] Ziqi Wang, Gu Xu, Hang Li, and Ming Zhang, "A Probabilistic Approach to String Transformation" IEEE Transaction on Knowledge and Data Engineering vol: PP NO:99 YEAR 2013

[2] A. R. Golding and D. Roth, "A winnow-based approach to context-sensitive spelling correction," Mach. Learn., vol. 34, pp.107–130, February 1999.

[3] 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: IEEE Computer Society, 2009,pp. 604–615.

[4] 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.

[5] M. Dreyer, J. R. Smith, and J. Eisner, "Latent-variable modelling of string transductions with finite-state methods," in Proceedings of the Conference on Empirical Methods in Natural Language Processing's. EMNLP '08. Stroudsburg, PA, USA: Association for Computational Linguistics, 2008, pp. 1080–1089.C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.

[6] N. Okazaki, Y. Tsuruoka, S. Ananiadou, and J. Tsujii, "A discriminative candidate generator for string transformations," in Proceedings of the Conference on Empirical Methods in Natural Language Processing, ser. EMNLP '08. Morristown, NJ, USA: Association for Computational Linguistics, 2008, pp. 447–456.

[7] 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

[8] J. Xu and G. Xu, "Learning similarity function for rare queries," in Proceedings of the fourth ACM international conference on Web search and data mining, ser. WSDM '11. New York, NY, USA: ACM,2011, pp. 615–624.

[9] R. Vernica and C. Li, "Efficient top-k algorithms for fuzzy search in string collections," in Proceedings of the First International Workshop on Keyword Search on Structured Data", ser. KEYS '09. New York, NY, USA: ACM, 2009, pp. 9–14.

[10] E. Brill and R. C. Moore, "An improved error model for noisy channel spelling correction," in Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, ser. ACL '00. Morristown, NJ, USA: Association for Computational Linguistics, 2000, pp. 286–293

[11] S. Tejada, C. A. Knoblock, and S. Minton, "Learning domain independent string transformation weights for high accuracy object identification," in Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, ser. KDD'02. New York, NY, USA: ACM, 2002, pp. 350–359.

[12] 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. pp. 1241–1249.

[13] Q. Chen, M. Li, and M. Zhou, "Improving query spelling correction using web search results," in Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, ser. EMNLP '07, 2007.

## AUTHOR'S PROFILE

**Dipika L. Tidke**.

P.G. student: Department of Computer Engineering, Late G.N. Sapkal College of Engineering, Anjaneri,City: Nasik, Country: India.

University: Savitribai Phule Pune University.

Email id:dipikatidke@gmail.com.

1. Paper published in International Journal of Engineering Research and Technology (IJERT)ISSN: 2278- 0181 "Survey Paper on Effective Web Search Through String Transformation".

2. Paper Presented in cPGCON 2014 "Practical Aspects of Effective Web Search Through String Transformation Technique".

**Prof . N.R. Wankhade**

Associate Professor: Department of Computer Engineering, Late G.N.Sapkal College of Engineering, Anjaneri, City: Nasik, Country: India. University: Savitribai Phule Pune University.

Email id:nileshrw_2000@yahoo.com

He has presented papers at National and International conferences and also published paper in national and international journals on various aspect of the computer engineering and networks.His research of interest include computer networks, network security, wireless sensor network.