Získávání paralelních textů z webu*

Hana Klempová[1], Michal Novák[1], Peter Fabian[1], Jan Ehrenberger[2], and Ondřej Bojar[1]

[1] Charles University in Prague, Institute of Formal and Applied Linguistics (ÚFAL)
hana.klempova@seznam.cz, mixnov@gmail.com, macbeth8@gmail.com, bojar@ufal.mff.cuni.cz
[2] Czech Technical University, FJFI
lankvil@seznam.cz

Abstrakt *Příspěvek se zaměřuje na vytváření paralelního česko-anglického korpusu pro účely strojového překladu vyhledáváním a stahováním paralelních textů z Internetu. Navrhujeme a vyhodnocujeme několik vlastních metod pro nalezení kandidátských webů, identifikaci jazyka stránek a především párování získaných dokumentů.*

1 Úvod

Texty dostupné elektronicky ve více jazycích, tzv. paralelní texty, představují velmi cenný zdroj dat pro tvorbu překladových slovníků nebo pro překladatele. Nenahraditelné jsou pro systémy (statistického) strojového překladu.

Práce Bojar et al. (2008) popisuje druhé vydání paralelního česko-anglického korpusu CzEng a uvádí vliv velikosti a typu použitých paralelních dat na kvalitu strojového překladu. Nejcennější je získat texty v daném oboru, který chceme překládat, jak se však ukazuje, i jakékoli texty mimo obor mohou kvalitu výstupu zlepšit.

Cílem této práce je navrhnout a otestovat metodu automatického získávání paralelního korpusu z webu. Proces vytváření korpusu sestává z následujících fází: hledání kandidátských webů, které pravděpodobně obsahují identické texty ve více jazycích (viz oddíl 2), procházení a stahovaní stránek (textů) z kandidátských webů (oddíl 3), automatické identifikace jazyka jednotlivých stránek (oddíl 4) a ze závěrečného párovaní stažených textů (oddíl 5).

Článek v příslušných oddílech přináší jak vyhodnocení úspěšnosti nalezení paralelních webů, tak i vyhodnocení párovaní nalezených dokumentů.

2 Hledání kandidátských webů

Prvním problémem, kterému čelíme při získávání paralelních korpusů z internetu je problém nalezení vhodných kandidátských stránek, které potenciálně obsahují odkaz na svůj překlad, nebo jsou samy vícejazyčné či jinak paralelní. Výběr vhodné metody je důležitý, protože ovlivňuje jednak množství získaných výsledků, ale také počet stránek, které bude nutné prohledat a mezi kterými se budou hledat páry. V této fázi upřednostňujeme přesnost (precision) nalezených výsledků před jejich množstvím (recall), které by však obsahovalo jen málo relevantních dokumentů a jehož zpracování by mohlo trvat neúměrně dlouhou dobu.

2.1 Metody hledání

Dosavadní články pojednávající o získávání paralelních korpusů se příliš touto fázi nezabývají, hledání kandidátských webů je většinou řešeno jednoduchým dotazem do webového vyhledávače, např. v práci Resnik and Smith (2003) je použito vyhledávače Altavista. Hlubší rozbor relevantních dotazů do vyhledávačů je učiněn v práci Saint-Amand (2008). My jsme použili právě tuto metodu v mírně zjednodušené verzi. Metoda spočívá v kladení vhodně zvolených ručně vytvořených dotazů na největší internetové vyhledávače – Yahoo.com a Google.com – s cílem hledat stránky v jednom jazyce obsahující odkaz na jinou jazykovou mutaci. Původní Saint-Amandova metoda spočívá v automatickém nalezení názvů jazyků na základě jejich zkratek („cs“ a „en“) a jejich křížové kombinaci s doménovými jmény typickými pro stránky v daném jazyce („.cz“ pro české stránky, „.uk“ ap. pro anglické stránky). Pro pár čeština-angličtina byly tedy vyrobeny dotazy ve tvarech: „english site:cz“, „česky site:uk“, „česky site:nz“, „čeština site:au“ a jejich různé jazykově zkřížené kombinace. Na rozdíl od Saint-Amanda jsme se rozhodli doplnit hledání výrazů „česky“ o název jazyka „čeština“ a prohledávání těchto výrazů jsme rozšířili také na všeobecné domény com, net a org. Při hledání je možné též využít složitějších dotazů používajících direktivy `filetype`, `inanchor`, `intext`, `inurl` (např. „česky inurl:lang=en“ pro stránky jejichž URL obsahuje náznak, že je stránka v angličtině), či se zkusit spolehnout na rozpoznávaní jazyků vyhledávače Google (direktiva `lang`). Posledně jmenovanou možnost jsme se po zběžném prohlédnutí nepříliš spolehlivých výsledků rozhodli zatím nepoužít, její pozdější bližší prozkoumání je však pravděpodobné.

