International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Note on the String Metric for Word Similarity

Sanil Shanker K. P.*[1], Megha Narayanan[1], Arunodhaya K. Nambiar[1]

Submitted:11/03/2024 **Revised**: 26/04/2024 **Accepted**: 03/05/2024

Abstract: This paper presents a string metric for measuring the similarity between words. The distance function satisfies the axioms of non-negativity, reflexivity, symmetry, and triangle inequality. A comparative study of the string metric is carried out with Hamming and Levenshtein distances for word matching task.

Keywords: *Distance measure, Hamming distance, Levenshtein distance, String metric, Word matching*

1. Introduction

Distance measure has unique importance for recognizing the strength of the clustering patterns which helps to study the bond between the associated structures. There are quite a few concepts of distance measures. Jaccard distance, which is the complement of the Jaccard coefficient, measures the dissimilarity between two data sets by subtracting the Jaccard coefficient from 1[1, 2]. Introduced by Hermann Minkowski, Manhattan distance is computed between two points as the sum of the absolute differences of the Cartesian coordinates[3, 4]. Instead of considering two points, Mahalanobis distance measures the distance between a point and a distribution [5].

String matching algorithms help to locate a pattern with some features within a given arrangement of symbols. A string metric helps to measure the distance between two text strings. For example, Hamming distance measures the number of positions with mismatched characters between two strings of equal length[6]. Levenshtein distance between two strings is the minimal number of edit operations like deletion, insertion, and alteration of a single character required to change one string to another [7]. To measure the edit distance between two sequences, two string metrics namely Jaro-Winkler distance, [8] and Damerau– Levenshtein distance are used [7, 9, 10]. Jaro-Winkler distance works for strings of equal or varying lengths; but by considering the fact that it does not obey the triangle inequality, this distance is not considered as metric in the mathematical sense [11, 12].

String metric, which is a distance measure that computes the distance between two strings, can be expressed in terms of matches and mismatches of the string elements. To qualify as distance measure, a metric must satisfy the axioms of non- negativity, reflexivity, symmetry, and triangle inequality [12]. Review on Logical matching strategy explores two real world applications, locating and comparing the sequential pattern of finite length [13]. Through this paper, we present a string-to-string distance measure based on logical match.

[1]*Department of Information Technology, Kannur University, Kerala, India*
** Corresponding Author Email: sanil@kannuruniv.ac.in*

2. Related work

String metric plays a vital role in text-linked research and applications in areas such as identifying the word similarity [14] comparing molecular sequences, and text mining. In general, there is a wide literature on quantifying the similarity between molecular sequences, such as Needleman-Wunsch [15] and Smith-Waterman [16], but there are very few publications concerning the measurement of likeness between short strings, for example, Levenshtein and Hamming distances. Studies on interrelated work admissible to explore the characteristics of earlier approaches and to recognize the specific problems for computing string similarity. Related works can be categorized into two types: one significant type of distance measure is edit distance, in which distance is the cost of the finest arrangement of edit operations. Conventional edit operations are character insertion, deletion, and substitution, and each operation must be allocated a cost. The Levenshtein distance assigns a unit cost to all edit operations [7]. On the other hand, Hamming distance measures the distance between two equal-length sequences of symbols as the number of positions at which the corresponding symbols are not same [6].

3. Method

The strings are arranged so that each symbol coincides with its corresponding index and then proceeds to match for computing the number of identical symbols. Here, the comparison of two text strings returns the number of matches by generating indices corresponding to the symbols.

3.1 Preliminary

3.1.1 Distance measure: Distance measure is a function D(P, Q) that takes two points as arguments and produces a real number[12]. The distance measure satisfies the

following axioms, i) $D(P, Q) \geq 0$ ii) $D(P, Q) = 0$, if and only if $P = Q$ iii) $D(P, Q) = D(Q, P)$ iv) $D(P, Q) \leq D(P, R) + D(R, Q)$

3.1.2 Computation of the number of matching symbols:

Let $P = p_1p_2p_3\ldots$ pn and $Q = q_1q_2\, q_3\ldots q_n$, be two strings of equal length, n (where $|P| = |Q| = n$). Then the number of matching symbols of strings P and Q can be computed by generating the indices [13].

Example: Let P and Q be two strings BELLA and BELLE respectively, where the length of the strings, $|P| = |Q| = 5$. Here, the word BELLA is constructed using the alphabet set Σ_P = {B, E, L, A} and the word BELLE is created by using the alphabet set Σ_Q = {B, E, L}. The string BELLA is arranged so that each symbol coincides with its corresponding index as in Table 1.

