Total rank distance and scaled total rank distance: two alternative metrics in computational linguistics

Anca Dinu
University of Bucharest,
Faculty of Foreign Languages/
Edgar Quinet 17,
Bucharest, Romania
anca_d_dinu@yahoo.com

Liviu P. Dinu
University of Bucharest, Faculty of
Mathematics and Computer Science/
Academiei 14, 010014,
Bucharest, Romania
ldinu@funinf.cs.unibuc.ro

Abstract

In this paper we propose two metrics to be used in various fields of computational linguistics area. Our construction is based on the supposition that in most of the natural languages the most important information is carried by the first part of the unit. We introduce total rank distance and scaled total rank distance, we prove that they are metrics and investigate their max and expected values. Finally, a short application is presented: we investigate the similarity of Romance languages by computing the scaled total rank distance between the digram rankings of each language.

1 Introduction

Decision taking processes are common and frequent tasks for most of us in our daily life. The ideal case would be that when the decisions can be taken deterministically, based on some clear, quantifiable and unambiguous parameters and classifiers. However, there are many cases when we decide based on subjective or sensorial criteria (e.g. perceptions), but which prove to function well. The domains in which decisions are taken based on perceptions vary a lot: the qualitative evaluation of services, management, financial predictions, sociology, information/intelligent systems, etc (Zadeh and Kacprzyk, 1999).

When people are asked to approximate the height of some individual, they prefer to use terms like: very tall, rather tall, tall enough, short, etc. We can expect the same linguistic variable to have a different metrical correspondence according to the community to which the individual belongs (i.e. an individual of 170 cm can be considered short by the Australian soldiers and tall by the Eskimos). Similar situations also arise when people are asked to hierarchically order a list of objects.

For example, we find it easy to make the top of the best five novels that we read, since number one is the novel that we like best and so on, rather than to say that we liked in the proportion of 40% the novel on the first position, 20 % the novel on the second place and so on. The same thing is happening when we try to talk about the style of a certain author: it is easier to say that the author x is closer to y than z, then to quantify the distance between their styles. In both cases we operate with a "hidden variable" and a "hidden metric".

Especially when working with perceptions, but not only, we face the situation to operate with strings of objects where the essential information is not given by the numerical value of some parameter of each object, but by the position the object occupies in the strings (according to a natural hierarchical order, in which on the first place we find the most important element, on the second place the next one and on the last position the least important element).

As in the case of perceptions calculus, in most of the natural languages, the most important information is also carried by the first part of the unit (Marcus, 1974). Cf. M. Dinu (1997), it is advisable that the essential elements of a message to be situated in the first part of the utterance, thus having the best chances to be memorized[1] (see Table 1).

Based on the remark that in most of the natural

[1] On the contrary, M. Dinu notices that at the other end, we find the wooden language from the communist period, text that was not meant to inform, but to confuse the receiver with an incantation empty of content, and that used the reversed process: to place the important information at the end of very long phrases that started with irrelevant information

Proceedings of the Workshop on Linguistic Distances, pages 109–116,
Sydney, July 2006.

The length of the phrase	Memorized words (%)		
	all	first half	second half
12	100 %	100 %	100 %
13	90 %	95 %	85 %
17	70 %	90%	50%
24	50 %	70 %	30 %
40	30 %	50 %	10 %

Table 1: The percentage of memorized words from phrases

languages the most important information is carried out by the first part of the unit, in this paper we introduce two metrics: total rank distance and scaled total rank distance.

Some preliminary and motivations are given in Section 2. In Section 3 we introduce total rank distance; we prove that it is a metric (Section 3.1), we investigate its max and expected values (Section 3.2) and its behavior regarding the median ranking problem (Section 3.3). An extension for strings is proposed in Section 4. Scaled total rank distance is introduced in Section 4, where we prove that it is a metric and we investigate its max and expected values. In Section 6 a short application is presented: we investigate the similarity of Romance languages by computing the scaled total rank distance between the digram rankings of each language. Section 7 is reserved to conclusions, while in Section 8 we give a mathematically addendum where we present the proofs of the statements.

2 Rank distance

By analogy to computing with words, natural language and genomics, we can say that if the differences between two strings are at the top (i.e., in essential points), the distance has to have a bigger value then when the differences are at the bottom of the strings.

On the other hand, many of the similarity measures used today (edit distance, Hamming distance etc.) do not take into account the natural tendency of the objects to place the most important information in the first part of the message.

