ISSN:0975-9646

String Transformation Using Top K Pruning Algorithm

Ms. Swapnali S. Maske[1], Mr. Prashant Jawalkar[2]

[1]*ME student,Department of computer engineering, JSPM's BSIOTR, wagholi, Pune.*

[2]*Assistant professor, Department of computer engineering, JSPM's BSIOTR, wagholi, Pune.*

Abstract-In this paper method is developed to solve two problems, spelling error correction and query reformulation of queries in web search. A search session in web search is comprised of a sequence of queries from the same user within a short time period. Many of search sessions in our data consist of misspelled queries and their corrections. We employed heuristics to automatically mine training pairs from search session data. Efficiency is vital for this task due to dictionary is extremely large and the response time must be very short. Proposed approach is very accurate and efficient improving upon existing methods in terms of accuracy and efficiency in different settings. Top k pruining algorithm is used to generate most likely spelling error correction.

1. INTRODUCTION

In String transformation, given an input string and set of operators we transform the input string into most likely output strings. Operator is nothing but transformation rule that defines replacement of input string with output string. Likelihood represents similarity, association and relevance between input and output strings. Precisely goal of this work is improving efficiency and accuracy. Work is divided into two tasks spelling error correction and query reformulation. In spelling error correction there are two steps candidate generation and candidate selection. In candidate generation most likely corrections for misspelled words are given and in query reformulation. In previous work on string transformation efficiency is not an important factor taken into consideration. In contrast, our work in this paper develops a model for string transformation which can achieve both high accuracy and efficiency. There are three fundamental problems with string transformation: (1) how to define a model which can achieve both high accuracy and efficiency, (2) how to accurately and efficiently train the model from training instances, (3) how to efficiently generate the top k output strings given the input string, with or without using a dictionary. The log linear model gives conditional probability distribution of an output string and a rule set given an input string. The learning estimates maximum likelihood. Thus, the model is trained toward the objective of generating strings with the largest likelihood given input strings. Top k pruining algorithm efficiently generates top k candidates. The experimental results on the two problems demonstrate that our method consistently and significantly performs better than the baseline methods of generative model and logistic regression model in terms of accuracy and efficiency.

1.1 Model for string transformation

String transformation model is proposed as shown in following figure in which there are three main phases: learning phase, generation phase and selection phase.

1.1.1 Learning Phase

Rule set is primary focus on learning phase. Here weights are estimated for transformation rules with user input. The type of transformation rules are stemming, prefix, suffix and acronym. Our model is designed for both accurate and efficient string transformation, with transformation rules and weight.

Model: Model consist of rules and weights. A rule is represented as $\alpha \rightarrow \beta$ which denotes an operation of replacing substring α in the input string with substring β, where $\alpha,\beta \epsilon$ {s/s= t, s=^t, s=t$, or s=^t$}Where ^ and $ are the start and end symbols respectively.

Training Model: Training data is given as a set of pairs T=sij,sojj=1N, Where sij is input string and soj is output string. We define likelihood on the basis of conditional probability of output strings given input strings.

1.1.2 Generation Phase

In this phase we generate most likely k output strings.

1.1.3 Selection Phase

In selection phase candidates are selected which having highest weights.

1.1.4 Spelling error correction

Spelling error correction is divided into two tasks: candidate generation and candidate selection. Brill and moore previously developed generative model but in this project we are using discriminative model which can work better than generative model because it is trained for enhancing accuracy. In this paper, we work on candidate generation, which can be applied to spelling error Correction for both high and low frequency words.

1.1.5 Query Reformulation

Query reformulation rewrites the original query with its similar queries and enhances the effectiveness of search. The weights of the transformation rules are calculated based on log likelihood ratio. 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 transforms NY Times to New York Times and makes matching between the query and document.

