Sam's String Metrics

Natural Language Processing Group,
Department of Computer Science,
University of Sheffield,
Regent Court, 211 Portobello Street, Sheffield, S1 4DP,
UNITED KINGDOM
Tel:+44(0)114-2228000 Fax:+44(0)114-22.21810
sam@dcs.shef.ac.uk

Links

HomePage
Research Links
Currently Reading
Handy Links
Publications
Funding
About Me

SimMetrics

In my investigations into string metrics, similarity metrics and the like I have developed an open source library of Similarity metrics called SimMetrics. SimMetrics is an open source java library of Similarity or Distance Metrics, e.g. Levenshtein distance , that provide float based similarity measures between String Data. All metrics return consistant measures rather than unbounded similarity scores. This open source library is hosted at http://sourceforge.net/projects/simmetrics/. The JavaDoc's of SimMetrics are detailed here. I would welcome collaborations and outside development on this open source project, if you want to help or simply leave a comment then please email me at reverendsam@users.sourceforge.net.

Similarity Metrics

- Hamming distance
- Levenshtein distance
- Needleman-Wunch distance or Sellers Algorithm
- Smith-Waterman distance
- Gotoh Distance or Smith-Waterman-Gotoh distance
- Block distance or L1 distance or City block distance
- Monge Elkan distance
- Jaro distance metric
- Jaro Winkler
- SoundEx distance metric
- Matching Coefficient
- Dice's Coefficient
- Jaccard Similarity or Jaccard Coefficient or Tanimoto coefficient
- Overlap Coefficient
- Euclidean distance or L2 distance
- Cosine similarity
- Variational distance
- Hellinger distance or Bhattacharyya distance
- Information Radius (Jensen-Shannon divergence)
- Harmonic Mean
- Skew divergence
- Confusion Probability
- Tau
- Fellegi and Sunters (SFS) metric
- TFIDF or TF/IDF
- FastA
- BlastP
- Maximal matches
- q-gram
- Ukkonen Algorithms

Other Points of Interest

Comparisons of similarity metrics

Workshops concerning Information Integration

Other links to papers of interest

Information Integration projects

Other Links

Hamming distance

This is defined as the number of bits which differ between two binary strings i.e. the number of bits which need to be changed (corrupted) to turn one string into the other. For example the bit strings

10011010 and
10001101 has a
hamming distance of 4bits, (as four bits are dissimilar).

The simple bitwise version can be simply calcualted from the following C code.

```
//given input
unsigned int bitstring1;
unsigned int bitstring2;

//bitwise XOR (bitstring1 is destroyed)
bitstring1 ^= bitstring2;

// count the number of bits set in bitstring1
unsigned int c; // c accumulates the total bits set in bitstring1
for (c = 0; bitstring1; c++)
{
  bitstring1&= bitstring1 - 1; // clear the least significant bit set
}
```

The simple hamming distance function can be extended into a vector space approach where the terms within a string are compared, counting the number of terms in the same positions. (this approach is only suitable for exact length comparisons). Such an extension is very similar to the **matching coefficient** approach.

This Metric is not currently included in the **SimMetric open source library** as it is a simplistic approach.

Levenshtein Distance

This is the basic edit distance function whereby the distance is given simply as the minimum edit distance which transforms string1 into string2. Edit Operations are listed as follows:

Copy character from string1 over to string2 (cost 0)
Delete a character in string1 (cost 1)
Insert a character in string2 (cost 1)
Substitute one character for another (cost 1)

```
                D(i-1,j-1) + d(si,tj) //subst/copy
D(i,j) = min D(i-1,j)+1 //insert
                D(i,j-1)+1 //delete
```

d(i,j) is a function whereby d(c,d)=0 if c=d, 1 else

There are many extensions to the Levenshtein distance function typically these alter the d(i,j) function, but further extensions can be made for instance, the **Needleman-Wunch distance** for which Levenshtein is equivalent if the gap distance is 1. The Levenshtein distance is calulated below for the term "sam chapman" and "sam john chapman", the final distance is given by the bottom right cell, i.e. 5. This score indicates that only 5 edit cost operations are required to match the strings (for example, insertion of the "john " characters, although a number of other routes can be traversed instead).

