אוניברסיטת בן-גוריון בנגב
جامعة بن غوريون في النقب
Ben-Gurion University of the Negev

When is the Normalized Edit Distance over Non-Uniform Weights a Metric?

Dana Fisman and Ilay Tzarfati

The Edit (Levenshtein) Distance

Types of operations:		Uniform weight
- Delete the letter σ	$\begin{bmatrix}\sigma\\ \varepsilon\end{bmatrix}$	1
- Insert the letter σ	$\begin{bmatrix}\varepsilon\\ \sigma\end{bmatrix}$	1
- Substitute the letter σ with σ'	$\begin{bmatrix}\sigma\\ \sigma'\end{bmatrix}$	1
- No change	$\begin{bmatrix}\sigma\\ \sigma\end{bmatrix}$	0

w1= aab_cde
aabcde
w2= abpcg _abpcg_

The edit path $\begin{bmatrix}a\\ \varepsilon\end{bmatrix}\begin{bmatrix}a\\ a\end{bmatrix}\begin{bmatrix}b\\ b\end{bmatrix}\begin{bmatrix}\varepsilon\\ p\end{bmatrix}\begin{bmatrix}c\\ c\end{bmatrix}\begin{bmatrix}d\\ g\end{bmatrix}\begin{bmatrix}e\\ \varepsilon\end{bmatrix}$

The edit path cost: 4

ED(w_1 ,w_2) = min {weight of ops needed to transforms w_1 to w_2 }

Issue with ED

w_1= aaaaa
w_2= abpcg

ED(w1,w2) =4

w_3= aaaaaaa...aaaaa
w_4= aaaaaaa...abpcg

ED(w3,w4) =4

But these are much more similar

How can we normalize ED?

- The sum edit-distance: $ED_{sum}(w_1, w_2)$

$$= \frac{ED(w_1,w_2)}{|w_1|+|w_2|}$$

- The max edit-distance: $ED_{max}(w_1, w_2)$

$$= \frac{ED(w_1,w_2)}{max(|w_1|,|w_2|)}$$

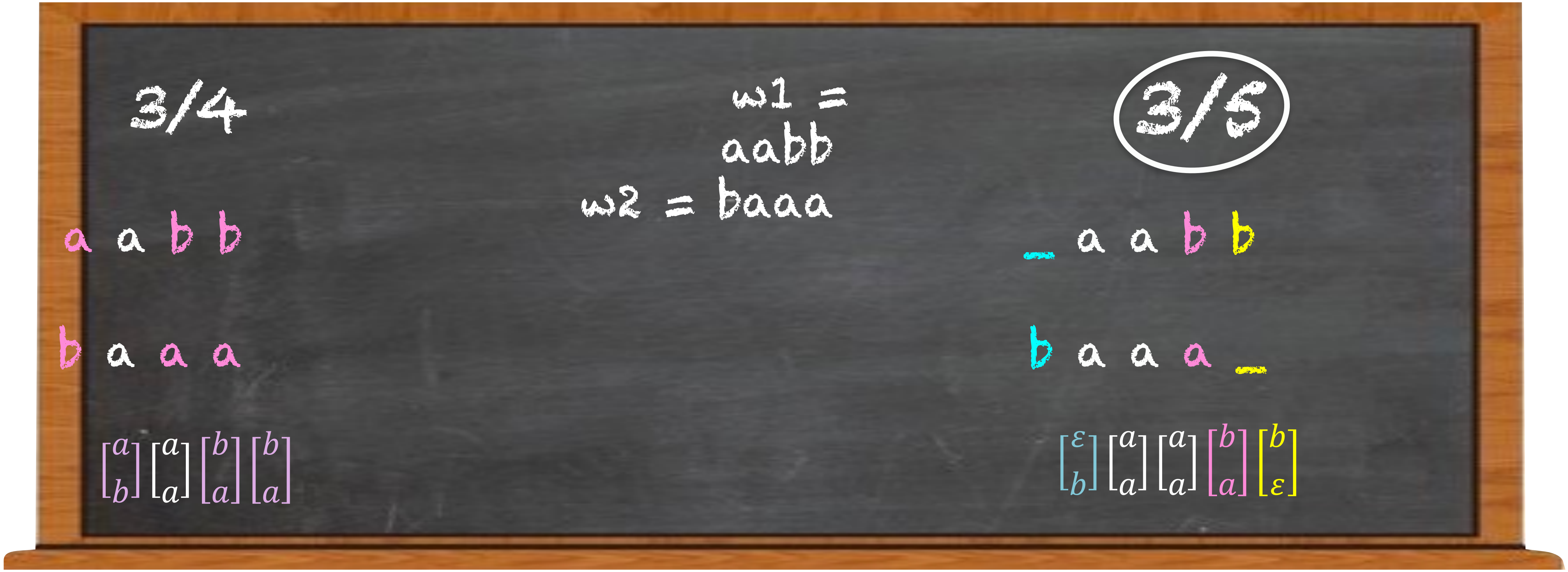
- The min edit-distance: $ED_{min}(w_1, w_2)$

