A Statistical Model for Flexible String Similarity

Atsuhiro Takasu
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
takasu@nii.ac.jp

Abstract

This paper proposes a statistical model for defining string similarity. The proposed model is based on hidden Markov model and defines string similarity as the combination of similarities of substrings. The proposed model has an advantage that the similarity is flexibly defined and complex parameters for the similarity are learnable from training data.

1 Introduction

String matching is an important problem. Many studies have been done on this problem and developed methods are applied to various problems such as approximate information retrieval from OCR processed and spoken documents, pattern extraction from sensor data and so on.

In order to measure the similarity of strings, edit distance and DP matching have been frequently used. In the edit distance, similarity is defined as the minimum number of operations of insertion, deletion and replacement required to convert one string to another. The DP matching allows us to use weights of the operations which enable flexible definition of string similarity according to objective problems. However, weights should be determined depending on the problem and it is hard work to find appropriate weights manually.

This problem motivates us to develop a statistical model which has high expressive power of string similarity and whose parameters can be learned from training data. This paper proposes a statical model called dual and variable length hidden Markov model (DVHMM) and applies it to structured string analysis.

2 Statistical Model for String Similarity

For a *base string* and a *compared string,* DVHMM defines their similarity by decomposing them into pairs of substrings and combining the similarities of the substrings.

A DVHMM is a form of Hidden Markov Model (HMM). Instead of producing a string, state transitions of DVHMM produce a pair of strings. Each state of the DVHMM defines the similarity of pairs of strings of fixed lengths in the form of output probability. A state is characterized by a pair of lengths. For example, a state characterized by a pair (2,1) of lengths defines similarities of two consecutive characters in a base string and one character in a compared string. A state (1,0) means that the length of the base (resp. compared) string is 1 (resp. 0), i.e., a delete operation, whereas a state (0,1) corresponds to an insert operation. A state (1,1) corresponds to both a replacement operation and equivalent mapping. Generally, the state corresponding to a pair *(i,j)* produces a pair of strings whose base string (resp. compared string) length is *i* (resp. *j)* with certain probability. States are categorized into *non-null* state that produces a pair of base and compared strings, and *null* state that produces null output and controls state transitions.

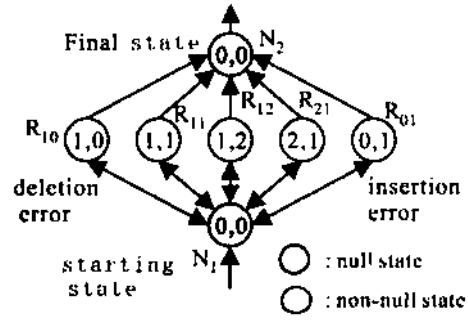

Figure 1: An example of DVHMM

Figure 1 shows an example of DVHMM that consists of five non-null states which define the similarity of pairs of substrings of length (1,0),(0,1),(1,1),(2,1) and (1,2). In this figure, output symbols are omitted due to space restrictions. Suppose the alphabet is {a,b}. Then, the output symbols of the state (2,1) are
{(aa,a),(aa,b),(ab,a),(ab,b),(ba,a), (ba,b),(bb,a),(bb,b)} and output probability is asigned to each output symbol.

DVHMM defines joint probability distribution of a pair of strings. Let us consider a base string "ab" and a compared string "a". Then, the DVHMM in Figure 1 produces the pair of strings (ab, a) by one of the following six sequences of state transitions.

$$N_1R_{10}N_1R_{10}N_1R_{01} \quad N_1R_{10}N_1R_{11}$$
$$N_1R_{10}N_1R_{01}N_1R_{10} \quad N_1R_{11}N_1R_{10}$$
$$N_1R_{01}N_1R_{10}N_1R_{10} \quad N_1R_{21}$$