This was the motivation we had in mind when we proposed Rank distance (Dinu, 2003) as an alternative similarity measure in computational linguistics. This distance had already been successfully used in computational linguistics, in such problems as the similarity of Romance languages (Dinu and Dinu, 2005), or in bioinformatics (in DNA sequence comparision problem, Dinu and Sgarro).

2.1 Preliminaries and definitions

To measure the distance between two strings, we use the following strategy: we scan (from left to right) both strings and for each letter from the first string we count the number of elements between its position in first string and the position of its first occurrence in the second string. We sum these scores for all elements and obtain the rank distance. Clearly, the rank distance gives a score zero only to letters which are in the same position in both strings, as Hamming distance does (we recall that Hamming distance is the number of positions where two strings of the same length differ).

On the other hand, the reduced sensitivity of the rank distance w.r.t. deletions and insertions is of paramount importance, since it allows us to make use of *ad hoc extensions to arbitrary strings*, such as its low computational complexity is not affected. This is not the case for the extensions of the Hamming distance, mathematically optimal but computationally heavy, which lead to the *edit-distance*, or *Levenshtein distance*, and which are at the base of the standard alignment principle. So, rank distance sides with Hamming distance rather than Levenshtein distance as far as computational complexity is concerned: the fact that in the Hamming and in the rank case the median string problem is tractable (Dinu and Manea), while in the edit case it is is NP-hard (Higuera and Casacuberta, 2000), is a very significant indicator.

The rank distance is an *ordinal* distance tightly related to the so-called *Spearman's footrule* (Diaconis and Graham, 1977) [2], which has long been used in non-parametric statistics. Unlike other ordinal distances, the Spearman's footrule is linear in n, and so very easy to compute. Its average value is at two-thirds of the way to the maximum value (both are quadratics in n); this is because, in a way, the Spearman footrule becomes rather "undiscriminating" for highly different orderings. Rank distance has the same drawbacks and the same advantages of Spearman's foootrule. As for "classical" ordinal distances for integers, with averages values, maximal values, etc., the reader is

[2]Both Spearman's footrules and binary Hamming distances are a special case of a well-known metric distance called sometimes taxi distance, which is known to be equivalent to the usual Euclidian distance. Computationally, taxi distance is obviously linear.

referred to the basic work (Diaconis and Graham, 1977).

Let us go back to strings. Let us choose a finite alphabet, say $\{N, V, A, O\}$ (Noun, Verb, Adjective, Object) and two strings on that alphabet, which for the moment will be constrained to be a permutation of each other. E.g. take two strings of length 6: $NNVAOO$ and $VOANON$; put indexes for the occurrences of repeated letters in increasing order to obtain $N_1N_2V_1A_1O_1O_2$ and $V_1O_1A_1N_1O_2N_2$. Now, proceed as follows: in the first sequence N_1 is in position 1, while it is in position 4 in the second sequence, and so the difference is 3; compute the difference in positions for all letters and sum them. In this case the differences are 3, 4, 2, 1, 3, 1 and so the distance is 14. Even if the computation of the rank distance as based directly on its definition may appear to be quadratic, in (Dinu and Sgarro) two algorithms which take it back to linear complexity are exhibit.

In computational linguistics the rank distance for strings *without repetitions* had been enough. In a way, *indexing* converts a sequence *with repetitions* into a sequence without repetitions, in which the k occurrence of a letter a are replaced by single occurrences of the k indexed letters a_1, a_2, ..., a_k. Let $u = x_1x_2 \dots x_n$ and $v = y_1y_2 \dots y_m$ be two strings of lengths n and m, respectively. For an element $x_i \in u$ we define its *order* or *rank* by $ord(x_i|u) = n+1-i$: we stress that the rank of x_i is its position in the string, counted from the **right** to the **left**, *after* indexing, so that for example the second O in the string $VOANON$ has rank 2.

Note that some (indexed) occurrences appear in both strings, while some other are *unmatched*, i.e. they appear only in one of the two strings. In definition (1) the last two summations refer to these unmatched occurrences. More precisely, the first summation on $x \in u \cap v$ refers to occurrences x which are common to both strings u and v, the second summation on $x \in u \setminus v$ refers to occurrences x which appear in u but not in v, while the third summation on $x \in v \setminus u$ refers to occurrences x which appear in v but not in u.

