www.ijecs.in

International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 4 Issue 7 July 2015, Page No. 13620-13625

Survey on String Transformation - A Query Based Approach

Shijina J Salim[1], Diliya M Khan[2]

[1]M.Tech CSE, LBSITW, Poojappura,
Thiruvananthapuram, Kerala University, India
shijina.j.s@gmail.com

[2]Assistant Professor, Dept. of IT, LBSITW, Poojappura,
Thiruvananthapuram, Kerala University, India
diliyakhan00@gmail.com

Abstract: ***Data mining is a powerful area, which computerizes the process of searching valuable information from a large database. Its wide range of applications promises a future, where the data grow rapidly. Many problems in natural language processing, data mining, information retrieval, and bioinformatics can be formalized as string transformation. Proposed system implements string transformation in data mining field with the help of efficient algorithms. As its name implies string transformation includes a set of operators to transform a given string into most likely output strings. Insertion, deletion, transposition, and substitution are the operators for transformation. Transformation rules and predefined rule indexes are used here to avoid unwanted searches and time delay. Here the users can view the formation of possible outcomes from the given string. By extracting these, proposed system finds most appropriate matches with respect to the given string and gives them as output within seconds. Another important feature is to provide query reformulation. By using efficient methods, proposed system can introduce query reformulation with useful description about the given query. Query reformulation is also a transformation technique and it deals with the term mismatch problem. Here similar query pairs can mine from training data. Proposed system tries to transform a given query to original query and therefore make a better match between the query and the document and also give a brief description about this like a search engine. Challenge is compounded by the fact is that new information from the field is being added to the database on a daily basis. For this purpose, proposed system use a dictionary method to add details to the database and the information retrieved by text mining approach. Text mining is a new area of computer science and a sibling of data mining which fosters strong connections with data mining and knowledge management. Proposed system is an efficient system and need less time for the retrieval of data.***

Keywords: String transformation, query reformulation

1. Introduction

Data mining is a step in the process of knowledge discovery from data (KDD). KDD concerns the acquisition of new, important, valid and useful knowledge from data. Data-mining tools promise to discover knowledge. It is a proactive process that automatically searches data for new relationships and anomalies on which to base business decisions in order to gain competitive advantage. Although data mining might always require some interaction between the investigator and the data-mining tool, it may be considered as an automatic process because 'data mining tools automatically search the data for anomalies and possible relationships, thereby identifying problems that have not yet been identified by the end user', while mere data analysis' relies on the end users to define the problem, select the data, and initiate the appropriate data analyses to generate the information that helps model and solve problems they uncovered'. Traditional query and report tools have been used to describe and extract what is in a database. The user forms a hypothesis about a relationship and verifies it or discounts it with a series of queries against the data. For example, an analyst might hypothesize that people with low income and high debt are bad credit risks and query the database to verify or disprove this assumption. Data mining can be used to *generate* a hypothesis.

String transformation is an essential problem in many applications. 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. The likelihood of transformation can represent similarity, relevance, and association between two strings in a specific application. Reformulation of queries in search is aimed at addressing the problem of mismatch. For example, if the query is "IOC" and the document only contains "Indian Oil Corporation", the query and the document does not fit well and the document does not classified high. Query Reformulation tries to transform "IOC" the "Indian Oil Corporation" and therefore make a better match between the query and the document.

2. Literature Review

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and resource availability. For this, a detailed study is need about the topic. Reference papers and websites have important role to make a successful analysis and plan the possible solutions. Once these things are satisfied, then next steps are to determine which operating system and language can be used for developing the tool. At the time of building the tool we need support from guide, reference papers, and books or from websites. So before building the proposed system, there need a deep study and a strong literature survey.

