ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 11, November 2014

A Tool for String Transformation Enabling Flexibility in Real World Applications

Gangadhar

Student, Department of CSE, RGMCET, Nandyal, India

Abstract: String is a sequence of characters whose manipulations have many real world utilities. Especially transforming a string from given input to various outputs has very important use in applications like bioinformatics, information retrieval, data mining and natural language processing. When an input string is given, the output has to be generated as multiple strings that can be used for solving problems such as query reformulation, correction of errors and so on. Recently Wang et al. proposed a probabilistic approach to string transformation. They proposed an algorithm for obtaining top-k candidates, a training model and log linear model for achieving string transformation which proved to be efficient and accurate. In this paper, we built an application, a prototype that can demonstrate the concept of string transformation. The experimental results reveal that the tool is useful in string transformation which can be used further in applications pertaining to error correction and query reformulation in online web search.

Keywords: String transformation, error correction, and query reformulation

I. INTRODUCTION

String manipulations and string transformations are widely used in all kinds of 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 is also used in data mining applications and record matching functionalities. It is widely used in online applications and e-Commerce applications for generating recommendations. Given a string as input and provided a set of operators, it is possible to convert input string into k output strings or tokens which can be used in many applications. 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

We focused on correlation of spelling errors without loss of generality. A string can have group of characters or groups of words. First of all, the given string is used as input and various rules are extracted. The rule extraction is done in the learning phase and then generation phase starts. The generated rules are used in this phase in order to generate top-k strings that have further utility in real world applications. The top k strings thus generated can be used in applications like spelling error correction, web search engines for correcting query or query reformulation and so on. The concept of candidate generation is nothing but the string transformation. The final candidate selection process produces strings which are really useful in real world applications. There are many prior works that focused on the related works [1], [2], [3], and [4].

In this paper we implemented string transformation approach using a prototype application that demonstrates taking input string and generating top k output strings that can have further utility in many applications. The remainder of the paper is structured as follows. Section II reviews literature on prior works. Section III provides the proposed system and prototype implementation. Section IV presents experimental results while section V concludes the paper.

II. RELATED WORKS

String manipulations and string transformations are widely used in all kinds of computer applications in the real world. This section reviews literature on the related works. There are many approaches to learning for string transformation as explored in [5], [6], [1], and [2]. Learning transformation rules is done in [6]. A log linear model for string transformation is demonstrated in [1]. An active learning approach that can compute weights for string transformation rules is explored in [5].

Many approximate string search methods were proposed in the literature [7], [8], [9], and [10]. There solutions used n-gram based algorithms. While trie based solution is found in [10]. For finding k-candidates using n-gram approach and with the help of similarity functions [11] and [12] can be found. Spelling error correction concept is explored in many research articles such as [13], [14] and [3]. A generative approach is followed in [13] while pronunciation model was introduced in [14]. Phrase based string transformation rules were proposed in [15] prior to generating number of candidates. Contextual substitution patterns were tried in [16] to generate string transformations from given input string.

III. PROPOSED SYSTEM AND PROTOTYPE APPLICATION

The proposed system is based on probabilistic string transformation approach [17] that is able to transform given input string into multiple output strings. This kind of

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 11, November 2014

output can be used for various real world applications such as bioinformatics, error correction and query reformulation. The prototype application built to demonstrate this is done using Java/J2EE platform. The application demonstrates two things such as learning and generation of top k candidates. The learning phase builds a model required for top k candidate generation while the generation model uses the rules that were extracted in learning phase in order to generate top-k candidates.

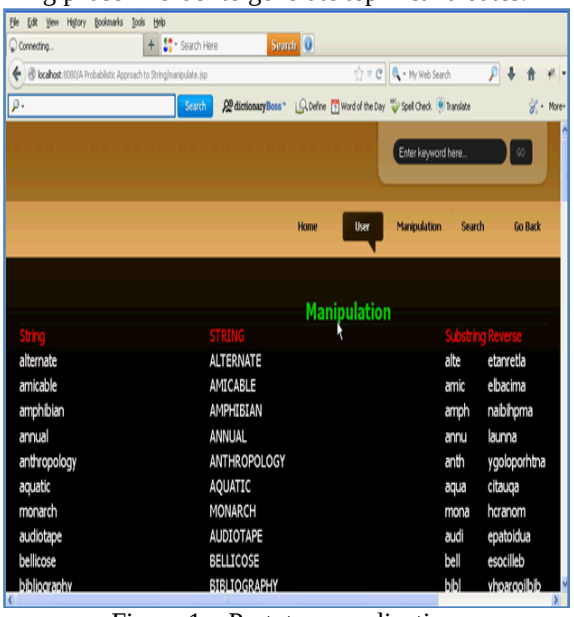

Figure 1 – Prototype application

As can be seen in Figure 1, it is evident that the application has functionalities for string transformation. The prototype application's main UI is presented which facilities to have two phases of functionality such as learning and generation of top-k candidates.

IV. EXPERIMENTAL RESULTS

Experiments are made in terms of various settings such as default settings, with different dictionary sizes and with different number of applicable rules. The top k value for candidate generation and the accuracy are considered for comparison of the approach with other approaches.

