Implementation Aspects of Effective Web Search through String Transformation Technique

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

Abstract- **Interaction of the end user with information retrieval system requires to execute a correct query to the system which not always feasible and if user is non-technical then formation of correct query will be difficult. So in this paper we are suggesting is that users' search should be satisfied in minimum time and addition to this should consume a less time and system should be user friendly and accurate. Main focus of our proposed system is if any user is entering a wrong or incorrect query then also we will try to correct it first of all and present the 'n' possible outcomes from that incorrect query. Query reforming of search is intended to address the problem of inconsistency of terminology. For example, if you have a query "TOI" and the document only contains "New York Times". Then the query and the document do not match well and the document will not be RANKED high. Attempts to transform query reformulation "TOI" to "Times of India" and thus make a better matching between the query and document. In the task, given a query one needs to generate all similar queries from the original query. Therefore here we are achieving the usability nature of the system and our system became user friendly as it does not insist end user to give correct query only. Though user to enter an incorrect query by applying the pruning strategy which will guarantee to generate the 'n' suitable string or queries which will be possibly relevant with the user's query. Proposed method is Applied to correct errors in spelling of queries as well as reformulation of queries in Web search. Experimental results on large scale data show that the Proposed approach is very accurate and efficient Improving upon Existing methods in terms of accuracy and efficiency in different settings.**

Keywords— String Transformation, Log Linear Model, Spelling Check, Query Reformulation

I. INTRODUCTION

Main task is to understand the natural language given by end user first of all, then need arises to process it to achieve the desired output computer interaction. & enabling computer systems to obtain meaning from human or natural language input. The need is to find a dictionary definition or specific vocabulary. When in operation, the weak means to fix each word in a sentence can not be possible. Performance is an important factor to take into consideration in our step. Off unused settings into consideration. At the end of a string for a raise this limit is only logical for input variability. Launch of multiple amendments and technical changes have created a better candidate. However, the development is not significant in any module, which has been linked to a number of processes the input that has to do with size. The implementation process typically leads to consumption time for action. Until the policy is weak in any sentence that has not been resolved to the full dynamic. Accuracy of pattern matching to ensure the implementation of the other small. The implementation process typically leads to consumption time for action. Until the policy is weak in any sentence that has not been resolved to the full dynamic. Accuracy of pattern matching to provide more results. String transformations have been proposed as a mechanism to make matching robust to such variations. However, in many domains, identifying an appropriate set of transformations is as challenging as possible transformations of the space is wide.

II. SURVEY OF LITERATURE

A. *String Transformation*

It can be described as follows; if a given an input string is 'S' and a set of operators included in the string, we can transform the input string to the 'n' most probable output strings [1]. Strings can be strings of words, characters, or any type of tokens. Each operator is a transformation rule or semantics which can define the substitute of a substring with another substring. The probability of transformation can represent similarity, Relevance, and association between two strings in a specific application.

B. *Approach towards string transformation*

As we know we can prepare a dictionary for matching the given input string against it. When a dictionary is used, assumption is that output strings must exist in the given dictionary; it may happen that size of the dictionary may be large. Another part which needs to be discussed is that we need to study correction of spelling errors in given first of all string which consists of characters need to be entered. Then next task to utilize a dictionary for finding the similar words or characters [11].

C. *Spell Check*

Correcting spelling errors in queries usually consists of two steps: candidate generation and candidate selection. Candidate generation is used to find the most likely corrections of a misspelled word from the dictionary [6]. In such a case, a string of characters is input and the operators represent insertion, deletion, and substitution of characters with or without surrounding characters, for example, "a"→ "e" and "lly" →"ly" [1]. Obviously candidate generation is an example of string transformation. Note that candidate generation is concerned with a single word; after candidate generation, the words in the context.

D. Query Reformulation

Query reforming of search is intended to address the problem of inconsistency of terminology. For example, if you have a query “TOI” and the document only contains “New York Times”. Then the query and the document do not match well and the document will not be RANKED high. Attempts to transform query reformulation “TOI” to “Times of India” and thus make a better matching between the query and document. In the task, given a query one needs to generate all similar queries from the original query.Query reformulation involves again writing the original query or string with its similar queries or words match with dictionary and enhancing the effectiveness of search. Most existing methods manage to mine transformation rules of queries from peers in the search logs. Given two words "CAT,FAR "determines if you can get from the first to the second via single transformations of valid words .... eg. gets you from one transformation to CAT ,CAR changing T to R, Then Reviews another gets you from CAR to FAR changing the C to F ... all are valid English words m CAT to CAR changing T to R, then another gets you from CAR to FAR changing the C to F...all are valid english words.

