Usability of String Distance Metrics for Name Matching Tasks in Polish

Jakub Piskorski*, **Marcin Sydow**†

*Joint Research Center of the European Commission, Web and Language Technology group of IPSC
Via Fermi 1, 21020 Ispra (VA), Italy, email:`Jakub.Piskorski@jrc.it`
†Polish-Japanese Institute of Information Technology (PJIIT), Department of Intelligent Systems
Koszykowa 86, 02-008 Warsaw, Poland, email:`msyd@pjwstk.edu.pl`

Abstract

This paper presents results of the numerous experiments on usability of well-established string distance metrics and some new variants thereof for various name matching tasks in Polish.

1. Introduction

A frequently appearing problem in the context of text processing technologies involves making a decision whether two distinct strings refer to the same real-world object. Name matching has been studied thoroughly in the past and approaches ranging from linguistically oriented ones (Morton, 1997) to very lightweight approximate-string matching techniques have been proposed.

In this paper, we focus on exploring the usability of the well-established string distance metrics and some new variants thereof for matching name occurrences in highly inflected languages. [1] In particular, we present results of numerous experiments carried out on a Polish proper-name dataset. Our work was inspired by the comprehensive studies on using string distance metrics for name matching tasks presented recently in (Cohen et al., 2003b; Cohen et al., 2003a; Christen, 2006). The main motivation of carrying out this research is the fact that processing highly inflective languages adds another complication to name matching. The intuitive way of combating the inflection problem would be to lemmatize names, and then to apply string-distance techniques which turned out to work fine for inflection-poor languages like English. Unfortunately, the lemmatization of proper names in Polish is knowledge intensive and accuracy figures of more than 80% have not been reported. For instance, lemmatization of full person names might depend on several factors, e.g., (a) the gender of the first name, (b) the part-of-speech information and gender of the word which constitutes the surname, (c) origin/pronunciation of the name, which clearly leaves a lot of space for ambiguities (Piskorski, 2005).

This paper is organized as follows. Section 2. introduces string distance metrics, which were used in our study. Next, in section 3. the results of the experiments are described. Finally, a summary is given in section 4.

2. String Distance Metrics

In our experiments on using character-level string metrics [2] for name matching we used mainly the metrics applied by the database community for record linkage. The point of departure constitutes the well-known *Levenshtein* edit distance metric given by the minimum number of character-level operations (insertion, deletion, or substitution) needed to transform one string into the other (Levenshtein, 1965). There are several extensions to this basic metric. The *Needleman-Wunsch* (Needleman and Wunsch, 1970) metric modifies the original one in that it allows for variable cost adjustment to the cost of a gap, i.e., insert/deletion operation and variable cost of substitutions. Another variant considered here is the *Smith-Waterman* metric (Smith and Waterman, 1981), which additionally uses an alphabet mapping to costs. We tested two settings for this metric namely, one which normalizes the *Smith-Waterman* score with the length of the shorter string and one which uses for the same purpose the *Dice coefficient*, i.e., the average length of strings compared (*Smith-Waterman-D*). A further extension of the *Smith-Waterman* metric introduces two extra edit operations, *open gap* and *end gap*. The cost of extending the gap is usually smaller than the cost of opening a gap, and this results in small cost penalties for gap mismatches than the equivalent cost under the standard edit distance metrics. We will refer to the aforesaid metric as *Smith-Waterman-AG*. Finally, we created a variant thereof, which uses a character substitution cost function adapted to Polish name declension [3] (*Smith-Waterman-AG-PL*). In general, the computation of most edit-distance metrics requires $O(|s| \cdot |t|)$. We have also considered the recently introduced *bag distance* metric (Bartolini et al., 2002) which is a good approximation of the previously mentioned edit distance metrics, and is calculated (in linear time) as $bag_{dist}(s,t) = \max(|M(s) \backslash M(t)|, |M(t) \backslash M(s)|)$, where $M(x)$ denotes the multiset of the characters in x.

Good results for name-matching tasks (Cohen et al., 2003b) have been reported using variants of the *Jaro* metric (Winkler, 1999), which is not based on the edit-distance model. It considers the number and the order of the common characters between two strings. Given two strings $s = a_1 \ldots a_K$ and $t = b_1 \ldots b_L$, we say that a_i in s

[1] This work has been supported by the MEI grant no. 3 T11C 007 27 and by the PJWSTI grant ST/SI/06/2006.