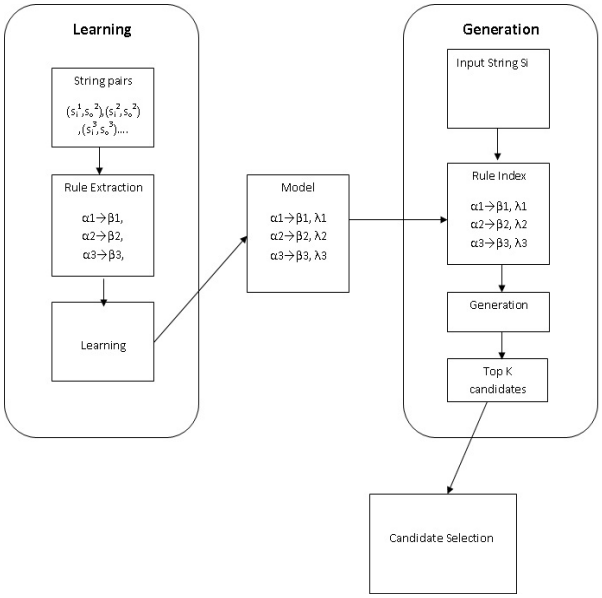

Fig 1. String transformation model

2. METHODOLOGY

In this paper we employ top k pruning technique to achieve efficiency in string transformation. It eliminates unlikely paths thus improve efficiency. Steps for top k pruning algorithm are given as follows:

Input: rule index Ir, input string s, candidate number k

Output: top k output strings in Stopk

Begin

Find all rules applicable to S from Ir with Aho corasic algorithm

Minscore=-1

Qpath=Stopk

Add (1,^,0) into Qpath

While Qpath is not empty do

Pickup a path (pos,string,score) from Qpath with heuristics

If score _ minscore then

Continue If pos== |Sj|AND string reaches $ then

If |Sk| <= k then

Remove candidate with minimum score from Stopk

Add candidate (string, score) into Stopk

Update minscore with minimum minimum score in Stopk

Foreach next substring c at pos do

α→β= corresponding rule of c

pos=pos+|α|

string=string+ β

score=score+ λα→β

Add(pos,string,score) into Qpath

If(pos,string,score)in Qpath then

Drop the path with similar score

Return Sk

In this algorithm a triple (pos, string, score) is used to denote each path generated corresponding to the position, the content and the score. Qpath is a priority queue. It stores paths, and is initialized with path (1, ^, 0). Stopk is a set that stores k candidates and scores (string, score). The algorithm picks up one path from Qpath each time and expands the path from its current position. Path is popped up from the priority queue when one path is processed. The algorithm uses the top k pruning strategy to eliminate unlikely paths and improve efficiency. If the score is smaller than the minscore of the top k list Stopk, then the path is discarded. The path with larger score is kept.

Advantages:

The algorithm uses top k pruning strategy to eliminate unlikely paths thus improves efficiency.

3. RESULTS AND DISCUSSION

To improve efficiency and accuracy is goal of this system. The system we developed in this study is evaluated using the following three metrics.

3.1.1 Accuracy

The number of correct outputs generated by the system divided by the total number of queries in the test set. AccuracyTopN(N=1,5,10,25,100)=Number of answers whose answer is in TopN/number of total samples

3.1.2 Precision

The number of correct spelling corrections for misspelled Queries generated by the system divided by the total number of corrections generated by the system.

Precision= Number of valid correction/(Number of valid correction + number of bad correction)

3.1.3 Recall

The number of correct spelling corrections for misspelled queries generated by the system divided by the total number of misspelled queries in the test set.

Recall= Number of valid correction/(Number of valid correction +Number of no correction + number of bad correction).

Following figure shows the results of accuracy and efficiency compared with the default setting. Default settings given as: 973,902 words in the dictionary, 10,597 rules for correction, and up to two rules used in one transformation. We made use of 100,000 word pairs mined from query sessions for training, and 10,000 word pairs for testing.

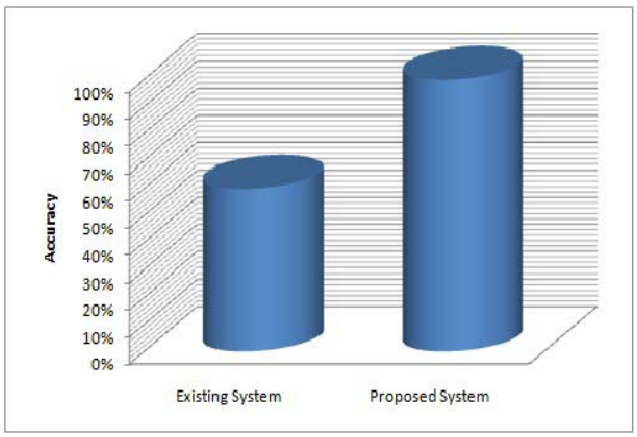

Fig. 2. Accuracy comparison between baselines and our method with Default settings.