Při použití databází vyhledávačů jako vstupní brány ke zdánlivě nekonečnému množství dat na webu

* Práce na tomto projektu je podporována granty FP7-ICT-2007-3-231720 (EuroMatrix Plus) a MSM 0021620838.

narážíme na vestavěné limity dotazovacích rozhraní. Při použití veřejného API k vyhledávači Yahoo.com je možné položit maximálně 5 000 dotazů denně; co je však horší, maximálně je možno se dostat k 1 000 výsledkům na jeden dotaz. Například na dotaz „english site:cz“ Yahoo.com zahlásí 40 milionů nalezených dokumentů, můžeme se však dostat jen k tisíci z nich. Ještě o něco horší je situace s vyhledávačem Google, který sice hlásí nalezení téměř 70 milionů dokumentů, ale při 614. výsledku vyhodnotí další položky jako podobné již zobrazeným a nevypíše je. Při použití klasického přístupu přes internetový prohlížeč, nebo pomocí SOAP API platí omezení na 1 000 výsledků na jeden dotaz obdobně jako u Yahoo, celkový počet povolených dotazů je však ještě menší. Mezi další možnosti, jak extrahovat z vyhledávačů více výsledků, patří použití různých triků s restrikcí pomocí data, či zaregistrování se pro použití nového AJAXového API.

K dalším možnostem hledání vhodných kandidátů patří implementace vlastního robota, který bude stránky procházet a hledat mezi nimi vícejazyčné weby. Tento postup je zevrubně popsán v Chen et al. (2003).

2.2 Vyhodnocení vhodnosti nalezených webů

Pomocí výše popsaných jednoduchých metod se nám povedlo získat pomocí vyhledávače Yahoo cca. 8 500 URL adres a pomocí Google cca 6 400 URL adres. Z nich jsme na ruční evaluaci vybrali náhodně 2 %, co činí 171, resp. 129 stránek. U stránek jsme kontrolovali, zda obsahují odkaz na paralelní texty; kromě toho jsme taky hodnotili celou doménu (druhého řádu) příslušnou k nalezené stránce. Přehled klasifikace stránek a domén uvádějí tabulky 1 a 2.

Jak je vidět v tabulce 3, z vyhledávače Yahoo je paralelních 35 % stránek, z Google je to jenom 15 %, celkově paralelních je 26 % stránek. Bez jakékoliv paralelní informace je v případě Yahoo 43 % stránek, v případě Google až 57 % stránek, což je v celkovém počtu 49 % stránek. Obtížně použitelných nebo zcela nepoužitelných stránek je tedy po sečtení necelých 75 %.

