INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN 2278 – 0211 (Online)

# A Log Linear Probabilistic Model for String Transformation Using Non Dictionary Approach

**Akshay Kumar N.**
M.Tech Scholar, PESIT, Bangalore, Karnataka, India

***Abstract:***
*A lot of problems in natural language processing, data mining, information retrieval, and bioinformatics can be legitimated as string transformation.The task of the string transformation is once the input string is given, the system generates the k most likely occuring output strings resultant to the input string.So this paper proposes a novel and probabilistic approach to string transformation which includes the use of a log linear model, a training method for the model and an algorithm for generating the top k candidates using a non-dictionary approach which helps the approach to be accurate as well as efficient. The log linear model can be stated as a conditional probability distribution of an output string along with a rule set for the transformation conditioned on an input string. The learning method employs maximum likelihood estimation for parameter estimation. The string generation is based on pruning algorithm which is guaranteed to generate the optimal top k candidates. The proposed method is applied to correction of spelling errors in string or queries.*

***Key words****: String Transformation, Log Linear Model, Spelling Error Correction*

## 1. Introduction

This paper addresses string transformation, which is an essential problem, in many applications. String transformation can be formulated to natural language processing, pronunciation generation, spelling error correction and word stemming. String transformation can be in use in mining of synonyms and database record matching in data mining. Since many of the applications are online based applications string transformation must be accurate as well as efficient. string transformation can be defined as one can transform the input string to k most likely output strings by applying a number of operators, once the input string and the set of operators is given. String can be words, characters and also tokens. The operator can be a transformation rule that defines the replacement of a substring with another substring by the likelihood of transformation which represents similarity, relevance and association between two strings.

String transformation can be performed in two different ways

- Using dictionary
- Using non-dictionary

In the first method, a string consists of characters. In the second method, a string is comprised of words. Spelling errors is of two types

- Typing or writing error
- Word error.

Typing or writing errors are those that are not in dictionary. They are majorly caused while typing or writing a word. For example perpul -> purple. Word errors are those that are in dictionary. For example piece -> peace. In this paper we majorly focus on typing or writing errors which includes non-dictionary approach.

Spelling errors in queries can be corrected using the following two steps: Candidate generation and Candidate selection. Candidate generation is used to find the words with similar spelling. In this kind of a case, a string of characters is input and the operators represent insertion, deletion, and substitution of characters with or without surrounding characters. They are done by using edit distance to error. Candidate generation is concerned with a single word. After candidate generation, the words in this perspective can be further leveraged to make the final candidate selection.

Efficiency is not an important factor taken into consideration in existing methods. Some work considered efficient generation of strings, assuming that the model is given [2]. Other work tried to learn the model with different approaches, such as a generative model [3], a logistic regression model [4] and a discriminative model [5]. 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 train the model accurately

and efficiently from training instances, (3) how to efficiently generate the top k output strings given the input string, with or without using a dictionary.

**2. Related work**

String transformation can be applied in many applications like data mining, natural language processing, information retrieval, and bioinformatics. The major difference between our work and the existing work is that we focus on enhancement of both accuracy and efficiency of string transformation.

String transformation is about generating one string from another string. Arasu et al. [6] proposed a method which can learn a set of transformation rules. The primary focus was on increasing the coverage of the rule set. Tejada et al. [7] 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 etal. [4] incorporated rules into an $L_1$-regularized logistic regression model as well as utilized the model for string transformation. Dreyer et al. [5] also proposed a log linear model for string transformation, with features representing latent alignments between the input and output strings.

**3. Proposed work**

In this paper, we propose a probabilistic approach to the task. Our method is novel and unique in the following aspects as shown in fig 1. It employs (1) a log-linear (discriminative) model, (2) an effective and accurate algorithm for model learning, and (3) an efficient algorithm for string generation. The log linear model can be 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 candidates generation using top k pruning. To find the best k candidates pruning is guaranteed without enumerating all the possibilities.

