Efficient Web Search Technique using String Transformation

Prof. Nitin Mishra¹, Anil E. Patil²

Associate Professor¹, P.G. student²
Department of Information Technology
NRI Institute of Science and Technology, Bhopal ¹, ²
Nitin.nriist@gmail.com,anilpatilonline@gmail.com,

Abstract

In today’s world, we deal with data mining, retrieval of information and various fields in which necessity to arises to deal with natural language processing. In that we transforming the query or we can say that the given string as an input. Once the string is given ‘a’ as input the system or application generates the ‘n’ more suitable output strings in reply to that string or query. It means that system will create the suitable ‘n’ strings – relation of probable strings. In this system, focused on the small amount of aspects of the natural language processing and string or transformation of query. We will also focus on the proposed system. In this approach we will see how a system ‘a’ can create the ‘n’ suitable relations of ‘a’ and ‘n’, and how those will be perfect and systematic. This means we will need to focus on the systems efficiency and accuracy through this novel process for retrieving the information. We can also try pruning strategy which give guarantee to create the ‘n’ suitable queries or string. In the proposed system, method is applied to correction of spelling in queries as well as reformulation of queries on web search. Experimental outcomes on large scale data show that the proposed system is very efficient, accurate and effective on improving upon existing methods with respect to accuracy and efficiency in individual settings.

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

1. Introduction

Many difficulties in natural language processing, information retrieval, bioinformatics and data mining, should be formalized as string transformation, which follows the process like. Given an input string, the system creates the ‘k’ most possible output strings related to the input string. In data mining, string transformation can be working in the mining of database record matching and synonyms. As many of the above are online or web applications, the transformation can be conducted not only effectively but also efficiently. String transformation should be defined in the following way. Given a string as input and a set of operators that manipulate, we are able to relocate the input string to ‘k’ most possibly output strings by applying a number of operators. Here the strings must be strings of characters, words, or any tokens. Each operator is a rule of transformation that defines the substring replacement with another substring. The like that of transformation can represent association, relevance, and similarity between two or both strings in a specific application. Although certain progress has been made, further examination of the task is still necessary, particularly from the lookout of enhancing both correctness and efficiency, which is exactly the goal of this work. String transformation can be focused on two different settings, depending on system whether or not a dictionary is used. When we used a dictionary, the output strings must exist in the respective dictionary, while the size of the dictionary can be precise large. Without loss of generality, we specifically study correction of spelling errors in queries as well as preparation of queries in web search in this paper. In the first process, a string contains of characters. In the second process, a string is concealed of words. The former needs to reveal a dictionary while the latter does not.

Following are the few process which can be reflect under this NLP (Natural language Processing):

1. Automatic summarization: as its name suggests it can produce a human readable outline of a given text. Frequently used to provide text outline which is known type, such as articles in the sports section of a newspaper.
2. **Machine translation**: Automatically convert text from one human language to other. This is one of the most difficult hazards, and is a member of a class of hazards colloquially termed "AI-complete", i.e. requiring all of the unlike types of knowledge that humans can possess (grammar, facts about the real world, semantics, etc.) in order to resolve properly.

3. **Relationship extraction**: Given an input text, recognize the relationships among named entities which will definitely help to find the more relevant data or records immediately which will increase the effectiveness of system.

4. **Word sense disambiguation**: Various words have more than one meaning; so need arises to select the meaning of the word which makes the most sense in context. For this hazard, we are typically given a list of words and related word senses; so we can formulate an index or a kind of dictionary that will help the end users.

2. **Related work**

   In 2002, “Pronunciation Modeling for Improved Spelling Correction” [3] Kristina Toutanova Robert and C. Moore developed a system which presents a method for separating word pronunciation information in to a noisy channel model to perform a spelling correction. This method construct a precise error model for a word pronunciations.[2] By modeling pronunciation same between words author achieved a substantial performance improvement over the preceding best performing models for spelling correction.