	s	**a**	**m**		**c**	**h**	**a**	**p**	**m**	**a**	**n**
s	0	1	2	3	4	5	6	7	8	9	10
a	1	0	1	2	3	4	5	6	7	8	9
m	2	1	0	1	2	3	4	5	6	7	8
	3	2	1	0	1	2	3	4	5	6	7
j	4	3	2	1	1	2	3	4	5	6	7
o	5	4	3	2	2	2	3	4	5	6	7

h	6	5	4	3	3	2	3	4	5	6	7
n	7	6	5	4	4	3	3	4	5	6	6
	8	7	6	5	5	4	4	4	5	6	7
c	9	8	7	6	5	5	5	5	5	6	7
h	10	9	8	7	6	5	6	6	6	6	7
a	11	10	9	8	7	6	5	6	7	6	7
p	12	11	10	9	8	7	6	5	6	7	7
m	13	12	11	10	9	8	7	6	5	6	7
a	14	13	12	11	10	9	8	7	6	5	6
n	15	14	13	12	11	10	9	8	7	6	5

This Metric is included in the **SimMetric open source library**.

Needleman-Wunch distance or Sellers Algorithm

This approach is known by various names, Needleman-Wunch, Needleman-Wunch-Sellers, Sellers and the Improving Sellers algorithm. This is similar to the basic edit distance metric, **Levenshtein distance**, this adds an variable cost adjustment to the cost of a gap, i.e. insert/deletion, in the distance metric. So the Levenshtein distance can simply be seen as the Needleman-Wunch distance with G=1.

```
               D(i-1,j-1) + d(si,tj) //subst/copy
D(i,j) = min D(i-1,j)+G //insert
               D(i,j-1)+G //delete
```

Where G = “gap cost” and d(c,d) is again an arbitrary distance function on characters (e.g. related to typographic frequencies, amino acid substitutibility, etc). The Needleman-Wunch distance is calulated below for the term "sam chapman" and "sam john chapman", with the gap cost G set to 2. The final distance is given by the bottom right cell, i.e. 10. This score indicates that only 10 edit cost operations are required to match the strings (for example, insertion of the "john " characters, although a number of other routes can be traversed instead).

	s	**a**	**m**		**c**	**h**	**a**	**p**	**m**	**a**	**n**
s	0	2	4	6	8	10	12	14	16	18	20
a	2	0	2	4	6	8	10	12	14	16	18
m	4	2	0	2	4	6	8	10	12	14	16
	6	4	2	0	2	4	6	8	10	12	14
j	8	6	4	2	1	3	5	7	9	11	13
o	10	8	6	4	3	2	4	6	8	10	12
h	12	10	8	6	5	3	3	5	7	9	11
n	14	12	10	8	7	5	4	4	6	8	9
	16	14	12	10	9	7	6	5	5	7	9
c	18	16	14	12	10	9	8	7	6	6	8
h	20	18	16	14	12	10	10	9	8	7	7
a	22	20	18	16	14	12	10	11	10	8	8
p	24	22	20	18	16	14	12	10	12	10	9
m	26	24	22	20	18	16	14	12	10	12	11
a	28	26	24	22	20	18	16	14	12	10	12
n	30	28	26	24	22	20	18	16	14	12	10

This Metric is included in the **SimMetric open source library**.

