Katedra Informatiky
Fakulta Matematiky, Fyziky a Informatiky
Univerzita Komenského, Bratislava

Miery podobnosti reťazcov

(Bakalárska práca)

Tamara Kuštárová

Vedúci: RNDr. Forišek

Bratislava, 2008

Čestne prehlasujem, že som túto bakalársku prácu vypracoval(a) samostatne s použitím citovaných zdrojov.

. . . . . .

Obsah

0.1 Abstrakt

Prvá kapitola popisuje základné metriky na porovnávanie reťazcov a ich vlastnosti. Medzi popisované metriky patrí Block distance [Pau04a], Dice distance [Kon03], Jaro distance [Jar89], Jaro-Winkler distance [Win99], Hamming distance [Pau04b], Levensthein distance [Dan97], Needleman distance [Nee70], Smith-Waterman distance [Smi81]. Taktiež definujem dve vlastné metriky na porovnávanie reťazcov a to Localized distance a Good distance. V tejto kapitole zároveň porovnávam uvedené metriky a vyhodnocujem, ktoré metriky sú vhodné na porovnávanie slovenských slov.

V druhej kapitole je popísaná metrika na porovnávanie textov, Monge-Elkan recursive scheme [Wil03a]. Túto metriku som upravila na MyMetric a obe som potom otestovala na niekoľkých vzorkách textov a navzájom porovnala.

Kľúčové slová:
porovnanie reťazcov, metrika, porovnanie textov

0.2 Úvod

Cieľom tejto bakalárskej práce je popísať existujúce metriky na porovnávanie reťazcov, porovnať ich s použitím vzorky slov a na základe porovnaných metrík zadefinovať vlastnú metriku, ktorá by zohľadňovala vlastnosti slov slovenského jazyka a dokázala rozlíšiť podobné a rozdielne slová. Potom tieto metriky aplikovať na porovnávanie textov, zadefinovať vlastnú metriku na porovnávanie textov, porovnať ich a vyhodnotiť.

Prvá kapitola popisuje jednotlivé metriky na porovnávanie textov. Pri charakteristikách jednotlivých metrík som použila vybrané vzorky slov, ktoré som navzájom porovnávala a výsledky zobrazila v tabuľkách a diagramoch. Z nazbieraných dát som vyabstrahovala vlastnosti ideálnej metriky pre slovenský jazyk, ktorá by zachytávala rozdiely a podobnosti slovenských slov.

V druhej kapitole sa zaoberám porovnávaním textov, ktoré je založené na porovnávaní reťazcov metrikami popísanými v prvej kapitole. Ako nosný prvok som zvolila Monge-Elkanovu rekurzívnu schému [Wil03a], ktorá porovnáva texty za pomoci sekundárnej metriky. Porovnávala som texty pomocou tejto metriky a analyzovala výsledky. Tiež som zadefinovala vlastnú metriku, ktorá je odvodená z Monge-Elkanovej rekurzívnej schémy a snaží sa o presnejšie zachytenie rozdielov a podobností dvoch textov. Navyše, ako sekundárnu metriku používa aj metriky, ktoré som odvodila v prvej kapitole pre slovenský jazyk.

Kapitola 1

Porovnávanie reťazcov

Rôzne vedecké odbory vyvinuli veľa techník na porovnávanie reťazcov, pričom každá z nich bola špecifická pre daný účel a zohľadňovala špecifické vlastnosti reťazcov vyskytujúcich sa v danom vednom odbore. V tejto kapitole uvádzam prehľad týchto techník porovnávania reťazcov spolu s ich použitím a aplikáciami, tiež poukazujem na vlastnosti každej uvedenej metriky a definujem dve vlastné metriky na porovnávanie reťazcov slovenského jazyka.

1.1 Popis použitých reťazcov na testovanie metrík na porovnávanie reťazcov.

Na porovnanie metrík som zvolila reťazce rôznych charakteristík. V Tabuľke 1.1 sa nachádza zoznam použitých slov na porovnávanie metrík spolu s ich charakteristikou.

Pre každú metriku, ktorú uvádzam, som pripojila tabuľku s vypočítanými vzdialenosťami pre slová zo vzorky. Tie vzdialenosti, ktoré nezodpovedali skutočnej podobnosti slov, tak, ako je všeobecne vnímaná, som zvýraznila tmavo-šedou farbou. Tieto zvýraznené slová poukazujú na nedostatky ana-

okno	okne	slová, ktoré vznikli skloňovaním jedno z druhého
pekný	pekná	slová v rôznom rode
zelenina	zeleninový	slová, ktoré sú odvodené z rovnakého slova, ale sú inými slovnými druhmi
logický	logika	
hotový	hotovosť	
počítač	počítadlo	slová, ktoré majú rovnaký slovný základ, ale nie úplne identický význam
hotový	pohotovosť	
spievať	zaspievame	slová, ktoré vznikli odvodením jedno z druhého s použitím prípon a predpôn
prepísaný	predpísaný	slová, ktoré nemajú podobný význam, ale obsahujú vo veľkej miere rovnaké znaky pričom jedno vzniklo vložením nejakých znakov do druhého slova
rozzúrený	rozkúrený	slová, ktoré nemajú podobný význam, ale obsahujú vo veľkej miere rovnaké znaky pričom prvé slovo vzniklo nahradením niekoľkých znakov v druhom slove inými
nemať	nedať	slová, z ktorých prvé vzniklo poprehadzovaním znakov v druhom slove
treba	tréma	
jeleň	nelej	slová, z ktorých prvé vzniklo miernou modifikáciou druhého slova
maklér	makrela	
odysea	odysae	slová, z ktorých prvé slovo vzniklo z druhého slova tak, že doň boli vnesené pravopisné chyby, napríklad výmena znakov
panelák	mrakodrap	slová, ktoré sú rozdielne, majú rozdielne znaky, ale podobný význam

Tabuľka 1.1: Vzorka porovnávaných slov

lyzovaných metrík.

1.2 Block Distance

Táto vzdialenosť je vypočítaná na základe sumy absolútnej hodnoty rozdielu znakov na korešpondujúcich pozíciách. Iné názvy pre túto vzdialenosť sú L1 Distance alebo City Block Distance [Pau04a]. Matematická formulácia tejto vzdialenosti je definovaná ako

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

Block Distance môže byť opísaná ako vzdialenosť definovaná počtom hrán medzi bodmi na mriežke, ktoré musia byť prejdené zo znaku na pozícii q do znaku na pozícii r.

Táto vzdialenosť reflektuje syntaktickú odlišnosť dvoch reťazcov, kladie dôraz na abecednú vzdialenosť korešpondujúcich slov. Vzdialenosť dvoch reťazcov vypočítaná pomocou Block distance je určená odlišnosťou znakov na korešpondujúcich pozíciách a tiež vzdialenosťou týchto znakov podľa ich pozície v abecede. Ak porovnávame dva reťazce, ktoré predstavujú určitú hodnotu, môže byť výhodné vedieť, ako ďaleko sú od seba jednotlivé znaky. Ale pri porovnávaní slov zo slovenského jazyka nie je podstatné, ako ďaleko od seba sú korešpondujúce znaky, ale skutočnosť, či sú identické, alebo nie.

Pri reťazcoch s posunutými znakmi, napríklad po vložení prípony alebo iného znaku táto vzdialenosť vypočíta rozdiel slov väčší, ako sa v skutočnosti môže javiť. Napríklad pri nasledovných slovách, uvedených v Tabuľke 1.2 sa vypočítaná vzdialenosť nerovná skutočnej podobnosti slov.

24	počítač	počítadlo
41	televízor	teleport
59	prvý	posledný
44	oproti	naproti
125	úctyhodný	ctihodný
52	cenný	cennosti
57	zelenina	zeleninový
75	hotový	pohotový
45	hotový	hotovosť
32	organizovať	organizátor
41	logický	logika
16	predpísaný	predpýsaný
19	hocikto	kocikdo
8	odysea	odysae
34	večera	evcera
13	maklér	makrela
8	jeleň	nelej
24	pekný	pekná
10	okno	okne
77	spievať	zaspievame
50	prvý	prvému
11	treba	tréma
9	nemať	nedať
94	predpísaný	prepísaný
15	rozzúrený	rozkúrený

Tabuľka 1.2: Block distance

Pri porovnaní slov, ktoré sa líšia len niekoľkými znakmi na korešpondujúcich pozíciach dá metrika Block distance výsledok zodpovedajúci skutočnej podobnosti. (napríklad rozzúrený, rozkúrený, pekný, pekná) Ale ak v slovách vynecháme niektorý znak, tak, že pôvodné znaky nebudú korešpondovať, napríklad predpísaný, prepísaný, kde sme v druhé slovo získali upravením prvého vynechaním znaku d, ich vzdialenosť nebude zodpovedať skutočnej vzdialenosti. Napríklad vzdialenosť pre predpísaný prepísaný je 94, pričom tieto slová sa líšia len v jednom znaku.

Ak porovnáme vzdialenosti slov okno, okne a jeleň, nelej, zistíme, že sú približne rovnaké, aj keď prvé dve slová sú si podobné, druhé dve však nie. Block distance nerozlišuje, či sa slová odlišujú v koreni slova, alebo v prípone. Táto vlastnosť je pre metriku porovnávajúcu slová slovenského jazyka veľmi dôležitá, keďže potrebujeme uznať rovnaké slová v iných pádoch a časoch za podobné.

Ak vezmeme dve identické slová a jedno z nich budeme postupne meniť a priebežne počítať ich vzdialenosť, zistíme, že po zmene niekoľkých znakov sú si slová stále podobné, avšak ich vzdialenosť sa rapidne zvýši po pridaní alebo odobraní niektorého znaku. Tým totiž z posunieme pôvodne korešpondujúce znaky, preto bude celá vzdialenosť počítaná akoby z iného slova. Metrika Block distance je podobná meraniu vzdialenosti dvoch vektorov, kde reťazec, resp. slovo predstavuje vektor a znaky v reťazci predstavujú jednotlivé zložky vektora.

Z predchádzajúcich úvah som usúdila, že spôsob merania vzdialenosti medzi jednotlivými znakmi ako výpočet ich rozdielu nie je vhodný pre porovnávanie slov. Pre reťazce používané v jazyku nie je podstatné, ăká je vzdialenosť medzi 'a' a 'z', ale to, že dané znaky sú rozdielne. Preto v ideálnej metrike

nebude rozlišovaný rozdiel medzi 'a','c' a 'a','p'.

Je teda zrejmé, že Block distance je vhodnejšia na porovnanie vektorov ako na porovnanie reťazcov textu.