   In 2005, “Learning a Spelling Error Model from Search Query Logs” [4] author Farooq Ahmad and Grzegorz Kondrak. This system viewed on occupying a language model and probabilistic error model unswerving from search query logs without any need of a misspelled words corpus paired with their corrections. The process of search query spelling correction analyzed, and an execution of the Expectation Maximization (EM) algorithm to grasp an error model is specified, with reference to same approaches. The character of search queries is analyzed in the context of spelling correction.

   In 2008, “A Discriminative Candidate Generator for String Transformations” [5] by Naoaki Okazaki, Sophia Ananiadou, Yoshimasa Tsuruoka and Jun’ichi Tsujii. This system represents a different method for generating candidate strings. In this system author utilized substitution of substring rules as features and score them by using an L1-regularized logistic regression model. They also proposed a system process to produce negative object that affect the decision boundary of the model. The essential benefit of this process is that candidate strings can be specified by an effective algorithm because the processes of string transformation are controllable in the model. Author demonstrated the remarkable results of the system in normalizing conjugation words and spelling variations. This system, addresses these challenges by exploring the distinguish by training of candidate generators. More interestingly, author developed a binary clustrization that, when given a source string “s”, finalized whether a candidate “t” is included in the candidate set or not. This approach appears straightforward, but it must solve two practical issues.

   In 2009, “Real-Word Spelling Correction using Google Web 1T 3-grams” [6] authors Aminul Islam and Diana Inkpen worked on real-word spelling correction can abruptly be cauterized into two simple module: methods based on semantic information or human-made lexical resources, and modules based on the machine learning or probability information. In this system falls into the latter category. The system method first regulate some predictable candidates and then finds the best one among the candidates or sorts them based on some weights. Author consider a string similarity function and a normalized frequency value function in their method. The following sections present a detailed description of each of these functions followed by the procedure to determine some probable candidates along with the procedure to sort the candidates.
In 2009, [7] “Fast Algorithms for Top-k Approximate String Matching”, by Zhenglu Yang, Jianjun Yu and Masaru Kitsuregawa, they introduce various efficient strategies for top-k approximate string search. The sum filtering and length-aware process are adapted to grab the top-k similar search issue, while the adaptive q-gram variety is employed to improve extra the performance of frequency counting. Based on these strategies, author propose two efficient algorithms in a general q-gram based framework [9]. Author conduct comprehensive experiments on three real data sets, and calculate the querying efficiency of the proposed approaches in terms of several aspects, such as the size of dataset, value of k, and q-gram dictionary selection. Author experimental results show that the system algorithms exhibit a superior performance.

In 2011[8] “Correcting Different Types of Errors in Texts” author Aminul Islam and Diana Inkpen they proposed method determines some probable candidates and then sorts those candidates. Author consider three similarity functions and one frequency value function in author’s method. One of the same functions, namely the string similarity function, is used to determine the candidate texts. The frequency value function and all the other similarity functions are used to sort the candidate texts.

In 2014, [1] “A Probabilistic Approach to String Transformation”, authors Ziqi Wang, Gu Xu, Hang Li, and Ming Zhang. They empirically calculated their method in spelling error correction of queries and reconstruction of queries in web search. The experimental results on two problems demonstrate that their method consistently and significantly performs improved than the baseline process of generative model and logistic regression model in terms of correctness and effectively. Authors have also applied their method to the Microsoft Speller Challenge and found that their approach can achieve a performance equivalent to those of the better performing systems in the challenge[10]. After observing these system, author work on query reformulation as an example of string transformation and they employ a more sophisticated model [11][12].

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Authors</th>
<th>Work done</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Kristina Toutanova Robert, C. Moore</td>
<td>Proposed a method which is using word pronunciation information to better spelling correction accuracy. It also substantially minimizes the error rate of the existing best spelling correction model.</td>
</tr>
<tr>
<td>2.</td>
<td>Farooq Ahmad, Grzegorz Kondrak</td>
<td>The proposed EM algorithm is efficient to learn an accurate error model without depending on a corpus of paired strings. The edit probabilities regulated using the EM algorithm are similar to error models existing and generated using other approaches.</td>
</tr>
<tr>
<td>3.</td>
<td>Naoaki Okazaki, Sophia Ananiadou, Yoshimasa Tsuruoka, Jun’ichi Tsujii</td>
<td>Proposed approach upgrade the detection recall or correction recall, while managing the respective precisions as high as possible are beneficial to the human correctors who post-edit the result of real-word spell checker. If there is not postediting, at least more errors get corrected automatically.</td>
</tr>
<tr>
<td>4.</td>
<td>Aminul Islam, Diana Inkpen</td>
<td>Proposed a system which consist of two algorithms in a general framework by combining those strategies. Extensive experiments are conducted onto three real data sets, the results show that their approaches which can efficiently answer the top-k queries of string.</td>
</tr>
</tbody>
</table>
5. Zhenglu Yang, Jianjun Yu, Masaru Kitsuregawa