Tabulka 1. Klasifikace stránek při ručním hodnocení.

n	obsahuje jenom jednu jazykovou verzi a v jejím okolí jsme nenašli žádnou stopu po paralelní verzi
x	obsahuje dva jazyky, jsou však oba přítomny v téže stránce
f	se vyskytuje ve vícero jazykových mutacích, každá z mutací má jiný obsah (typický příklad: Wikipedie)
l	sice obsahuje přepínač jazyků, obě verze však mají totožný obsah (tedy česká i anglická verze obsahují tentýž český text)
m	má překlad do jiných jazykových verzí, ale jenom strojový, většinou Google Translate
s	sice paralelní, ale strukturně se překlady značně liší
p	paralelní stránka

Tabulka 2. Klasifikace domén při ručním hodnocení.

ugc	uživatelsky tvořený obsah – typicky přeložené jen části obecného rozhraní stránek, např. blogy, vlastní youtube kanály a pod.
p/f	část domény je paralelní, část obsahuje pro různé jazyky odlišné texty
p/l	část domény je paralelní, část jednojazyčná, tvářící se, že obsahuje jazyků více
p/n	část domény je paralelní, část jednojazyčná
p	paralelní stránka

Tabulka 3. Hodnocení stránek podle vyhledávačů (hodnoty v procentech).

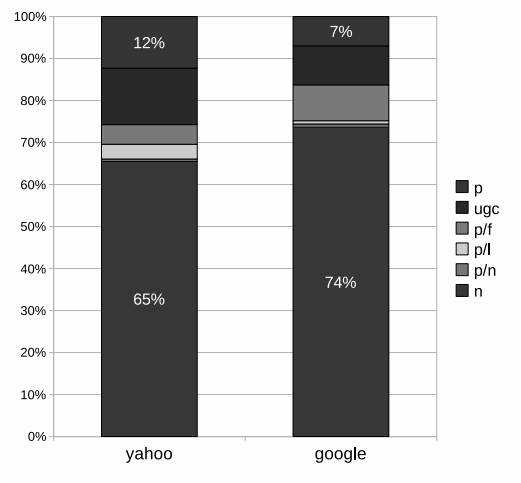
Vyhledávač	**Zastoupení typů stránek**						
	n	x	m	f	l	s	p
Yahoo	42,7	4,1	0,6	11,7	5,3	0,6	35,1
Google	57,4	0,8	0,8	11,6	6,2	8,5	14,7
Celkem	49,0	2,7	0,7	11,7	5,7	4,0	26,3

Z grafu na obr. 1, který obsahuje hodnocení domén druhého řádu, je patrné, že poměr čistě paralelních domén v celém hodnoceném souboru domén, příslušným k náhodně vybraným stránkám, je v případě vyhledávače Yahoo 12 %, v případě Google cca 7 %, celkově tato skupina webů tvoří 10 % za všech domén v našem testovacím vzorku. Je však potřeba si uvědomit, že i části domén hodnocených p/f, p/l a p/n jsou vhodnými kandidáty, protože jisté jejich části jsou též paralelní. Domény kompletně a částečně paralelní tvoří více než 19 % kontrolovaných domén.

Prezentované výsledky pokládáme za poměrně slibné, pro větší reprezentativnost by bylo ovšem vhodné ručně ohodnotit větší skupinu stránek a ověřit též anotátorskou shodu více anotátorů. Je možné, že další zlepšení úspěšnosti by přinesly sofistikovaněji kladené dotazy na vyhledávače nebo specializované metody hledání paralelních stránek pomocí webových robotů.

3 Stažení a očištění textů

Abychom získali co nejvíce paralelních textů, nalezené stránky pokládáme pouze za doporučené startovní body a pomocí standardní UNIXové utility `wget` s přepínačem `-m` stáhneme všechny dokumenty z dané domény dosažitelné rekurzivně přes odkazy z počáteční stránky. Tato metoda je časově i prostorově poměrně náročná. Větší domény mohou obsahovat množství archivního obsahu a jejich stahování, i když nejsou



Obrázek 1. Hodnocení domén druhého rádu podle vyhledávačů.

třeba vůbec paralelní, tak může celý proces významně zpomalit a zaplnit disk. Navíc je pak nutné tyto stránky zkoumat alespoň pomocí rozpoznávače jazyků pro zjištění, zda obsahují češtinu a angličtinu (případně jiné jazyky, o které se zajímáme). Na druhou stranu můžeme mít poměrně velkou jistotu, že pokud se v dané doméně nějaké paralelní stránky nacházejí, tato metoda na ně narazí a stáhne je.