1.3 Dice Distance

Dice distance [Kon03] je vzdialenosť definovaná ako podiel dvojnásobku spoločných znakov a súčtu dĺžky reťazcov. Pre reťazce

$q = q_1, q_2, \ldots, q_m$

$r = r_1, r_2, \ldots, r_n$

bude ich vzdialenosť definovaná:

$$D(q, r) = \frac{2x}{(m + n)}$$

kde $x = |\{\ i \mid r_i \in \{q_1, \ldots, q_m\}\}|$

Dice distance je vzdialenosť definovaná na porovnávanie textových reťazcov. Pri porovnaní slov, ktoré majú určitý počet spoločných znakov bude výsledok indikovať podobnosť. V prípadoch, kedy Block distance nedá správny výsledok, bude hodnota Dice distance presnejšie určovať vzdialenosť dvoch reťazcov. Oproti tomu v slovách, ktoré pozostávajú z rovnakých znakov, ale v inom poradí, nebude Dice distance presná. Napríklad pri slovách jeleň a nelej bude Dice distance indikovať takmer rovnosť slov, pričom tieto slová nie sú podobné. Dice distance tiež nerozlišuje, či sa zmena v slovách nachádza na konci, začiatku alebo v strede slov, čo väčšinou má vplyv na význam slova a jeho podobnosť so slovami od neho odvedenými pomocou prípon a predpôn, čo je bežne vo veľkom počte jazykov. špeciálne pre slovenský jazyk je vhodné, aby boli rozlišované zmeny v koreni slova a zmeny zavedené skloňovaním a časovaním.Tabuľka 1.3 ukazuje porovnanie vzorky slov pomocou Dice distance so zvýraznenými problematickými typmi slov, pri ktorých nedáva Dice distance správne výsledky.

0,706	televízor	teleport
0.333	prvý	posledný
0,75	počítač	počítadlo
0.769	oproti	naproti
0.824	úctyhodný	ctihodný
0,615	cenný	cennosti
0,777	zelenina	zeleninový
0,857	hotový	pohotový
0,714	hotový	hotovosť
0,727	organizovať	organizátor
0,769	logický	logika
0,909	večera	včera
0,769	maklér	makrela
0,8	jeleň	nelej
0,8	pekný	pekná
0,75	okno	okne
0,706	spievať	zaspievame
0,6	prvý	prvému
0,6	treba	tréma
0,8	nemať	nedať
0,947	predpísaný	prepísaný
0,888	rozzúrený	rozkúrený

Tabuľka 1.3: Dice distance

Dice distance vracia výsledky v intervale <0,1>, pričom 0 predstavuje rozdielne slová a 1 identické. Dice distance počíta počet spoločných znakov, preto slová predpísaný a prepísaný uzná za podobné, čo Block distance neuznala. Pretože Dice distance nerozoznáva, v akom poradí sú znaky použité, uzná slová ako večera, včera za podobné a tiež bude pokladať za podobné slová, ktoré obsahujú tie isté znaky, alebo z veľkej miery tie isté znaky, ale v inom poradí, napríklad jeleň, nelej, maklér, makrela.

Dice distance je však miera podobnosti, ktorá dokáže realisticky zachytiť slová, ktoré majú rovnaký alebo podobný slovný základ, preto je vhodná na porovnávanie reťazcov. Nedokáže však rozlíšiť, či sa dané reťazce rozlišujú v slovnom základe, alebo v prípone, či predpone. Preto je Dice distance vhodnejšia na porovnávanie reťazcov v jazykoch, ktoré nemajú tak veľa tvarov pri skloňovaní a časovaní ako slovenský jazyk. Ale aj v jazykoch bez skloňovania nereflektuje Dice distance vzdialenosť slov presne, pretože neberie do úvahy slovné odvodenia, napríklad perfect a perfectly by mali byť viac podobné ako perfect a imperfect, ale ich vzdialenosť je rovnaká.

1.4 Hamming Distance

Hammingova distance [Pau04b] bola navrhnutá Richardom Hammingom (1959) ako jeden z kódov na detekciu a opravovanie chýb. Je používaná v telekomunikáciách na počítanie nesprávne doručených bitov v binárnom slove fixnej dĺžky ako odhad komunikačných chýb, preto je tiež niekedy nazývaná signal distance. Táto vzdialenosť je tiež používaná v informačnej teórii, teórii kódovania a kryptológii. Vzdialenosť Hamming distance je definovaná ako počet zmien potrebných na pretransformovanie prvého reťazca na druhý reťazec, pričom sa predpokladá, že dĺžka reťazcov je rovnaká. Hamming distance nedefinuje výpočet vzdialenosti pre slová, ktoré nemajú rovnakú dĺžku, pre tieto slová je teda nepoužiteľná a namiesto nej sa odporúča použiť Levensthein distance, ktorá je podobná Hamming distance, ale naviac je aplikovateľná na slová s rôznou dĺžkou. Táto metrika je definovaná nasledovne:

$$D(s_1, s_2) = \sum_{i=0}^{i=length(s_1))} d(i, i)$$

kde $d(i, i) = 0$ ak znak na pozícii i v reťazci s_1 sa rovná znaku na pozícii j v reťazci s_2, inak $d(i, i) = 1$.

Hamming distance korektne zachytí slová, ktoré majú rovnaké, alebo podobné znenie, dokáže zaregistrovať tlačové a pravopisné chyby a odraziť ich vo výpočte vzdialenosti. Ak máme dve slová obsahujúce rovnaké znaky, ale v rôznom poradí, Hamming Distance korektne vypočíta vzdialenosť týchto slov a z výsledku bude zrejmé, že tieto slová si nie sú podobné.

Nedostatok tejto metriky spočíva v tom, že nedokáže rozoznávať, či sa zmena nachádza vo vnútri slova, alebo na konci, čo môže byť pre význam slova rozhodujúce.

Tabuľka 1.4 zobrazuje niektoré typy slov, v ktorých Hamming distance obstojí a iné, v ktorých nie je použiteľná.

4	organizovať	organizátor
2	jeleň	nelej
1	pekný	pekná
1	okno	okne
2	treba	tréma
1	nemať	nedať
1	rozzúrený	rozkúrený

Tabuľka 1.4: Hamming distance

Hamming distance je vo veľkej miere obmedzená tým, že môže porovnávať iba reťazce rovnakej dĺžky. Je to veľmi jednoduchá metrika, zobrazuje počet korešpondujúcich znakov, v ktorých sa reťazce líšia. Vhodná môže byť pri porovnávaní reťazcov, ktoré majú predpísanú rovnakú dĺžku, napríklad ak chceme zistiť, na koľkých miestach sa líšia dva reťazce, napríklad telefónne čísla, kódy a podobne. Ak však chceme porovnávať slová v jazyku, potrebujeme metriku, ktorá dokáže porovnať aj slová s rôznou dĺžkou. Preto bola Hamming distance rozšírená na Levensthein distance.

1.5 Levensthein distance

Vzdialenosť Levensthein distance [Dan97] je veľmi podobná Hamming distance. Je definovaná ako počet znakov, ktoré musia byť zmenené na pretransformovanie prvého reťazca na druhý reťazec. Oproti Hamming distance navyše poskytuje možnosť pridať alebo zmazať písmeno, teda nemusí nutne platiť, že dĺžka reťazcov je rovnaká. Posledná vlastnosť robí Levensthein distance prakticky použiteľnú pre všetky reťazce, nielen pre tie s rovnakou dĺžkou.

Matematická formulácia Levensthein distance je:

$$D(i,j) = min \begin{cases} D(i-1,j-1) + d(s_i,t_j) & \text{substitúcia, kopírovanie} \\ D(i-1,j)+1 & \text{vloženie potrebného znaku do 1. slova} \quad S = s_1 \dots s_n, T = t \\ D(i,j-1)+1 & \text{vymazanie znaku z 1. slova} \end{cases}$$

vzdialenosť S a T je D(length(S),length(T))

kde

$$d(s_i,t_j) = \begin{cases} 0 & \text{keď } s_i = t_j \\ 1 & \text{inak} \end{cases}$$

Levensthein distance korektne zachytí slová, ktoré majú rovnaké, alebo podobné znenie, dokáže zaregistrovať tlačové chyby a odraziť ich vo výpočte vzdialenosti. Ak máme dve slová obsahujúce rovnaké znaky, ale v rôznom poradí, Levensthein Distance korektne vypočíta vzdialenosť týchto slov a z výsledku bude zrejmé, že tieto slová si nie sú podobné.

Oproti Hamming distance je výhoda Levensthein distance v tom, že dokáže počítať aj vzdialenosti slov, ktoré nemajú rovnakú dĺžku.

Nedostatok tejto metriky spočíva v tom, že nedokáže rozoznávať, či sa zmena nachádza vo vnútri slova, alebo na okraji, čo môže byť pre význam slova rozhodujúce. Tabuľka 1.5 zobrazuje niektoré typy slov, v ktorých Levensthein distance obstojí a iné, v ktorých nie je použiteľná.

Pretože Levensthein distance poskytuje možnosť vložiť alebo vynechať písmeno pri porovnávaní, určí táto metrika slová predpísaný a prepísaný ako podobné. Taktiež táto metrika určí, že slová nemať a nedať sú si rovnako podobné ako slová pekný a pekná. Chceli by sme, aby bol rozdiel medzi vzdialenosťami týchto slov, teda aby boli rozlíšené zmeny v strede slova a zmeny na konci slova.

3	počítač	počítadlo
4	televízor	teleport
6	prvý	posledný
2	oproti	naproti
2	úctyhodný	ctihodný
4	cenný	cennosti
3	zelenina	zeleninový
2	hotový	pohotový
3	hotový	hotovosť
4	organizovať	organizátor
2	logický	logika
4	maklér	makrela
2	jeleň	nelej
1	pekný	pekná
1	okno	okne
4	spievať	zaspievame
3	prvý	prvému
2	treba	tréma
1	nemať	nedať
1	predpísaný	prepísaný
1	rozzúrený	rozkúrený

Tabuľka 1.5: Levensthein distance

1.6 Needleman Distance

Needleman metrika [Nee70] je metrika podobná Levensthein distance, avšak navyše pridáva cenu pridania alebo vymazania ľubovoľného znaku. Iný názov pre Needleman distance je Needleman-Wunsch distance. Táto metrika je používaná v bioinformatike na zarovnávanie proteínových a nukleových postupností. Algoritmus bol navrhnutý v roku 1970 Saulom Needlemanom, Christianom Wunschom. Matematická formulácia tejto metriky je nasledovná:

$$D(i,j) = min \begin{cases} D(i-1,j-1) + d(s_i,t_j) & \text{substitúcia, kopírovanie} \\ D(i-1,j) + G & \text{vloženie znaku} \\ D(i,j-1) + G & \text{vymazanie znaku} \end{cases}$$

kde

$$d(s_i,t_j) = \begin{cases} 0 & \text{keď } s_i = t_j \\ 1 & \text{inak} \end{cases}$$

G je cena pridania, alebo vymazania znaku

Pri Needleman metrike je možné ohodnotiť pridanie alebo vymazanie znaku ľubovoľnou hodnotou, poskytuje teda väčšiu variabilitu ako Levensthein distance. V Tabuľke 1.6 sú zobrazené hodnoty pre vzorové slová vypočítané metrikou Needleman distance pre hodnoty G = 0.5, G = 1, G = 1.5.