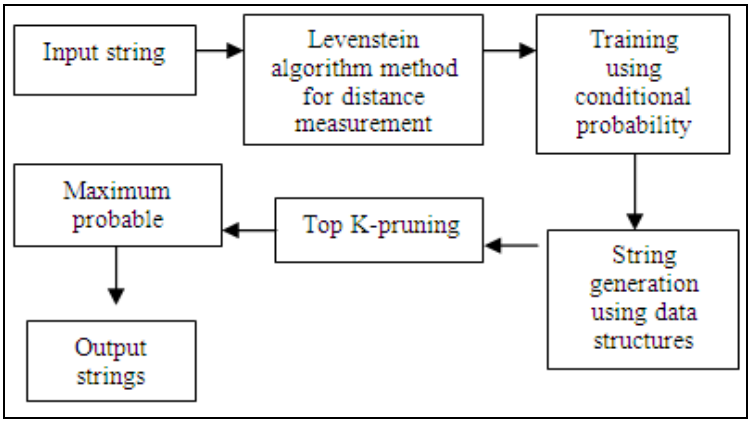


*Figure 1: Proposed architecture*

*3.1. Levenshtein Distance*

Levenshtein distance or evolutionary distance is a concept from information retrieval and it describes the number of edits (insertions, deletions and substitutions) that have to be made in order to change one string to another. It is the most common measure to expose the dissimilarity between two strings.

To compute the Levenshtein distance *ed(x,y)* between strings *x* and *y*, a matrix *M* 1...*m*+1,1...*n*+1 is constructed where *M* is the minimum number of edit operations needed to match *x 1...i* to *y1...j* Each matrix element *M i,j* is calculated as per Equation , where δ (a , b) = 0 if *a* = *b* and 1 otherwise. The matrix element $M_{1,1}$ is the Levenshtein distance between two empty strings.

$$M_{1,1} \leftarrow 0$$

$$M_{i,j} \leftarrow \min \begin{cases} M_{i-1,j} + 1 \\ M_{i,j-1} + 1 \\ M_{i-1,j-1} + \delta(x_i, y_j) \end{cases}$$

*3.2. Log-Linear Model*

A log-linear model consists of the following components:

- A set X of possible inputs.
- A set Y of possible labels. The set Y is assumed to be finite.
- A positive integer d specifying the number of features and parameters in the model.

- A function f : X × Y → $R^d$ that maps any (x, y) pair to a feature-vector f(x, y).
- A parameter vector v Є $R^d$.

For any x Є X, y Є Y, the model defines a conditional probability

$$p(y|x;v) = \frac{\exp(v \cdot f(x,y))}{\sum_{y' \in \mathcal{Y}} \exp(v \cdot f(x,y'))}$$

Here exp(x) = $e^x$, and v*f(x; y) = $\sum_{k=1}^{d} v_k f_k(x,y)$ is the inner product between v and f(x; y). The term p(y|x, v) is intended to be read as "the probability of y conditioned on x, under parameter values v".

$$p(y|x;v) = \frac{\exp(v \cdot f(x,y))}{\sum_{y' \in \mathcal{Y}} \exp(v \cdot f(x,y'))}$$

**4. Conclusion**

Thus in this paper we have tried to reduce the problems with information processing by making use of a new statistical learning approach to string transformation**.** Our method is novel as well as unique, providing more accuracy and efficiency in specific application like spelling error correction. Our method is particularly more useful when the problem occurs on a large scale datasets.

**5. Acknowledgement**

I take this opportunity to express my profound gratitude and deep regards to my guide Ms. Sangeetha. J, Associate Professor, Dept of ISE, PESIT, Bangalore, for her exemplary guidance and constant encouragement throughout.

**6. References**

1. Ziqi Wang, Gu Xu, Hang Li and Ming Zhang," A Probabilistic Approach to String Transformation", in Proceedings of the 2013 IEEE Transactions on Knowledge and Data Engineering, vol. 26, pp. 1-14.
2. 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.
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. 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.
5. 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.
6. Arasu, S. Chaudhuri, and R. Kaushik, "Learning string trans- formations from examples," Proc. VLDB Endow., vol. 2, pp. 514– 525, August 2009.
7. 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