Definition 1 *The rank distance between two strings without repetitions u and v is given by:*

$$\Delta(u,v) = \sum_{x \in u \cap v} |ord(x|u) - ord(x|v)| + + \sum_{x \in u \setminus v} ord(x|u) + \sum_{x \in v \setminus u} ord(x|v) \quad (1)$$

Example 1 *1. Let $u = abcde$ and $v = beaf$ be two strings without repetitions. $\Delta(u,v) = |ord(a|u) - ord(a|v)| + |ord(b|u) - ord(b|v)| + |ord(e|u) - ord(e|v)| + ord(c|u) + ord(d|u) + ord(f|v) = 3 + 0 + 2 + 3 + 2 + 1 = 11$.*

2. Let $w_1 = abbab$ and $w_2 = abbbac$ be two strings with repetitions. Their corresponding indexed strings will be: $\overline{w_1} = a_1b_1b_2a_2b_3$ and $\overline{w_2} = a_1b_1b_2b_3a_2c_1$, respectively. So, $\Delta(w_1, w_2) = \Delta(\overline{w_1}, \overline{w_2}) = 8$.

Remark 1 *The ad hoc nature of the rank distance resides in the last two summations in **(1)**, where one compensates for unmatched letters, i.e. indexed letters which appear only in one of the two strings.*

Deletions and insertions are less worrying in the rank case rather than in the Hamming case: if one incorrectly moves a symbol by, say, one position, the Hamming distance loses any track of it, but rank distance does not, and the mistake is quite light. So, generalizations in the spirit of the edit distance are unavoidable in the Hamming case, even if they are computationally very demanding, while in the rank case we may think of *ad hoc* ways-out, which are computationally convenient.

3 Total Rank Distance

We remind that one of the goals of introducing rank distance was to obtain a tool for measuring the distance between two strings which is more sensitive to the differences encountered in the beginning of the strings than in the ending.

Rank distance satisfies in a good measure the upper requirement (for example it penalizes more heavily unmatched letters in the initial part of strings), but some black points are yet remaining. One of them is that rank distance is invariant to the transpositions on a given length.

The following example is eloquent:

Example 2 *1. Let $a = (1,2,3,4,5)$, $b = (2,1,3,4,5)$, $c = (1,2,4,3,5)$ and $d = (1,2,3,5,4)$ be four permutations. Rank distance between a and each of b, c or d is the same, 2.*

2. The same is happening with $a = (1,2,3,4,5,6,7,8)$ and $b = (3,2,1,4,5,6,7,8)$, $c = (1,4,3,2,5,6,7,8)$, or $d = (1,2,3,4,5,8,7,6)$ (here rank distance is equal to 4).

In the following we will repair this inconvenient, by introducing the *Total Rank Distance*, a measure which gives us a more comprehensive information (compared to rank distance) about the two strings which we compare.

Since in many situations occurred in computational linguistics, the similarity for strings *without repetitions* had been enough, in the following we introduce first a metric between rankings[3] and then we generalize it to strings.

3.1 Total rank distance on permutations

Let A and B be two rankings over the same universe U, having the same length, *n*. Without loss of generality, we suppose that $U = \{1, 2, \ldots, m\}$.

For each $1 \leq i \leq n$ we define the function δ by:

$$\delta(i) \stackrel{\text{def}}{=} \Delta(A_i, B_i). \tag{2}$$

where A_i and B_i are the partial rankings of length i obtained from the initial rankings by deleting the elements below position i (i.e. the top i rankings).

Definition 2 *Let A and B be two rankings with the same length over the same universe, U. The Total Rank Distance between A and B is given by:*

$$D(A, B) = \sum_{i=1}^{n} \delta(i) = \sum_{i=1}^{n} \Delta(A_i, B_i).$$

Example 3

1. *Let a, b, c and d be the four permutations from Example 2, item 1. The total rank distance between a and each of b, c, d is: $D(a, b) = 10$, $D(a, c) = 6$, $D(a, d) = 4$.*
2. *The visible differences are also in the item 2 of the upper example if we apply total rank distance: $D(a, b) = 30$, $D(a, c) = 28$, $D(a, d) = 10$.*