Otázkou je, aká hodnota Gap je vhodná pre použitie v Needleman distance. V Tabuľke 1.6 sa nachádzajú tri hodnoty G, konkrétne 0.5, 1, 1,5. Ak použijeme hodnotu 0,5 pre G, bude doplnenie ľubovoľného znaku menej ohodnotené ako porovnanie existujúcich znakov, teda ak si vezmeme slová

G = 0,5	G = 1	G = 1.5		
2	3	4	počítač	počítadlo
2,5	4	5,5	televízor	teleport
4	6	7	prvý	posledný
2	2	2	oproti	naproti
2	2	2	úctyhodný	ctihodný
2,5	4	5,5	cenný	cennosti
2	3	4	zelenina	zeleninový
2	2	2	hotový	pohotový
2	3	4	hotový	hotovosť
4	4	4	organizovať	organizátor
1,5	2	2,5	logický	logika
2,5	4	4,5	maklér	makrela
2	2	2	jeleň	nelej
1	1	1	pekný	pekná
1	1	1	okno	okne
3,5	4	4,5	spievať	zaspievame
2	3	4	prvý	prvému
2	2	2	treba	tréma
1	1	1	nemať	nedať
0,5	1	1,5	predpísaný	prepísaný
1	1	1	rozzúrený	rozkúrený

Tabuľka 1.6: Needleman distance

predpísaný a prepísaný, ich vzdialenosť bude menšia ako pri hodnote G = 1, alebo G = 1,5. Ale nízka hodnota pre G má svoje výhody, hlavne pri slovách, ktoré sa odlišujú prefixom, alebo sufixom, v tom prípade sa do nich doplnia vhodné znaky a výsledná podobnosť týchto slov bude väčšia, teda hodnota Needleman distance bude nižšia ako pri hodnote G = 1, alebo G = 1,5. Príkladom takýchto slov sú napríklad spievať, zaspievame. Ak použijeme hodnotu G = 1, výsledné vzdialenosti vypočítané Needleman distance budú zhodné so vzdialenosťami vypočítanými Levensthein distance, ktorá má štandardne hodnotu G = 1. V prípade, že by sme zvolili hodnotu G vyššiu ako 1, znamená to, že vloženie znaku do slova považujeme za operáciu, ktorá viac narušuje podobnosť slov ako porovnanie prítomných znakov. Tento prístup by bol vhodný, keby sme chceli dosiahnuť, aby napríklad slová predpísaný, prepísaný si boli navzájom menej podobné, ako slová rozzúrený, rozkúrený. Pretože pre hodnotu G = 0,5 majú reťazce porovnávané pomocou Needleman distance vzdialenosť, ktorá viac korešponduje s ich skutočnou podobnosťou, použila som hodnotu G = 0,5 pre ďalšie porovnania. Defaultná hodnota, ktorá sa pre G používa pri hľadaní zarovnaní nukleotidových reťazcov je 2. Pri použití v bioinformatike však vloženie ďalšieho znaku (génu) predstavuje väčšie narušenie reťazca, ako pri slovenskom jazyku.

1.7 Jaro Distance

Jaro distance [Jar89], navrhnutá v roku 1989, autor Jaro. Jaro distance je vzdialenosť definovaná nasledovne:

$$d_j = \frac{1}{3}\Big(\frac{|m_1|}{|s_1|} + \frac{|m_2|}{|s_2|} + \frac{|m_1| - |t_{1,2}|}{|m_1|}\Big)$$

m_1 je definované ako počet znakov z reťazca 1, ktoré sa nachádzajú v reťazci 2 nie ďalej, ako

$$\left\lfloor \frac{max(|s_1| + |s_2|)}{2} \right\rfloor - 1$$

od pozície daného znaku v reťazci 1. a m_2 je počet znakov z reťazca 2, ktoré sa nachádzajú v reťazci 1 nie ďalej ako

$$\left\lfloor \frac{max(|s_1| + |s_2|)}{2} \right\rfloor - 1$$

od pozície daného znaku v reťazci 2.
s_1 a s_2 sú dĺžky reťazcov
$t_{1,2}$ je počet transpozícií v reťazci 1 vzhľadom na reťazec 2.

Jaro distance je vzdialenosť, ktorá rozširuje Dice distance, pričom ale zohľadňuje, pomery počtu spoločných znakov a dĺžku reťazcov. Taktiež berie do úvahy pomer spoločných znakov a transpozícií, čo predstavuje výskyt opakujúcich sa spoločných znakov. V Tabuľke 1.7 je prehľad typov slov a ich vzdialenosti podľa Jaro distance.

0,841	počítač	počítadlo
0,75	televízor	teleport
0,458	prvý	posledný
0,683	oproti	naproti
0,718	úctyhodný	ctihodný
0,767	cenný	cennosti
0,858	zelenina	zeleninový
0,778	hotový	pohotový
0,786	hotový	hotovosť
0,797	organizovať	organizátor
0,816	logický	logika
0,783	maklér	makrela
0,733	jeleň	nelej
0,867	pekný	pekná
0,833	okno	okne
0,625	spievať	zaspievame
0,75	prvý	prvému
0,733	treba	tréma
0,867	nemať	nedať
0,819	predpísaný	prepísaný
0,905	rozzúrený	rozkúrený

Tabuľka 1.7: Jaro distance

Metrika Jaro distance zobrazuje vzdialenosti slov na interval <0,1>, kde 0 predstavuje rozdielne slová, ktoré nemajú žiadny spoločný znak a 1 predstavuje identické slová. Počítanie transpozícií je vhodné, ak chceme zistiť, či niektoré znaky v slovách neboli vymenené, napríklad keď kontrolujeme pravopis a potrebujeme zistiť, ktoré slovo chcel používateľ napísať, ale pomýlil sa. Jaro distance teda dokáže nájsť podobnosť slov ako odysea a odysae, kde ich vzdialenosť bude 0,944. Preto je Jaro distance vhodná na rozpoznávanie spellingových chýb. Pri porovnávaní slov, a ich významu práve potrebujeme rozlíšiť slová, ktoré majú rozdielne poradie znakov, hlavne vo vnútri slova, pretože to môže meniť ich význam. Rovnako ako pri slovách nemať a nedať, ak sú v slovách niektoré znaky nekorešpondujúce, ich vzdialenosť bude väčšia ako keby mali vymenené poradie znakov. Pre porovnávanie slov v textoch, o ktorých predpokladáme, že sú správne napísané vyžadujeme, aby vzdialenosti slov nemať, nedať a nemať, nemať boli približne rovnaké, ale Jaro distance určí vzdialenosť prvých dvoch slov ako 0,867 a vzdialenosť druhých dvoch slov bude 0,933.

1.8 Jaro-Winkler distance

Jaro-Winkler distance [Win99] je vzdialenosť odvodená od Jaro distance. Jej výpočet prebieha podobne, avšak navyše zohľadňuje dĺžku spoločného prefixu a tak dáva výrazne presnejšie výsledky pre reťazce, ktoré majú spoločný prefix. Táto metrika teda pravdivo zachytí slová identické na začiatku a rozdielne na konci. Preto je vhodná pre ohýbané slová a aj odvodené slová, identifikuje spoločný prefix a vzdialenosť týchto slov vypočíta ako modifikáciu Jaro distance s ohľadom na spoločný prefix. Jaro-Winkler distance je matematicky definovaná nasledovne:

$$d_w = d_j + lp(1 - d_j)$$

kde
d_j je Jaro distance daných reťazcov
l je dĺžka spoločného prefixu
p je prefix scale - hodnota určujúca ako veľmi je metrika nastavená v prospech spoločného prefixu. Táto hodnota je defaultne 1.

Tým, že metrika Jaro-Winkler berie do úvahy spoločný prefix, stáva sa vhodnou na meranie vzdialeností slovenských slov, rovnako ako aj slov v jazykoch, kde je flexia, teda skloňovanie a časovanie. Tabuľka 1.8 zobrazuje správanie sa Jaro-Winkler distance pri rôznych typoch slov.

Ak berieme hodnotu d=0,1, dostaneme hodnotu podobnosti slov o dosť vyššiu ako pri Jaro distance, stále na intervale $\langle 0, 1\rangle$. 0,1 je štandardná prefixová hodnota. Pri vyhodnocovaní vzdialeností slov z Tabuľky 1.8 zistíme,

0,937	počítač	počítadlo
0,85	televízor	teleport
0,513	prvý	posledný
0,683	oproti	naproti
0,718	úctyhodný	ctihodný
0,86	cenný	cennosti
0,958	zelenina	zeleninový
0,778	hotový	pohotový
0,786	hotový	hotovosť
0,939	organizovať	organizátor
0,889	logický	logika
0,848	maklér	makrela
0,733	jeleň	nelej
0,92	pekný	pekná
0,833	okno	okne
0,625	spievať	zaspievame
0,825	prvý	prvému
0,786	treba	tréma
0,893	nemať	nedať
0,873	predpísaný	prepísaný
0,934	rozzúrený	rozkúrený

Tabuľka 1.8: Jaro-Winkler distance

že Jaro distance považuje za podobné slová rozzúrený a rozkúrený, taktiež predpísaný, prepísaný, ktoré chceme aby boli odlišné. Teda Jaro distance uznáva slová, ktoré majú v koreni zmenených niekoľko znakov za podobné. Výhodou však je, že považuje slová pekný, pekná a tiež okno, okne za podobné. Slová zelenina, zeleninový, ktoré sa líšia v troch znakoch uznal za podobné, tiež organizovať a organizátor. Ich vzdialenosť závisí aj od počtu znakov v slovách, preto Jaro-Winkler distance považuje slová okno, okne za približne rovnako podobné ako logický, logika, aj keď prvý pár slov sa líši v 1 znaku a druhý pár sa líši v 2 znakoch, Jaro-Winkler metrika totiž ich vzdialenosti normalizuje vzhľadom na dĺžku slova.

