The International Journal Of Engineering And Science (IJES)
|| Volume || 3 || Issue || 5 || Pages || 13-16 || 2014 ||
ISSN (e): 2319 – 1813 ISSN (p): 2319 – 1805

An Advanced Optimistic Approach for String Transformation

[1,] Amith Kumar V Pujari, [2,] Prof. H R Shashidhar
[1,] M.Tech Scholar
RNS Institute of Technology Bangalore, Karnataka, India
[2,] Asst. Prof. Dept of CSE
RNS Institute of Technology Bangalore, Karnataka, India

ABSTRACT
String transformation can be formalized as the part of coding bio-informatics, information retrieval, and in data mining. However, in part of our paper we focus generating the k most likely output strings corresponding to the input string. This concept proposes a novel and probabilistic approach to string transformation, which is both accurate and efficient. The approach mainly focuses on three methods dynamic programming, modeling and pruning. In the dynamic programming we use the concept of the edit distance which is dynamic programming tool that estimates the score to each transformation so that the probability in the linear model can be estimated well. In the linear model, a modeling method for training the model, and an algorithm for generating the top k candidates, whether there is or is not a predefined dictionary. The linear model is defined as a conditional probability distribution of an output string and a rule set for the transformation conditioned on an input string. The learning method employs maximum likelihood estimation for parameter estimation. The string generation algorithm based on pruning is guaranteed to generate the optimal top k candidates. The proposed method is applied to correction of spelling errors in queries as well as reformulation of queries on dataset. Experimental results on large scale data show that the proposed approach is good and efficient improving upon existing methods in terms of accuracy and efficiency in different settings

KEYWORDS—*String Transformation, Edit distance algorithm , Linear Model, Aho-Corasick trees, Spelling Error Correction, Query Reformulation.*

Date of Submission: 19 May 2014 Date of Publication: 30 May 2014

I. INTRODUCTION

The string transformation is an essential problem in many applications. In natural language processing, pronunciation generation, spelling error correction, word transliteration, and word stemming can all be formalized as string transformation. String transformation can also be used in query reformulation and query suggestion in search. In data mining, string transformation can be employed in the mining of synonyms and database record matching.

As many of the above are online applications, the transformation must be conducted not only accurately but also efficiently. String transformation can be defined in the following way. Given an input string and a set of operators, one can transform the input string to the k most likely output strings by applying a number of operators. Here the strings can be strings of words, characters, or any type of tokens. Each operator is a transformation rule that defines the replacement of a substring with another substring.

The likelihood of transformation can represent similarity, relevance, and association between two strings in a specific application. Although certain progress has been made, further investigation of the task is still necessary, particularly from the viewpoint of enhancing both accuracy and efficiency, which is precisely the goal of this work.

String transformation can be conducted at two different settings, depending on whether or not a Dictionary is used. When a dictionary is used, the output strings must exist in the given dictionary, while the size of the dictionary can be very large. Without loss of generality, one can specifically studies correction of spelling errors in queries as well as reformulation of queries in web search in this Concept.

In the first task, a string consists of characters. In the second task, a string is comprised of words. The former needs to exploit a dictionary while the latter does not. 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. 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".

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 (i.e., in the query) can be further leveraged to make the final candidate selection. Query reformulation in search is aimed at dealing with the term mismatch problem. For example, if the query is "NY Times" and the document only contains "New York Times", then the query and document do not match well and the document will not be ranked high.

Query reformulation attempts to transform "NY Times" to "New York Times" and thus make a better matching between the query and document. In the task, given a query (a string of words), one needs to generate all similar queries from the original query (strings of words).The operators are transformations between words in queries such as "tx"!"texas" and meaning of " ! " definition of.

II. RELATED WORK

Arasu et al. proposed a method which can learn a set of transformation rules that explain most of the given examples. Increasing the coverage of the rule set was the primary focus. Tejada et al. proposed an active learning method that can estimate the weights of transformation rules with limited user input. The types of the transformation rules are predefined such as stemming, prefix, suffix and acronym. Okazaki et al. incorporated rules into an L1-regularized logistic regression model and utilized the model for string transformation.

Later, Brill and Moore developed a generative model including contextual substitution rules. Toutanova and Moore further improved the model by adding pronunciation factors into the model. Duan and Hsu also proposed a generative approach to spelling correction using a noisy channel model. They also considered efficiently generating candidates by using a tree. However these works on string transformation can be categorized into two groups.

Some work mainly considered efficient generation of strings. Other work tried to learn the model with different approaches. However, efficiency is not an important factor taken into consideration in these methods. The existing work is not focusing on enhancement of both accuracy and efficiency of string transformation.