[3] A ranking is an ordered list of objects. Every ranking can be considered as being produced by applying an ordering criterion to a given set of objects. More formally, let U be a finite set of objects, called the universe of objects. We assume, without loss of generality, that $U = \{1, 2, \ldots, |U|\}$ (where by $|U|$ we denote the cardinality of U). A ranking over U is an ordered list: $\tau = (x_1 > x_2 > \ldots > x_d)$, where $\{x_1, \ldots, x_d\} \subseteq U$, and $>$ is a strict ordering relation on $\{x_1, \ldots, x_d\}$, (*an ordering criterion*. It is important to point the fact that $x_i \neq x_j$ if $i \neq j$. For a given object $i \in U$ present in τ, $\tau(i)$ represents the position (or rank) of i in τ. If the ranking τ contains all the elements of U, than it is called a *full ranking*. It is obvious that all full rankings represent all total orderings of U (the same as the permutations of U). However, there are situations when some objects cannot be ranked by a given criterion: the ranking τ contains only a subset of elements from the unverse U. Then, τ is called *partial ranking*. We denote the set of elements in the list τ with the same symbol as the list.

The following theorem states that our terminology *total rank distance* is an adequate one:

Theorem 1 *Total rank distance is a metric.*

Proof:

It is easy to see that $D(A, B) = D(B, A)$.

We prove that $D(A, B) = 0$ iff $A = B$. If $D(A, B) = 0$, then $\Delta(A_i, B_i) = 0$ for each $1 \leq i \leq n$ (since Δ is a metric, so a nonnegative number), so $\Delta(A_n, B_n) = \Delta(A, B) = 0$, so $A = B$.

For the triangle inequality we have: $D(A, B) + D(B, C) = \sum_{i=1}^{n} \Delta(A_i, B_i) + \sum_{i=1}^{n} \Delta(B_i, C_i)$

$= \sum_{i=1}^{n} (\Delta(A_i, B_i) + \Delta(B_i, C_i))$

$\geq \sum_{i=1}^{n} \Delta(A_i, C_i) = D(A, C).$ □

3.2 Expected and max values of the total rank distance

Let S_n be the group of all permutations of length n and let A, B be two permutations from S_n. We investigate the max total rank distance between A and B and the average total rank distance between A and B.

Proposition 1 *Under the upper hypothesis, the expected value of the total rank distance between A and B is:*

$$E(D) = \frac{(n^2 - 1)(n + 2)}{6}.$$

Proposition 2 *Under the same hypothesis as in the previous proposition, the max total rank distance between two permutations from S_n is:*

$$\max_{A,B \in S_n} D(A, B) = \frac{n^2(n + 2)}{4}$$

and it is achieved when a permutation is the reverse of the other one.

3.3 On the aggregation problem via total rank distance

Rank aggregation is the problem of combining several ranked lists of objects in a robust way to produce a single ranking of objects.

One of the most natural way to solve the aggregation problem is to determine the median (sometimes called *geometric median*) of ranked lists via a particular measure.

Given a multiset T of ranked lists, a median of T is a list L such that

$$d(L,T) = \min_X d(X,T),$$

where d is a metric and X is a ranked list over the universe of T.

Depending on the choice of measure d, the upper problem may contain many unpleasant surprises. One of them is that computing the median set is NP-complete for some usual measure (including edit-distance or Kendal distance) even for binary universe.

We will show in the following that the median aggregation problem via Total rank distance can be computed in polynomial time.

Theorem 2 *Given a multiset T of full ranked lists over the same universe, the median of T via total rank distance can be computed in polynomial time, namely proportional to the time to find a minimum cost perfect matching in a bipartite graph.*

Proof: Without loss of generality, we suppose that the universe of lists is $U = \{1, 2, \ldots, n\}$. We define a weighted complete bipartite graph $G = (N, P, W)$ as follows. The first set of nodes $N = \{1, 2, \ldots, n\}$ denotes the set of elements to be ranked in a full list. The second set of nodes $P = \{1, 2, \ldots, n\}$ denotes the n available positions. The weight $W(i, j)$ is the contribution, via total rank distance, of node i to be ranked on place j in a certain ranking.

We can give a close formula for computing the weights $W(i, j)$ and this ends the proof, because we reduced the problem to the solving of the minimum cost maximum matching problem on the upper bipartite graph ((Fukuda and Matsui, 1994), (Fukuda and Matsui, 1992), (Dinu and Manea)).

□

4 An extension to strings of total rank distance

We can extend total rank distance to strings.

Similar to the extensions of rank distance to strings, we index each letter in a word with the number of its previous occurrences.