2.1 Online Spelling Correction for Query Completion

This model proposes a transform based transformation model that is capable of capturing users spelling behavior. Also estimate the transformation model using clicks on search engine recourse links, which represent user confirmed query misspellings. Search algorithm used here is configured to deal with partial queries, so that online search is possible. In this paper, they model search queries with a generative model, where the intended query is transformed through a noisy channel into a potentially misspelled query. The distribution from which the target query is selected is estimated from the search engine query log based on frequency. Thus, they are more likely to suggest more popular queries. For the noisy channel, this describes the distribution of spelling errors.

Main problem with this function is that it does not deal the untransformed part of the input query. Therefore, this can design a better heuristic by taking into consideration the upper bound of the transformation probability. Advantages of it are making use of both absolute pruning and relative pruning method to improve the search efficiency and accuracy, user can add or drop letters unintentionally by using the process and suggesting algorithm is not only accurate, but also efficient here. Disadvantages of it are inside the process need suggestions incurs a cost, as users spend more time looking at them instead of completing their task, irrelevant suggestions risks annoying users to terminate the process, online correction has many merits that cannot be achieved by offline correction and the algorithm is not sufficiently robust and scalable for online spelling correction for query completion.

2.2 A Unified and Discriminative Model for Query Refinement

This Model describes a new CRF model for performing query refinement, called CRF-QR. The model is unique in that it predicts a sequence of refined query words as well as corresponding operations given a sequence of query words. And show that employing a unified and discriminative model in query refinement is effective. They propose exploiting a unified and discriminative model in query refinement, specifically conducting various query refinement tasks in a unified framework, and employing a special model called CRF-QR to accomplish the tasks. One advantage of employing this model is that the accuracy of query refinement can be enhanced. This is because the tasks of query refinement are often mutually dependent, and need to be addressed at the same time. Lexicon-based feature representing whether a query word or a refined query word is in a lexicon or a stop word list. Position-based feature representing whether a query word is at the beginning, middle, or end of the query. Corpus-based feature representing whether the frequency of a query word or a query word in the corpus exceeds a certain threshold. Query-based feature representing whether the query is a single word query or multi-word query. Pros of this method are progress will get guaranteed that the global optimal solution will be found because the log-likelihood function is convex and heuristics used to reduce the number of possible sequences inside the process. Cons are efficiency of this model performance is comparatively low than our string transformation system and the use of the basic model is not enough for the correction and stemming process. Word stemming is not easy to make a refinement judgment, because the effectiveness of a refinement also depends on the contents of document collection.

2.3 Space-Constrained Gram-Based Indexing For Efficient Approximate String Search

This paper describes how to reduce the size of such index structures, while still maintaining a high query performance. The setting of approximate string search is unique in that a candidate result needs to occur at least a certain number of times among all the inverted lists, and not necessarily on all the inverted lists. The first approach is based on the idea of discarding some of the lists. They study several technical challenges that arise naturally in this approach. One issue is how to compute a new lower bound on the number of common grams shared by two similar strings, the formula of which becomes technically interesting. These models partition an inverted list into fixed-size segments and compress each segment with a word-aligned integer coding scheme. Also studied how to adopt existing inverted-list compression techniques to achieve the goal, and proposed two novel methods for achieving the goal. The trie based pruning is not included in this model; hence performance of this model is not so efficient. This method used two steps to discovering candidate gram pairs and selecting some of them to combine. It has construct to be more effective because data sets used different reduction ratios for equal the limitation of technique. Cons are estimation is not 100% accurate, and an inaccurate result could greatly affect the accuracy of the estimated post-processing time. This will affect the quality of the selected non-whole lists and this estimation may need to be done repeatedly when choosing lists to discard, and therefore needs to be very efficient but it has failed to do that.

2.4 Exploring Distributional Similarity Based Models for Query Spelling Correction

The paper concentrate on the problem of learning improved query spelling correction model by integrating distributional similarity information automatically derived from query logs. The key contribution of work is identifying that they can successfully use the evidence of distributional similarity to achieve better spelling correction accuracy.