II. Proposed System

After studying the various aspects of the string formation or even we can say the string reformation, proposed the following system which we feel will definitely give the effectiveness and accuracy. This we can present with the help of following model.

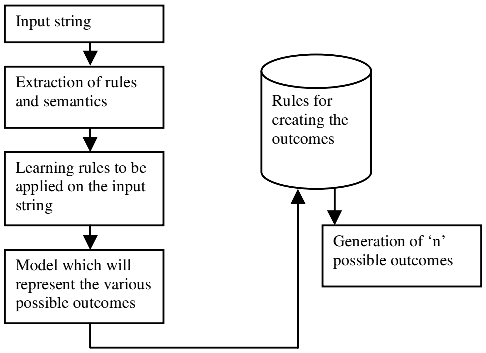

Input string
Extraction of rules and semantics
Learning rules to be applied on the input string
Model which will represent the various possible outcomes
Rules for creating the outcomes
Generation of ‘n’ possible outcomes

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 need to focus on this definitely as it is related with the input string.

- Ambiguity: Like other languages, English spelling had not been updated and therefore are just some of the principles of value. As a result, the spelling is necessary to concentrate on the rules that are weak and puzzled many exceptions.
- Letter redundant: the letter may have several letters that look as already other characters spelling of certain words - such as the tongue and stomach so unindicative pronounce their changes. clearly spell to change the shape of the word. [12] Similarly, the unusual spelling of common words such as will perform and make it difficult to resolve the issue without introducing changes evident with the presence 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 following set in which all the possible outcomes will be stored and it will look like as follows:

tests1 = {‘access’: 'acess', 'accessing': 'accesing', 'accommodation': 'accomodation acommodation acomodation', 'account': '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]. The string generation problem amounts to that of finding the top k output strings given the input string. To further improve the efficiency of pruning algorithm, we need to limit the search space and prune unpromising paths early. In practice, carefully designed beam pruning methods can usually achieve significant improvement in efficiency without causing much loss in accuracy. For absolute pruning, we limit the number of paths to be explored at each position in the target query. With relative pruning, we only explore the paths that have probabilities higher than a certain percentage of the maximum probability at each position. The threshold values are carefully designed to achieve the best efficiency without causing a significant drop in accuracy. In practice we find relative pruning to be generally more effective for pruning unpromising paths. In our system, we make use of both absolute pruning and relative pruning to improve search
efficiency and accuracy.

E. Predicted Modules in proposed system

➢ First module which handle the input string to be entered by the end user.

➢

- Second module which will immediately check whether the entered string is correct with respect to syntax and semantics.
- Third module will suggest corrected spellings with respect to user's query or string.
- Fourth module will retrieve the 'n' possible outcomes of user's query or string.
- Fifth module will retrieve the relevant documents from the database which will satisfy the user's request.

These above mentioned modules will handle the overall system smoothly and effectively.

F. Practical Aspects related with our proposed System

1) Lexical Database

String transformation can be conducted at two different settings, depending on the dictionary. When dictionary is used, the output string must exist in the given dictionary, while the size of the dictionary can be very large and the response time is very short. To create a dictionary it follows these steps: add words to the dictionary arrange the words alphabetically and add description related to that word.

2) String Comparison

As user will enter string as a input then if in case that query will be wrong then in drop down list we will display few string which are matching with respect to that user's query. Therefore pseudo code will look like something as follows in which user's query will be compared and correct query will be displayed on the screen [4].public double getSimilarity (String str1, String str2);For this one more case need to be considered like following:
getSimilarity ("Professor", "Teacher").

3) Proposed System Architecture

what we wish to do with this approach is that as soon as user will submit one query which may be in natural language our system will check whether that query is correct or not. Once this phase over with the help of dictionary concept we will try to form the 'n' possible outcomes and will be presented to the user. So as soon as user will selects any one of them it will be consider as input to next step and web search will be performed. Our system's intention is that within less time user's search session should be completed. And at the same time we have to maintain the effectiveness and accuracy of the system.

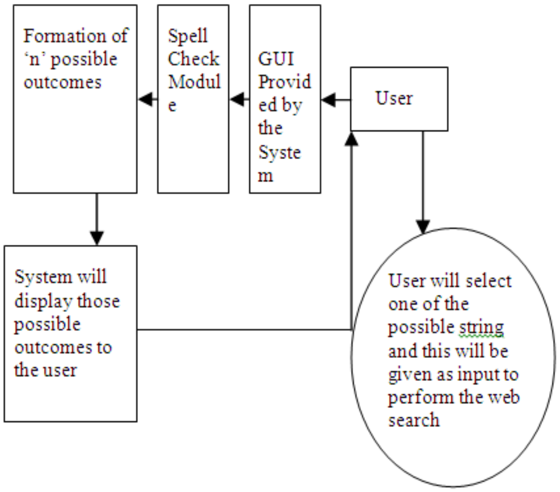

Formation of 'n' possible outcomes
Spell Check Modul e
GUI Provid ed by the Syste m
User
System will display those possible outcomes to the user
User will select one of the possible string and this will be given as input to perform the web search