Sofistikovanějšího postupu je využito v práci Saint-Amand (2008) – z nalezené stránky se pokusí dostat na její paralelní verzi (verzi v jiném jazyce, v originální práci se vyhledává více jazykových verzí najednou). Pokud se mu to povede, vytvoří z paralelních stránek stromové struktury na základě HTML a přiřadí tak k sobě i odkazy na další navzájem paralelní stránky. Pomocí zarovnaných odkazů je daná doména přezkoumána a nalezené paralelní stránky uloženy. Tento přístup ušetří mnoho místa a času, na druhou stranu je zde možnost, že paralelní stránky nebudou nalezeny, a nevyužijeme tak celý potenciál dané domény.

Metoda zvolená pro očištění textů od nadbytečných HTML značek byla jednoduchá – námi implementovaný odstraňovač tagů se ukázal jako nejvhodnější řešení, jelikož jiná řešení (např. pomocí PERLovského modulu `HTML::Parser`) často odstranila i text.

4 Identifikace jazyka

Stažené a očištěné texty je nutné rozdělit do skupin podle jazyků, ve kterých jsou tyto texty napsány. Fáze párování (viz oddíl 5) funguje jenom za předpokladu, že texty jsou správně faktorizovány podle jazyka.

Identifikaci jazyka lze provádět např. statistickými metodami založenými na porovnání frekvencí výskytu

Tabulka 4. Úspěšnost identifikátoru jazyka na anglických a českých textech o velikosti 200 znaků (100 pokusů) a 400 znaků (50 pokusů).

Jazyk	**Úspěšnost dle velikosti vzorku**	
	200 znaků	400 znaků
angličtina	95 %	98 %
čeština	97 %	100 %

znakových n-tic (n-gramů) v textu s frekvencemi v trénovacích datech, pro které je známo na jakém jazyku byly natrénovány (modely). Metoda popsaná v Saint-Amand (2008) trénuje tyto modely pro velkou množinu jazyků z dat internetové encyklopedie Wikipedia. Četnosti jsou zaznamenávány na plném textu očištěném od HTML, z důvodu jazykové nezávislosti se neprovádí žádná tokenizace (dělení na slova) ani další jiné úpravy textu, které by při identifikaci mohly pomoci. N-gramová metoda byla popsána i dřív v práci Dunning (1994), kde jsou navrženy také možnosti vylepšení prostřednictvím Markovových řetězců nebo Bayesovských rozhodovacích metod. Tento přístup však v sobě skrývá problém malé reprezentativnosti natrénovaného modelu. Kdyby se totiž jako česká trénovací data vybral text, který pojednává o Praze, mohlo by se stát, že mezi nejčastějšími trigramy bude „Pra“ nebo „ Pr“, přičemž obecně jsou v češtině nejčastější trigramy „ch “, „ ní“ nebo „ př“.

Zmíněný problém řeší větší množství trénovacích dat. Čím více dat, tím reprezentativnější je model. Náš systém však používá identifikátor jazyka, na jehož natrénování postačí menší objem textu. Princip této metody spočívá v tom, že četnosti n-gramů se nepočítají na celém textu, t.j. na všech slovních výskytech, ale jen na slovních typech. Text se převede na množinu použitých slov (slovních forem) a na této množině se spočte n-gramová statistika. Tento postup vyžaduje tokenizaci textu, což v našem případě (čeština a angličtina) není obtížná úloha. Pro jazyky jako čínština nebo japonština je použití popsaného postupu problematičtější.