They present efficient methods that are able to take advantage of distributional similarity information. This system extends a string edit based error Model with probabilities within a generative source channel model and it explores the effectiveness of approach by integrating distributional similarity based features. This method used the standard algorithm to search for the best output of source channel model and distributional similarity known to achieve better spelling correction accuracy. This paper reports that un-weighted edit distance will cause the overall accuracy of their speller's output and also probability renormalization and smoothing problem has been thrown on to the process.

2.5 A Discriminative Candidate Generator for String Transformations

This paper addresses challenges by exploring the discriminative training of candidate generators. More specifically, they build a binary classifier that, when given a source strings, decides whether a candidate t should be included in the candidate set or not. This approach appears straightforward, but it must resolve two practical issues. First, the task of the classifier is not only to make a binary decision for the two strings s and t, but also to enumerate a set of positive strings for the string s. Another issue arises when they prepare a training set. A discriminative model requires a training set in which each instance (pair of strings) is annotated with a positive or negative label. They design features that express transformations from a source string s to its destination string t. And also present an algorithm that utilizes the feature weights to enumerate candidates of destination strings efficiently. Here inserting the vocabulary into a suffix array, this is used to locate every occurrence on process easily. This approach appears straightforward and they can proceeds the cross word evaluation. Generation algorithm of substitution rules had produced inappropriate rules that transform a string incorrectly in this paper also finding distance or similarity metrics did not specifically derive destination strings to which the classifier is likely to assign positive labels It could not use the efficient algorithm as a candidate generator with all required features. Due to this will lead to loss some words. Some process is not suited for a candidate generator because the processes of string transformations are intractable in their discriminative models.

2.6 Learning string transformations from examples

This paper 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. This paper analyzes the difference between the strings and used the hypothesis that consistent differences occurring across many examples are indicative of a transformation rule. Based on this intuition, they formulated a rule learning problem where seek a concise set of transformation rules that accounts for a large number of differences.

Most approaches to record matching rely on textual similarity of the records, typically computed using a similarity function such as edit distance to determine if two records are matches or not. However, textual similarity can be an imperfect indicator of whether or not two records are matches; in particular, two matching records can be textually dissimilar. It analyzed differences between the strings and it found a significant impact on record matching quality. It is not clear what the level of human intervention necessary is with the transformations learned. Sometimes it helps improve the record matching quality as expected but sometimes it does not.

2.7 Learning Domain-Independent String Transformation Weights for High Accuracy Object Identification

It is a domain-independent approach for incorporating the users' knowledge into an object identification system. The task of object identification occurs when integrating information from multiple websites. The same data objects can exist in inconsistent text formats across sites, making it difficult to identify matching objects using exact text match. This paper discusses extensions to the Active Atlas system, which allow it to learn to tailor the weights of a set of general transformations to a specific application domain through limited user input. The experimental results demonstrate that this approach achieves higher accuracy and requires less user involvement than previous methods across various application domains. Here high accuracy objects identification system requires less user involvement than previous methods across various application domains and working on larger data sets. Here transformations and mapping rules to a specific application domain through limited user input and cannot minimize noise or error in the labels provided by the user.

2.8 Efficient String Matching: An aid to Bibliographic Search

This paper describes a simple, efficient algorithm to locate all occurrences of any of a finite number of keywords in a string of text. The algorithm consists of constructing a finite state pattern matching machine from the keywords and then using the pattern matching machine to process the text string in a single pass. Construction of the pattern matching machine takes time proportional to the sum of the lengths of the keywords. The number of state transitions made by the pattern matching machine in processing the text string is independent of the number of keywords. The algorithm has been used to improve the speed of a library bibliographic search program by a factor of 5 to 10. The pattern matching scheme described in this paper is well suited for large numbers of keywords in text strings and also no additional information needs to be added to the text string, searches can be made over arbitrary files. The time spent in constructing the pattern matching machine was insignificant and making state transitions was also insignificant compared to the time spent reading and unpacking the text string.

3. Problem Definition

Previous works on string transformation can be categorized into two groups. Some work considered efficient generation of strings, assuming that the model is given. Other work tried to learn the model with different approaches, such as generative model, a logistic regression model and discriminative model. However efficiency is not an important factor taken into consideration in these methods.