Smith-Waterman distance

Specific details can be found for this approach in the following paper:

Smith, T. F. and Waterman, M. S. 1981. **Identification of common molecular subsequences**

Again similar to the to **Levenshtein distance**, this was developed to identify optimal allignments between related DNA and protein sequencies. This has two main adjustable parameters a function for an alphabet mapping to cost values for substitutions and costs, (the d function). This also allows costs to be attributed to a gap G, (insert or delete). The final similarity distance is computed from the maximum value in the path where

```
             0 //start over
             D(i-1,j-1) -d(si,tj) //subst/copy
D(i,j) = max D(i-1,j)-G //insert
             D(i,j-1)-G //delete
```

Distance is maximum over all i,j in table of D(i,j)

G = 1 //example value for gap
d(c,c) = -2 //context dependent substitution cost
d(c,d) = +1 //context dependent substitution cost

The Smith-Waterman distance is calulated below for the term "aaaa mnop zzzz" and "bbbb mnop yyyy", with the gap cost G set to 0.5 and where d(c,c) = -2, d(b,c) = 1.

The final distance is given by the higest valued cell, i.e. 6. This score indicates that the longest approximately matching string terminates in the cell with the highest value so the sequence " mnop " matches in both strings.

	a	**a**	**a**	**a**		**m**	**n**	**o**	**p**		**z**	**z**	**z**	**z**
b	0	0	0	0	0	0	0	0	0	0	0	0	0	0
b	0	0	0	0	0	0	0	0	0	0	0	0	0	0
b	0	0	0	0	0	0	0	0	0	0	0	0	0	0
b	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	1	0.5	0	0	0	1	0.5	0	0	0
m	0	0	0	0	0.5	2	1.5	1	0.5	0.5	0	0	0	0
n	0	0	0	0	0	1.5	3	2.5	2	1.5	1	0.5	0	0
o	0	0	0	0	0	1	2.5	4	3.5	3	2.5	2	1.5	1
p	0	0	0	0	0	0.5	2	3.5	5	4.5	4	3.5	3	2.5
	0	0	0	0	1	0.5	1.5	3	4.5	6	5.5	5	4.5	4
y	0	0	0	0	0.5	0	1	2.5	4	5.5	5	4.5	4	3.5
y	0	0	0	0	0	0	0.5	2	3.5	5	4.5	4	3.5	3
y	0	0	0	0	0	0	0	1.5	3	4.5	4	3.5	3	2.5
y	0	0	0	0	0	0	0	1	2.5	4	3.5	3	2.5	2

An extension of this technique was made by **Gotoh** in 1982 which also allows affine gaps to be taken into consideration. This extension was incorporated into the better known **Monge Elkan distance**.

This Metric is included in the **SimMetric open source library**.

Gotoh or Smith-Waterman-Gotoh distance

This is an extension from the Smith Waterman distance by allowing affine gaps within the sequence. This work comes from the following paper:

Gotoh, O. "An Improved Algorithm for Matching Biological Sequences". Journal of Molecular Biology. 162:705-708, 1981.

The Affine Gap model includes variable gap costs typically based upon the length of the gap *l*.

- Comparing two sequences A (=a_1 a_2 a_3 ... a_n) and B (=b_1 b_2 b_3 ... b_m)
- $D_{ij}=\max\{D_{i-1,j-1}+d(a_i,b_j), \max\{D_{i-k,j}-W_k\}, \max\{D_{i,j-l}-W_l\}, 0\}$
- D_{ij} is the maximum similarity of two segments *ending* in a_i and b_j respectively.

Affine gap costs are specified in two ways one with a cost for starting a gap and secondly with a cost for continuation of a gap.

This Metric is included in the SimMetric open source library.

Block Distance or L1 distance or City Block distance

This distance metric is variously named as block distance, L1 distance or city block distance

This is a vector based approach so where 'q' and 'r' are defined in n-dimensional vector space The L1 or block distance is calcualted from summing the edge distances.

$$L_1(q,r) \qquad = \quad \sum_y |q(y) - r(y)|$$

This can be decribed in two dimensions with discrete-valued vectors, when we can picture the set of points within a grid, the distance value is simply the number of edges between points that must be traversed to get from 'q' to 'r' within the grid. This is the same problem as getting from corner a to b in a rectilinear streetmap, hence the name "city-block metric".