All the above do not satisfy the triangle inequality! (hence is not a metric)

How can this be solved?
Marzal and Vidal suggested
dividing by the length of the edit path

[Marzal & Vidal 93]

Normalizing using path length



3/4

w1 = aabb

w2 = baaa

3/5

a a b b

b a a a

$\begin{bmatrix} a \\ b \end{bmatrix} \begin{bmatrix} a \\ a \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix}$

_ a a b b

b a a a _

$\begin{bmatrix} \varepsilon \\ b \end{bmatrix} \begin{bmatrix} a \\ a \end{bmatrix} \begin{bmatrix} a \\ a \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} \begin{bmatrix} b \\ \varepsilon \end{bmatrix}$

$_2$) = min { cost(P): P is an edit path transforming w_1 to w_2 }

Cost(P) = $\frac{weight(P)}{length(P)}$

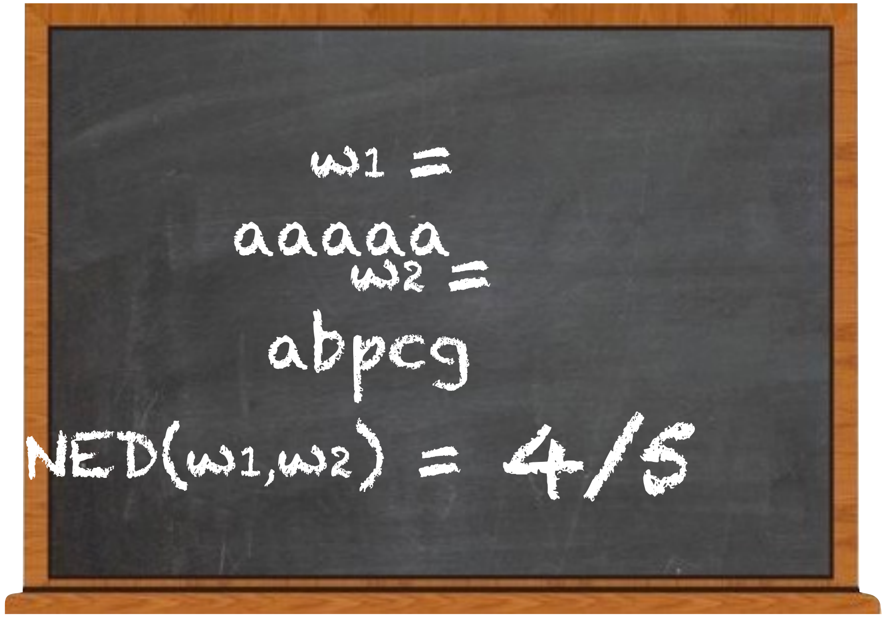
© Ilay Tzarfati

[Marzal & Vidal 93]