[2] Distance (similarity) metrics map a pair of strings s and t to a real number r, where a smaller (larger) value of r indicates greater similarity.

[3] There are three different scores for substitution operation: (a) exact match between characters (score +5), (b) approximate match between similar characters (+3), where for Polish two characters are considered similar if they both appear in one of the sets: {a,e,i,o,u,y,a,ą,ę,ó},{c,ć},{s,ś},{n,ń},{l,ł},{k,c},{t,c}, (c) mismatch of characters (-5)

is *common* with t if there is a $b_j = a_i$ in t such that $i - R \leq j \leq i + R$, where $R = \lfloor \max(|s|, |t|)/2 \rfloor - 1$. Further, let $s' = a'_1 \dots a'_K$ be the characters in s which are common with t (with preserved order of appearance in s) and let $t' = b'_1 \dots b'_L$ be defined analogously. A *transposition* for s' and t' is defined as the position i such that $a'_i \neq b'_l$. Let us denote the number of transposition for s' and t' as $T_{s',t'}$. The *Jaro* similarity is then calculated as:

$$J(s,t) = \frac{1}{3} \cdot (\frac{|s'|}{|s|} + \frac{|t'|}{|t|} + \frac{|s'| - \lfloor T_{s',t'}/2 \rfloor}{|s'|})$$

A *Winkler* variant thereof boosts the *Jaro* similarity for strings with agreeing initial characters. It is calculated as:

$$JW(s,t) = J(s,t) + \delta \cdot \mathit{boost}_p(s,t) \cdot (1 - J(s,t))$$

,where δ denotes the common prefix adjustment factor (default: 0.1) and $\mathit{boost}_p(s,t) = \min(|lcp(s,t)|, p)$. Here $lcp(s,t)$ denotes the longest common prefix between s and t. For multi-token strings we extended boost_p to boost^*_p. Let $s = s_1 \dots s_K$ and $t = t_1 \dots t_L$, where s_i (t_i) represent i-th token of s and t respectively, and let without loss of generality $L \leq K$. boost^*_p is calculated as:

$$\mathit{boost}^*_p(s,t) = \frac{1}{L} \cdot \sum_{i=1}^{L-1} \mathit{boost}_p(s_i, t_i) + \frac{\mathit{boost}_p(s_L, t_L..t_K)}{L}$$

We denote the metric which uses boost^*_p as JWM. The time complexity of '*Jaro*' metrics is $O(|s| \cdot |t|)$.

The *q-gram* metric (Ukkonen, 1992) is based on the intuition that two strings are similar if they share a large number of character-level q-grams. Let $G_q(s)$ denote the multiset of all q-grams of a string s obtained by sliding a window of length q over the characters of s. [4] The q-gram metric is calculated as:

$$q\text{−}grams(s,t) = \frac{|G_q(s) \cap G_q(t)|}{\max(|G_q(s)|, |G_q(t)|)}$$

An extension to this metric is to add positional information, and to match only common q-grams that occur within a maximum distance to each other (*positional q-grams*) (Gravano et al., 2001). Further, (Keskustalo et al., 2003) introduced *skip-gram* metric. It is based on the idea that in addition to forming bigrams of adjacent characters, bigrams that skip characters are considered. *Gram classes* are defined that specify what kind of skip-grams are created, e.g. $\{0, 1\}$ class means that normal bigrams are formed, and bigrams that skip one character. The q-gram type metrics can be computed in $O(\max\{|s|, |t|\})$.