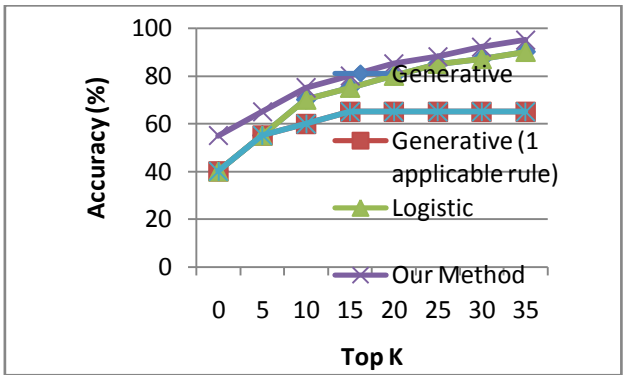

Figure 1 – Accuracy comparison with default settings

As can be seen in Figure 1, it is evident that the top k values are presented in horizontal axis while the vertical axis represents accuracy in generating top k candidates. The results revealed that the proposed approach exhibits more accuracy.

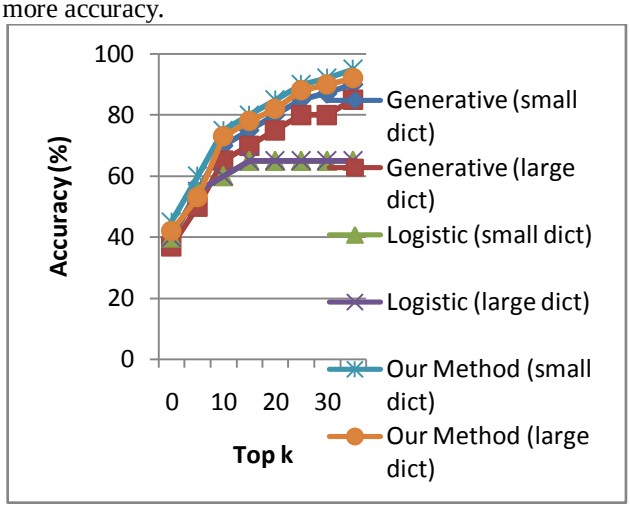

Figure 2 – Accuracy comparison with different dictionary sizes

As can be seen in Figure 2, it is evident that the top k values are presented in horizontal axis while the vertical axis represents accuracy in generating top k candidates. The results revealed that the proposed approach exhibits more accuracy. The experiments are done with different dictionary sizes.

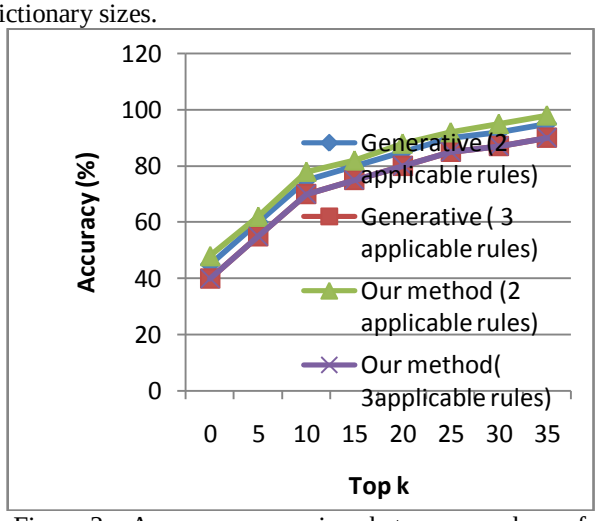

Figure 3 – Accuracy comparison between numbers of applicable rules

As can be seen in Figure 3, it is evident that the top k values are presented in horizontal axis while the vertical axis represents accuracy in generating top k candidates. The results revealed that the proposed approach exhibits more accuracy. The experiments are done with different number of applicable rules.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we studied the problem of string transformation and its real world utility and usefulness in various applications. Strings play an important role in software applications where query processing, analyzing case studies etc. are to be done using string transformations. Recently Wang et al. [17] proposed a probabilistic string transformation approach which is unique. They also proposed an algorithm for generating

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 11, November 2014

top-k candidate strings. Given an input string, the approach is to generate many output strings that can be further used in applications like error correction and query reformulations. In this paper, we built a prototype application that demonstrates this proof of concept. The empirical results reveal that the application is capable of transforming string which can be further utilized in various real world applications. In our future work we build some applications that can be integrated with the prototype built in this paper.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

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

[7] 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.

[8] 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 SIGMOD international conference on Management of data, ser. SIGMOD '08. New York, NY, USA: ACM, 2008, pp. 353–364.

[9] 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.

[10] 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.

[11] Z. Yang, J. Yu, and M. Kitsuregawa, "Fast algorithms for top-k approximate string matching," in Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, ser. AAAI '10, 2010, pp. 1467–1473.

[12] 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.

[13] 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.

[14] K. Toutanova and R. C. Moore, "Pronunciation modeling for improved spelling correction," in Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ser. ACL '02. Morristown, NJ, USA: Association for Computational Linguistics, 2002, pp. 144–151.

[15] R. Jones, B. Rey, O. Madani, and W. Greiner, "Generating query substitutions," in Proceedings of the 15th international conference on World Wide Web, ser. WWW '06. New York, NY, USA: ACM, 2006, pp. 387–396.

[16] 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.

[17] Ziqi Wang, Gu Xu, Hang Li, and Ming Zhang. (2013). A Probabilistic Approach to String Transformation. IEEE. 2 (99), p1-14.