

# String Resemblance Systems: A Unifying Framework for String Similarity with Applications to Literature and Music

Masayuki Takeda

Department of Informatics, Kyushu University 33, Fukuoka 812-8581, Japan  
and Japan Science and Technology Corporation  
takeda@i.kyushu-u.ac.jp

## 1 Introduction

Identification of similar objects from a large collection of objects is one fundamental technique in several different areas in computer science, e.g., the case-based reasoning and the machine discovery. *Strings* are the most basic representations of objects inside computers, and thus string similarity is one of the most important topics in computer science.

Similarity measure must be sensitive to the kind of differences we wish to quantify. The *weighted edit distance* is one such framework in which the measure can be varied by altering weight assignment to each edit operation depending on symbols involved. However, it does not suffice to solve ‘real problems’ (see e.g., [2]). It is considered that two objects have necessarily a common structure if they seem similar, and the degree of similarity depends upon how valuable the common structure is. Based on this intuition, we present a unifying framework, named *string resemblance system* (SRS, for short). In this framework, similarity of two strings can be viewed as the maximum score of pattern that matches both of them. The differences among the measures are therefore the choices of (1) *pattern set* to which common patterns belong, and (2) *pattern score function* which assigns a score to each pattern.

For example, if we choose the set of *patterns with variable length don't cares* and define the score of a pattern to be the number of symbols in it, then the obtained measure is the length of the longest common subsequence (LCS) of two strings. In fact, the strings *acdeba* and *abdac* have a common pattern *a\*d\*a\** which contains three symbols. With this framework one can easily design and modify his/her measures. In this paper we briefly describe SRSs and then report successful results of applications to literature and music.

## 2 Unifying Framework for String Similarity

In practical applications such as biological sequence comparisons, it is often preferred to measure *similarity* rather than distance between two given strings. We shall regard a distance measure as a similarity measure by multiplying the distance values by  $-1$ . Gusfield [2] pointed out that in dealing with string similarity

the language of alignments is often more convenient than the language of edit operations. Here, we generalize the alignment based scheme and propose a new scheme which is based on the notion of *common patterns*. Before describing our scheme, we need to introduce some notation. The set of strings over an alphabet  $\Sigma$  is denoted by  $\Sigma^*$ . The length of a string  $u$  is denoted by  $|u|$ . The string of length 0 is called the *empty string*, and denoted by  $\varepsilon$ . Let  $\Sigma^+ = \Sigma^* - \{\varepsilon\}$ . Let us denote by  $\mathbf{R}$  the set of real numbers.

**Definition 1.** A string resemblance system (SRS) is a 4-tuple  $\langle \Sigma, \Pi, L, \Phi \rangle$ , where:

1.  $\Sigma$  is an alphabet;
2.  $\Pi$  is a set of descriptions called patterns;
3.  $L$  is a function called interpretation that maps a pattern in  $\Pi$  to a language over  $\Sigma$ , i.e., a subset of  $\Sigma^*$ ;
4.  $\Phi$  is a function that maps a pattern in  $\Pi$  to a real number called score.

The similarity between strings  $x$  and  $y$  with respect to  $\langle \Sigma, \Pi, L, \Phi \rangle$  is defined by

$$\mathbf{SIM}(x, y) = \sup\{\Phi(\pi) \mid \pi \in \Pi \text{ and } x, y \in L(\pi)\}.$$

We would assume that, for any  $x, y \in \Sigma^*$ , the set  $\{\Phi(\pi) \mid \pi \in \Pi \text{ and } x, y \in L(\pi)\}$  is non-empty, bounded upwards, and contains the least upper bound as a member. This assumption guarantees that for any  $x, y \in \Sigma^*$  there always exists a pattern  $\pi \in \Pi$  common to  $x$  and  $y$  that maximizes the score  $\Phi(\pi)$ . Thus, computation of similarity is regarded as *optimal pattern discovery* in our framework. In this sense our framework bridges a gap between similarity computation and pattern discovery.