Considering the declension paradigm of Polish we also considered a basic and time efficient metric based on the longest common prefix information, which would intuitively perform well in the case of single-token names. It is calculated as: $CP_\delta(s,t) = (|lcp(s,t)| + \delta)^2 / |s| \cdot |t|$. The symbol δ in $CP_\delta(s,t)$ is an additional parameter for favoring certain suffix pairs in s (t). We have experimented with two variants, CP_{δ_1} and CP_{δ_2}. In CP_{δ_1} the value of δ is set to 0. In CP_{δ_2}, as a result of empirical study of the data and the declension paradigm δ has been set to 1 if s ends in: *o*,*y*,*ą*,*ę*, and t ends in an *a*. Otherwise δ is set to 0. For coping with multi-token strings, we tested a similar metric called *longest common substrings* distance (*LCS*), which recursively finds and removes the longest common substring in the two strings compared. Let $lcs(s,t)$ denote the 'first' longest common substring for s and t and let s_{-p} denote a string obtained via removing from s the first occurrence of p in s. The *LCS* metric is calculated as:

$$LCS(s,t) = \begin{cases} 0 \text{ if } |lcs(s,t)| \leq \phi \\ |lcs(s,t)| + LCS(s_{-lcs(s,t)}, t_{-lcs(s,t)}) \end{cases}$$

The value of ϕ is usually set to 2 or 3. The time complexity of *LCS* is $O(|s| \cdot |t|)$. We extended LCS by additional weighting of the $|lcs(s,t)|$. The main idea is to penalize longest common substrings which do not match the beginning of a token in at least one of the compared strings. Let α be the maximum number of non-whitespace characters, which precede the first occurrence of $lcs(s,t)$ in s or t. Then, $lcs(s,t)$ is assigned the weight:

$$w_{lcs(s,t)} = \frac{|lcs(s,t)| + \alpha - \max(\alpha, p)}{|lcs(s,t)| + \alpha}$$

where p has been experimentally set to 4. We denote the 'weighted' variant of LCS as $WLCS$.

Finally, for multi-token strings we tested the recursive schema, known also as *Monge-Elkan* distance (Monge and Elkan, 1996). Let us assume that the strings s and t are broken into substrings (tokens), i.e., $s = s_1 \dots s_K$ and $t = t_1 \dots t_L$. The intuition behind *Monge-Elkan* measure is the assumption that s_i in s corresponds to a t_j with which it has highest similarity. The similarity between s and t equals the mean of these maximum scores. Formally, the *Monge-Elkan* metric is defined as follows, where sim denotes some secondary similarity function.

$$Monge\text{−}Elkan(s,t) = \frac{1}{K} \cdot \sum_{i=1}^{K} \max_{j=1 \dots L} sim(s_i, t_j)$$

Inspired by the multi-token variants of the *JW* metric presented in (Christen, 2006) we introduced two additional metrics, which are similar in spirit to the *Monge-Elkan* metric. The first one, *Sorted-Tokens* is computed in two steps: (a) firstly the tokens constituting the full strings are sorted alphabetically, and (b) an arbitrary metric is applied to compute the similarity of the 'sorted' strings. The second metric, *Permuted-Tokens* compares all possible permutations of tokens constituting the full strings and returns the maximum calculated similarity value.

3. Experiments

This section describes our experiments on using different string metrics for the entity matching task. We define the problem as follows. Let A, B and C be three sets of strings over some alphabet Σ, with $B \subseteq C$. Further, let $f : A \rightarrow B$ be a function representing a mapping of inflected forms into their corresponding base forms. Given,

[4] Since q-grams at the beginning and the end of the string can have fewer than q characters, the strings are extended by adding $q - 1$ unique initial and trailing characters to a string.

A and C (the latter representing the search space), the task is to construct an approximation of f, namely $\widehat{f} : A \to C$. If $\widehat{f}(a) = f(a)$ for $a \in A$, we say that $\widehat{f}$ returns a correct answer for a, otherwise, $\widehat{f}$ is said to return an incorrect answer. Secondly, we defined an additional task consisting of constructing another approximation of f, namely function $f^* : A \to 2^C$, where f^* is said to return a correct answer for $a \in A$ if $f(a) \in f^*(a)$.