1.9 Smith-Waterman distance

Metrika [Smi81] definovaná pánmi Temple Smith a Michael Waterman v roku 1981 je založená na Levensthein distance optimalizovaná na porovnávanie postupností proteínov a nukleotidov (DNA). Hľadá optimálne zarovnanie DNA reťazcov, použitím hľadania miesta, kde končí najdlhšia spoločná podpostupnosť. reťazcov a tiež pridáva možnosť ohodnotenia pridania alebo vymazania znaku. Hlavným rozdielom oproti Needleman-Wunsch metrike je, že záporné hodnoty v matici výpočtu sú nastavené na nulu, teda dostaneme vždy nezápornú hodnotu vzdialenosti. Smith-Waterman distance je definovaná nasledovne:

$$D(i,j) = max \begin{cases} D(i-1,j-1) - d(s_i,t_j) & \text{substitúcia, kopírovanie} \\ D(i-1,j) - G & \text{vloženie znaku} \\ D(i,j-1) - G & \text{vymazanie znaku} \end{cases}$$

kde

$$d(s_i,t_j) = \begin{cases} -2 & \text{keď } s_i = t_j \\ 1 & \text{inak} \end{cases}$$

G je cena pridania, alebo vymazania znaku
Potom vzdialenosť slov s a t je maximum z tabuľky D(i, j) cez všetky i a j. V Tabuľke 1.9 sa nachádza prehľad hodnôt vypočítaných Smith-Waterman metrikou.

Pri Smith-Waterman metrike je vzdialenosť dvoch slov rovná maximálne súčtu dĺžky slov, teda pre identické slová by bola Smith-Waterman metrika rovná dvojnásobku dĺžky slov. Čím je Smith-Waterman distance medzi dvoma slovami menšia, tým sú si tieto slová menej podobné. Táto vzdialenosť

12	počítač	počítadlo
9	televízor	teleport
2	prvý	posledný
10	oproti	naproti
13	úctyhodný	ctihodný
8	cenný	cennosti
14	zelenina	zeleninový
12	hotový	pohotový
10	hotový	hotovosť
14	organizovať	organizátor
9	logický	logika
6	maklér	makrela
6	jeleň	nelej
8	pekný	pekná
6	okno	okne
12	spievať	zaspievame
6	prvý	prvému
4	treba	tréma
7	nemať	nedať
17	predpísaný	prepísaný
15	rozzúrený	rozkúrený

Tabuľka 1.9: Smith-Waterman distance

však závisí aj od dĺžky porovnávaných slov, preto vzdialenosť slov okno, okne je 6, aj keď sa líšia len v jednom znaku a vzdialenosť slov rozzúrený a rozkúrený je 15 a tiež sa líšia len v jednom znaku.

1.10 Localized Distance

Localized distance je metrika, ktorú som navrhla a použila na porovnávanie slovenských slov. Je založená na skutočnosti, že slovenský jazyk je jazykom s bohatou flexiou a podobné slová s rovnakým významom majú často odlišnú príponu na konci slov. Preto táto metrika ohodnocuje rozdiely, ktoré sa nachádzajú na začiatku slov väčšou váhou ako zmeny na konci slov. Podľa zoznamu najbežnejších slovenských prípon určí akú časť slova tvorí prípona a akú časť tvorí slovotvorný základ. Rozdiely v slovotvornom základe sú ohodnotené hodnotou 1 a rozdiely v prípone len hodnotou 0.1. S pomocou tejto modifikácie dokážeme zachytiť podobnosť slov, ktoré iné metriky zachytiť nedokážu. V Tabuľke 1.10 je zobrazené správanie sa Localized distance pri ohodnoteniach zmien na konci slova 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, pričom zmeny sa začiatku slova sú pri každom z nich ohodnotené 1.

Pri určovaní ideálnych hodnôt ohodnotenia zmien na konci slova a na začiatku slova som použila túto vzorku slov:
Slová, ktoré chceme, aby boli blízko seba, teda aby ich vzdialenosť bola malá:
izba, izbe
ohodnotiť, ohodnotenie
preteky, pretekár
Slová, ktorých vzdialenosť chceme aby bola približne na hranici medzi podobnosťou a rozdielnosťou
svieti, svitá
stôl stolička
kvietok, kvetináč
Slová, ktoré chceme aby boli mierne rozdielne:
neskoro, skoro

0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1		
0	0,1	3,9	4,1	4,3	4,5	4,7	4,9	5,1	5,3	5,5	izba	izbe
0,5	0,75	1	1,25	1,5	1,75	2	2,25	2,5	2,75	3	ohodnotiť	ohodnotenie
0	0,2	0,4	0,6	0,8	1	1,2	1,4	1,6	1,8	2	preteky	pretekár
0,5	0,75	1	1,25	1,5	1,75	2	2,25	2,5	2,75	3	svieti	svitá
2,5	2,75	3	3,25	3,5	3,75	4	4,25	4,5	4,75	5	stôl	stolička
3,5	3,8	4,1	4,4	4,7	5	5,3	5,6	5,9	6,2	6,5	kvietok	kvetináč
4,0	4,1	4.2	4,3	4,4	4,5	4,6	4,7	4,8	4,9	5	neskoro	skoro
1	1	1	1	1	1	1	1	1	1	1	bežať	ležať
4,5	4,6	4,7	4,8	4,9	5	5,1	5,2	5,3	5,4	5,5	trasa	okrasa
3,5	3,8	4,1	4,4	4,7	5	5,3	5,6	5,9	6,2	6,5	vodovod	závod
4,5	4,6	4,7	4,8	4,9	5	5,1	5,2	5,3	5,4	5,5	vysmieval	vysával
1	1,2	1,4	1,6	1,8	2	2,2	2,4	2,6	2,8	3	musím	moslim
5	5,2	5,4	5,6	5,8	6	6,2	6,4	6,6	6,8	7	ekvivalent	poviedka
8,5	8,6	8,7	8,8	8,9	9	9,1	9,2	9,3	9,4	9,5	problémy	samozrejme
8	8,25	8,5	8,75	9	9,25	9,5	9,75	10	10,25	10,5	terminológia	definícia

Tabuľka 1.10: Localized distance

bežať, ležať
trasa, okrasa
Slová, ktoré chceme aby boli pomerne rozdielne:
vodovod, závod
vysmieval, vysával
musím, moslim
Slová, ktoré chceme aby boli veľmi rozdielne:
ekvivalent, poviedka
problémy, samozrejme
terminológia, definícia

Pri týchto porovnaniach vidíme, že čím väčšie ohodnotenie zmien na konci slova použijeme, tým nameraná vzdialenosť stúpa. Ohodnotenie 0 zmien na konci slova znamená, že zmeny, alebo rozdiely medzi slovami, ktoré sa nachádzajú na konci slov, nebudeme vôbec brať do úvahy. V tom prípade uznáme slová izba, izbe za úplne zhodné. Toto nie je vhodné, pretože tieto slová rovnaké nie sú, ideálne by bolo, keby ich vzdialenosť bola blízko hodnoty, ktorá určuje ekvivalenciu slov, v tomto prípade 0. Ohodnotenie 1 zmien na konci slova predstavuje metriku Levensthein distance. Táto metrika bola použitá na porovnanie koreňov slova a tiež koncov slova, ak bude však ohodnotenie zmien na konci slova rovné ohodnoteniu zmien v koreni slova, budú vzdialenosti rovnaké, ako by slová boli porovnané pomocou Levensthein distance. Toto však nie je vhodné na porovnávanie slovenských slov, keď potrebujeme rozlíšiť zmeny, ktoré sa nachádzajú v koreni slova a zmeny, ktoré sa nachádzajú na konci slova.

Vhodné ohodnotenie zmien na konci slova sa teda nachádza niekde medzi 0 a 1. Pretože v slovenskom jazyku zmeny na konci slova vo veľkom množstve

prípadov predstavujú ohýbanie alebo časovanie rovnakého slova, pre slovenský jazyk zvolím za najvhodnejšie ohodnotenie 0,1, ktoré zmeny na konci slov považuje za málo dôležité, oproti tomu, zmeny v koreni slova, ohodnotené 1, považuje za dôležité.

Ak sa pozrieme na ohodnotenie slov bežať a ležať, ktorých vzdialenosť je 1, je zrejmé, že potrebujeme, aby tieto slová boli považované za rozdielne. Ak sa totiž slová líšia minimálne v jednom znaku v koreni slova, môžeme predpokladať, že sú rozdielne. Teda ak vzdialenosť dvoch slov je v intervale $\langle 0, 1)$, budeme ich považovať, pri porovnávaní pomocou Localized distance, za podobné, ak vzdialenosť dvoch slov bude väčšia, alebo rovná 1, budeme ich považovať za rozdielne.

V Tabuľke 1.11 sa nachádza porovnanie vzorových slov, porovnaných pomocou Localized distance, s ohodnotením zmien v koreni slova ako 1, a ohodnotením zmien na konci slova ako 0,1.

0,1	pekný	pekná
0,1	okno	okne
6,7	spievať	zaspievame
0,65	prvý	prvému
4,7	oproti	naproti
7,6	úctyhodný	ctihodný
0,7	cenný	cennosti
0,65	zelenina	zeleninový
3,65	hotový	pohotový
0,65	hotový	hotovosť
2,2	organizovať	organizátor
1,6	logický	logika
1,25	maklér	makrela
1,1	jeleň	nelej
2	treba	tréma
1	nemať	nedať
5,6	predpísaný	prepísaný
1	rozzúrený	rozkúrený
0,65	počítač	počítadlo
1,8	televízor	teleport
2,75	prvý	posledný

Tabuľka 1.11: Localized distance

1.11 Vhodná metrika na porovnanie reťazcov

Podľa vlastností predchádzajúcich metrík na porovnávanie reťazcov môžem určiť, ktoré vlastnosti sú pre porovnanie reťazcov vhodné a ktoré nie. Metrika, ktorá bude vedieť rozoznávať podobnosť slovenských slov by mala osobitne hodnotiť odlišnosti medzi slovami, ktoré sa nachádzajú v predpone, koreni slova a prípone. Táto metrika bude schopná rozdeliť slovo na predponu, koreň slova a príponu. Potom pre dve zadané slová osobitne porovná tieto tri časti, pričom rozdiely v každej časti budú mať inú váhu, najviac ohodnotené budú rozdiely v koreni slova - najčastejšie platí, že slová s rovnakým koreňom sú si podobné. Najmenej ohodnotené budú rozdiely v prípone, pretože tie zodpovedajú skloňovaniu a časovaniu, prípadne odvodeným slovám. Predpona bude ohodnotená menej ako koreň slova, ale zároveň viac ako prípona, pretože predpona môže buď mierne pozmeniť význam slova, alebo ho znegovať. Iným prístupom by bolo neoddeľovanie predpony, ale hľadanie najdlhšej spoločnej podpostupnosti v časti slova tvorenej predponou a koreňom slova. Táto podpostupnosť by potom bola identifikovaná ako koreň slova.