Table 1. Arrangement of string P with alphabet set Σ_P

Indices	B	E	L	A
5				#
4			#	
3			#	
2		#		
1	#			

Here, the symbol # is used to indicate the index where the alphabet is placed. Indices of the string P can be represented as: <B(1); E(2); L(3, 4); A(5) >. The string BELLE is arranged so that each symbol coincides with its corresponding index as in Table 2.

Table 2. Arrangement of string Q with alphabet set Σ_Q

Indices	B	E	L
5		#	
4			#
3			#
2		#	
1	#		

Here, the indices of the string Q can be represented as: <B(1); E(2,5); L(3, 4)>. On comparison, only the 5^{th} index of the strings P and Q does not have a match as mentioned in Table 3 and the remaining four indices have matched, so the total number of matches, (say k) equals 4.

Table 3. Comparison of strings P and Q to find matches

Indices	Symbol	Locations		Match / Mismatch	Number of match
1	B	1		1^{st} Match	1
2	E	2	5	2^{nd} Match	1
3	L	3		3^{rd} Match	1
4	L	4		4^{th} Match	1
5	A			1^{st} Mismatch	

3.2 String distance measure

The distance function is defined as the difference between the length of the string and total number of matches, where the number of matches is determined by matching the indices. Let $P = p_1p_2p_3\ldots\ p_n$ and $Q = q_1q_2q_3\ldots q_n$ be two strings of equal length n, (where $|P| = |Q| = n$). The distance function, $D(P, Q) = n - k$, where k is the number of matching symbols of string P while comparing with the string Q.

Axiom-1: $D(P, Q) \geq 0$ (Non-negativity). The distance measure using matches satisfies $D(P, Q) \geq 0$.

Proof sketch: As the number of matches between the strings are always less than or equal to the length of the strings, the distance between the strings becomes non-negative. Therefore, $D(P, Q) = n - k \geq 0$. This implies distance is non- negative.

Example: Let P and Q be the strings BELLA and BELLE respectively, where $|P| = |Q| = 5$. By using the number of matching symbols, k can be computed as 4 (as per 3.1.2).

P	B	E	L	L	A
Q	B	E	L	L	E

Distance is, n – (number of matching symbols) = 5- 4 = 1. The result shows that distance is non- negative. This satisfies $D(P, Q) \geq 0$.

Axiom-2: $D(P, Q) = 0$, if and only if $P = Q$ (Reflexivity).

The distance measure using matches satisfies $D(P, Q) = 0$, if and only if $P = Q$.

Proof sketch: Distance between the strings equals to zero only if the number of matches between the strings and length of the string are equal, that is, only if both strings are identical. Therefore, the number of mismatching symbols in between the strings P and Q becomes zero. That

is, D(P, Q) = n- k = k- k = 0, since the number of matching symbols, k = n.

Example: If the two strings are identical, then P = Q.

P	B	E	L	L	A
Q	B	E	L	L	A

The string P is arranged so that each symbol coincides with its corresponding index as in Table 4.

Table 4. Arrangement of string P with alphabet set Σ_P

Indices	B	E	L	A
5				#
4			#	
3			#	
2		#		
1	#			

Indices of the string P are represented as: <B(1); E(2); L(3, 4); A(5)>. The string Q is arranged so that each symbol coincides with its corresponding index as in Table 5.

Table 5. Arrangement of string Q with alphabet set Σ_Q

Indices	B	E	L	A
5				#
4			#	
3			#	
2		#		
1	#			

Indices of the string Q are represented as: <B(1); E(2); L(3, 4); A(5) >. Here, number of matching symbols while comparing the strings P and Q can be computed as in Table 6.

Table 6. Comparison of strings P and Q to find matches

Indices	Symbol	Locations	Match/ Mismatch	Number of match
1	B	1	1st Match	1
2	E	2	2nd Match	1
3	L	3	3rd Match	1
4	L	4	4th Match	1
5	A	5	5th Match	1

From Table 6, total number of matches = 5. This implies that the number of matches, k equals to the length of the strings P and Q (where, |P|= |Q|) and n- k = 0. That is, 5-5= 0, where n= 5 and k= 5. This satisfies D(P, Q) = 0, if and only if P = Q.

Axiom-3: D(P, Q) = D(Q, P) (Symmetry). The distance measure satisfies D(P, Q) = D(Q, P).

Proof sketch: In both the cases (the string P compares with the string Q or the string Q compares with the string P), the number of matching symbols, k remains the same. This implies the distance is symmetric.

Example: Let P and Q be the strings BELLA and BELLE respectively, where |P|= |Q|= 5.