This Metric is included in the SimMetric open source library.

Monge Elkan distance

This distance is often incorrectly reffered to as an implementation of the Smith-Waterman distance approach. As in the original work of Monge and Elkan they called this Smith-Waterman, but actually, this uses an affine gap model, so it's not Smith-Waterman at all but rather the extended Gotoh distance. Specific details can be found regarding the Monge Elkan edit distance function in the following paper:

Alvaro E. Monge and Charles P. Elkan. The field matching problem: Algorithms and applications. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, (KDD), 1996

The Monge Elkan approach makes other additional tests by taking the semantic similarity of a number of fields aor sub fields into consideration. Each sub field is evaluated against the most similar sub field in the comparison string using the Gotoh distance between the fields this is then combined using a matching Algorithm (displayed below).

$$match(A,B) = \frac{1}{|A|}\sum_{i=1}^{|A|}\max_{j=1}^{|B|} match(A_i, B_j).$$

This recursive field matching algorithm has quadratic time complexity. As every atomic entry in A and B must be compared against each other.

This Metric is included in the SimMetric open source library.

Jaro distance metric

The Jaro distance metric takes into account typical spelling deviations, this work comes from the following paper.

Jaro, M. A. 1989 "Advances in record linking methodology as applied to the 1985 census of Tampa Florida". Journal of the American Statistical Society 64:1183-1210

this was later clarified in the subsequent paper

Jaro, M. A. 1995 "Probabilistic linkage of large public health data file" Statistics in Medicine 14:491-498

Briefly, for two strings *s* and *t*, let *s'* be the characters in *s* that are "common with" *t*, and let *t'* be the charcaters in *t* that are "common with" *s*; roughly speaking, a character *a* in *s* is "in common" with *t* if the same character *a* appears in about the place in *t*.

Let *Ts,t* measure the number of transpositions of characters in *s'* relative to *t'*. The Jaro similarity metric for *s* and *t* is

$$Jaro(s,t) = \frac{1}{3} \cdot \left(\frac{|s'|}{|s|} + \frac{|t'|}{|t|} + \frac{|s'| - T_{s',t'}}{2|s'|} \right)$$

This approach has since been expanded to slightly modified approach called **Jaro Winkler**.

This Metric is included in the **SimMetric open source library**.

Jaro Winkler

This is an extension of the **Jaro distance metric**, from the work of Winkler in 1999. This extension modifies the weights of poorly matching pairs *s, t* that share a common prefix. The output score is simply adjusted as follows,

JaroWinklerScore(s,t) = JaroScore(s,t) + (prefixLength PREFIXSCALE * (1.0f - JaroScore(s,t)));*

where

prefixLength is the legth of common prefix at the start of the string,

PREFIXSCALE is a constant scaling factor for how much the score is adjusted upwards for having common prefix's.

This adjustment gives more favourable ratings to strings that match from the begining for a set prefix length.

This Metric is included in the **SimMetric open source library**.

Soundex distance metric

Soundex is a coarse phonetic indexing scheme, widely used in genealogy. This approach focuses upon individuals names and as such has not been provably applied to a more general context.

Soundex allows phonetic mispellings to be easily evaluated, for instance the names *John*, *Johne* and *Jon* are often geneologically the same person. This is a term based evaluation where each term is given a Soundex code, each soundex code consists of a letter and three numbers between 0 and 6, e.g. "Chapman" is "C155". The letter is always the first letter of the surname. The numbers hash together the rest of the name. This approach is very promising for disabiguation of translitterated/misspelt names, i.e non english names represented as ASCII are frequently misrepresented. The basic algorithm for soundex is now detailed.