3.1. Test Data

For the experiments on name matching we have used three resources: (a) a lexicon of the most frequent Polish first names (PL-FIRST-NAMES) consisting of pairs $(in, base)$, where in is an inflected form and $base$ stands for the corresponding base form, (b) an analogous lexicon of inflected forms of country names in Polish (PL-COUNTRIES) [5], and (c) a similar lexicon of inflected full person names (first name + surname) (PL-FULL-NAMES). The latter resource was created semi-automatically as follows. We have automatically extracted a list of 22485 full person-name candidates from a corpus of 15,724 on-line news articles from *Rzeczpospolita*, one of the leading Polish newspapers, via using PL-F-NAMES lexicon and an additional list of 58038 uninflected foreign first names. Subsequently, we have selected an excerpt of circa 1900 entries (inflected forms) from this list. $1/3$ of this excerpt are the most frequent names appearing in the corpus, $1/3$ are the most rare names, and finally $1/3$ of the entries were chosen randomly.

In the basic experiments we simply used the base forms as the search space, however we produced some variants of PL-FIRST-NAMES and PL-FULL-NAMES resources via enriching the search space by adding base forms of foreign first names and a complete list of full names extracted from the *Rzeczpospolita* corpus respectively. Table 1 gives an overview of our test datasets. [6]

Dataset	**#inflected**	**#base**	**search space**
PL-F-NAMES	5941	1457	1457
PL-F-NAMES-2	5941	1457	25490
PL-COUNTRIES	1765	220	220
PL-FULL-NAMES	1900	1219	1219
PL-FULL-NAMES-2	1900	1219	2351
PL-FULL-NAMES-3	1900	1219	20000

Table 1: Dataset used for the experiments

3.2. Evaluation Metrics

Since for a given string more than one answer can be returned, we measured the accuracy in three ways. Firstly, we calculated the accuracy with the assumption that a multi-result answer is not correct and we defined (*all-answer accuracy*) (AA) measure which penalizes the accuracy for multi-result answers. Secondly, we measured the accuracy of single-result answers (*single-result accuracy* (SR)) disregarding the multiple-result answers. Finally, we used a somewhat weaker measure which treats a multi-result answer as correct if one of the results in the answer is correct (*relaxed-all-answer accuracy* (RAA)).

Let s denote the number of strings, for which a single result was returned. Analogously, m is the number of strings for which more than one result was returned. Further, let s_c and m_c denote the number of correct single-result answers returned and the number of multi-result answers containing at least one correct result respectively. The accuracy metrics are computed as: $AA = s_c/(s+m)$, $SR = s_c/s$, and $RAA = (s_c + m_c)/(s + m)$.

3.3. Matching First Names

First experiment was run on the PL-F-NAME dataset with the non-recursive string distance metrics. The results of the accuracy evaluation [7] are given in table 2. The first three columns give the accuracy figures, whereas the columns labeled with **AV** and **MAX** give the average and maximum number of results returned in an answer.

Metric	**AA**	**SR**	**RAA**	**AV**	**MAX**
Bag Distance	0,476	0,841	0,876	3,02	19
Levenshtein	0.708	0.971	0.976	2.08	8
Needleman-Wunsch	0.728	0.833	0.826	3	20
Smith-Waterman	0.625	0.763	0.786	3.47	74
Smith-Waterman-AG	0.603	0.728	0.749	3.36	74
Jaro	0,775	0,820	0,826	2,06	4
Jaro-Winkler	0,820	0,831	0,831	2,03	3
2-grams	0,701	0,972	0,978	2,10	17
3-grams	0,712	0,974	0,981	2,10	14
4-grams	0,714	0,974	0,981	2,09	14
pos 2-grams	0,717	0,975	0,982	2,09	15
pos 3-grams	0,721	0,976	0,982	2,09	14
pos 4-grams	0,712	0,976	0,982	2,093	14
$\{0, 2\}$ skip grams	0,873	0,935	0,936	2,14	6
LCS	0,696	0,971	0,977	12,69	550
WLCS	0,731	**0,983**	**0,986**	2,97	549
CP_{δ_1}	0.829	0.843	0.844	2.11	3
CP_{δ_2}	**0.947**	0.956	0.955	2.18	3

Table 2: Results for PL-F-NAMES

Interestingly, the simple linguistic-aware common prefix-based measure turned out to work best in the **AA** category, whereas *WLCS* metric is the most accurate one in the **SR** and **RAA** categories. Thus, a combination of the two seems to be a reasonable solution to further improve the performance, i.e., if *WLCS* returns a single answer, return it, otherwise return the answer of CP_{δ_2}. Further, the time-efficient skip grams metric performed surprisingly good in the **AA** category. Noteworthy, circa 10% of the inflected first name forms in Polish are ambiguous w.r.t. gender (e.g., *Stanisława* - genitive form of the male name *Stanisław* vs. nominative form of the female name *Stanisława*), which illustrates additional complexity.