NED- Normalized Edit Distance

w_1,w_2) = min { cost(P): P is an edit path transforming w_1 to w_2 }

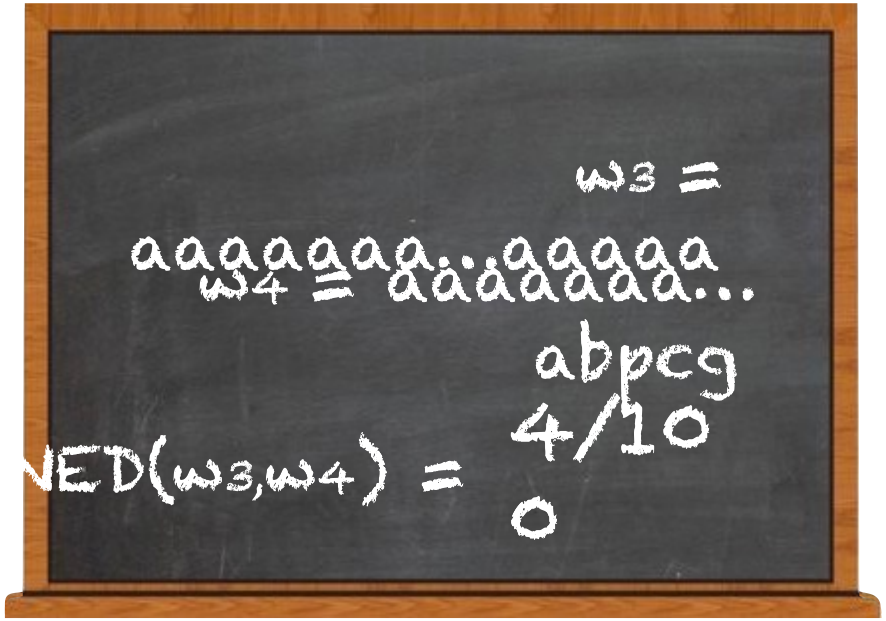
Cost(P) = $\frac{weight(P)}{length(P)}$



w1 = aaaaa

w2 = abpcg

NED(w1,w2) = 4/5



w3 = aaaaaaa...aaaaa

w4 = aaaaaaa... abpcg

NED(w3,w4) = 4/10 0

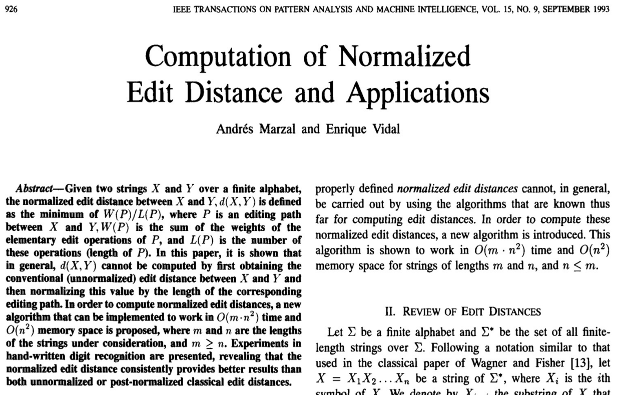
Matches our intuition ☺

But is it a metric ???

© Ilay Tzarfati

Is NED a Metric ?

Normalized Edit Distance (NED)



926 IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 15, NO. 9, SEPTEMBER 1993

Computation of Normalized Edit Distance and Applications

Andrés Marzal and Enrique Vidal

***Abstract*—Given two strings X and Y over a finite alphabet, the normalized edit distance between X and $Y, d(X, Y)$ is defined as the minimum of $W(P)/L(P)$, where P is an editing path between X and $Y, W(P)$ is the sum of the weights of the elementary edit operations of P, and $L(P)$ is the number of these operations (length of P). In this paper, it is shown that in general, $d(X, Y)$ cannot be computed by first obtaining the conventional (unnormalized) edit distance between X and Y and then normalizing this value by the length of the corresponding editing path. In order to compute normalized edit distances, a new algorithm that can be implemented to work in $O(m \cdot n^2)$ time and $O(n^2)$ memory space is proposed, where m and n are the lengths of the strings under consideration, and $m \geq n$. Experiments in hand-written digit recognition are presented, revealing that the normalized edit distance consistently provides better results than both unnormalized or post-normalized classical edit distances.**