> A soundex code consists of the first letter of the surname, followed by three digits. The digits are based on the consonants as in the following table, (this can differ in implementations):
>
> 1) B,P,F,V
> 2) C,S,K,G,J,Q,X,Z
> 3) D,T
> 4) L
> 5) M,N
> 6) R
>
> The vowels are not used. If two or more adjacent (not separated by a vowel) letters have the same numeric value, only one is used. This also applies if the first (which is used for the letter in the code), and the second letter in name have the same value; the second letter would not be used to generate a digit. If there are not three digits after the consonants are convert, the code is filled out with zeros. The name Sue has no consonants after the S, so the

soundex code would be S000.

This Metric is included in the **SimMetric open source library**.

Matching Coefficient

The Matching Coefficient is a very simple vector based approach which simply counts the number of terms, (dimensions), on which both vectors are non zero. So for vector set X and set Y the matching coeffient is |X & Y|. This can be seen as the vector based count of coreferent terms. This is similar to the vector version of the simple **hamming distance** although position is not taken into account.

This Metric is included in the **SimMetric open source library**.

Dice’s Coefficient

Dice coefficient is a term based similarity measure (0-1) whereby the similarity measure is defined as twice the number of terms common to compared entitys divided by the total number of terms in both tested entities. The Coefficient result of 1 indicates identical vectors as where a 0 equals orthogonal vectors.

Dices coefficient = (2*Common Terms) / (Number of terms in String1 + Number of terms in String2)

This Metric is included in the **SimMetric open source library**.

Jaccard Similarity or Jaccard Coefficient or Tanimoto coefficient

This is another token based vector space similarity measure like the **cosine distance** and **the matching coefficient**. Jaccard Similarity uses word sets from the comparison instances to evaluate similarity. The jaccard similarity penalizes a small number of shared entries (as a portion of all non-zero entries) more than the **Dice coeffient**. The Jaccard similarity is frequently used as a similarity measure for chemical compounds. This similarity measure was first details in 1912 in the following paper.

Jaccard 1912, "The distribution of the flora of the alpine zone", New Phytologist 11:37-50

Each instance is represented as a Jaccard vector similarity function. The Jaccard between two vectors X and Y is

(X*Y) / (|X||Y|-(X*Y))

where (X*Y) is the inner product of X and Y, and |X| = (X*X)^1/2, i.e. the Euclidean norm of X.

This can more easily be described as (|X & Y|) / (| X or Y |)

This approach has been extend by a number of specialisations/alterations not considered here.

This Metric is included in the **SimMetric open source library**.

Overlap Coefficient

This is a measure whereby if a set X is a subset of Y or the converse then the similarity coefficient is a full match. This is similar to **Dice coefficient**. Overlap coefficient is defined as so.

overlap_coefficient(q,r) = (| q & r |) / min{ | q | , | r | }

This Metric is included in the **SimMetric open source library**.

Euclidean distance or L2 distance

This approach again works in vector space similar to the **matching coefficient** and the **dice coefficient**, however the similarity measure is not judged from the angle as in **cosine rule** but rather the direct euclidean distance between the vector inputs. Below is the standard Euclidean distance formula between vectors q and r.

$$\mathrm{euc}(q,r) \quad = \quad \left(\sum_y (q(y) - r(y))^2\right)^{1/2}$$

This Metric is included in the **SimMetric open source library**.

Cosine similarity

Cosine similarity is a common vector based similarity measure similar to **dice coefficient**. Whereby the input string is transformed into vector space so that the Euclidean cosine rule can be used to determine similarity. The cosine similarity is often paired with other approaches to limit the dimensionality of the problem. For instance with simple strings at list of stopwords are used to exclude from the dimensionality of the comparison. In theory this problem has as many dimensions as terms exist.

$$\cos(q,r) \quad = \quad \sum_y q(y)r(y) \Big/ \sqrt{\sum_y q(y)^2 \sum_y r(y)^2}$$

This Metric is included in the **SimMetric open source library**.