First, we extent the total rank distance to rankings with unequal lengths as it follows:

Definition 3 *Let u and v be two rankings of length $|u|$ and $|v|$, respectively. We can assume that $|u| < |v|$. The total rank distance between u and v is defined by:*

$$D(u,v) = \sum_{i=1}^{|u|} \Delta(v_i, u_i) + \sum_{i=|u|+1}^{|v|} \Delta(v_i, u).$$

Theorem 3 *The total rank distance between two rankings with unequal lengths is a metric.*

To extent the total rank distance to strings, firstly we index both strings and than we apply the upper definition to the newly obtained strings (which are now rankings).

Example 4 *Let $u = aabca$, $v = aab$ and $w = bca$ be three strings. We obtained the following results:*

1. *Rank distance: $\Delta(u,v) = \Delta(a_1a_2b_1c_1a_3, a_1a_2b_1) = 9$ and $\Delta(u,w) = \Delta(a_1a_2b_1c_1a_3, b_1c_2a_1) = 9$;*
2. *Total rank distance: $D(u,v) = D(a_1a_2b_1c_1a_3, a_1a_2b_1) = 13$ and $D(u,w) = D(a_1a_2b_1c_1a_3, b_1c_2a_1) = 33$.*

What happens in item 1 is a consequence of a general property of rank distance which states that $\Delta(uv, u) = \Delta(uv, v)$, for any nonempty strings u and v.

Total rank distance repairs this fact, as we can see from item 2; we observe that the total rank distance is more sensitive than rank distance to the differences from the first part of strings.

5 Scaled Total Rank Distance

We use the same ideas from Total rank distance, but we normalize each partial distance. To do this, we divide each rank distance between two partial rankings of length i by $i(i+1)$, which is the maximal distance between two rankings of length i (it corresponds to the case when the two rankings have no common elements).

Definition 4 *The Scaled Total Rank distance between two rankings A and B of length n is:*

$$S(A,B) = \sum_{i=1}^{n} \frac{\Delta(A_i, B_i)}{i(i+1)}.$$

Theorem 4 *Scaled total rank distance is a metric.*

Proof: The proof is similar to the one from the total rank distance. □

Remark 2 *It is easy to see that $S(A,B) \leq H(A,B)$, where $H(A,B)$ is the Hamming distance.*

Example 5 *Let* $A = (a, b, c, d, e)$, $B = (b, a, c, d, e)$ *and* $C = (a, b, d, e, c)$ *be three permutations. We have the following values for* Δ, *D and S, respectively:*

1. *Rank distance:* $\Delta(A, B) = 2$, $\Delta(A, C) = 4$, *so* $\Delta(A, B) < \Delta(A, C)$.
2. *Total Rank Distance:* $D(A, B) = 2 + 2 + 2 + 2 + 2 = 10$, $D(A, C) = 0 + 0 + 2 + 4 + 4 = 10$, *so* $D(A, B) = D(A, C)$.
3. *Scaled Total Rank Distance:* $S(A, B) = \frac{2}{2} + \frac{2}{6} + \frac{2}{12} + \frac{2}{20} + \frac{2}{30} = \frac{5}{3}$, $S(A, C) = \frac{0}{2} + \frac{0}{6} + \frac{2}{12} + \frac{4}{20} + \frac{4}{30} = \frac{1}{2}$, *so* $S(A, B) > S(A, C)$.

It is not hard to see that $S(A, B) \leq n$, so we can normalize scaled total rank distance by dividing it to n.

We obtained the following two values for max and average values of scaled total rank distance:

Proposition 3

1. *If* $n \to \infty$, *then* $\max_{A,B \in S_n} \frac{1}{n} S(A, B) = \frac{7}{2} - 4 \ln 2$.
2. *The average value of scaled total rank distance is:* $E(S) = \frac{2(n-1)}{3}$. *When* $n \to \infty$, $\frac{E(S)}{n} \to \frac{2}{3}$.

Remark 3 *It is a nice exercise to show that* $\frac{7}{2} - 4 \ln 2 \leq 1$.