Konkrétněji k modelům našeho identifikátoru jazyka:

- Byly natrénovány modely pro 21 evropských jazyků, včetně těch, které by mohly být eventuelně zaměněny s češtinou nebo angličtinou.
- Na natrénování byly použity různě dlouhé texty z Wikipedie, nejkratší obsahoval cca 50 tis. znaků.
- Text byl rozsekán na všech neabecedních znacích, čímž se jednak odstranila všechna čísla a speciální znaky, a jednak se tím text tokenizoval.
- Každé slovo bylo ohraničeno z obou stran závorkami, abychom odlišili prefixové a sufixové n-gramy, tj. n-gramy na počátku a konci slov.

– Výsledný model pro daný jazyk je seznam 100 nejčastějších trigramů na slovních typech, t.j. každé slovo započteno jenom jednou.

Samotná identifikace probíhá tak, že se stejným způsobem spočte model z textu, jehož jazyk se snažíme zjistit (testovaný text/model). Testovaný model je porovnán s každým z natrénovaných modelů a je vybrán takový jazyk, jehož model je nejpodobnější testovanému modelu. Pro míru podobnosti mohou být použity všechny metriky pro určení vzdálenosti mezi vektory (např. Euklidova). My jsme použili míru vyjádřenou vzorcem

$$p = 1 - \frac{|l_1 - t_1| + ... + |l_n - t_n|}{2} \quad (1)$$

kde $l_1...l_n$ jsou relativní frekvence jednotlivých trigramů v natrénovaném modelu a $t_1...t_n$ jsou relativní frekvence stejných trigramů v testovaném modelu.

Úspěšnost identifikátoru jsme testovali na českých a anglických textech z internetových novin. V prvním testu jsme texty rozdělili na 100 částí o 200 znacích, v druhém testu na 50 částí o 400 znacích. Výsledky ukázaly, že už na takto malých kusech textu náš identifikátor jazyka dosahuje úspěšnosti nad 95 % a s rostoucí velikostí textu roste úspěšnost identifikace. Podrobnější výsledky jsou v tabulce 4.

5 Párování dokumentů

V dosavadních pracích byly popsány různé metody párování dokumentů. Nejrychlejší jsou metody založené na metainformacích stažených stránek, např. párování na základě podobnosti URL adres (Resnik and Smith, 2003; Chen and Nie, 2000; Chen et al., 2003). Takovou metodou je i párování na základě porovnání délek dokumentů (v Resnik and Smith (2003) a Chen and Nie (2000)), které je však vhodnější pro párování vět.

Paralelní dokumenty mívají často stejnou nebo podobnou strukturu HTML značek, čehož využívají strukturní metody, např. využitím linearizovaného HTML v Resnik and Smith (2003) nebo porovnáním stromové struktury v Saint-Amand (2008).

Nejpomalejší, ale zároveň nejúčinnější jsou metody založené na obsahové podobnosti dokumentů. Pro příbuzné jazyky nebo jazyky s některými slovy příbuznými je možné použít metodu, která pomocí Levensteinovy vzdálenosti určuje podobnost textů nebo záchytných slov (použito v Resnik and Smith (2003)). Záchytná slova jsou zadána slovníkem, mít kompletní slovník však není potřeba. Práce Chen and Nie (2000) také používá tuto metodu, jelikož však zkoumá velmi odlišné jazyky (angličtina, čínština), je tato metoda ekvivalentní strukturní metodě linearizace HTML.

Konečně musíme zmínit i metody, využívající slovník při měření obsahové podobnosti. Tyto metody mohou text reprezentovat jako vektor četností jednotlivých typů slov a porovnávat podobnost vektorů (Yang and Li, 2003) nebo párovat kandidátské texty slova na slovo a podobnost měřit mírou spárování (Resnik and Smith, 2003). Práce Yang and Li (2003) používá i druhou metodu a navíc řeší specifika čínštiny.