Then, the joint probability of (ab,a) is obtained by summing

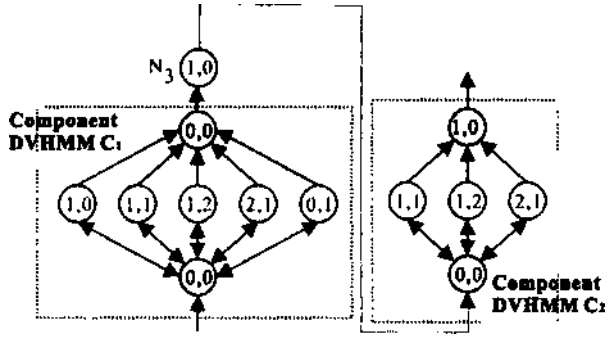

Figure 2: An example of DVHMM for strings consisting of two components

the probabilities of all the five sequences of transitions. Formally, for a base string b and a compared string c, the joint probability of them is obtained by

$$\sum_{\mathbf{q}\in\mathbf{Q}} P(\mathbf{b},\mathbf{c},\mathbf{q}) \qquad (1)$$

where Q is a set of state transitions producing (b,c) and P(b, c, q) is the joint probability of (b, c) along the state transition q. The joint probability is used as the similarity of a pair of strings.

DVHMM inherits HMM's ability to represent syntactical structure. This ability enables DVHMM to change similarity definition depending on the part of strings. Let us consider strings consisting of two components delimited by special characters. For example, base string consists of two parts delimited by colon and compared string is concatenation of two parts without delimiters. Then, the DVHMM depicted in Figure 2 defines the similarity of these strings, i.e., a component DVHMM C_1 defines similarity of prefix component whereas a component DVHMM C_2 defines the similarity of suffix component. State N_3 defines the delimiter, which produce a pair of colon and null string with probability 1.0. For a given pair ("a:ab", "ab") of string, the DVHMM may separate the pair into ("a","a") as a prefix component and ("b","b") as a suffix component, and similarities of these components are measured by different similarity measures, i.e., *C1* and *C2.* respectively. In this case, similarity can be calculated by the joint probability of the pair of string produced by the most likely state transition, i.e.,

$$\max_{\mathbf{q}\in\mathbf{Q}} P(\mathbf{b},\mathbf{c},\mathbf{q}) \qquad (2)$$

where Q is the same set in expression (1).

Probabilities of DVHMM can be estimated from training data by the expectation-maximization technique efficientlytTakasu, 2002]. For a given pair (b,c) of strings, the similarity (2) can be calculated efficiently by a dynamic programming algorithm. We omit the details of the algorithms due to space restriction.

Compared with DP matching, DVHMM enables to

- define the weights in a more detailed manner, and
- calculate the weights systematically.

3 Preliminary Experiment

We applied the proposed method to bibliographic matching which matches references in academic articles with records in bibliographic databases. Reference information is effectively used in digital libraries such as CiteSeer [Lawrence, 1999]. In bibliographic references, bibliographic components such as author's name and article title are located in a specific order and separated by delimiters. The syntactical structure of bibliographic reference can be represented with graphical structure of finite automaton where each state represents a bibliographic component. A DVHMM for bibliographic references is obtained by replacing each state with a component DVHMM like Figure 2 where each component DVHMM defines the similarity of the corresponding bibliographic component. For example, because journal names are often abbreviated, higher probabilities are assigned to deletion errors in the DVHMM for journal name whereas low probabilities are assigned to the deletion errors for page because it is seldom abbreviated in references. In this way, the similarity is changed depending on the bibliographic components, and consequently, more accurate similarity model can be constructed for structured strings.

We carried out an experiment on approximate string matching. In this experiment, we prepared 1,575 references appeared in Japanese academic articles and corresponding records in a bibliographic database, then applied 5-fold cross-validation, i.e., divided them into 5 groups, trained DVHMM using 4 groups of them, and made bibliographic matching using remaining one group. Matching result is ranked records in the database according to the similarity. We used edit distance as baseline similarity and compared the performance of the proposed method with the edit distance. The following table shows the average accuracy that the correct record is ranked within top *m* records.

rank	1	5	10
ED	92.33	94.56	95.23
DVHMM	94.57	95.10	95.47

As shown in the table the proposed method achieved higher matching accuracy.

4 Conclusion

This paper proposed a statistical model for defining string similarity. The proposed model can represent similarity of strings with the similarities of substrings such as traditional edit distance and DP matching. However, the proposed model can define the similarity more flexible way. This paper shows the proposed model can be applied to the matching of structured strings.

References

[Lawrence, 1999] S. Lawrence, C. L. Giles, and K. D. Bollacker. Digital libraries and autonomous citation indexing. *IEEE Computer,* 32(6):67-71, June 1999.

[Takasu, 2002] A. Takasu and K. Aihara. DVHMM: Variable Length Text Recognition Error Model. In *Proceedings of International Conference on Pattern Recognition* 2002.