String transformation, the proposed system has many applications in data mining, natural language processing, information retrieval, and bioinformatics. String transformation has been studied in different specific tasks such as database record matching, spelling error correction, query reformulation and synonym mining. The major difference between proposed work and the existing work is that the work focuses on enhancement of both accuracy and efficiency of string transformation. Figure 1 shows the expected model.

In proposed system, first module that handles the input string to be entered by the end user, second module will check if the string entered is correct with respect to syntax and semantics. Third module suggests spell correction for the user's Query. In spelling Error Correction, if a user wants to check the spelling, he/she can check it and correct it automatically. Through the efficient algorithms and mathematical models, it is simple, accurate and less time consuming offline method. Edit distance and the context details are clubbed here to get better result. Fourth module gives possible outcomes of user’s query. It will be useful for every user. It is impossible to construct a dictionary of queries, so similar query pairs were mined from search log data. It 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”. Fifth Module retrieves relevant documents from the database that will satisfy the user's request.

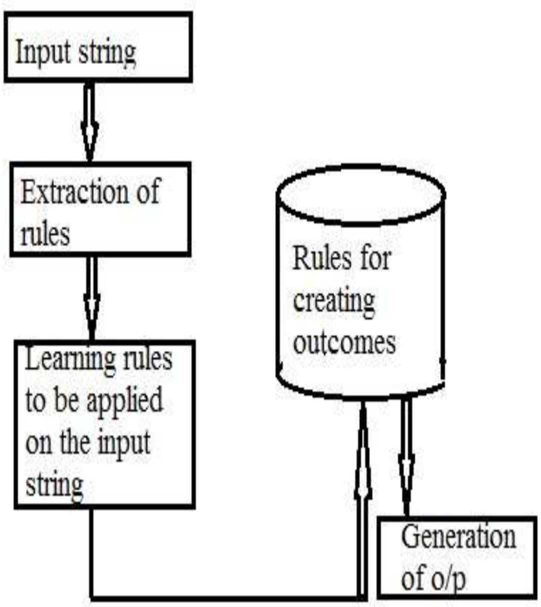
Input string
Extraction of rules
Learning rules to be applied on the input string
Rules for creating outcomes
Generation of o/p

Figure 1: Expected Model

4. Conclusion

We proposed a new statistical learning approach to string transformation. Proposed method is novel and unique in its model, learning algorithm, and string generation algorithm. Two specific applications are addressed with this 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 the proposed method improves upon the baselines in terms of accuracy and efficiency. Proposed method is particularly useful when the problem occurs on a large scale. No another cost factors need to get knowledge. Another big factor is that there is less need of time for the retrieval of data and also for string transformation.

References

[1] 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. 1025–1032.

[2] A. R. Golding and D. Roth, "A winnow-based approach to context-sensitive spelling correction," Mach. Learn., vol. 34, no. 1–3, pp. 107–130, Feb. 1999.

[3] 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. 379–386.

[4] A. V. Aho and M. J. Corasick, "Efficient string matching: an aid to bibliographic search," *Commun. ACM, vol. 18, no. 6, pp. 333–340,* Jun. 1975.

[5] N. Okazaki, Y. Tsuruoka, S. Ananiadou, and J. Tsujii, "A candidate generator for string transformations," in *Proc. Conf. Empirical Methods Natural Language Processing,* Morristown, NJ, USA, 2008, pp. 447–456.

[6] 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. 1080–1089.

[7] A. Arasu, S. Chaudhuri, and R. Kaushik, "Learning string transformations from examples," *Proc. VLDB Endow., vol. 2, no.* 1, pp. 514–525, Aug. 2009.

[8] S. Tejada, C. A. Knoblock, and S. Minton, "Learning domainindependent string transformation weights for high accuracy object identification," in *Proc. 8th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, New York, NY, USA, 2002,* pp. 350–359