Naše metody pro nalezení paralelních textů v množině českých a anglických textů byly navrženy nezávisle a implementovány v aplikaci WebCorpus, podrobněji viz Klempová (2009). Využíváme kombinaci jednoduché strukturní metody a slovníkové obsahové metody. Algoritmus lze rozdělit do několika základních kroků:

1. Zmapování struktury souboru (pozice prázdných a neprázdných řádků).
2. Segmentace a tokenizace souborů (na každém řádku souboru právě jedna věta, slova a interpunkční znaménka jsou oddělena mezerou).
3. Výběr n nejčastějších slov pro každý soubor.
4. Přeložení anglických slov pomocí slovníku.
5. Vyhledání základních tvarů pro česká slova.
6. Párování souborů jednou z několika implementovaných metod.

Kroky 1 až 5 provádíme dávkově předem, viz následující oddíl. Jádro párování (krok 6) pak popisuje oddíl 5.2.

5.1 Zpracování souborů

Prvním krokem algoritmu je zmapování struktury souboru. Výsledkem této operace je binární vektor, kde na i-té pozici je 0, právě když je i-tý řádek souboru prázdný, a 1 v opačném případě (viz též obrázek 2). Motivací k této metodě bylo pozorování, že stránky, které jsou si navzájem překladem, jsou velmi často stejně členěné. Nespornou výhodou této možnosti porovnání souborů je její časová nenáročnost.

Následujícím krokem předzpracování je segmentace a tokenizace textu. V této části dochází k restrukturalizaci textu tak, aby na každém řádku byla právě jedna věta a všechna slova a interpunkční znaménka byla oddělena mezerou. K tokenizaci textu byl použit String::Tokenizer a následná segmentace probíhá pomocí regulárních výrazů. Největším úskalím je v této úloze rozeznat, kdy je tečka použita jako součást zkratky a kdy ukončuje větu.

Následující zpracování souborů se liší podle toho, zda se jedná o anglický či český text. V případě anglických souborů se nejprve nalezne n nejčastějších slov (default je 80). Tyto výrazy se přeloží ve slovníku a odfiltrují se slova, která příliš nevypovídají o významu daného textu (např. některá zájmena či předložky).

V případě českých souborů se také nejprve nalezne n nejčastějších slov, místo překladů však ke slovům jen nalezneme základní tvary pro odstranění morfologické bohatosti. K tomu používáme námi implementovaný modul, který pracuje s morfologickým slovníkem z práce Hajič (2001)[3].

5.2 Hodnocení dvojic textů

Hledání paralelních textů probíhá vždy samostatně v rámci každého staženého webu.

Pro ohodnocení vzájemné paralelnosti dvojic souborů si program vytvoří dvě matice. V první z nich, nazvěme ji matice S, je vyjádřena podobnost souborů na základě binární charakteristiky struktury staženého souboru. Technicky se jedná o porovnání dvou řetězců pomocí editační (Levenshteinovy) vzdálenosti implementované v modulu `String::Similarity`, který vrací hodnotu v intervalu <0,1> vyjadřující míru jejich podobnosti. Ilustrace je na obrázku 2.

mountain hotel JARMILKA - animation services - Liberec

mountain hotel JARMILKA

home
accommodation
restaurant

Liberec

The town of Liberec
The town Liberec is considered to be a significant administrative, cultural, industrial and sports centre not only of the

Jizera Mountains but also of the whole region. It lies in a valley between the hills of the Jizera Mountains and the Jested Ridge.

The ZOO Liberec

The ZOO Liberec

1100001000111011110111000100001

Obrázek 2. Ukázkový text stránky po odstranění HTML značek a jeho binární charakteristika struktury (znak 0 na místě prázdného řádku, znak 1 na místě neprázdného řádku). Pokud bychom k tomuto anglickému souboru měli paralelní český s touto binární charakteristikou 1100001100111001110111001100000101 (odlišné znaky jsou podtržené), vyšla by pomocí String::Similarity míra podobnosti 0.89.