Vhodná metrika založená len syntaktických porovnaniach zrejme neexistuje, pretože v slovenskom jazyku ale tiež aj v iných svetových jazykoch sa nachádzajú slová s rovnakou syntaxou ale rozdielnou sémantikou a nie sú vymedzené presné pravidlá na určovanie významu slova pomocou jeho syntaxe. Preto metrika zvládajúca rozlišovať všetky podobné, či rozdielne slová by musela byť založená na báze slovníka obsahujúceho všetky slová v jazyku v nejakej dátovej štruktúre, napríklad grafe, ktorý by určoval ich významovú podobnosť a tiež by musela dokázať rozlišovať kontext, v ktorom boli slová použité. Ľubovoľná metrika založená na porovnávaní syntaxe, nedokáže napríklad rozlíšiť medzi slovami koruna a koruna, aj keď tieto slová môžu byť použité v rozdielnom kontexte a teda aj význame. Metrika založená

na sémantickom porovnávaní slov by mala však tieto významy rozlíšiť.

Táto práca sa však zaoberá syntaxou slov a jej cieľom je zaznamenať syntaktické rozdiely medzi slovami. Vymedzuje požiadavky metriky porovnávajúcej syntax slov, ktorá určovať syntaktickú podobnosť slov a bude zodpovedať Slovenskému jazyku.

Táto metrika rozdelí každé z porovnávaných slov na tri časti, príponu, koreň slova a predponu. Potom osobitne porovná prípony, korene slov a predpony. Tieto vzdialenosti sčíta tak, že každej priradí rôznu váhu, podľa jej dôležitosti vzhľadom na zmeny významu, ktoré predstavuje. Zmeny v koreni slova budú mať najvyššiu váhu, pretože indikujú rozdielne významy slov, oproti tomu zmeny na konci slova často predstavujú zmeny zavedené skloňovaním, časovaním a odvodzovaním. Pri týchto operáciách sa význam slova modifikuje, ale nemení úplne, preto tieto zmeny budú mať najmenšiu váhu. Predpona bude ohodnotená tak, aby odlišnosti v predponách porovnávaných slov boli zaznamenané výraznejšie ako odlišnosti v príponách, avšak menej výrazne ako odlišnosti v koreni slov. Ďalšou vlastnosťou tejto metriky by mala byť možnosť pridať alebo vymazať písmeno pri porovnávaní koreňov slova, ale za určitú cenu.

Po nájdení najdlhšej spoločnej podpostupnosti (nie nutne súvislej) v koreni slova by potom znaky, ktoré by nekorešpondovali boli zarátané ako $+1$ v sume celkovej vzdialenosti. Podobne by bola porovnaná aj prípona, ale nekorešpondujúce znaky by boli zarátané ako $+0.1$ v sume celkovej vzdialenosti.

Pri porovnávaní znakov x, y sa použije nasledujúce pravidlo:

$$d(x,y) = \begin{cases} 1 & \text{ak x=y} \\ 0 & \text{ak x} \neq \text{y} \end{cases}$$

Podľa predchádzajúcich požiadaviek bude vhodnou metrikou na porovnanie reťazcov vhodná modifikovaná Levensthein distance. Použijem ju tak, že osobitne porovnám koreň slova a osobitne príponu. Tieto hodnoty potom sčítam a normalizujem tak, sa zobrazili na interval $\langle 0, \rangle 1$. Predponu som ohodnotila hodnotou 0,5, zmeny v koreni slova predstavovali 1 a zmeny v prípone slova boli ohodnotené 0,1. Pri ďalšom porovnávaní budem pre túto metriku používať názov Good distance.

Túto metriku som implementovala a v Tabuľke 1.12 sa nachádzajú vypočítané vzdialenosti na už pred tým použitej vzorke slov.

0.1	pekný	pekná
0.1	okno	okne
4.4	spievať	zaspievame
2.2	prvý	prvému
1.0	oproti	naproti
3.0	úctyhodný	ctihodný
2.3	cenný	cennosti
2.2	zelenina	zeleninový
2.2	hotový	hotovosť
3.2	organizovať	organizátor
1.1	logický	logika
2.3	maklér	makrela
1.1	jeleň	nelej
1.1	treba	tréma
1.0	nemať	nedať
2.0	predpísaný	prepísaný
1.0	rozzúrený	rozkúrený
0.3	počítač	počítadlo
4.1	televízor	teleport
5.1	prvý	posledný

Tabuľka 1.12: Good distance

Kapitola 2

Porovnávanie textov

Pri porovnávaní textov môžeme aplikovať dva prístupy.

V prvom z nich porovnáme texty nejakou vopred vybratou metrikou, ktorá bude jednotlivé slová v textoch považovať za ucelenú jednotku. Pri abstraktom pohľade na túto skupinu metrík zistíme, že fungujú rovnako ako metriky na porovnávanie reťazcov, pričom jednotlivé slová považujú za znaky a texty za reťazce. Medzery predstavujú oddeľovače znakov resp. slov. Tento spôsob je vhodný, ak potrebujeme, aby sa slová v porovnávaných textoch vyskytovali bezo zmeny, teda nie upravené skloňovaním alebo časovaním, prípadne odvodením slova podobného významu. Môže byť použitý napríklad na porovnanie dvoch sekvencií DNA kódov, alebo dvoch zdrojových súborov.

Ak však chceme zohľadniť jazykovú flexiu a matchovať slová, ktoré sú nielen identické, ale aj tie, ktoré sú podobné, je vhodnejšie použiť druhý prístup. Tento prístup spočíva vo výbere sekundárnej metriky, ktorá je aplikovaná na jednotlivé slová. Primárna metrika potom určuje, ktoré slová podľa výskytu v texte budú porovnané sekundárnou metrikou a celková vzdialenosť textov je určená ako suma vzdialeností vypočítaných sekundárnou metrikou a môže

byť normalizovaná na interval <0,1>.

2.1 Popis použitých vzoriek textov

Na porovnanie textov pomocou vzdialeností Monge-Elkan recursive, MyMetric, so sekundárnymi vzdialenosťami definovanými v prvej kapitole som použila niekoľko vzoriek textov. Na kalibráciu premenných použitých v týchto metrikách som použila texty IndianaJones1 [Ind08a], IndianaJones2 [Ind08b], MačacieZlato1 [Mac08a], MačacieZlato2 [Mac08b]. Sú to popisy filmov Indiana Jones a Mačacie Zlato, každý z dvoch rôznych zdrojov. Tieto texty uvádzam v prílohe.

Na porovnanie a vyhodnotenie kalibrácie som použila texty PočasieZápad [Poc08b], PočasieStred [Poc08a], Praha [Pra08] a Koruna [Kor08]. PočasieZápad, PočasieStred sú texty s predpoveďou počasia pre západné a pre stredné Slovensko, ktoré sú bežný čitateľ považuje za podobné. Praha hovorí o tom, že Praha sa nedostala medzi kandidátske mestá na Olympijské hry 2016 a text Koruna hovorí o posilnení slovenskej koruny za posledné obdobie. Posledné dva texty hovoria o odlišných témach, nepovažujeme ich za podobné, zatiaľ čo predpoveď počasia pre západné Slovensko v texte PočasieZápad sa veľmi podobá predpovedi počasia pre stredné Slovenskov texte PočasieStred.

2.2 Monge-Elkan recursive scheme

Táto schéma [Wil03a] vyvinutá na porovnávanie sekvencií znakov používa prístup primárnej a sekundárnej metriky na výpočet vzdialeností dvoch textov, pričom priamo určuje len vlastnosti primárnej metriky, sekundárnu metriku ponecháva na výber používateľa a teda môže mať ľubovoľné vlastnosti. Práve v tom spočíva jej výhoda, keď že umožňuje vytvoriť metriku s vlastnosťami, ktoré potrebujeme. Matematicky je Monge-Elkan rekurzívna schéma definovaná ako:

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

kde $match(A_i, B_j)$ predstavuje vzdialenosť medzi reťazcami na pozícii A_i a B_j. Táto vzdialenosť je meraná nejakou sekundárnou metrikou, ktorá môže byť ľubovoľná metrika. Na porovnávanie som použila metriky uvedené v prvej kapitole, pričom výsledná metrika mala potom vlastnosti konkrétnej metriky na porovnávanie reťazcov kombinované s vlastnosťami Monge-Elkan distance.

Monge-Elkan vzdialenosť hľadá ku každému slovu v prvom texte slovo z druhého textu, ktoré mu je najpodobnejšie. Táto vlastnosť vytvára dve charakteristiky Monge-Elkan distance.

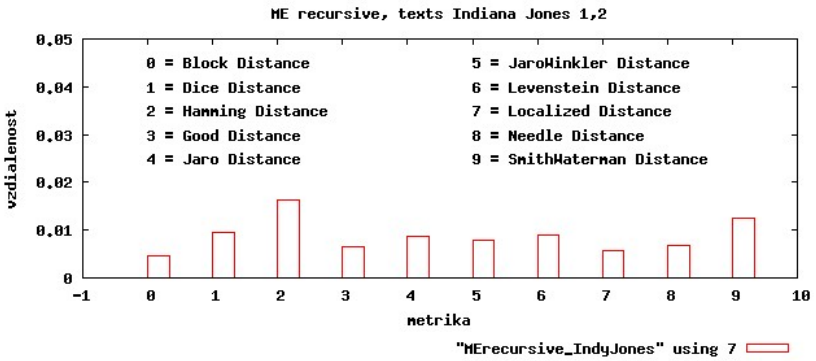
Prvá je, že slová nehľadá v ich príbuznom kontexte, teda môžu sa vyskytovať kdekoľvek v texte, nehľadá ich v nejakom okolí prvého slova. Výsledkom tohoto je, že Monge-Elkan dokáže zachytiť dva podobné texty, ktoré sa líšia napríklad usporiadaním odsekov, alebo viet. Ďalším aspektom však je, že kedže nehľadá podobné slová v nejakom okolí prvého slova, môžu slová, ktoré Monge-Elkan distance našla a považuje za najviac podobné byť použité

v úplne inom kontexte, napríklad jedno na začiatku prvého textu, kde táto časť hovorí o téme č.1 a druhé na konci textu, kde táto časť hovorí o téme č.2. Je zrejmé, že síce slová podobné byť môžu o podobnosti textov to veľa hovoriť nemusí.