**Definition 2.** An SRS  $\langle \Sigma, \Pi, L, \Phi \rangle$  is said to be homomorphic if

1.  $\Pi = (\Sigma \cup \Delta)^*$ , where  $\Delta$  is a set of wildcards.
2.  $L : \Pi \rightarrow 2^{\Sigma^*}$  is a homomorphism such that  $L(c) = \{c\}$  for any  $c \in \Sigma$  and  $L(\pi_1\pi_2) = L(\pi_1)L(\pi_2)$  for any  $\pi_1, \pi_2 \in \Pi$ .
3.  $\Phi : \Pi \rightarrow \mathbf{R}$  is a homomorphism such that  $\Phi(\pi_1\pi_2) = \Phi(\pi_1) + \Phi(\pi_2)$  for any  $\pi_1, \pi_2 \in \Pi$ .

Note that when  $\Sigma$  is fixed, a homomorphic SRS is determined by specifying (1) the set  $\Delta$  of wildcards, (2) the values  $L(\gamma)$  for all  $\gamma \in \Delta$ , and (3) the values  $\Phi(\gamma)$  for all  $\gamma \in \Sigma \cup \Delta$ .

The class of homomorphic SRSs covers most of the known similarity (dissimilarity) measures. For example, the edit distance falls into this class. Let  $\Delta = \{\psi\}$  where  $\psi$  is the wildcard that matches the empty string and any symbol in  $\Sigma$ , namely,  $L(\psi) = \Sigma \cup \{\varepsilon\}$ . Let  $\Phi(\psi) = -1$  and  $\Phi(c) = 0$  for all  $c \in \Sigma$ . Then, the similarity measure defined by this homomorphic SRS is the same as the edit distance except that the values are non-positive. Similarly, the Hamming distance can be defined by using the wildcard  $\phi$  that matches any symbol in  $\Sigma$ .

We can define the LCS measure by using the wildcard  $\star$  that matches any string in  $\Sigma^*$ . Namely, the homomorphic SRS such that (1)  $\Delta = \{\star\}$ , (2)  $L(\star) =$

$\Sigma^*$ , and (3)  $\Phi(\star) = 0$  and  $\Phi(c) = 1$  for any  $c \in \Sigma$  gives the LCS measure. Although another definition is possible for this measure which uses the wildcard  $\psi$  with  $L(\psi) = \Sigma \cup \{\varepsilon\}$ , but the common patterns obtained are much simpler.

The weighted edit distance can also be defined as a homomorphic SRS in which the wildcards  $\phi(a|b)$  ( $a, b \in \Sigma \cup \{\varepsilon\}$  and  $a \neq b$ ) such that  $L(\phi(a|b)) = \{a, b\}$  are introduced, and  $\Phi(\phi(a|b))$  is the weight assigned to each pair of  $a$  and  $b$ .

Next, we extend the class of homomorphic SRSs by easing the restriction on the pattern score functions as follows. A pattern score function  $\Phi$  defined on  $\Pi = (\Sigma \cup \Delta)^*$  is said to be *semi-homomorphic* if there exists a subset  $\mathcal{D}$  of  $\Pi$  with  $\varepsilon \notin \mathcal{D}$  and  $\Pi = \mathcal{D}^*$ , and a function  $g : \mathcal{D} \rightarrow \mathbf{R}$  such that, for any  $\pi \in \Pi$ ,

$$\Phi(\pi) = \max \left\{ \sum_{i=1}^{\ell} g(\pi_i) \mid \ell \geq 0, \pi_i \in \mathcal{D} \ (i = 1, \dots, \ell), \text{ and } \pi = \pi_1 \cdots \pi_\ell \right\}.$$

**Definition 3.** An SRS  $\langle \Sigma, \Pi, L, \Phi \rangle$  is said to be semi-homomorphic if

1.  $\Pi = (\Sigma \cup \Delta)^*$ , where  $\Delta$  is a set of wildcards.
2.  $L : \Pi \rightarrow 2^{\Sigma^*}$  is a homomorphism such that  $L(c) = \{c\}$  for any  $c \in \Sigma$  and  $L(\pi_1\pi_2) = L(\pi_1)L(\pi_2)$  for any  $\pi_1, \pi_2 \in \Pi$ .
3.  $\Phi : \Pi \rightarrow \mathbf{R}$  is semi-homomorphic.