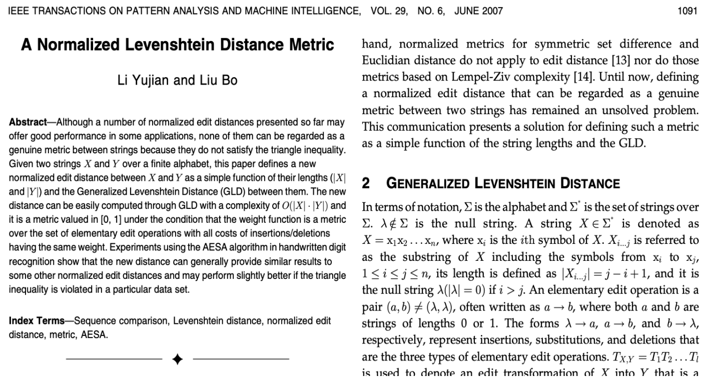
properly defined *normalized edit distances* cannot, in general, be carried out by using the algorithms that are known thus far for computing edit distances. In order to compute these normalized edit distances, a new algorithm is introduced. This algorithm is shown to work in $O(m \cdot n^2)$ time and $O(n^2)$ memory space for strings of lengths m and n, and $n \leq m$.

II. Review of Edit Distances

Let Σ be a finite alphabet and Σ^* be the set of all finite-length strings over Σ. Following a notation similar to that used in the classical paper of Wagner and Fisher [13], let $X = X_1X_2 \dots X_n$ be a string of Σ^*, where X_i is the ith symbol of X. We denote by

[Marzal & Vidal 93]

Generalized Edit Distance (GED)



IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 29, NO. 6, JUNE 2007 1091

A Normalized Levenshtein Distance Metric

Li Yujian and Liu Bo

Abstract—Although a number of normalized edit distances presented so far may offer good performance in some applications, none of them can be regarded as a genuine metric between strings because they do not satisfy the triangle inequality. Given two strings X and Y over a finite alphabet, this paper defines a new normalized edit distance between X and Y as a simple function of their lengths ($|X|$ and $|Y|$) and the Generalized Levenshtein Distance (GLD) between them. The new distance can be easily computed through GLD with a complexity of $O(|X| \cdot |Y|)$ and it is a metric valued in [0, 1] under the condition that the weight function is a metric over the set of elementary edit operations with all costs of insertions/deletions having the same weight. Experiments using the AESA algorithm in handwritten digit recognition show that the new distance can generally provide similar results to some other normalized edit distances and may perform slightly better if the triangle inequality is violated in a particular data set.

Index Terms—Sequence comparison, Levenshtein distance, normalized edit distance, metric, AESA.

✦

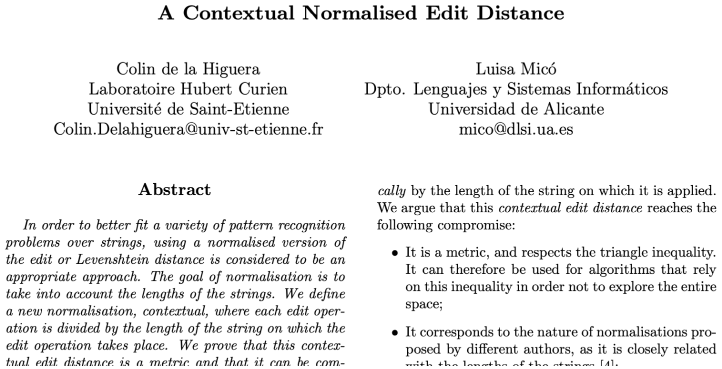
hand, normalized metrics for symmetric set difference and Euclidian distance do not apply to edit distance [13] nor do those metrics based on Lempel-Ziv complexity [14]. Until now, defining a normalized edit distance that can be regarded as a genuine metric between two strings has remained an unsolved problem. This communication presents a solution for defining such a metric as a simple function of the string lengths and the GLD.

2 Generalized Levenshtein Distance

In terms of notation, Σ is the alphabet and Σ^* is the set of strings over Σ. $\lambda \notin \Sigma$ is the null string. A string $X \in \Sigma^*$ is denoted as $X = x_1x_2 \dots x_n$, where x_i is the ith symbol of X. $X_{i \dots j}$ is referred to as the substring of X including the symbols from x_i to x_j, $1 \leq i \leq j \leq n$, its length is defined as $|X_{i \dots j}| = j - i + 1$, and it is the null string $\lambda(|\lambda| = 0)$ if $i > j$. An elementary edit operation is a pair $(a, b) \neq (\lambda, \lambda)$, often written as $a \rightarrow b$, where both a and b are strings of lengths 0 or 1. The forms $\lambda \rightarrow a$, $a \rightarrow b$, and $b \rightarrow \lambda$, respectively, represent insertions, substitutions, and deletions that are the three types of elementary edit operations. $T_{X,Y} = T_1T_2 \dots T_l$ is used to denote an edit transformation of X into Y that is a