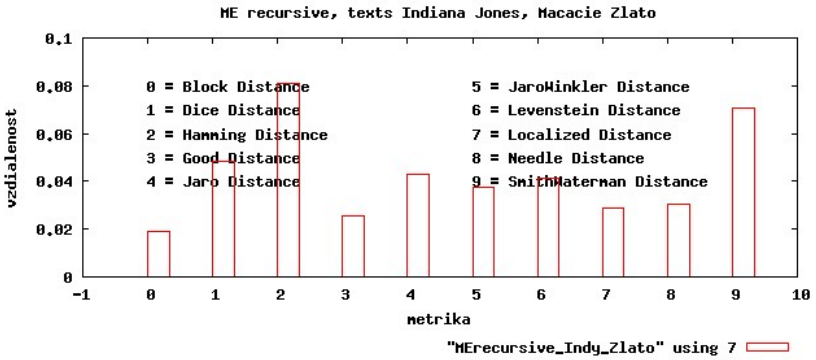
Druhou charakteristikou Monge-Elkan distance je, že keďže pre každé slovo porovná jeho vzdialenosť s každým slovom, časová zložitosť tejto metriky je $O(N \cdot M \cdot \sum_{n=0,m=0}^{n=N,m=M} \psi(n,m))$, kde $\sum_{n=0,m=0}^{n=N,m=M} \psi(n,m))$ je suma porovnania slov s indexom n a m pomocou sekundárnej metriky. Táto časová zložitosť môže byť dosť veľká na to, aby mohla byť táto vzdialenosť použitá na dlhé texty, prípadne práce. Ak napríklad chceme porovnať dva dokumenty, z ktorých každá má 5000 slov (bežné pri 25 stranovej práci) a 26000 znakov, teda jedno slovo má približne 5 znakov, potom na porovnanie týchto prác potrebujeme 625 000 000 operácií, ak ako sekundárnu vzdialenosť použijeme takú, čo porovnáva každý znak s každým.

Významnou črtou Monge-Elkan distance je, že používa sekundárnu vzdialenosť, ktorá vracia výsledky v intervale $\langle 0, 1\rangle$. Väčšina porovnávaných a bežne používaných slov však nedosahuje príliš vysoké hodnoty porovnania, teda aj keď majú rozdielny význam, množina znakov, z ktorej môžu byť vytvorené je obmedzená na počet znakov slovenskej abecedy. To znamená, že ľubovoľné dve bežne používané slová budú mať vzdialenosť bližšiu 0.5, nie 1, aj keď majú odlišný význam. Je to spôsobené tým, že niektoré hlásky, prípadne časti môžu mať rovnaké. Nízke hodnoty v sekundárnej vzdialenosti sa potom odrazia v tom, že pri porovnávaní textov a normalizovaní ich vzdialeností dosiahneme to, že vzdialenosť textov sa bude pohybovať približne od 0 po 0.2 a teda nebude v celom intervale $\langle 0, 1\rangle$.

Nasledujúce grafy zobrazujú vzdialenosti troch vzoriek textu. Prvé texty, ich porovnanie je zobrazené na Obrázku 2.1, IndianaJones1, Indiana-



Obrázok 2.1: Monge-Elkan recursive



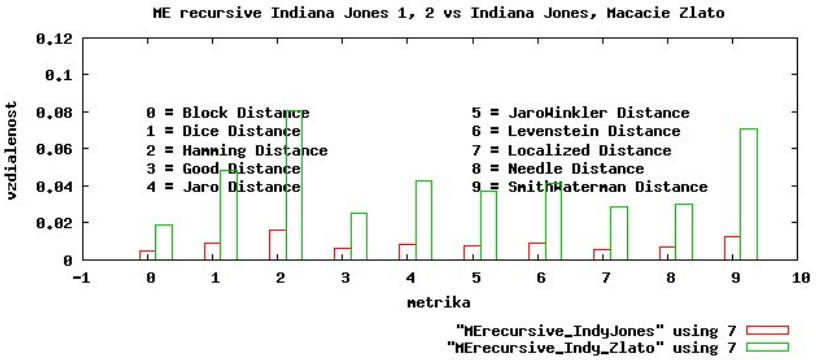
Obrázok 2.2: Monge-Elkan recursive

JONES2, oba obsahujú stručný popis filmu Indiana Jones, z dvoch rôznych zdrojov. Všetky použité články sa nachádzajú v prílohe. Obrázok 2.2 zobrazuje porovnanie textov INDIANAJONES1 a MAČACIEZLATO1.

Pri porovnaní hodnôt nachádzajúcich sa v Obrázkoch 2.1, 2.2 vidíme, že všeobecne hodnoty pre porovnanie textov na Obrázku 2.2 sú vyššie ako pre texty na Obrázku 2.1. Aj po prečítaní textov je zrejmé, že texty zobrazené na

Obrázku 2.1, teda IndianaJones1, IndianaJones2 sú si podobné. Texty z Obrázku 2.2 hovoria o úplne iných témach, po prečítaní je zrejmé, že tieto texty si nie sú podobné. Podľa týchto textov určím hodnoty, ktoré budú určovať hranicu podobnosti a rozdielnosti pre Monge-Elkan recursive pri použití rôznych sekundárnych metrík.

Na Obrázku 2.3 sa nachádza porovnanie vzdialeností vypočítaných Monge-Elkan recursive metrikou už pred tým zobrazených v Obrázkoch 2.1 a 2.2. Vidíme tu dve porovnania, pričom prvé je pre texty, ktoré sú si podobné a druhé pre rozdielne texty. Určené hranice sa nachádzajú v Tabuľke 2.1.



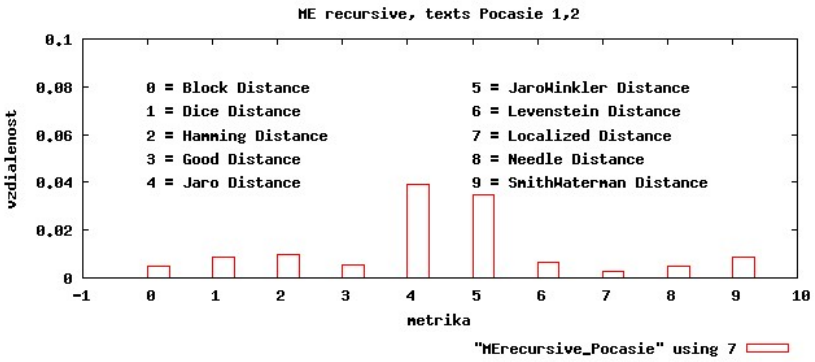
Obrázok 2.3: Monge-Elkan recursive

Teraz sa pozrieme na porovnanie dvoch párov textov, POČASIE1, POČASIE2 a PRAHA, KORUNA. Pri porovnaní aplikujeme hodnoty určené pomocou predchádzajúcich porovnaní. Obrázky 2.4 a 2.5 zobrazujú tieto porovnania a Obrázok 2.6 ich porovnáva navzájom. V Grafe 2.6 vidíme zelenou zvýraznenú hranicu určujúcu podobnosť a rozdielnosť textov. Ako je vidieť z Obrázka, pre všetky použité metriky korektne určuje táto hranica podobnosť textov POČASIE1, POČASIE2 a rozdielnosť textov PRAHA, KORUNA.

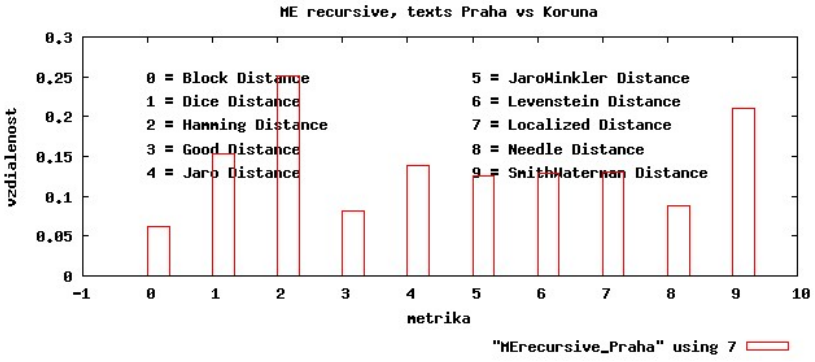
Pri pohľade na vzdialenosti textov počítané pomocou Monge-Elkan recursive, je vidno, že vzdialenosti ľubovoľných textov sú si pomerne blízke, pohybujú sa približne na intervale $\langle 0, 0.5\rangle$. Je to spôsobené tým, že keďže Monge-Elkan recursive hľadá čo najpodobnejšie slovo ku každému slovu z prvého textu a do celkovej vzdialenosti započítava ich podobnosť, potom je zrejmé, že táto celková vzdialenosť, keď ju vydelíme počtom slov v textoch, dostaneme hodnotu, ktorá je bližšie k 0 ako k 1.

0.01	Block distance
0.03	Dice distance
0.05	Hamming distance
0.015	Good distance
0.05	Jaro distance
0.045	Jaro Winkler distance
0.02	Levensthein distance
0.015	Localized distance
0.015	Needleman distance
0.04	Smith-Waterman distance