The experiental results in Fig. 2. shows that our method always performs better compared with the baselines. As k increases the performance of logistic becomes saturated, because the method allows the use of one rule each time. We observe that there are many word pairs in the data that need to be transformed with multiple rules. We compared three methods by using one rule. Our method works better than the baselines, especially when k is small. The experimental results in Fig.3. shows that running time of our method is remarkably less than generative and logistic method. So we can conclude that our method works efficiently with top k pruning strategy.

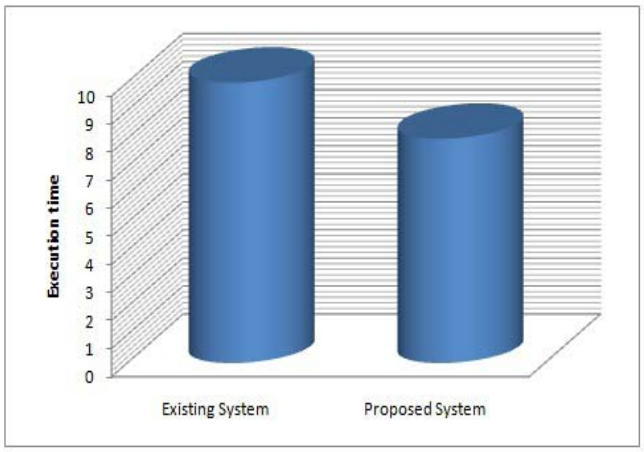

Fig.3. EEfficiency comparison between baselines and our method with Default settings.

4. CONCLUSION

In this paper we have proposed a new statistical method for string transformation. This method is unique in its model, learning algorithm and string transformation algorithm. Two specific applications are addressed with this method namely spelling error correction and query reformulation in web search. This paper focuses on accuracy and efficiency of string transformation. This method is particularly useful when the problem occurs on a large scale.

REFERENCES

[1] Ziqi Wang, Gu Xu, Hang Li, and Ming Zhang, "A Probabilistic Approach to String Transformation", in IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 26, NO. 5, pp. 1063-1075.

[2] M. Li, Y. Zhang, M. Zhu, and M. Zhou, "Exploring distributional similarity based models for query spelling correction" ,in Proc. 21st Int. Conf. Computational Linguistics and the 44th Annu. Meeting Association for Computational Linguistics, Morristown, NJ, USA, 2006, pp. 10251032.

[3] H. Duan and B.-J. P. Hsu, "Online spelling correction for query completion" , in Proc. 20th Int. Conf. World Wide Web, New York, NY, USA, 2011, pp. 117126.

[4] J. Guo, G. Xu, H. Li, and X. Cheng, "A unified and discriminative model for query refinement" , in Proc. 31st Annu. Int. ACM SIGIR Conf. Research Development Information Retrieval, New York, NY, USA, 2008, pp. 379386.

[5] A. Behm, S. Ji, C. Li, and J. Lu, "Space-constrained gram-based indexing for efficient approximate string search" , in Proc. 2009 IEEE Int. Conf. Data Engineering, Washington, DC, USA, pp. 604 615.

[6] N. Okazaki, Y. Tsuruoka, S. Ananiadou, and J. Tsujii, "A discriminative candidate generator for string transformations" , in Proc. Conf. Empirical Methods Natural Language Processing, Morristown, NJ, USA, 2008, pp. 447456.

[7] E. Brill and R. C. Moore, "An improved error model for noisy channel spelling correction",in Proc. 38th Annual Meeting Association for Computational Linguistics, Morristown, NJ, USA, 2000, pp. 286293.

[8] M. Dreyer, J. R. Smith, and J. Eisner, "Latent-variable modeling of string transductions with finite-state methods" ,in Proc. Conf. Empirical Methods Natural Language Processing, Stroudsburg, PA, USA, 2008, pp. 10801089.

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

[10] 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. 350359.

[11] M. Hadjieleftheriou and C. Li, "Efficient approximate search on string collections" , Proc. VLDB Endow., vol. 2, pp. 16601661, August 2009.

[12] 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. 303314.

[13] 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. 353364.

[14] A. R. Golding and D. Roth, "A winnow-based approach to contextsensitive spelling correction", Mach. Learn., vol. 34, pp.107130, February 1999.