ISSN (Online) : 2319 – 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)
On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

Top K Pruning Approach to String Transformation

A. Meenahkumary[1], V. Manjula[2], B. Divyabarathi[3], V. Nirmala[4]

Student, Department of CSE, Manakula Vinayagar Institute of Technology, Puducherry, India[1, 2, 3]

Assistant Professor, Department of CSE, Manakula Vinayagar Institute of Technology, Puducherry[4]

Abstract-- Many problems in natural language processing, data mining, information retrieval, and bioinformatics can be formalized as string transformation, which is a task as follows. Given an input string, the system generates the k most likely output strings corresponding to the input string. This paper proposes a novel and probabilistic approach to string transformation, which is both accurate and efficient. The approach includes the use of a log linear model, a method for training the model, and an algorithm for generating the top k candidates, whether there is or is not a predefined dictionary. The log 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 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.

I. INTRODUCTION

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 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, we specifically study correction of spelling errors in queries as well as reformulation of queries in web search in this paper. 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 doesnot. 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, cf.,

II. EXISTING WORK

Efficiency is not an important factor taken into consideration in these methods. Some work mainly considered efficient generation of strings, assuming that the model is given of misspelling and correction can be frequently observed in web search log data, it has been proposed to mine spelling-error and correction pairs by using search log data. The mined pairs can be directly used in spelling error correction. Methods of selecting

ISSN (Online) : 2319 – 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)
On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

spelling and correction pairs with maximum entropy model [24] and similarity functions [25], [26] have been developed. Only high frequency pairs can be found from log data, however. In this paper, we work on candidate generation at the character level, which can be applied to spelling error correction for both high and low frequency words. 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, we are able to transform the input string to the k most likely output strings by applying a number of Either efficiency is achieved or effectiveness is achieved not both. In the existing system the following algorithm are used Generative model. Logistic Regression Model.A discriminative model there are also methods for finding the top k candidates by using n-grams. Efficiency is the major focus for these methods and the similarity functions in them are predefined. In contrast, our work in this paper aims to learn and utilize a similarity function which can achieve both high accuracy and efficiency. There are two possible settings for string transformation.

One is to generate strings within a dictionary, and the other is to do so without a dictionary. In the former, string transformation becomes approximate string search, which is the problem of identifying strings in a given dictionary that are similar to an input string.

III. PROPOSED SYSTEM

In this paper, we propose a probabilistic approach to the task. Our method is novel and unique in the following aspects. It employs a log-linear (discriminative) model for string transformation, an effectiveand accurate algorithm for model learning, and an efficient algorithm for string generation.

3.1 SPELLING ERROR CORRECTION

Spelling error correction normally consists of candidate generation and candidate selection. The former task is an example of string transformation. Candidate generation is usually Sometimes a dictionary is utilized in string transformation in which the output strings must exist in the dictionary, such as spelling error correction, database record matching, and synonym mining. In the setting of using a only concerned with a single word. For single-word candidate generation, a rule-based approach is commonly used. The use of edit dictionary, we can further enhance the efficiency. Specifically, we index the dictionary in a trie, such that each string in the dictionary corresponds to the path from the root node to a leaf node. When we expand a path (substring) in candidate generation, we match it against the trie, and see whether the expansions from it are legitimate paths. If not, we discard the expansions and avoid generating unlikely candidates. In other words, candidate generation is guided by the traversal of the trie.

3.2 QUERY REFORMULATION

Query reformulation involves rewriting the original query with its similar queries and enhancing the effectiveness of search. Most existing methods manage to mine transformation rules from pairs of queries in the search logs. One represents an original query and the first identifies phrase-based transformation rules from query pairs, and then segments the input query into phrases, and generates a number of candidates based on substitutions of each phrase using the rules. The weights of the transformation rules are calculated based on log likelihood ratio. A query dictionary is used in this case

3.3 EFFICIENT DICTIONARY MATCHING ALGORITHM

Sometimes a dictionary is utilized in string transformation in which the output strings must exist in the dictionary, such as spelling error correction, database record matching, and synonym mining.

IV. CONCLUSION

In this paper, we have proposed a new statistical learning approach to string transformation. Our method is novel and unique in its model, learning algorithm, and string generation algorithm. Two specific applications are addressed with our method, namely spelling error correction of queries and query reformulation in web search. Experimental results on two large data sets and Microsoft Speller Challenge show that our method improves upon the baselines in terms of accuracy and

ISSN (Online) : 2319 – 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)
On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

efficiency. Our method is particularly useful when the-problem occurs on a large scale.

REFERENCES

[1] M. Li, Y. Zhang, M. Zhu, and M. Zhou, "Exploring distributional similarity based models for query spelling correction," in Proceed-ings 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.

[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] 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.

[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] 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.

[6] N. Okazaki, Y. Tsuruoka, S. Ananiadou, and J. Tsujii, "A discrim-inative candidate generator for string transformations," in Pro-ceedings 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] 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 Com-putational Linguistics, 2008, pp. 1080–1089.

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

[9] 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.

[10] M. Hadjieleftheriou and C. Li, "Efficient approximate search on string collections," Proc. VLDB Endow., vol. 2, pp. 1660–1661, August 2009.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

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

[16] 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.

[17] 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.

[18] E. S. Ristad and P. N. Yianilos, "Learning string-edit distance," IEEE Trans. Pattern Anal. Mach. Intell., vol. 20, pp. 522–532, May 1998.

[19] J. Oncina and M. Sebban, "Learning unbiased stochastic edit distance in the form of a memoryless finite-state transducer," in In Workshop on Grammatical Inference Applications: Successes and Future Challenges, 2005.

[20] A. McCallum, K. Bellare, and F. Pereira, "A conditional random field for discriminatively-trained finite-state string edit distance," in Proceedings of the 21st Conference on Uncertainty in Artifical Intelligence, ser. UAI '05, 2005, pp. 388–395.