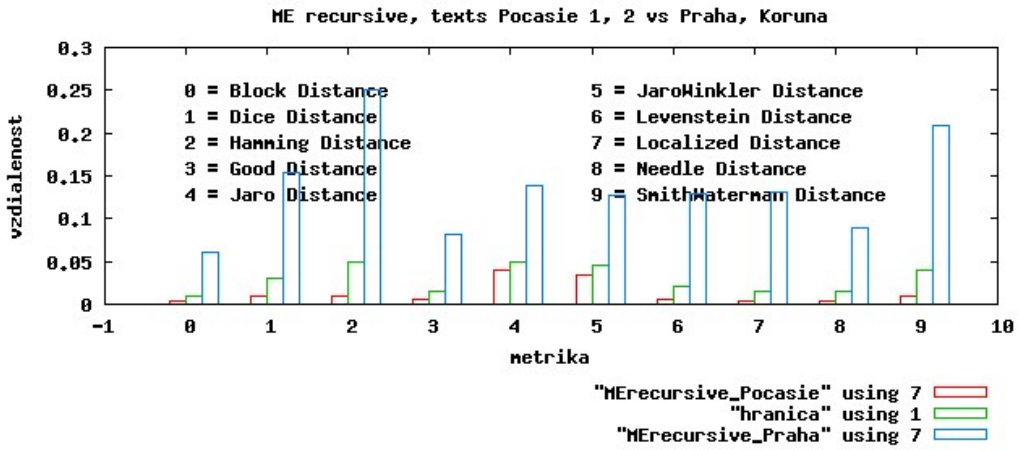
Tabuľka 2.1: Monge-Elkan recursive



Obrázok 2.4: Monge-Elkan recursive



Obrázok 2.5: Monge-Elkan recursive

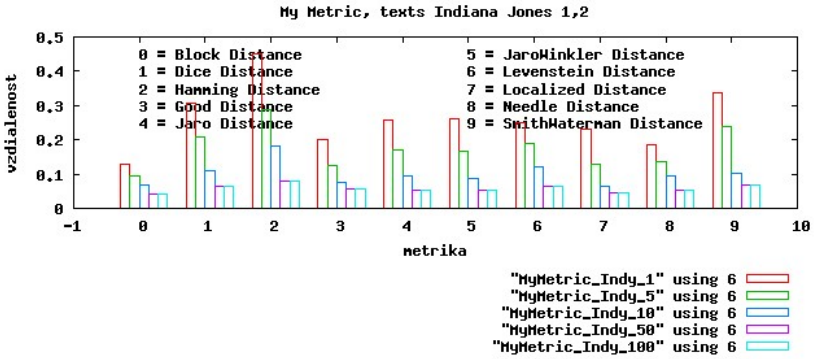


Obrázok 2.6: Monge-Elkan recursive

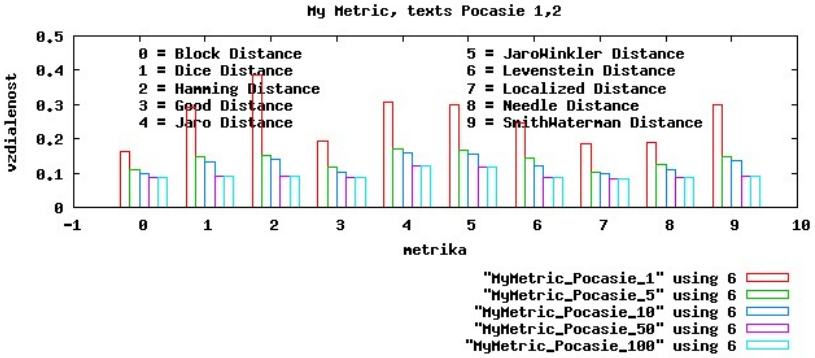
Metrika Monge-Elkan recursive má jeden významný nedostatok. Pre každé slovo z prvého textu totiž hľadá slovo v druhom texte, ktoré mu je najbližšie. Takto môže nájsť podobné slová, ale použité v inom kontexte, teda miera podobnosti textov nemusí byť realistická. Preto som túto metriku upravila na metriku MyMetric, ktorá k porovnávaniu slov pristupuje podobne ako Monge-Elkan recursive, avšak pre slovo z prvého textu hľadá čo najpodobnejšie slovo v druhom texte, ale iba v určitom rozmedzí, alebo tolerancii vzdialenosti týchto slov.

2.3 MyMetric

Metrika MyMetric funguje na princípe Monge-Elkan recursive distance. Teda používa sekundárnu vzdialenosť, ktorá môže byť ľubovoľne zvolená na nájdenie dvojíc slov, každé z jedného porovnávaného textu, takých, čo sú si najbližšie. Avšak kým Monge-Elkan recursive pre jedno slovo z prvého textu hľadá najpodobnejšie slovo z druhého textu v celom druhom texte, MyMetric toto obmedzuje a podobné slovo hľadá iba v určitej vzdialenosti od miesta,

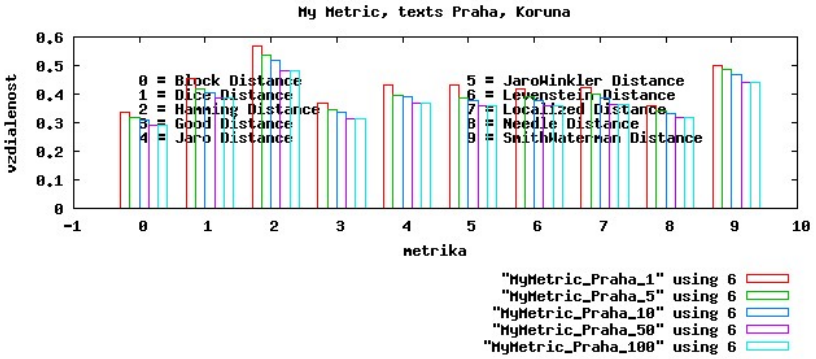


Obrázok 2.7: MyMetric



Obrázok 2.8: MyMetric

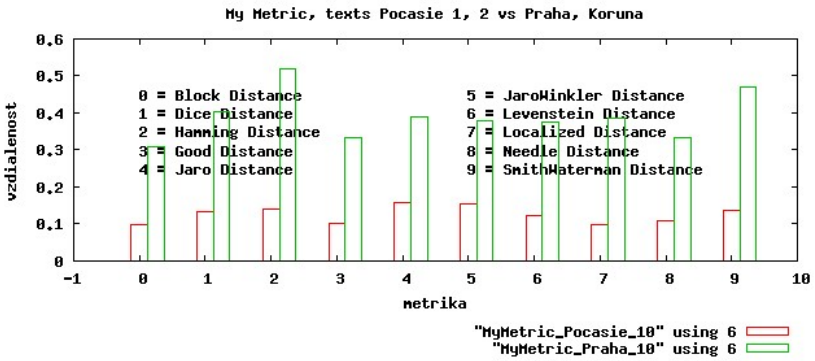
ktoré zodpovedá pozícii slova v prvom texte. Táto vzdialenosť je konfigurovateľná v programe, ktorý som napísala a na Obrázku 2.7 sa nachádzajú výsledky porovnávania textov IndianaJones1, IndianaJones2, použitých aj pri Monge-Elkan recursive. V Obrázku 2.8 sú porovnané texty Počasie1, Počasie2 a v Obrázku 2.9 texty Praha, Koruna.



Obrázok 2.9: MyMetric

Ako je vidno z grafov, MyMetric hľadá najpodobnejšie slovo vo vzdialenosti 1, 5, 10, 50 a 100 slov od pozície zodpovedajúcej slovu v prvom texte. Túto vzdialenosť nazvem tolerancia. Z grafov je tiež zrejmé, že čím je použitá väčšia tolerancia, tým sú si texty viac podobné. Je to prirodzené, lebo čím pripustíme väčšiu toleranciu, tým je prehľadávame väčší priestor slov a je väčšia pravdepodobnosť, že nájdeme podobnejšie slovo ako pri nízkej tolerancii. Toto má, samozrejme, výhody aj nevýhody. Rozlíšime výskyty podobných slov, ale použitých v inom kontexte. Môže to byť užitočné, ak vieme, že texty, ktoré chceme porovnať, majú podobné aj časti, teda napríklad usporiadanie odsekov hovoriacich o rovnakých alebo podobných témach je rovnaké. Nevýhodou je, že ak máme v textoch poprehadzované časti, odseky a podobne, nájdeme málo podobných slov a budeme tieto texty považovať za odlišné.

Hodnota tolerancie môže byť volená pri porovnaní rôzne, ale v ďalších porovnaniach používam hodnotu 10, ktorá hľadá podobné slová vo vzdialenosti 10 od výskytu slova v prvom texte. Na Obrázku 2.10 uvádzam porovnanie vzdialeností textov POČASIE1,POČASIE2 a PRAHA,KORUNA. Predpokladám, že texty POČASIE1, POČASIE2 sú si podobné a texty PRAHA,KORUNA



Obrázok 2.10: MyMetric

sú odlišné.

Odlišnosť porovnaných vzoriek je zrejmá aj z porovnania pomocou MyMetric. Tu boli texty POČASIE1,POČASIE2 uznané za podobné a texty PRAHA,KORUNA za rozdielne.

2.4 Porovnanie Monge-Elkan recursive a MyMetric

Ak chceme určiť, ktorá vzdialenosť je vhodnejšia na porovnanie textov, bude to Monge-Elkan recursive. Práve vlastnosť, že Monge-Elkan recursive dokáže vypočítať korektnú vzdialenosť aj textov s preusporiadanými odsekmi a poprehadzovanými vetami ju robí použiteľnou na veľké množstvo rôznych textov. Metrika, ktorú som navrhla ako alternatívu Monge-Elkan recursive, by nedokázala korektne určiť vzdialenosť textov, z ktorých prvý by mal dĺžku napríklad 500 slov a druhý by bol odlišný, s dĺžkou 500, za ktoré by sme ešte pridali kópiu druhého textu. Tieto texty by nepovažovala za podobné, pričom korektná metrika by ich vzdialenosť mala určiť niekde na hranici

medzi podobnosťou a odlišnosťou.

Kapitola 3

Záver

V prvej kapitole som uviedla metriky na porovnávanie textov Block distance, Dice distance, Hamming distance, Jaro distance, Jaro-Winkler distance, Levensthein distance, Needleman distance, Smith-Waterman distance. Tieto metriky som otestovala na vzorke slov a porovnala výsledky. Určila som, kedy je výhodné použiť jednotlivé metriky a tiež som poukázala na vlastnosti týchto metrík, ktoré nie sú vhodné na porovnávanie slov slovenského jazyka. Na základe týchto pozorovaní som zadefinovala dve metriky, Localized distance a Good distance, ktoré majú vlastnosti potrebné na rozoznávanie podobnosti slovenských slov. Tieto metriky som otestovala na vzorke slov, ktorú som používala aj pre ostatné metriky a výsledky, ktoré som dostala presnejšie zachytávali podobnosť slov vo vzorke. Pre metriku Localized distance som experimentálne určila hodnotu zmien na konci slova. Tento výsledok som potom overila na vzorke slov a zistenú hodnotu som použila aj ako ohodnotenie zmien na konci slova pre Good distance.

Metriky, ktoré som uviedla v tejto práci zohľadňujú syntaktickú podobnosť slov. Pre rozoznávanie podobnosti všetkých slov v jazyku však metriky, ktoré analyzujú syntaktickú podobnosť slov nerozoznávajú slová použité v rôznom

kontexte. Ak by sme chceli zadefinovať metriku, ktorá by rozoznávala takéto slová, museli by sme zájsť do sémantickej roviny jazyka a porovnávať kontexty, v ktorých sú slová použité.

V druhej kapitole som uviedla metriku na porovnávanie textov, Monge-Elkan recursive scheme. Pomocou tejto metriky a s použitím všetkých metrík zadefinovaných v prvej kapitole ako sekundárnych metrík som porovnala a vyhodnotila niekoľko vzoriek textov. Taktiež som zadefinovala vlastnú MyMetric distance, ktorá slúži na porovnávanie textov a je odvodená z Monge-Elkan recursive scheme. Pomocou tejto metriky som porovnala niekoľko vzoriek textov a vyhodnotila vlastnosti tejto metriky. Nakoniec som experimentálne určila hodnoty hranice pre Monge-Elkan recursive scheme, ktorá určuje, či budú dva texty považované za podobné, alebo rozdielne. Túto hranicu som určila pre každú sekundárnu metriku a tiež overila na ďalšej vzorke textov.