Clearly, the **AA** accuracy in the experiment run on the PL-F-NAME-2 (with the large search space) was significantly worse. However, the **SR** accuracy for some of the metrics is still acceptable. The top ranking metrics with respect to **SR** and **AA** accuracy are given in table 3.

[5] This resource contains a larger number of multi-words since full names of some countries consist of more than one token.

[6] Pairs, where inflected form is identical with the base form have been excluded from the experiments since in such a case finding an answer is straightforward.

[7] In case of metrics which have parameters, e.g., positional q-grams, we just give the best result achieved.

Metric	SR	AA	Metric	SR	AA
WLCS	**0.893**	0.469	2-grams	0.810	0.398
CP_{δ_2}	0.879	**0.855**	LCS	0.768	0.340
pos 2-grams	0.876	0.426	CP_{δ_1}	0.668	0.600
skip grams	0.822	0.567	JW	0.620	0.560

Table 3: Top results for PL-F-NAMES-2

3.4. Matching Country Names

The next test was carried out on the PL-COUNTRIES, which contains many multi-token strings, where the number of tokens the strings contain of varies. We considered also *Monge-Elkan* metric, *Sorted-Tokens* and *Permuted-Tokens* to better cope with multi-token strings. The aforementioned metrics were tested with different 'internal' metrics. The results are given in table 4, 5, 6 and table 7.

Metric	AA	SR	RAA	AV	MAX
Bag distance	0,369	0,461	0,402	2,6	7
Levenshtein	0.564	0.590	0.586	2.94	12
Needleman-Wunsch	0.720	0.779	0.763	2.95	11
Smith-Waterman	**0.904**	**0.936**	**0.928**	3.34	10
Smith-Waterman-D	0.849	0.858	0.858	2	2
Smith-Waterman-AG	0.799	0.805	0.802	2.45	4
Smith-Waterman-AG-PL	0.793	0.797	0.797	2.22	3
Jaro	0.432	0.437	0.436	2	2
JW	0,452	0,457	0,452	2,06	3
JWM	0,453	0,458	0,453	2,06	3
2-grams	0,665	0,693	0,689	2,72	13
pos 2-grams	0,425	0,470	0,440	4,08	11
skip-grams	0,662	0,681	0,672	2,13	3
LCS	0.749	0.781	0.783	54.61	189
WLCS	0.530	0.545	0.550	80.16	189
CP_{δ_1}	0.416	0.421	0.420	2.35	3

Table 4: Results for PL-COUNTRIES with 'basic' metrics

Internal Metric	AA	SR	RAA	AV	MAX
Bag Distance	0,461	0,602	0,526	3,05	8
Levenshtein	0.573	0.639	0.593	2.79	4
Needleman-Wunsch	0.532	0.663	0.577	3.08	11
Smith-Waterman	0.205	0.494	0.291	4.94	10
Smith-Waterman-D	0.620	0.672	0.627	2.94	4
Smith-Waterman-AG	0.607	0.633	0.615	3.02	4
Smith-Waterman-AG-PL	0.584	0.605	0.591	3	4
Jaro	0,552	0,624	0,563	3,02	5
JW	0.557	0.623	0.563	3.07	5
4-grams	0,625	0,813	0,665	3,11	8
pos 4-grams	0,629	0,838	0,668	3,12	8
$\{0, 1\}$-skip-grams	0.619	0.664	0.637	2.94	4
$\{0, 1, 2\}$-skip-grams	0.610	0.691	0.630	2.97	4
LCS	0,620	0,813	0,672	4,59	189
WLCS	0,636	0,837	0,688	4,55	189
CP_{δ_1}	**0.694**	**0.868**	**0.716**	3.08	4
CP_{δ_2}	0.631	0.845	0.669	3.13	4

Table 5: Results for PL-COUNTRIES with *Monge-Elkan*

Surprisingly, the best results were achieved by the *Smith-Waterman* metrics. On the contrary, *Monge-Elkan* performed rather badly (probably due to the varying number of tokens the names cosist of). Using CP_{δ_1} as internal metric yielded the best results. The results for *Sorted-Tokens* and *Permuted-Tokens* were significantly better, with *Smith-Waterman* being the the best internal metric.