Computation of the weighted edit distance between two given strings  $x$  and  $y$  can be viewed as computation of the lowest scoring paths from node  $(0, 0)$  to node  $(|x|, |y|)$  in the *weighted edit graph* (see, e.g., [2]), a directed (acyclic) weighted graph where the vertices are the  $(|x| + 1) \times (|y| + 1)$  points of the grid with rows  $0, \dots, |x|$  and columns  $0, \dots, |y|$ . The computation can be done by standard dynamic programming in  $O(|x||y|)$  time.

A similar discussion is possible for (semi-)homomorphic SRSs, with appropriate modifications in the definition of weighted edit graph. The construction time of such graph depends upon the response time for membership query “ $w \in L(\gamma)$ ” for a wildcard  $\gamma$  in  $\Delta$  and upon that of  $\Phi$ . However, once such graph is constructed, the best score can be computed in linear time with respect to the number of edges in the graph, which varies depending upon  $\Delta$  (and upon  $\mathcal{D}$  in the case of semi-homomorphic SRSs). It would be interesting to reveal the hierarchy of subclasses of SRSs from the viewpoint of computational complexity, but this is beyond the scope of the present paper.

As demonstrated so far, we can handle a variety of string (dis)similarity by changing the pattern set  $\Pi$  and the pattern score function  $\Phi$ . The pattern sets discussed above are, however, restricted to the form  $\Pi = (\Sigma \cup \Delta)^*$ , where  $\Delta$  is a set of wildcards. Here we shall mention pattern sets of other types. An *order-free pattern* is a multiset  $\{u_1, \dots, u_k\}$  such that  $k > 0$  and  $u_1, \dots, u_k \in \Sigma^+$ , and is denoted by  $\pi[u_1, \dots, u_k]$ . The language of pattern  $\pi[u_1, \dots, u_k]$  is defined to be the union of the languages  $\Sigma^*u_{\sigma(1)}\Sigma^* \cdots \Sigma^*u_{\sigma(k)}\Sigma^*$  over all permutations  $\sigma$  of  $\{1, \dots, k\}$ . For example, the language of the pattern  $\pi[abc, de]$  is  $\Sigma^*abc\Sigma^*de\Sigma^* \cup \Sigma^*de\Sigma^*abc\Sigma^*$ . The membership problem for order-free patterns is NP-complete, and therefore the similarity computation is impractical generally. However the problem is polynomial-time solvable when  $k$  is fixed.

The pattern languages, introduced by Angluin [1], is also interesting for our framework. A *pattern* is a string in  $\Pi = (\Sigma \cup V)^+$ , where  $V$  is an infinite set  $\{x_1, x_2, \dots\}$  of variables and  $\Sigma \cap V = \emptyset$ . The language of a pattern  $\pi$  is the set of strings obtained by replacing variables in  $\pi$  by non-empty strings. The membership problem is NP-complete for the class of patterns as shown in [1], but it is polynomial-time solvable when the number of variables occurring more than once within  $\pi$  is bounded by a fixed number  $k$ .

### 3 Discovery from Literary Works

Waka is a form of traditional Japanese poetry with a 1300-year history. A Waka poem has five lines and thirty-one syllables, arranged thus: 5-7-5-7-7. In [5] we attempted to semi-automatically discover similar poems from an accumulation of about 450,000 Waka poems in a machine-readable form. One reasonable approach is to arrange all possible pairs of poems in decreasing order of their similarity, and to scholarly scrutinize a first part. One of the aims here is to discover unheeded instances of Honkadori (poetic allusion), one important rhetorical device in Waka poems based on specific allusion to earlier famous poems.

We tested three similarity measures for dealing with similarity between Waka poems, which were newly designed along with our framework. The first measure is based on line-order alternation and on the *modified LCS measure* for quantifying affinity between lines, which is defined as a semi-homomorphic SRS such that  $\Delta = \{\star\}$ ,  $L(\star) = \Sigma^*$ , and the pattern score function  $\Phi$  defined by  $\mathcal{D} = \Sigma^+ \cup \{\star\}$ , and  $g(\pi) = |\pi| - s$ , if  $\pi \in \Sigma^+$ ; otherwise,  $g(\pi) = 0$ , where  $s$  ( $0 < s \leq 1$ ) is a penalty for break in continuity of symbols. This measure was proved suitable for finding instances of poetic allusion.