Kapitola 4

Prílohy

4.1 Použité texty

4.1.1 IndianaJones1

Indiana Jones - univerzitný profesor a muž, ktorý objavil legendárnu Archu, zachránil posvätné Sankarove kamene a našiel Svätý grál bude v novom dobrodružstve pátrať po ďalšej slávnej relikvii - záhadnej Krištáľovej lebke. Príbeh sa odohráva symbolicky presne 19 rokov po Poslednej krížovej výprave, v dobe rodiacej sa Studenej vojny. Indiana Jones je v čom profesorom archeológie na prahu penzie, ktorý si napriek búrlivej minulosti zo všetkého najviac užíva kľud a nebezpečne sa začína podobať svojmu otcovi. Staré inštinkty sa v ňom prebudia vo chvíli, keď sa objavia stopy vedúce k legendárnej Krištáľovej lebke a hlavne informácie, že po tomto údajne mocnom artefakte prahnú sovietski agenti. Indy preto strasie únavu a vydá sa na ďalšiu mimoriadne dobrodružnú cestu, ktorá ho zavedie až do vnútra peruánskej džungle, v ktorej sa má táto vzácna relikvia ukrývať. Jeho úhlavným nepriateľom bude tento raz všetkého schopná ruská agentka Irina Spalko, sprevádzaná armádou hrdlorezov. Ani Indy nie je na svoju výpravu sám. Na jeho strane

stojí spriaznený dobrodruh Mac, ktorý má však tendenciu každú chvíľu ho podraziť, ďalej napomádovaný frajer Mutt Williams, ktorý si permanentne uťahuje z jeho veku, a jeho prvá a zrejme životná láska Marion.

4.1.2 IndianaJones2

Príbeh sa odohráva symbolicky presne 19 rokov po Poslednej krížovej výprave, v dobe rodiacej sa Studenej vojny. Indiana Jones je v ňom profesorom archeológie na prahu penzie, ktorý si napriek búrlivej minulosti zo všetkého najviac užíva kľud a nebezpečne sa začína podobať svojmu otcovi. Staré inštinkty sa v ňom prebudia vo chvíli, keď sa objavia stopy vedúce k legendárnej Krištáľovej lebke a hlavne informácie, že po tomto údajne mocnom artefakte prahnú sovietski agenti. Indy preto strasie únavu a vydá sa na ďalšiu mimoriadne dobrodružnú cestu, ktorá ho zavedie až do vnútra peruánskej džungle, v ktorej sa má táto vzácna relikvia ukrývať. Jeho úhlavným nepriateľom bude tento raz všetkého schopná ruská agentka Irina Spalko(Cate Blanchett), sprevádzaná armádou hrdlorezov. Ani Indy nie je na svoju výpravu sám. Na jeho strane stojí spriaznený dobrodruh Mac (Ray Winston), ktorý má však tendenciu každú chvíľu ho podraziť, ďalej napomádovaný frajer Mutt Williams (Shia LaBeouf), ktorý si permanentne uťahuje z jeho veku, a jeho prvá a zrejme životná láska Marion (Karen Allen).

4.1.3 MačacieZlato1

Ben "Finn"Finnegan (Matthew McConaughey) je sympatický moderný lovec pokladov, ktorý je posadnutý hľadaním legendárneho Kráľovninho vena z 18-teho storočia a 40 debien klenotov nevyčísliteľnej hodnoty, ktoré sa v roku 1715 stratili v mori. Pátranie ho pripraví o všetko, čo má, vrátane manželky Tess (Kate Hudson).

4.1.4 MačacieZlato2

Ben "Finn"Finnegan (Matthew McConaughey) je sympatický moderný lovec pokladov, ktorý je posadnutý hľadaním legendárneho Kráľovninho vena z 18-teho storočia a 40 debien klenotov nevyčísliteľnej hodnoty, ktoré sa v roku 1715 stratili v mori. Pátranie ho pripraví o všetko, čo má, vrátane manželky Tess (Kate Hudson). Krátko potom, ako si Tess začne budovať svoj vlastný život ako asistentka miliardára Nigela Honeycutta (Donald Sutherland), objaví Finn dôležitú stopu o mieste, kde sa poklad nachádza. Na Tesiino zdesenie sa Finn objaví na palube Nigelovej jachty a svojím šarmom presvedčí magnáta a jeho mediálne preslávenú dcéru Gemmu (Alexis Dziena), aby sa k nemu v honbe za španielskym bohatstvom pridali. Dokonca ani Tess už dlhšie nedokáže odolať šanci odhaliť poklad, ktorý im tak dlho unikal.

4.1.5 Koruna

Silná úroveň slovenskej meny pred vstupom do eurozóny je blízko férovej hodnoty, pretože Slovensko v posledných rokoch prudko zvýšilo produktivitu, uviedla analytička Claire Dissauxová. Nečakaný 15-percentný posun centrálnej parity slovenskej koruny môže vyvolať špekulatívne hry na devízových trhoch okolitých štátov, avšak analytici varujú, že nemusí ísť o také "jednoznačné stávky", ako sa na prvý pohľad zdá.

4.1.6 Praha

Praha sa nedostala medzi kandidátske mestá na usporiadanie OH 2016. Rozhodlo o tom hlasovanie výkonného výboru MOV v Aténach. Zo siedmich uchádzačov môžu o usporiadanie hier naďalej bojovať Chicago, Tokio, Rio de Janeiro a Madrid. S Prahou vypadli z hry Baku a Dauhá.

4.1.7 Počasie1

V piatok 6.6.2008 bude na strednom Slovensku polojasno a búrky. Denné teploty sa budú pohybovať v intervale od 23 do 30 °C. V sobotu 7.6.2008 očakávame polojasno s búrkami . Denné teploty sa budú pohybovať v intervale od 22 do 28 °C. V nedeľu 8.6.2008 čakajme polojasno a búrky . Denné teploty sa budú pohybovať v intervale od 25 do 29 °C. V pondelok 9.6.2008 predpokladáme, že bude polojasno a búrky. Denné teploty sa budú pohybovať v intervale od 26 do 30 °C. V utorok 10.6.2008 očakávame polojasno s dažďovými prehánkami . Denné teploty sa budú pohybovať v intervale od 26 do 30 °C. V stredu 11.6.2008 čakajme polojasno a búrky . Denné teploty sa budú pohybovať v intervale od 26 do 30 °C. Vo štvrtok 12.6.2008 bude polojasno s búrkami. Denné teploty sa budú pohybovať v intervale od 26 do 30 °C.

4.1.8 Koruna

V piatok 6.6.2008 bude na západnom Slovensku polooblačno a dažďové prehánky. Denné teploty sa budú pohybovať v intervale od 23 do 30 °C. V sobotu 7.6.2008 očakávame polooblačno s búrkami . Denné teploty sa budú pohybovať v intervale od 22 do 28 °C. V nedeľu 8.6.2008 čakajme polooblačno a búrky . Denné teploty sa budú pohybovať v intervale od 25 do 29 °C. V pondelok 9.6.2008 predpokladáme, že bude polooblačno a dažďové prehánky. Denné teploty budú v intervale od 26 do 30 °C. V utorok 10.6.2008 očakávame polojasno . Denné teploty sa budú pohybovať v intervale od 26 do 30 °C. V stredu 11.6.2008 čakajme polojasno a dažďové prehánky . Denné teploty sa budú pohybovať v intervale od 26 do 30 °C. Vo štvrtok 12.6.2008 bude polooblačno s dažďovými prehánkami . Denné teploty sa budú pohybovať v intervale od 26 do 30 °C.

Literatúra

[Dan97] Dan Gusfield. Algorithms on strings, trees, and sequences: computer science and computational biology. in *Cambridge University Press, New York, NY, USA*, , 1997.

[Ind08a] Indiana Jones a kralovstvo kristalovej lebky 1, 2008. http://www.palacecinemas.sk/movie.asp.

[Ind08b] Indiana Jones a kralovstvo kristalovej lebky 2, 2008. http://www.kinanitra.sk/.

[Jar89] Jaro, M. A. Advances in record linking methodology as applied to the 1985 census of Tampa Florida. 1989.

[Kon03] Kondrak G, Marcu D, Knight K. Cognates Can Improve Statistical Translation Models. 2003.

[Kor08] Posilnenie koruny, 2008. http://www.sme.sk/.

[Mac08a] Macacie zlato1, 2008. http://www.palacecinemas.sk/movie.asp.

[Mac08b] Macacie zlato2, 2008. http://amfiko2007.spontane.sk/amfiko.php.

[Nee70] Needleman S, Wunsch C. A general method applicable to the search for similarities in the amino acid sequence of two proteins. 1970.

[Pau04a] Paul M. Sant. Dictionary of Algorithms and Data Structures. in *Dictionary of Algorithms and Data Structures* [online], Paul E. Black, ed., U.S. National Institute of Standards and Technology, 2004. (accessed 19 April 2006) http://www.nist.gov/dads/HTML/manhattanDistance.html.

[Pau04b] Paul M. Sant. Dictionary of Algorithms and Data Structures. in *Dictionary of Algorithms and Data Structures* [online], Paul E. Black, ed., U.S. National Institute of Standards and Technology, 2004. (accessed 19 April 2006) http://www.nist.gov/dads/HTML/HammingDistance.html.

[Poc08a] Predpoved pocasia pre stredne Slovensko, 2008. http://www.sme.sk/.

[Poc08b] Predpoved pocasia pre zapadne Slovensko, 2008. http://www.sme.sk/.

[Pra08] Praha a Olympijske hry 2016, 2008. http://www.sme.sk/.

[Smi81] Smith TF, Waterman MS. Identification of Common Molecular Subsequences. 1981.

[Wil03a] William W. Cohen, Pradeep Ravikumar, Stephen E. Fienberg. A Comparison of String Distance Metrics for Name-Matching Tasks. *II Web Acapulco 2003*, 2003. www.cs.cmu.edu/ wcohen/postscript/ijcai-ws-2003.pdf.

[Wil03b] William W. Cohen, Pradeep Ravikumar, Stephen E. Fienberg. A Comparison of String Metrics for Matching Names and Records. *IEEE Transactions on Information Theory*, 2003. cmu.edu/ pradeepr/papers/kdd03.pdf.

[Win99] Winkler, W. E. The state of record linkage and current research problems. 1999.