This can be represented as: D(BELLA, BELLE)

P	B	E	L	L	A
Q	B	E	L	L	E

Here, the number of matching symbols, k = 4. Similarly, D(BELLE, BELLA)

Q	B	E	L	L	E
P	B	E	L	L	A

Here also, the number of matching symbols, k = 4. In both cases, the number of matching symbols, k remains the same. This implies that the distance is symmetric, D(P, Q) = D(Q, P).

Axiom-4: D(P, Q) ≤ D(P, R) + D(R, Q) (Triangle inequality). The distance measure satisfies D(P, Q) ≤ D(P, R) + D(R, Q).

Proof sketch: Let P be $p_1p_2p_3\ldots p_n$, Q be $q_1q_2q_3\ldots q_n$ and R be $r_1r_2\ r_3\ldots r_n$ be three strings of equal lengths n (where, |P| = |Q| = |R| = n). Let k be the number matching symbols between the strings P and Q, k_1 be the number of matches between the strings P and R, and k_2 be the number of matching symbols between the strings R and Q. Here, the value of k will always be less than or equal to the sum of k_1 and k_2 for any three strings with equal length. Here, $n - k \leq n - k_1 + n - k_2$ implies $n - k \leq (n + n) - (k_1 + k_2)$; that is, $n - k \leq 2n - (k_1 + k_2)$. That is the number of mismatching symbols (n - k) between strings P and Q is less than or equals to the sum of the number of mismatching symbols ($n – k_1$) between the strings P and R; and the number of mismatching symbols ($n – k_2$) between the strings R and Q. This proves that the distance measure satisfies triangle inequality.

Example: Let P, Q, and R be the strings BELLA, BELLE, and BELAA respectively. The distance between the strings P and Q, D(P, Q) ≤ D(P, R) + D(R, Q). This implies the number of mismatching symbols between the strings P and Q ≤ Number of mismatching symbols between the strings P and R + Number of mismatching symbols between the strings R and Q. This implies, D(BELLA, BELLE) ≤

D(BELLA, BELAA) + D(BELAA, BELLE); this implies the number of mismatching symbols, $1 \leq 1 + 2$.

That is,

P	B	E	L	L	A
Q	B	E	L	L	E

≤

P	B	E	L	L	A
R	B	E	L	A	A

+

R	B	E	L	A	A
Q	B	E	L	L	E

This satisfies triangle inequality, D(P, Q) ≤ D(P, R) + D(R, Q).

4. Discussion And Conclusion

The results show the realization of the method for comparing text strings of various sizes with real data. To evaluate the proposed string metric, the method was tested with English words[17], and we validated the difference with the Hamming and Levenshtein distances. In section 3, axioms 1 through 4 satisfy the norms of distance measure; thus, we computed the distance by taking pairs of strings. Table 7 exemplifies the proposed string metric for word matching task. Here, the number of matches is calculated between the strings P and Q. From the set of strings in Table 7, it is evident that all pairs of strings obey the axioms of non-negativity, reflexivity, symmetry, and triangle inequality. The comparison results of the proposed distance measure with Hamming distance and Levenshtein distance is illustrated in Table 8. Alphabets represented red in color indicates the mismatch, while those in black color symbolize the match between the strings. Hamming distance and proposed distance determines distance between two strings of equal length, whereas Levenshtein distance can measure the distance even if the strings have different lengths. Hamming distance measures the minimum number of positions by which the strings differ whereas the proposed metric computes the distance by using the number of matches. For any two strings, the minimum number of positions determined in Hamming distance is same as the total number of mismatches calculated by the proposed distance measure. From Table 8, we can observe that the distance calculated by Hamming distance and the proposed distance for same pair of strings are the same. On the other hand, Levenshtein distance incorporates operations other than substitution; thus, Levenshtein distance for any two strings is not always equal to Hamming or the proposed distance measure for the same pair of strings. In the string metric based on Logical matching strategy, matches between the symbols are computed and the distance is calculated subsequently.