[Yujian & Bo '07]

Contextual Edit Distance (CED)



A Contextual Normalised Edit Distance

Colin de la Higuera
Laboratoire Hubert Curien
Université de Saint-Etienne
Colin.Delahiguera@univ-st-etienne.fr

Luisa Micó
Dpto. Lenguajes y Sistemas Informáticos
Universidad de Alicante
mico@dlsi.ua.es

Abstract

In order to better fit a variety of pattern recognition problems over strings, using a normalised version of the edit or Levenshtein distance is considered to be an appropriate approach. The goal of normalisation is to take into account the lengths of the strings. We define a new normalisation, contextual, where each edit operation is divided by the length of the string on which the edit operation takes place. We prove that this contextual edit distance is a metric and that it can be computed by the length of the string on which it is applied. We argue that this *contextual edit distance* reaches the following compromise:

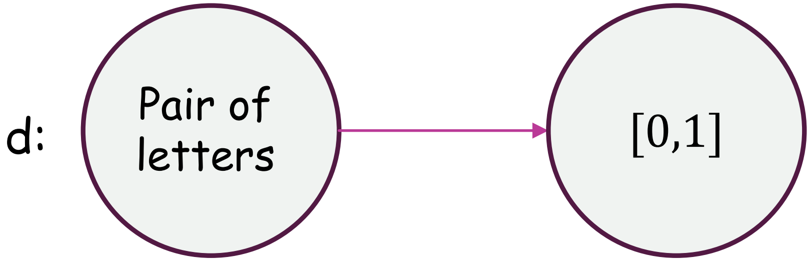
- It is a metric, and respects the triangle inequality. It can therefore be used for algorithms that rely on this inequality in order not to explore the entire space;
- It corresponds to the nature of normalisations proposed by different authors, as it is closely related with the lengths of the strings [4];

[de la Higuera & Mico'08]

The Problem

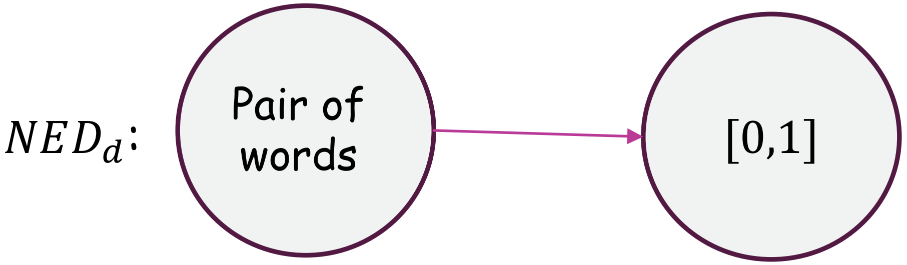
Weight function :

$$d: (\Sigma \cup \{\varepsilon\}) \times (\Sigma \cup \{\varepsilon\}) \to [0,1]$$