Fig 3.1 Proposed System Architecture

4) Log linear model

A log-linear model is a mathematical model that takes the form of a function whose logarithm is a first-degree polynomial function of the parameters of the model, which makes it possible to apply (possibly multivariate) linear regression [5].In our method, the model is a log linear model representing the rules and weights, the learning is driven by maximum likelihood estimation on the training data, and the generation is efficiently conducted with top *k* pruning.

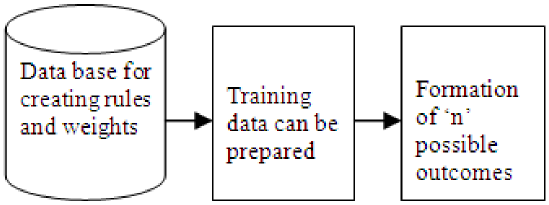

Data base for creating rules and weights
Training data can be prepared
Formation of 'n' possible outcomes

Fig 4.1 formation of 'n' possible outcomes

5) Word pair mining

Word pair mining is nothing but identifying the pair of words. Idea is to find out the equivalent pairs such as after giving a query which is in natural language system will first of all check the accuracy and correctness of that word and if it is wrong system will automatically assume the correct and will present the possible strings to be formed from that wrong query. Then user will select any one of them this approach will be known as word pair mining.

.

III. MATHEMATICAL MODEL

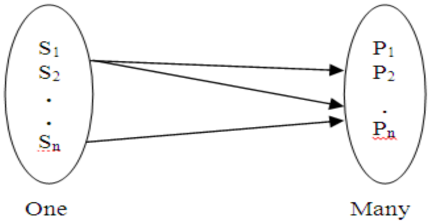
Here we are presenting mathematical model for our proposed system is as follows:

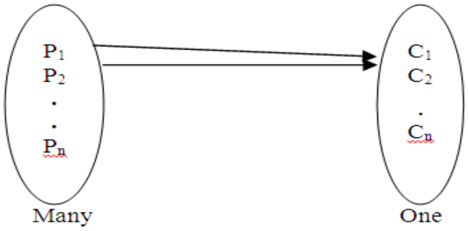
1. U={S, P, C, M, T}

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.

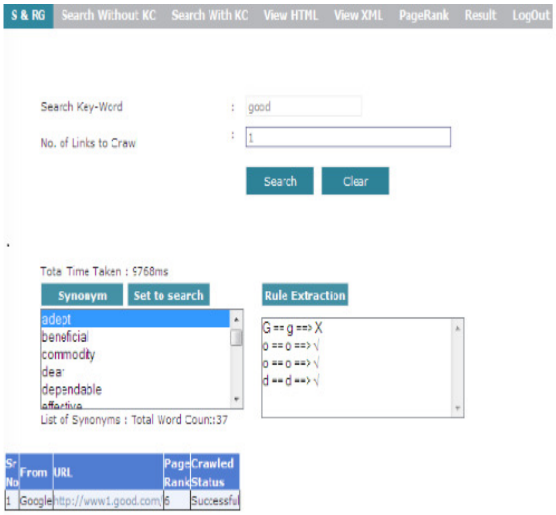
2. Let $f_p(S) \longrightarrow P$
Where f_w is a function that takes string patterns and provide it.
Let $f_c(P) \longrightarrow C$
Where f_c is a function that checks the spelling..
Let $f_m(C) \longrightarrow M$
Where f_m is a function for String matching.
Let $f_t(M) \longrightarrow T$
Where f_t is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





IV. Results And Analysis

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



Search Key-Word : good

No. of Links to Craw : 1

Tota Time Taken : 9768ms

adept
beneficial
commodity
dear
dependable
effective

List of Synonyms : Total Word Coun::37

G == g ==> X
o == o ==> √
o == o ==> √
d == d ==> √

Sr No	From	URL	Page Rank	Crawled Status
1	Google	http://www1.good.com/	6	Successful