Variational distance

The Variational distance metric is a measure used to quantify the difference (sometimes called divergence) between probability distributions. This is a form of the Kullback-Leibler divergence. The formula is displayed beneath but not mentioned further here.

$$V(P\|Q) = \sum_{i=1}^{n} |p_i - q_i|,$$

This Metric is not included in the **SimMetric open source library**. If you have an implementation of this please add it to the **SimMetric Library** or mail **reverendsam@users.sourceforge.net**

Hellinger distance or Bhattacharyya distance

HDE, Hellinger distance estimation, also know as also known as Bhattacharyya distance is detailed in the following source,

A. Basu, I. R. Harris, and S. Basu. Minimum distance estimation: The approach using density-based distances. In G. S. Maddala and C. R. Rao, editors, Handbook of Statistics, volume 15, pages 21-48. North-Holland, 1997.

The Hellinger distance is defined as follows.

$$B(P\|Q) = \sum_{i=1}^{n} \sqrt{p_i q_i}$$

This approach is extended with a number of optermisations, MHDE, Minimum Hellinger Distance Estimation and AMHDE, Approximate Minimum Hellinger Distance Estimation. This distance measure is used frequently within machine learning approaches but is not discussed further here.

This Metric is not included in the **SimMetric open source library**. If you have an implementation of this please add it to the **SimMetric Library** or mail **reverendsam@users.sourceforge.net**

Information Radius (Jensen-Shannon divergence)

The Information Radius is otherwise known as the Jensen-Shannon divergence also as the Jensen-Shannon Distance. This measure is used to quantify the difference (sometimes called divergence) between two (or more) probability distributions. This is a form of the Kullback-Leibler divergence, the formula is displayed beneath.

$$\mathrm{JS}(q,r) \quad = \quad \tfrac{1}{2}\left[D\left(q \,\middle\|\, \mathrm{avg}(q,r)\right) + D\left(r \,\middle\|\, \mathrm{avg}(q,r)\right)\right]$$

This is beyond the scope of this metric overview so for more information see further sources.

This Metric is not included in the **SimMetric open source library**. If you have an implementation of this please add it to the **SimMetric Library** or mail **reverendsam@users.sourceforge.net**

Harmonic Mean

The Harmonic Mean metric is a measure used to quantify the difference (sometimes called divergence) between probability distributions. This could more accurately be called the Harmonic Geometric Mean. This is a form of the Kullback-Leibler divergence. The formula is displayed beneath but not mentioned further here.

$$M(P||Q) = \sum_{i=1}^{n} \frac{2p_i q_i}{p_i + q_i}.$$

This Metric is not included in the **SimMetric open source library**. If you have an implementation of this please add it to the **SimMetric Library** or mail **reverendsam@users.sourceforge.net**

Skew divergence

Skew Divergence is discussed in the following paper:

"**On the Effectiveness of the Skew Divergence for Statistical Language Analysis**" by Lillian Lee, Artificial Intelligence and Statistics 2001, pp. 65-72.

Skew divergence is defined by the following formula

$$s_\alpha(q,r) \quad = \quad D\left(r \parallel \alpha q + (1-\alpha) r\right)$$

This is again extended from the Kullback-Leibler divergence so is not mentioned any further here.

This Metric is not included in the **SimMetric open source library**. If you have an implementation of this please add it to the **SimMetric Library** or mail **reverendsam@users.sourceforge.net**

Confusion Probability

The confusion probability, which estimates the substitutability of two given words, is again an approximation of the Kullback-Leibler divergence, the formular is detailed beneath but not discussed further here.

$$\mathrm{conf}(q,r,P(x')) \quad = \quad P(x') \sum_y q(y) r(y) / P(y)$$

This Metric is not included in the **SimMetric open source library**. If you have an implementation of this please add it to the **SimMetric Library** or mail **reverendsam@users.sourceforge.net**

Tau

The Tau metric, is again an approximation of the Kullback-Leibler divergence, the formular is detailed beneath but not discussed further here.

$$\tau(q,r) \quad = \quad \sum_{y_1,y_2} \mathrm{sign}\left[(q(y_1) - q(y_2))(r(y_1) - r(y_2))\right] \Big/ \left(2\binom{|V|}{2}\right)$$

This Metric is not included in the **SimMetric open source library**. If you have an implementation of this please add it to the **SimMetric Library** or mail **reverendsam@users.sourceforge.net**

Fellegi and Sunters (SFS) metric

This method was developed for relation databases merge/purge problem back in 1959 by Newcombe et al. This work is detailed in the follwing paper:

Newcombe, H. B., Kennedy, J. M., Axford, S. J., and James, A. P. (1959), "Automatic Linkage of Vital Records," Science, 130 954-959.