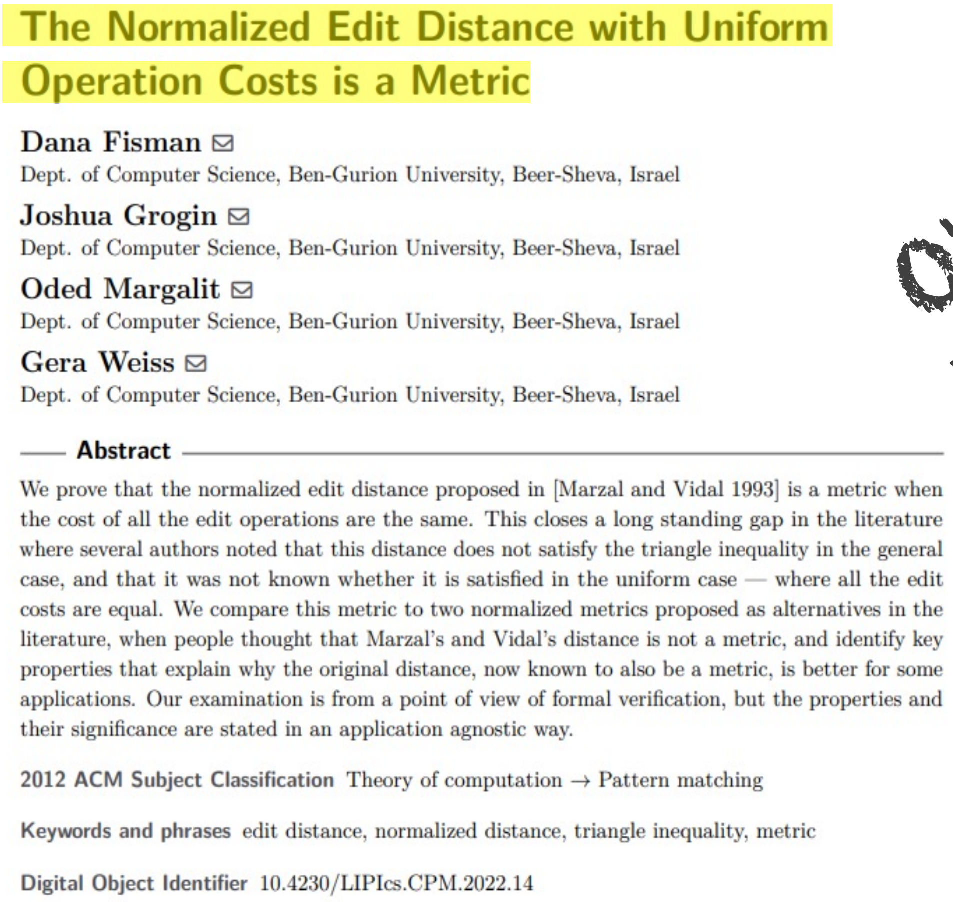
NED_d that induced by d:

$$NED_d: \Sigma^* \times \Sigma^* \to [0,1]$$



When is NED_d a metric ?

If d is uniform, it is !



The Normalized Edit Distance with Uniform Operation Costs is a Metric

Dana Fisman ✉
Dept. of Computer Science, Ben-Gurion University, Beer-Sheva, Israel

Joshua Grogin ✉
Dept. of Computer Science, Ben-Gurion University, Beer-Sheva, Israel

Oded Margalit ✉
Dept. of Computer Science, Ben-Gurion University, Beer-Sheva, Israel

Gera Weiss ✉
Dept. of Computer Science, Ben-Gurion University, Beer-Sheva, Israel

Abstract

We prove that the normalized edit distance proposed in [Marzal and Vidal 1993] is a metric when the cost of all the edit operations are the same. This closes a long standing gap in the literature where several authors noted that this distance does not satisfy the triangle inequality in the general case, and that it was not known whether it is satisfied in the uniform case — where all the edit costs are equal. We compare this metric to two normalized metrics proposed as alternatives in the literature, when people thought that Marzal's and Vidal's distance is not a metric, and identify key properties that explain why the original distance, now known to also be a metric, is better for some applications. Our examination is from a point of view of formal verification, but the properties and their significance are stated in an application agnostic way.

2012 ACM Subject Classification Theory of computation → Pattern matching

Keywords and phrases edit distance, normalized distance, triangle inequality, metric

Digital Object Identifier 10.4230/LIPIcs.CPM.2022.14

OK...
But in the
rest of the
cases ???

Our main result

A necessary and sufficient condition for $\mathrm{NED}_{\mathrm{d}}$ to be a Metric.

Surprisingly, d being a metric is neither sufficient nor necessary.

Essential edit operations

An edit operation is essential, if there exist optimal edit path that use it.

$$a \xrightarrow[\begin{bmatrix}a\\ \varepsilon\end{bmatrix}\ \begin{bmatrix}\varepsilon\\ b\end{bmatrix}]{\begin{bmatrix}a\\ b\end{bmatrix}} b$$

$\forall a \in \Sigma$, (a,ε) and (ε,a) are essential.

d(a,b) ≥ d(a,ε) + d(ε,b) iff (a,b) is inessential.