III. PROPOSED WORK

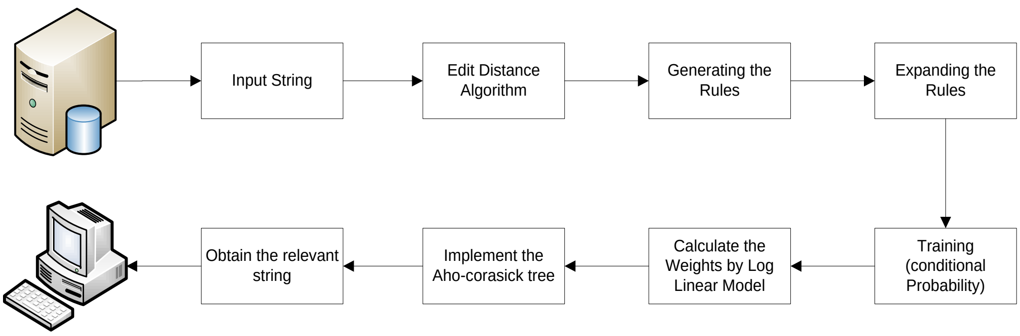

Figure 1 : Proposed Architecture

This paper addresses string transformation, which is an essential problem, in many applications. In natural language processing, pronunciation generation, spelling error correction, word transliteration, and word stemming can all be formalized as string transformation. String transformation can also be used in query reformulation and query suggestion in search. String transformation can be defined in the following way.

Given an input string and a set of operators, we are able to transform the input string to the k most likely output strings by applying a number of operators. Here the strings can be strings of words, characters, or any type of tokens. Each operator is a transformation rule that defines the replacement of a substring with another substring.

Initially we align the string by using the number of iterations obtained by using edit distance algorithm followed by the generation of the rules and expanding its rules without the loss of generality. Then a conditional probability of identifying the maximum likelihood of the strings generated for the respective strings are obtained and their weights are calculated by the linear method. Later in the generation phase, aho-corasick tree is generated arranging the rules and weights.

The linear model is defined as 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 toward the objective of generating strings with the largest likelihood given input strings.

The generation algorithm efficiently performs the top k candidate's generation using top k pruning. It is guaranteed to find the best k candidates without enumerating all the possibilities. An Aho-Corasick tree is employed to index transformation rules in the model. When a dictionary is used in the transformation, a tree is used to efficiently retrieve the strings in the dictionary.

The likelihood of transformation can represent similarity, relevance, and association between two strings in a specific application. Although certain progress has been made, further investigation of the task is still necessary, particularly from the viewpoint of enhancing both accuracy and efficiency, which is precisely the goal of this work.

IV. RESULTS AND DISCUSSIONS

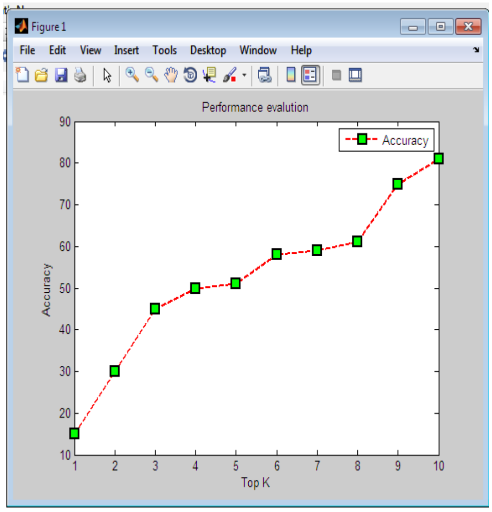

Figure 2: Accuracy Comparison of our model with a constant dataset

The above graph shows that there is significant improvement in the accuracy when the concept was implemented on a constant dataset and hence proves that the concept is very much applicable for a real-time usage. The date-set size that I have used consists of two thousand words. The above results are based on the same data-set.

V. CONCLUSION

This paper addresses string transformation, which is an essential problem, in many applications. In natural language processing, spelling error correction, word transliteration, and word stemming can all be formalized as string transformation. Our concept in general poses an additional concept of the abbreviation expansion thus covering all the functional and non-functional requirements. This concept has been implemented using a friendly interface on a pre-defined dataset and thus results obtained are as expected.

VI. ACKNOWLEDGEMENT

I take this Opportunity to express my profound gratitude and deep regards to my guide Prof. H R Shashidhar , Assistant Professor , RNS Institute of technology, Bangalore, for his exempla nary guidance, and constant encouragement throughout.

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:pp no:99 year 2013

[2]. 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.

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

[4]. J. Guo, G. Xu, H. Li, and X. Cheng, "A unified and discriminative model for query refinement," in Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, ser. SIGIR '08. New York, NY, USA: ACM, 2008, pp. 379–386.

[5]. 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.

[6]. 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.

[7]. 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.

[8]. M. Dreyer, J. R. Smith, and J. Eisner, "Latent-variable modeling of string transductions with finite-state methods," in Proceedings of the Conference on Empirical Methods in Natural Language Processing, ser. EMNLP '08. Stroudsburg, PA, USA: Association for Computational Linguistics, 2008, pp. 1080–1089.

[9]. Arasu, S. Chaudhuri, and R. Kaushik, "Learning string transformations from examples," Proc. VLDB Endow., vol. 2, pp. 514–525, August 2009.S. Tejada, C. A. Knoblock, and S. Minton, "Learning domainindependent 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.

[10]. A. V. Aho and M. J. Corasick, "Efficient string matching: an aid to bibliographic search," Commun. ACM, vol. 18, pp. 333–340, June 1975.

[11]. 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