Proof: $\frac{7}{2} - 4 \ln 2 \leq 1$ *iff* $1 \leq 4(\ln 4 - 1)$. *But* $4(\ln 4 - 1) > 4(\ln 4 - \ln 3)$. *From Lagrange Theorem, there is* $3 < \xi < 4$ *such that* $\ln 4 - \ln 3 = \frac{1}{\xi}$, *so* $4(\ln 4 - \ln 3) = \frac{4}{\xi} > 1$, *so* $4(\ln 4 - 1) > 4(\ln 4 - \ln 3) > 1$. □

6 Application

We present here a short experiment regarding the similarity of Romance languages. The work corpus is formed by the representative vocabularies of the following six Romance languages: Romanian, Italian, Spanish, Catalan, French and Portuguese languages (Sala, 1988). We extracted the digrams from each vocabularies and then we constructed a ranking of digrams for each language: on the first position we put the most frequent digram of the vocabulary, on the second position the next frequent digram, and so on.

We apply the scaled total rank distance between all pairs of such classifications and we obtain a series of results which are presented in Table 2.

Some remarks are immediate:

- If we analyze the Table 2, we observe that every time Romanian finds itself at the biggest distance from the other languages. This fact proves that the evolution of Romanian in a distanced space from the Latin nucleus has lead to bigger differences between Romanian and the rest of the Romance languages, then the differences between any other two Romance languages.
- The closest two languages are Portuguese and Spanish.
- It is also remarkable that Catalan is equally distanced from Portuguese and Spanish.

Table 2: Scaled total rank distances in Romance languages

	Ro	It	Sp	Ca	Po	Fr
Ro	0	0.36	0.37	0.39	0.41	0.36
It	0.36	0	0.21	0.24	0.26	0.30
Sp	0.37	0.21	0	0.20	0.18	0.27
Ca	0.39	0.24	0.20	0	0.20	0.28
Po	0.41	0.26	0.18	0.20	0	0.30
Fr	0.36	0.30	0.27	0.28	0.30	0

The upper remarks are in concordance with the conclusions of (Dinu and Dinu, 2005) obtained from the analise of the syllabic similarity of the Romance languages, where the rank distance was used to compare the rankings of syllables, based on the frequency of syllables for each language.

During the time, different comparing methods for natural languages were proposed. We mention here the work of Hoppenbrouwers and Hoppenbrouwers (2001). Their approach was the following: using the letter frequency method for each language variety the unigram frequencies of letters are found on the basis of a corpus. The distance between two languages is equal to the sum of the differences between the corresponding letter frequencies. They verify that this approach correctly shows that the distance between Afrikaans and Dutch is smaller than the distance between Afrikaans and the Samoan language.

7 Conclusions

In this paper we provided some low-complexity metrics to be used in various subfields of computational linguistics: total rank distance and scaled total rank distance. These metrics are inspired from the natural tendency of objects to put the main information in the first part of the units. Our analyze was especially concentrated on the mathemat-

ical and computational properties of these metrics: we showed that total rank distance and scaled total rank distance are metrics, computed their expected and max values on the permutations group and showed that total rank distance can be used in classification problem via a polynomial algorithm.

8 Mathematical addendum

This addendum may be skipped by readers who are not interested in mathematical technicalities; below some statements are sketched and other are unproved, but then the proofs are quite straightforward.

Proposition 1:

Proof: It is not hard to see that $D(A, S_n) = D(B, S_n)$ for any two permutation $A, B \in S_n$. So, the expected value can be computed by computing first $D(A, S_n)$ for a convenable permutation and then by dividing the upper sum to $n!$. If we choose $A = e_n$ (i.e. the identical permutation of the group S_n), then the expected value is:

$$E(D) = \frac{1}{n!} \sum_{\sigma \in S_n} D(e_n, \sigma).$$

The upper sum can be easily computed if we take into account the fact that each number $1, 2, \ldots, n$ appears the same number of times (i.e. (n-1)!) on the ranks $1, 2, \ldots n$. So, we obtain that the expected value is equal to:

$$E(D) = \frac{(n^2 - 1)(n + 2)}{6}.$$

□

Proposition 2:

Proof: W.l.g. we can suppose that first permutation is the identical one, i.e. e_n (otherwise we will relabelled it). To compute the max value, the following preliminary results must be proven (we skipped the proofs).

We say that an integer from σ is *low* if its position is $\leq \frac{n}{2}$ and it is *high* if its position is $> \frac{n}{2}$.