However the title of this approach goes to the more often cited Fellegi and Sunters who published the following paper ten years later.

Fellegi, I. P., and Sunter, A. B. (1969), "A Theory for Record Linkage," Journal of the American Statistical Association, 40, 1183-1210.

Fellegi and Sunter or SFS is an estimator taking into account the relevance of related fields, for example peoples names may be combined if not only the names are similar but if the Date of Birth and email addresses are also. This approach has been used extensively for many years in database applications.

This Metric is not included in the **SimMetric open source library**. Although is employeed at a higher level in Sam chapman's work. If you have an compatible implementation of this please add it to the **SimMetric Library** or mail **reverendsam@users.sourceforge.net**

TFIDF or TF/IDF

This is not typically considered to be a similarity metric as it is typically used to consider the relevance of pages for web page searches. This is again a vector based approach whereby weights are applied to terms in respect to their frequency within the predefined corpus, (typically the internet) and the inverse frequency within the test string or document. This provides a relevance metric for the string to the given query, hence why this technique is often used in searching rather than a similarity metric explicitly. This technique may however be used in terms of a similarity measure however this technique is not detailed any further here.

This Metric is not included in the **SimMetric open source library**. If you have an implementation of this please add it to the **SimMetric Library** or mail **reverendsam@users.sourceforge.net**

FastA

This is a specific and optermised technique for Sequence similarity and homology searching against nucleotide and protein databases. This technique has four basic stages culminatinmg in the **Needleman-Wunch** metric. As this is such a specific technique this similarity measure is not discussed further here.

This Metric is not included in the **SimMetric open source library**. If you have an implementation of this please add it to the **SimMetric Library** or mail **reverendsam@users.sourceforge.net**

BlastP

Blast or, Basic Local Alignment Search Tool has a number of varients Blast, Gapped BLAST, BlastP, BlastPSI and so on. These methodologies are specific and optermised techniques for Sequence similarity and homology searching specific against nucleotide and protein databases. As this is such a specific domain this similarity measure is not discussed further here.

This Metric is not included in the **SimMetric open source library**. If you have an implementation of this please add it to the **SimMetric Library** or mail **reverendsam@users.sourceforge.net**

Maximal matches

This is often used within the protien and DNA sequence community to refer to a technique of finding the maximum matching sequences of protein or DNA with tested sequences. This technique is in fact identical apart from implementation issues to the **q-Gram** metric so is not discussed fiurther here.

This Metric is not included in the **SimMetric open source library**. If you have an implementation of this please add it to the **SimMetric Library** or mail **reverendsam@users.sourceforge.net**

q-gram

Q-grams are typically used in approximate string matching by “sliding” a window of length q over the characters of a string to create a number of 'q' length grams for matching a match is then rated as number of q-gram matches within the second string over possible q-grams.

The intuition behind the use of q-grams as a foundation for approximate string processing is that when two strings s1 and s2 are within a small edit distance of each other, they share a large number of q-grams in common.

Consider the following example. The positional q-grams of length q=3 for string "sam chapman" are f(1,##s), (2,#sa), (3,sam), (4,am), (5,m c), (6, ch), (7,cha), (8,hap), (9,apm), (10,pma), (11,man), (12,an%), (13,n%%),

where '#' and '%' indicate the begining and end of the string. So getting the q-grams for two query strings allows the count of identical q-grams over the total q-grams available. this method is frequently employed as a 'fuzzy' search in databases to allow non exact matching of queries. A rating is also often given by the ratio of the q-grams diatnce metric.