The second and the third measures are based on the order-free patterns defined in the previous section, in order to cope with word-order alternation. These two measures differ in the respect that the pattern score function of the third measure depends on the *rarity* of common pattern within a given large collection of poems, whereas that of the second one is defined syntactically. The idea of rarity is proved to be effective in identifying only close affinities which are hardly seen elsewhere, possibly excluding known stereotype expressions.

The first measure is especially favored by Waka researchers and used in discovering affinities of some unheeded poems with some earlier ones. The discovered affinities raise an interesting issue for Waka studies: (1) We have proved that one of the most important poems by Fujiwara-no-Kanesuke, one of the renowned thirty-six poets, was in fact based on a model poem found in Kokin-Shū. The same poem had been interpreted just to show “frank utterance of parents’ care for their child.” Our study revealed the poet’s techniques in composition half hidden by the heart-warming feature of the poem by extracting the same structure between the two poems. (2) We have compared Tametada-Shū, the mysterious anthology unidentified in Japanese literary history, with a number of private anthologies edited after the middle of the Kamakura period (the 13th-century) using the same method, and found that there are about 10 pairs of similar poems between Tametada-Shū and Sōkon-Shū, an anthology by Shōtetsu. The result

suggests that the mysterious anthology was edited by a poet in the early Muramachi period (the 15th-century). There have been surmised dispute about the editing date since one scholar suggested the middle of Kamakura period as a probable one. We have had a strong evidence about this problem.

## 4 Finding Affinities from Musical Scores

Any monophonic score can be regarded as a string of ordered pairs consisting of the pitch of the note and its length. Mongeau and Sankoff [4] proposed a dissimilarity measure for monophonic scores, which is a variant of the weighted edit distance where additional two edit operations, *fragmentation* and *consolidation*, are allowed to associate multiple notes with a single note or vice versa. It is reported in [4] that the measure arranges the variations on a theme by Mozart in a reasonable order which coincides with subjective impressions. However, it turned out from our experimental results that a problem arises when dealing with the mixtures of variations on several themes.

In [3] we tested three similarity measures and showed that the third one could cope with this problem. As a preprocessing, each note in a musical score is replaced with a sequence of notes of a unit length (16th note) to obtain simply a string of pitches. The measures are respectively based on three measures to quantify the affinities between two phrases of uniform length, each falls into the class of the semi-homomorphic SRSs. The set  $\Pi = (\Sigma \cup \{\phi\})^*$  is commonly used in the three. While the pattern score function of the first measure is the one which simply counts up matches (i.e., the number of symbols in a pattern), those of the second and third measures are sensitive to the continuity of matches. More precisely, in the second measure it is defined by  $\mathcal{D} = \Sigma^+ \cup \{\phi\}$ , and  $g(\pi) = |\pi|$ , if  $\pi \in \Sigma^+$  and  $|\pi| \geq s$ ; otherwise,  $g(\pi) = 0$ , where  $s$  is a threshold. In the third measure,  $\mathcal{D} = \{\pi \in (\Sigma \cup \{\phi\})^+ \mid \pi \text{ does not contain } \phi^{t+1}\}$ , and  $g(\pi)$  is the number of symbols within  $\pi$ , if  $|\pi| \geq s$ ; otherwise,  $g(\pi) = 0$ , where  $s, t$  are thresholds. Despite its simplicity, the third measure is better than Mongeau and Sankoff's one in the sense that it is able to exclude variations on other themes.

## References

1. D. Angluin. Finding patterns common to a set of strings. *J. Comput. Sys. Sci.*, 21:46–62, 1980.
2. D. Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, New York, 1997.
3. T. Kadota, A. Ishino, M. Takeda, and F. Matsuo. On melodic similarity. *IPSJ SIG Notes*, 2000(49):15–24, 2000. (in Japanese).
4. M. Mongeau and D. Sankoff. Comparison of musical sequences. *Computers and the Humanities*, 24(3):161–175, 1990.
5. K. Tamari, M. Yamasaki, T. Kida, M. Takeda, T. Fukuda, and I. Nanri. Discovering poetic allusion in anthologies of classical Japanese poems. In *Proc. 2nd International Conference on Discovery Science (DS'99)*, pages 128–138, 1999.