Let $\sigma \in \mathcal{S}_n$ be a permutation. We construct the set Θ_σ as following:

$$\Theta_\sigma = \{\tau \in \mathcal{S}_n \mid \forall x \in \{1 \ldots n\},\ x \text{ is low in } \tau \text{ iff } x \text{ is high in } \sigma \text{ and viceversa}\}$$

Result 1 *For each $\sigma \in \mathcal{S}_n$ and every two permutation $\tau,\ \pi$ in Θ_σ we have: $D(\sigma, \tau) = D(\sigma, \pi)$.*

Result 2 *For each $\sigma \in \mathcal{S}_n$ and every two permutation τ, π such that $\pi \in \Theta_\sigma$ and $\tau \notin \Theta_\sigma$, we have: $D(\sigma, \tau) < D(\sigma, \pi)$.*

To prove Result 2 we use the following Lemma:

Lemma 1 (Dinu, 2003) *If $a > b$, then the function $f(x) = |x - b| - |x - a|$ is an increasing one.*

Result 3 *Let $\sigma \in \mathcal{S}_n$ be a permutation. The maximum total rank distance is reached by the permutation τ where $ord(x|\tau) = n + 1 - ord(x|\sigma)$, $\forall x \in V(\mathcal{P}_n)$. Under this conditions the maximum total rank distance is:*

$$\max_{A,B \in S_n} D(A, B) = \frac{n^2(n + 2)}{4} \tag{3}$$

In other words, we obtained a more general result:

Theorem 5 *For a given permutation σ, the maximum rank distance is achieved by all permutations from Θ_σ and it is equal to (3).*

□

Proposition 3:

Proof:

1. Similar to Proposition 2, given a permutation $\sigma \in S_n$, the max value is reached by its invert. So, to give a close formula for the max value it is enough to compute $S(e_n, e_n^{-1})$. To make easier our life, we can suppose that $n = 2k$.

   $S(e_n, e_n^{-1}) = k + \sum_{i=1}^{k} \frac{2i^2 + (k-i)(k-i+1)}{(k+i)(k+i+1)} = \ldots = 4k - \frac{2k^2}{2k+1} - 2(4k + 1)(\sum_{i=1}^{k} \frac{1}{k+i} - \frac{k}{2k+1})$;

   When $k \to \infty$, $\sum_{i=1}^{k} \frac{1}{k+i} \to \ln 2$, so $\frac{S(e_n, e_n^{-1})}{n} = \frac{7}{2} - 4 \ln 2$ □

2. To compute the expected value we use the same motivation as in expected total rank distance. The rest is obvious.

Acknowledgements 1 *We want to thank to reviewers for their comments and suggestions. Research supported by CNR-NATO and MEdC-ANCS.*

References

P. Diaconis, R.L. Graham, 1977. *Spearman footrule as a Measure of Disarray*, Journal of Royal Statistical Society. Series B (Methodological), Vol. 39, No. 2, 262-268.

L. P. Dinu, 2003. *On the classification and aggregation of hierarchies with different constitutive elements*, Fundamenta Informaticae, 55(1), 39-50.

A. Dinu, L.P. Dinu, 2005. *On the Syllabic Similarities of Romance Languages*. In Proc. CICLing 2005, Lecture Notes in Computer Science, Volume 3406, pp. 785-789.

L.P. Dinu, F. Manea. *An efficient approach for the rank aggregation problem.* Theoretical Computer Science (to appear).

L.P. Dinu, A. Sgarro. *A low-complexity distance for DNA strings*, Fundamenta Informaticae (to appear).

M. Dinu, 1997. *Comunicarea* (in Romanian). Ed. Ştiinţifică, Bucureşti.

K. Fukuda, T. Matsui, 1992. *Finding all minimum cost perfect matchings in bipartite graphs*, Networks, 22, 461-468.

K. Fukuda, T. Matsui, 1994. *Finding all the perfect matchings in bipartite graphs*, Appl. Math. Lett., 7(1), 15-18.

C. de la Higuera, F. Casacuberta, 2000. *Topology of strings: Median string is NP- complete,* Theoretical Computer Science, 230:39-48.

C. Hoppenbrouwers, G. Hoppenbrouwers, 2001. *De indeling van de Nederlandse streektalen. Dialecten van 156 steden en dorpen geklasseerdvolgens de FFM*. Koninklijke Van Gorcum, Assen.

S. Marcus, 1974. Linguistic structures and generative devices in molecular genetics. Cahiers Ling. Theor. Appl., 11, 77-104.

M. Sala, (coord.) 1982. Vocabularul reprezentativ al limbilor romanice, Bucureşti.

L.A. Zadeh, J. Kacprzyk, 1999. *Computing with words in information/intelligent systems 1: Foundations, 2: Application*. Physica-Verlag, Heidelberg and New York.