For a wider discussion on q-grams and adding them to database applications see the following paper:

"**Using q-grams in a DBMS for Approximate String Processing**" by Luis Gravano, Panagiotis G, Ipeirotis H. V, Jagadish, Nick Koudas S. Muthukrishnan, Lauri Pietarinen and Divesh Srivastava.

See aslo **Maximal matches** for the protein or DNA sequence version of this technique.

This Metric is included in the **SimMetric open source library**.

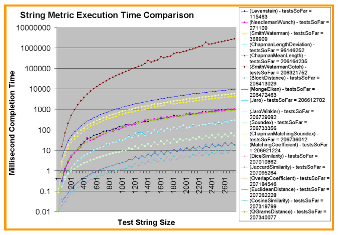
Ukkonen's Algorithms

Ukkonens Algorithms concern transposition invariant string matching. This means that strings must match when all the characters of either of them can be shifted by some amount *t*. By shifting he means that the strings are sequences of numbers and we add or subtract *t* from each character of one of them. For example the ASCII string "ABBA" is shifted by one to the string "BCCB". This is a very specific similarity technique so this is not discussed further here.

This Metric is not included in the **SimMetric open source library**. If you have an implementation of this please add it to the **SimMetric Library** or mail **reverendsam@users.sourceforge.net**

Comparisons of similarity metrics

Here is an in depth **performance evaluation** of the execution **cost of string metrics** for varrying size input.



This clearly demonstrates the increased cost in variable affine gap costs in the **smith waterman gotoh** metric.

The following papers compare similarity metrics showing the strengths for techniques for a task.

Big comparison paper is "**A comparison of string metrics for matching names and records**" by cohen

another comparison paper is "**On the Effectiveness of the Skew Divergence for Statistical Language Analysis**" by Lillian Lee, Artificial Intelligence and Statistics 2001, pp. 65-72.

Workshops concerning Information Integration

http://www.kr.tuwien.ac.at/colognet_ws/program.html

http://www.dagstuhl.de/02181/

http://www.isi.edu/info-agents/workshops/ijcai03/iiweb.html
http://www.isi.edu/info-agents/workshops/ijcai03/proceedings.htm

http://www.isi.edu/ariadne/aiii98-wkshp/proceedings.html

http://www.cs.man.ac.uk/img/FMII-2001/

http://www.informatik.uni-trier.de/~ley/db/conf/wiiw/wiiw2001.html

http://www.comp.nus.edu.sg/~icom/misc/iiwas99/

Other links to papers of interest

http://www.cs.rit.edu/~amt/datamining/LinksToPapers.htm

http://paul.rutgers.edu/~weiz/readinglist.html

Information Integration projects

Systems of interest for Information Integration

- SIMS/Ariadne - University of Southern California/ISI
- TSIMMIS - Stanford
- Information Manifold - AT&T Research
- Garlic - IBM Almaden
- Tukwila - University of Washington
- DISCO - University of Maryland/INRIA and Dyade
- HERMES - University of Maryland
- InfoMaster - Stanford University
- InfoQuilt - University of Georgia
- MIX - UCSD
- Digital Libraries Project - Stanford University
- Internet Softbot - University of Washington
- WebWatcher Project - Carnegie Mellon University

Industrial Implementations

- IBM DB2 DataJoiner
- Enterprise data integration middleware - DataJoiner, (above), functionality now incorporated in IBM DB2 UDB
- Cohera - (link removed bought into PeopleSoft and since disappeared from the web?)
- Nimble Technologies
- WhizBangLabs
- Fetch
- Ensosys Markets Inc
- Mergent Inc
- Flamingo

Other Links

A gloassary of record linking terms

Other Similarity metric types and research

Music Similarity Metrics

Mail me for any further details at
Sam Chapman (c) 2006 sam@dcs.shef.ac.uk