Fig.4.1.Search Word

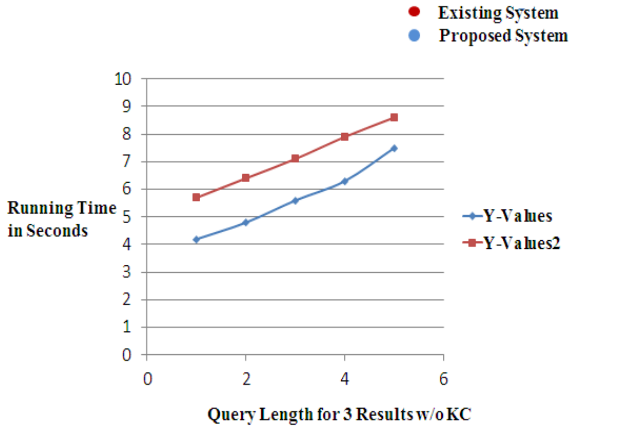


Fig. 4.2 Query Length for 3 Results w/o KC

Here in this fig. 4.2: Result is plotted for total of 3 Results with Query Length against project execution time in seconds without keyword count.

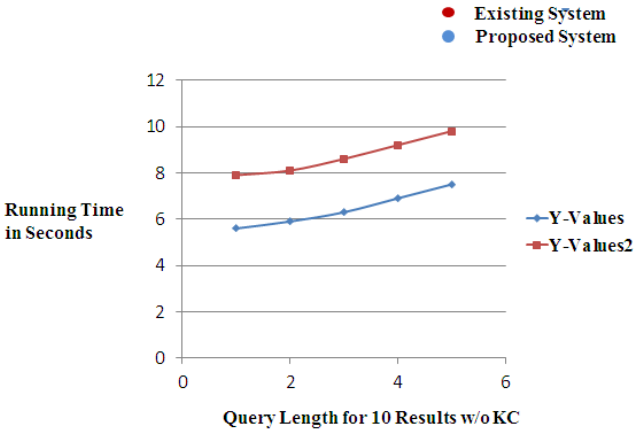


Fig. 4.3 Query Length for 10 Results w/o KC

Here in this fig. 4.3: Result is plotted for total of 10 Results with Query Length against project execution time in seconds without keyword count.

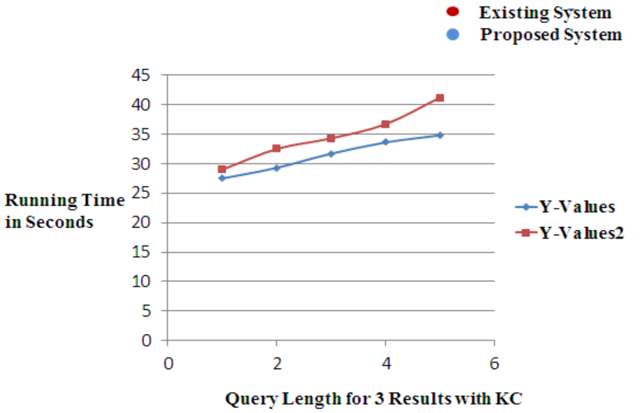


Fig. 4.4 Query Length for 3 Results with KC

Here in this fig. 4.4: Result is plotted for total of 3 Results with Query Length against project execution time in seconds with keyword count.

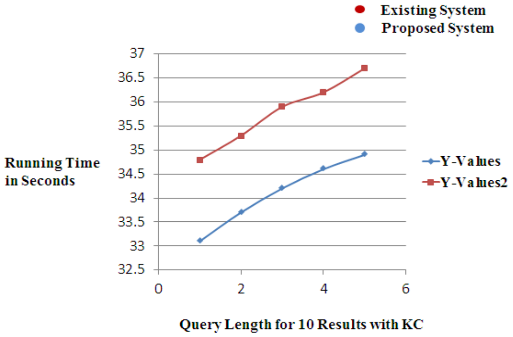


Fig. 4.5 Query Length for 10 Results with KC

Here in this fig. 4.5: Result is plotted for total of 10 Results with Query Length against project execution time in seconds with keyword count.

V. CONCLUSION & FUTURE SCOPE

After discussing above mentioned points now we proposed our contribution what we wish to do in our proposed system. As almost information retrieval application will require an internet connection in live state; what happens if sometimes that connection will not be available? So our one proposed approach is to store those input strings and possible outcome with respect to that query. Means even in offline mode an end user can find out the possible outcomes. If that query or string is already executed on the system then our system will prompt a message to user that query or string is already executed. Even for reducing the time required to form the possible outcomes we will mainly focus on the string entered by user. Meaning is that our system within a less time should check the spelling of that string. If it wrongs instead of warning user about the same, system should give correct spellings in drop down list immediately. If the input string is large enough then also we are thinking on applying the algorithms to remove the frequently occurring word in that string. So what will remain will be the important words or tokens. Then by using those tokens we will form 'n' possible outcomes.

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 aspects of the computer engineering and networks. His research of interest include computer networks, network security, wireless sensor network.