Internal Metric	AA	SR	RAA	AV	MAX
Bag Distance	0,370	0,461	0,402	2,6	7
Levenshtein	0.614	0.656	0.640	2.56	11
Needleman-Wunsh	0.483	0.527	0.518	3.25	13
Smith-Waterman	**0.898**	**0.931**	**0.919**	2.84	10
Smith-Waterman-D	0.835	0.891	0.838	2.01	3
Smith-Waterman-AG	0.801	0.826	0.802	2.06	4
Smith-Waterman-AG-PL	0.784	0.800	0.786	2.06	3
Jaro	0,757	0,767	0,768	2,19	3
JW	0,769	0,774	0,772	2,44	3
JWM	0,770	0,774	0,773	2,44	3
4-grams	0,768	0,821	0,789	2,51	12
pos 4-grams	0,742	0,804	0,765	9,48	19
skip-grams	0,709	0,729	0,722	2,02	3
LCS	0,738	0,829	0,768	5,32	189
WLCS	0,741	0,817	0,750	5,86	189

Table 6: Results for PL-COUNTRIES with *Sorted-Tokens*

Internal Metric	AA	SR	RAA	AV	MAX
Bag Distance	0,370	0,461	0,402	2,6	7
Levenshtein	0.543	0.603	0.566	2.3	14
Needleman-Wunsh	0.618	0.663	0.650	2.71	11
Smith-Waterman	**0.895**	**0.921**	**0.916**	2.93	10
Smith-Waterman-D	0.798	0.803	0.801	2	2
Smith-Waterman-AG	0.760	0.766	0.763	2.5	5
Smith-Waterman-AG-PL	0.749	0.754	0.752	2.2	3
Jaro	0,786	0,800	0,793	2,21	4
JW	0,790	0,803	0,793	2	2
JWM	0,866	0,872	0,866	2	2
4-grams	0,767	0,793	0,782	2,92	8
pos 4-grams	0,769	0,796	0,785	2,90	9
skip-grams	0,693	0,718	0,707	2,09	4
LCS	0,710	0,732	0,732	14,93	189
WLCS	0,781	0,801	0,801	16,64	189

Table 7: Results PL-COUNTRIES with *Permuted-Tokens*

3.5. Matching Full Person Names

Finally, we have made experiments for full person names, each represented as two tokens. It is important to note that the order of the first name and the surname in some of the entities in our test datasets is swapped, which poses a complicacy since some surnames may also function as a first name. Nevertheless, the results of the experiment on PL-FULL-NAMES given in table 8 are nearly optimal. JWM, $WLCS$, LCS, *skip grams* and *Smith-Waterman* were among the 'best' metrics. The recursive metrics scored in general only slightly better than the basic metrics. The best results oscillating around 0.97, 0.99, and 0.99 for the three accuracy metrics were obtained using *LCS*, *WLCS*, *JWM*, CP_δ and some *Smith-Waterman* variants as internal metrics.

Internal Metric	AA	SR	RAA	AV	MAX
Bag Distance	0,891	0,966	0,966	3,13	11
Levenshtein	0.951	0.978	0.970	4.59	18
Smith-Waterman	0,965	0,980	0,975	3,5	5
Smith-Waterman-D	0.972	0.985	0.980	3.62	5
Smith-Waterman-AG	0.970	0.982	0.975	3.75	5
Smith-Waterman-AG-PL	0.970	0.982	0.978	3.75	5
Needleman-Wunsch	0.896	0.956	0.935	2.88	11
Jaro	0,957	0,970	0,964	3,54	5
JW	0,952	0,964	0,958	3,74	5
JWM	0,962	0,974	0,968	3,74	5
2-grams	0,957	0,988	0,987	3,915	10
pos 3-grams	0,941	0,974	0,966	4,32	23
skip-grams	0,973	0,991	0,990	5,14	10
LCS	0,971	0,992	0,990	5,7	13
WLCS	**0,975**	**0,993**	**0,992**	6,29	12

Table 8: Results for PL-FULL-NAMES

We have further compared the performance of the aforementioned 'recursive' metrics on PL-FULL-NAMES-2, which has a larger search space. The most significant results for the **AA** accuracy are depicted in table 9. The JWM and *Smith-Waterman-D* metric seem to be the best choice as an internal metric, whereas $WLCS$, CP_{δ_2} and *Jaro* perform slightly worse.