d(a,ε) = d(ε,a) =0.4

d(b,ε) = d(ε,b) =0.3

d(a,b) = d(b,a) =0.8

w1=a

w2=b

$\begin{bmatrix}a\\ \varepsilon\end{bmatrix}\begin{bmatrix}\varepsilon\\ b\end{bmatrix}$

$NED_d(\text{w1,w2}) = \frac{0.4+0.3}{2} = 0.35$

In the example both (a,b) , (b,a) are inessential

The Necessary and sufficient condition for NED_d to be a Metric

Weight function : Let $d: (\Sigma \cup \{\varepsilon\}) \times (\Sigma \cup \{\varepsilon\}) \to [0,1]$

Let $a,c \in \Sigma \cup \{\varepsilon\}$ and $b \in \Sigma$

✓ 1. identity:

$d(a,c) = 0$ iff $a=c$

✓ 2. symmetry:

$d(a,c) = d(c,a)$

? 3. Relaxed triangle inequality :

$d(a,b) + d(b,c) \geq \min \{ d(a,c), d(a,\varepsilon) + d(\varepsilon,c) \}$

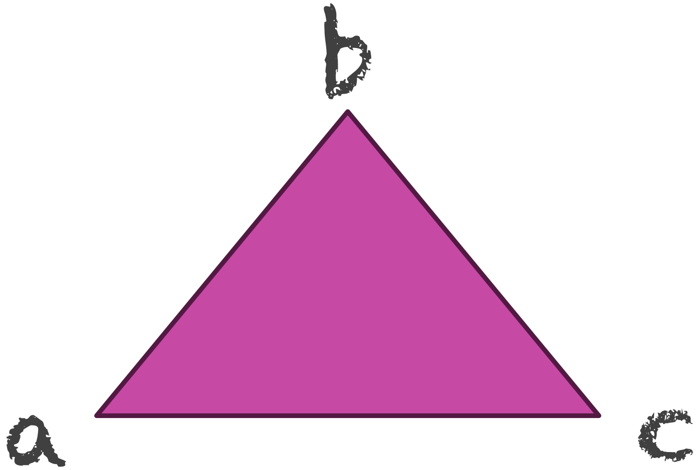
? 4. At least half:

$d(\varepsilon,b) = d(b,\varepsilon) \geq \frac{1}{2}$

A d satisfying those properties, is termed fi

Sketch of the idea: The (Relaxed) Triangle Inequality

$\min\{d(a,c), d(a,\varepsilon) + d(\varepsilon,c)\} \leq d(a,b) + d(b,c)$



Obviously to satisfy the triangle inequality we need:
$d(a,c) \leq d(a,b) + d(b,c)$

But since we know that d(a,c) can be replace with $d(a,\varepsilon)$, $d(\varepsilon,c)$ we require that:

$$\min\{d(a,c), d(a,\varepsilon) + d(\varepsilon,c)\} \leq d(a,b) + d(b,c)$$

Sketch of the idea: At least half

$d(b,\varepsilon) = d(\varepsilon, b) \geq \frac{1}{2}$

We want to make sure that inflating the edit path is not worthwhile .

w_1 = a $w_2 = ac^{100}$ w_3 = b

$d(\varepsilon,c)$ = $d(c,\varepsilon)$ = 0.2,
$d(a,b)$ = $d(b,a)$ = 0.55 ,
$d(a,\varepsilon)$ = $d(\varepsilon,a)$ = $d(b,\varepsilon)$ = $d(\varepsilon,b)$ = 0.6 ...

Optimal edit path $[w_1 \to w_3]$: $\begin{bmatrix}a\\b\end{bmatrix}$
cost : 0.55.

edit path $[w_1 \to w_2]$:

$$\begin{bmatrix}a\\a\end{bmatrix}\begin{bmatrix}\varepsilon\\c\end{bmatrix}\ldots\begin{bmatrix}\varepsilon\\c\end{bmatrix}\begin{bmatrix}\varepsilon\\c\end{bmatrix}$$

cost $\frac{100*0.2}{101}$ = 0.198

edit path $[w_2 \to w_3]$:

$$\begin{bmatrix}a\\b\end{bmatrix}\begin{bmatrix}c\\\varepsilon\end{bmatrix}\ldots\begin{bmatrix}c\\\varepsilon\end{bmatrix}\begin{bmatrix}c\\\varepsilon\end{bmatrix}$$

cost $\frac{100*0.2+0.5}{101}$ = 0.203

NED_d is not Metric
0.198 + 0.203 < 0.55

1. identity:

$d(a,c) = 0$ iff $a=c$

2. symmetry:

$d(a,c) = d(c,a)$

3. Relaxed triangle inequality :

$d(a,b) + d(b,c) \geq \min \{ d(a,c),$
$d(a,\varepsilon) + d(\varepsilon,c)\}$

4. At least half:

$d(\varepsilon,b) = d(b,\varepsilon) \geq \frac{1}{2}$

Are necessary.