Proposed a system, Recognition (OCR) errors, which automatically-mark (based on grammar and semantics) subjective examination papers, etc.

6. Ziqi Wang, Gu Xu, Hang Li, Ming Zhang

Proposed a new approach of statistical learning for string transformation. This method is novel in its model and unique, learning algorithm, and a string generation algorithm. Two different applications are addressed with their novel method, query reformulation in web search and namely spelling error correction of queries.

<table>
<thead>
<tr>
<th>TABLE I EXISTING WORK</th>
</tr>
</thead>
<tbody>
<tr>
<td>3. Proposed system</td>
</tr>
</tbody>
</table>

The proposed system contains of five modules are as follows:

a) First module - handle the input string to be entered by the end user.
b) Second module - will immediately check whether the entered string is correct with respect to syntax and semantics.
c) Third module - will suggest corrected spellings with respect to user,s query or string.
d) Fourth module - will retrieve the „n” possible outcomes of user,s query or string.

d) 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 efficiently and smoothly. The overview of system is as shown in the fig. 1. There are two processes, generation and learning. In the process of learning, first rules are extracted from training pairs of string. After that the string transformation model is developed using the learning system, consisting of weights and rules. So the process of generation, a new input given as a string, the system of generation produces the top k output candidates string by using to the model (rules and weights) which is stored in the rule index. In this process, the model is a log linear model representing the weights and rules, the learning is determined by maximum similar estimation on the training data, and the generation is conducted effectively with top „k’ pruning.

![Proposed basic model](image)

Without avoiding of simplification, we presume that the large number of rules can be applicable to „n” input string predefined. As a result, the number of possible outcomes for an input string can be generated. This is reasonable because the dissimilarity between an input string and output string shouldn’t be so enormous. In error correction of spelling, for example, the number of potential spelling errors in a word can be rather small. In proposed system will use String searching algorithms, known as string matching algorithms, are a notable class of string algorithms those will try to find out a place where one or several strings are found within a larger string or text. A search session while searching some information from web which comprises of a sequence of queries from the similar user within a less time period. Many of search sessions in data contains of misspelled queries as well as their corrections. When the algorithm extends a node, it results all the dictionary entries that end at the current character place in the input text. That is up to printing every node touched by following the dictionary beginning from that node, suffix links, and continuing until it reaches a node without dictionary suffix link. After that node that one is printed, if it is an entry of dictionary.

**Pruning** is a method in machine learning that decreases the bulk of decision trees by
removing sections of the tree which provide little power to categorize objects. The another objective of pruning is reducing the complexity of the last classifier as well as better predictive correctness by the decrease of overfitting and elimination of sections of a classifier that can be based on noisy or wrong data. Pruning of noisy data is important to fully automate the process of data grouped and learning for object recognition. The labours in this way have been focused on creating complicated features, particularly with respect to the targeted area. Our key idea is that automatic elimination of non-relevant and possible outcomes can be done without incorporating very complicated steps. Therefore in this way we can reduce the ‘n’ possible outcomes at a certain level so that what will remain will be the perfect one.

4. Conclusion

After discussing above mentioned points now, we proposed our contribution what we wish to do in our proposed system. An internet connection is require for all information retrieval applications in live state. So, our one proposed system is to store those input strings and gives possible outcome with respect to that query and even if the system is in offline mode an end user can find out the possible outcomes. Even for reducing the time required to form the possible outcomes we will mainly focus on the string entered by user. So our system should check the spelling of that string within a less time. 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 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.

10. References