Internal M.	**Monge-Elkan**	**Sorted-Tok.**	**Permuted-Tok.**
Bag Distance	0,868	0,745	0,745
Jaro	0,974	0,961	0,968
JWM	**0,976**	**0,976**	0,975
SmithWaterman	0.902	0.972	0.967
Smith-Waterman-D	0.974	**0.976**	**0.976**
Smith-Waterman-AG	0.958	0.966	0.955
Smith-Waterman-AG-PL	0.965	0.971	0.961
Needleman-Wunsch	0.808	0.903	0.857
3-grams	0,848	0,930	0,911
pos 3-grams	0,855	0,928	0,913
skip-grams	0,951	0,967	0,961
LCS	0,941	0,960	0,951
WLCS	0,962	0,967	0,967
CP_{δ_1}	0.969	n.a.	n.a.
CP_{δ_2}	0.974	n.a.	n.a.

Table 9: Results of **AA** accuracy for PL-FULL-NAMES-2

In our last experiment we selected the 'best' metrics so far and tested them against PL-FULL-NAMES-3 (largest search space). The top results for non-recursive metrics are given in Table 10. *Smith-Waterman-D* and JWM turned out to achieve the best scores in the **AA** accuracy, whereas $WLCS$ is far the best metric w.r.t. **SR** accuracy. The top scores achieved for the recursive metrics on PL-FULL-NAMES-3 were somewhat better. In particular, *Monge-Elkan* performed best with CP_{δ_2} as internal metric (0.937 **AA** and 0.947 **SR**) and slightly worse results were obtained with *JWM* and *WLCS*. *Sorted-Tokens* scored best in **AA** and **SR** accuracy with *Smith-Waterman-D* (0.904) and *WLCS* (0.949), resp. For *Permuted-Tokens* using *JWM* and *WLCS* yielded the best results, namely 0.912 (**AA**) and 0.948 (**SR**), resp. Interestingly, the *Smith-Waterman* metrics used as internal metric resulted in lower values of **SR** accuracy than *WLCS* and *JWM*. Comparing all the results for PL-FULL-NAMES-3 reveals that a further improvement could be achieved via combining of *WLCS* and *Monge-Elkan* with CP_{δ_2}, i.e., if *WLCS* returns a single answer, return it, otherwise return the answer of *Monge-Elkan* with CP_{δ_2}. The top results for recursive metrics in **AA** and **SR** accuracy are summarized in table 11.

Metrics	**AA**	**SR**	**RAA**	**AV**	**MAX**
Levenshtein	0.791	0.896	0.897	2.20	9
Smith-Waterman	0.869	0.892	0.889	2.35	6
Smith-Waterman-D	**0.899**	0.911	0.910	2.08	3
Smith-Waterman-AG	0.840	0.850	0.850	2.04	3
Smith-Waterman-AG-PL	0.842	0.857	0.854	2.09	3
JW	0.791	0.807	0.802	2.11	3
JWM	0.892	0.900	0.901	2.11	3
skip-grams	0.852	0.906	0.912	2.04	4
LCS	0.827	0.925	0.930	2.48	46
WLCS	0.876	**0.955**	**0.958**	2.47	44

Table 10: Results for PL-FULL-NAMES-3

Metric	**AA**	**Metric**	**SR**
ME & CP_{δ_2}	**0.937**	ST & WLCS	**0.949**
ME & JWM	0.923	PT & WLCS	0.948
ME & CP_{δ_1}	0.921	ME & CP_{δ_2}	0.947
PT & JWM	0.914	ME & WLCS	0.939
ST & Smith-Waterman-D	0.911	ME & JWM	0.936
ME & Smith-Waterman-D	0.908	ME & CP_{δ_1}	0.935
ST & JWM	0.904	PT & JWM	0.927
PT & Smith-Waterman-D	0.899	ST & Smith-Waterman-D	0.924