Sufficient condition

d is fine→ NED_d is metric

Let $w_1, w_2, w_3 \in \Sigma^*$:

1. $NED_d\ (w_1, w_2) = 0$ iff $w_1 = w_2$ ✓

2. $NED_d\ (w_1, w_2) = NED_d\ (w_2, w_1)$ ✓

3. $NED_d\ (w_1, w_2) + NED_d\ (w_2, w_3) \geq NED_d\ (w_1, w_3)$?

© Ilay Tzarfati

NED_d follows the triangle inequality

$$\begin{bmatrix} a \\ b \\ a \\ b \\ c \end{bmatrix}$$

Compose:

input : P(w1->w2) , P(w2->w3)

Output: P(w1->w3)

d fine -> cost(P(w1->w3)) $\leq$ cost(P(w1->w2)) + cost(P(w2->w3))

Proof idea:

w1 = aaa

w2 = bb

w3 = cccc

P (1->2) $= \begin{bmatrix} a \\ b \end{bmatrix} \begin{bmatrix} a \\ \varepsilon \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}$

P (2->3) $= \begin{bmatrix} \varepsilon \\ c \end{bmatrix} \begin{bmatrix} b \\ c \end{bmatrix} \begin{bmatrix} b \\ c \end{bmatrix} \begin{bmatrix} \varepsilon \\ c \end{bmatrix}$

```
a a a
b _ b
_ b b _
 c c c c
```

Problem

P (1->2) $= \begin{bmatrix} \varepsilon \\ \varepsilon \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \begin{bmatrix} a \\ \varepsilon \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \begin{bmatrix} \varepsilon \\ \varepsilon \end{bmatrix}$

P (2->3) $= \begin{bmatrix} \varepsilon \\ c \end{bmatrix} \begin{bmatrix} b \\ c \end{bmatrix} \begin{bmatrix} \varepsilon \\ \varepsilon \end{bmatrix} \begin{bmatrix} b \\ c \end{bmatrix} \begin{bmatrix} \varepsilon \\ c \end{bmatrix}$

P (1->3) $= \begin{bmatrix} \varepsilon \\ c \end{bmatrix} \begin{bmatrix} a \\ c \end{bmatrix} \begin{bmatrix} a \\ \varepsilon \end{bmatrix} \begin{bmatrix} a \\ c \end{bmatrix} \begin{bmatrix} \varepsilon \\ c \end{bmatrix}$

```
  a a a
_ b _ b
_
_ b _ b
c c _ c
c
```

alignm ent

Examples for fine d's

Distances in [0,n] : Let $d\colon [0,n]\times[0,n] \to [0,1]$ be defined as follows :

$$d(n_1, n_2) = \frac{|n_1 - n_2|}{n+1}$$

Distances in $\mathbb{N}$: Let $d\colon \mathbb{N}\times\mathbb{N} \to [0,1]$ be defined as follows :

$$d(n_1, n_2) = 1 - \frac{1}{|n_1 - n_2| + 1}$$

Distances in $\Sigma = 2^k$: Let $d\colon 2^k\times 2^k \to [0,1]$ be defined as follows :

$$d(v_1, v_2\ ) = \frac{HD(v_1, v_2)}{k}$$

NED_d is metric

Applications in Formal Verification

- FV requires ω-words
-

  Robustness requires NED between omega-words
- [FGW23] suggested ωNED
- ωNED can be generalized to ωNED_d
- Same algorithms are applicable.

Paper results

- We have shown necessary and sufficient conditions for NED_d to be metric.
- We have shown several fine d's .
- We have shown that NED_d can also be used for formal verification.

Thanks for listening ! Any questions ?