Table 7. Distance between words using the proposed metric

String P	String Q	String R	k	D(P,Q)	D(Q, P)	D(P,Q) ≤ D(P, R) + D(R, Q)
TEA	TEE	TOE	2	1	1	1≤ 2+1
TEA	TEA	TEA	3	0	0	0≤ 0+0
DEAR	DEER	READ	3	1	1	1≤ 2+3
HEEL	HEAL	HELL	3	1	1	1≤ 1+1
TIRE	TIER	TREE	2	2	2	2 ≤ 2+2
STAN	TANK	SKAT	0	4	4	4 ≤ 2+4
STEEL	STEAL	STALE	4	1	1	1≤ 3+3
THERE	THEIR	THREE	3	2	2	2 ≤2+3
CENTER	CENTRE	RENTER	4	2	2	2 ≤1+3
WATCH	DREAM	LEARN	0	5	5	5 ≤ 5+5

Table 8. Comparison of Hamming, Levenshtein, and the proposed distance measure

String P	String Q	Hamming Distance	Levenshtein Distance	Proposed Distance
TEA	TEE	1	1	1
TEA	TEA	0	0	0
DEAR	DEER	1	1	1
HEEL	HEAL	1	1	1

TIRE	TIER	2	2	2
STAN	TANK	4	2	4
STEEL	STEAL	1	1	1
THERE	THEIR	2	2	2
CENTER	CENTRE	2	2	2
WATCH	DREAM	5	5	5

Acknowledgements

This work is supported by University Grants Commission (Grant Number: F.2264-MRP-15-16-KLCA028-UGC-MRP)

Author contributions

Sanil Shanker K P: Investigation, Conceptualization, Methodology, Literature review, Writing- Original draft preparation.

Megha Narayanan: Literature review, Writing-Reviewing

Arunodhya K Nambiar: Writing- Reviewing

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Sven Kosub. A note on the triangle inequality for the Jaccard distance, Pattern Recognition Letters, 2019 (120): 36-38. https://doi.org/10.1016/j.patrec.2018.12.007

[2] Tibrewal B., Chaudhury G.S., Chakraborty S., Kairi A. Rough Set-Based Feature Subset Selection Technique Using Jaccard's Similarity Index. In: Chakraborty M., Chakrabarti S., Balas V., Mandal J. (eds) Proceedings of International Ethical Hacking Conference 2018. Advances in Intelligent Systems and Computing, vol 811. Springer, Singapore. 2019. https://doi.org/10.1007/978-981-13-1544-2_39.

[3] Kretz T., Bönisch C., Vortisch P. Comparison of Various Methods for the Calculation of the Distance Potential Field. In: Klingsch W., Rogsch C., Schadschneider A., Schreckenberg M. (eds) Pedestrian and Evacuation Dynamics 2008. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-04504-2_29.

[4] José M. Merigó, and Anna M. Gil-Lafuente. Using the OWA Operator in the Minkowski Distance, World Academy of Science, Engineering and Technology. 2008: 21.

[5] P. Mahalanobis. On the generalized distance in statistics Proc. Nat. Inst. Sci. India (Calcutta) 1936(2): 49–55.

[6] Hamming, Richard W. Error detecting and error correcting codes, The Bell system technical journal.1950:147-160. DOI: 10.1002/j.1538-7305.1950.tb00463.x

[7] Levenshtein, Vladimir. Binary codes capable of correcting spurious insertions and deletion of ones, Problems of information Transmission. 1965: 8-17.

[8] Cohen, William, Pradeep Ravikumar, Stephen Fienberg. A comparison of string metrics for matching names and records, KDD workshop on data cleaning and object consolidation. 2003 (3).

[9] Zhao C., Sahni S. String correction using the Damerau- Levenshtein distance, BMC bioinformatics. 2019: 1-28. https://doi.org/10.1186/s12859-019-2819-0

[10] Fred J Damerau. A technique for computer detection and correction of spelling errors, Communications of the ACM.1964:171-176. https://doi.org/10.1145/363958.363994

[11] Van der Loo, Mark PJ. The stringdist package for approximate string matching, R J. 6.1. 2014.

[12] Rajaraman, Anand, and Jeffrey David Ullman. Mining of massive datasets. Cambridge University Press. 2011.

[13] Sanil Shanker KP, Elizabeth Sherly, Jim Austin. A note on two applications of Logical Matching Strategy, Applied Artificial Intelligence. 2011: 708-720.

[14] Carla Pires, Afonsa Cavaco & Marina Vigário. Towards the Definition of Linguistic Metrics for Evaluating Text Readability. Journal of Quantitative Linguistics. 2017: 319-349. https://doi.org/10.1080/09296174.2017.1311448

[15] Needleman S B and Wunsch C D. A general method applicable to the search for similarities in the amino acid sequence of two proteins, J. Mol. Biol.. 197(48): 443–453.

[16] Smith T F and Waterman M S. Identification of Common Molecular Subsequences, J. Mol. Bio. 1981: 195–197.

[17] Simpson, J. A., Weiner, E. S. C., and Oxford University Press. The Oxford English Dictionary. Oxford: Clarendon Press. 1989.