Table 11: Results for PL-FULL-NAMES-3 with *Monge-Elkan*, *Sorted-Tokens* and *Permuted-Tokens*

4. Summary

In this paper we investigated the usability of string distance metrics for matching Polish names of different type. For first names, simple common prefix (CP_{δ_2}) metric obtains the best results for all-answer accuracy, whereas the $WLCS$ measure provides the best score for the single-result accuracy. A further improvement could be achieved via combining the aforementioned metrics. As for country names, consisting of varying numbers of tokens, the *Smith-Waterman* metrics perform best among the basic metrics and also as an internal metric for *Sorted-Tokens* and *Permuted-Tokens*. In case of full person names, the basic metrics which yielded the best results include $WLCS$, JWM, *Smith-Waterman-D* and CP_{δ_2}. However, the best overall all-answer accuracy was achieved with the *Monge-Elkan* with CP_{δ_2}, whereas for single-result accuracy $WLCS$ performed best. Again, combining the two latter metrics would possibly improve the accuracy.

The results presented here constitute a kind of handy guideline for developing a fully-fledged solution to reference matching for Polish. To the authors knowledge the presented work is the first comprehensive comparison of various string distance metrics applied to name matching tasks in Polish. In proximate step, we will test statistical significance of the results presented in this paper.

5. References

Bartolini, I., P. Ciacca, and M. Patella, 2002. String mathcing with metric trees using an approximate distance. In *Proceedings of SPIRE, LNCS 2476, Lissbon, Portugal*.

Christen, P., 2006. A Comparison of Personal Name Matching: Techniques and Practical Issues. Technical report, TR-CS-06-02, Computer Science Laboratory, The Australian National University, Canberra, Australia.

Cohen, E., P. Ravikumar, and S. Fienberg, 2003a. A Comparison of String Metrics for Matching Names and Records. KDD Workshop on Data Cleaning and Object Consolidation.

Cohen, W., P. Ravikumar, and S. Fienberg, 2003b. A comparison of string metrics for matching names and records. In *Proceedings of the KDD2003*.

Gravano, L., P. Ipeirotis, H. Jagadish, S. Koudas, N. Muthukrishnan, L. Pietarinen, and D. Srivastava, 2001. Using q-grams in a DBMS for Approximate String Processing. *IEEE Data Engineering Bulletin*, 24(4):28–34.

Keskustalo, H., A. Pirkola, K. Visala, E. Leppanen, and K. Jarvelin, 2003. Non-adjacent digrams improve matching of cross-lingual spelling variants. In *Proceedings of SPIRE, LNCS 22857, Manaus, Brazil*.

Levenshtein, V., 1965. Binary Codes for Correcting Deletions, Insertions, and Reversals. *Doklady Akademii Nauk SSSR*, 163(4):845–848.

Monge, A. and C. Elkan, 1996. The Field Matching Problem: Algorithms and Applications. In *Proceedings of Knowledge Discovery and Data Mining 1996*.

Morton, T., 1997. Coreference for NLP Applications. In *Proceedings of ACL 1997*.

Needleman, S. and C. Wunsch, 1970. A General Method Applicable to Search for Similarities in the Amino Acid Sequence of Two Proteins. *Journal of Molecular Biology*, 48(3):443–453.

Piskorski, J., 2005. Named-Entity Recognition for Polish with SProUT. In Leonard Bolc, Zbigniew Michalewicz, and Toyoaki Nishida (eds.), *LNCS Vol 3490: Proceedings of IMTCI 2004, Warsaw, Poland.*.

Smith, T. and M. Waterman, 1981. Identification of Common Molecular Subsequences. *Journal of Molecular Biology*, 147:195–197.

Ukkonen, E., 1992. Approximate String Matching with q-grams and Maximal Matches. *Theoretical Computer Science*, 92(1):191–211.

Winkler, W., 1999. The state of record linkage and current research problems. Technical report, Statistical Research Division, U.S. Bureau of the Census, Washington, DC.