Na základě nejčastějších slov souboru je vytvořena matice L. Porovnání probíhá nejprve vždy tak, že pro

[3] `http://ufal.mff.cuni.cz/pdt/ /Morphology_and_Tagging/Morphology/index.html`

Tabulka 5. Ukázka výpočtu indexu D a H pro daná slova a jejich četnosti.

Slovo	Četnost	Index D	Index H
27 liberec	16	27	27
26 město	8	26	26
25 rok	6	23	25
24 centr	6	23	25
23 století	6	23	25
22 jenž	5	21	22
21 zahrada	5	21	22
20 hora	4	19	20
19 m2	4	19	20
18 ZOO	3	13	18
17 animační	3	13	18
...			
14 aquapark	3	13	18
13 služba	3	13	18
12 radnice	2	3	12
11 jizerský	2	3	12
...			
4 hodně	2	3	12
3 daleko	2	3	12
2 lázeň	1	1	2
1 kulturní	1	1	2

každé slovo z n nejčastějších slov českého souboru hledáme ekvivalent mezi překlady n nejčastějších slov anglického souboru. Vedle n nejčastějších slov si také pamatujeme jejich četnosti. Nabízí se tedy otázka, jak s informací o četnosti při hodnocení dvojice textů naložit. Označme c četnost slova v českém souboru a e četnost shodného slova v anglickém souboru. Potom existuje hned několik možností, jak bodové ohodnocení určit:

1. První a nejjednodušší možností je četnosti ignorovat a za každé shodné slovo ohodnotit dvojici souborů po jednom bodě.
2. Druhou variantou je poměr součtu četností a dvojnásobku jejich rozdílu:

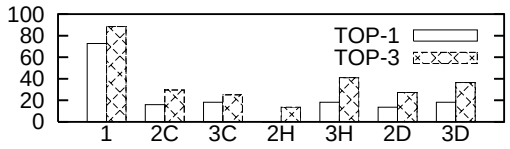
$$\frac{c+e}{2*|c-e+1|} \tag{2}$$

   Tímto způsobem se nejvíce ocení slova, která jsou velmi častá v obou textech.
3. Dále se samozřejmě nabízí klasický aritmetický průměr četností:

$$\frac{c+e}{2} \tag{3}$$

Další variantou je dosadit na místo skutečné četnosti (označme C) námi navržené indexy D a H, které vyjadřují dolní a horní index slova při uspořádání dle četnosti, příklad viz tabulka 5. Motivací k výpočtu indexů D a H nám byla snaha co nejvíce rozlišit bodové hodnocení dvojice textů v případech shody v hodně častých a naopak málo častých slovech.

K tomu, abychom mohli vybrat nejefektivnější z popsaných metod, bylo nutné provést experiment. Sadu testovacích dat tvořilo 44 dvojic českých a anglických textů, které si jsou překladem a byly staženy ze tří



Obrázek 3. Procento správně spárovaných dvojic dokumentů při použití jednotlivých metrik.

menších webů. Všechny soubory prošly analýzou, která našla n nejčastějších slov včetně jejich četností. Poté byly všechny české a anglické dvojice porovnány pomocí všech implementovaných metrik. Cílem bylo nalézt tu z nich, která nejlépe ohodnocuje dvojice textů, které si jsou překladem. Nejprve jsme tedy sledovali, v kolika případech bude správná dvojice danou metrikou hodnocena nejvýše (TOP-1). Poté jsme uvažovali, zda je správná dvojice alespoň mezi třemi nejlépe hodnocenými páry (TOP-3). Výsledky obou testů zachycuje obrázek 3. Číslo určuje testovanou metriku dle výše uvedeného seznamu, písmeno dosazenou četnost, resp. index, například metrika 3C znamená výpočet aritmetického průměru z původních četností.

Z grafu jasně plyne, že nejlépe hodnotila správné dvojice textů první nejjednodušší metrika, která přidělí dvojici souborů jeden bod za každé shodné slovo bez ohledu na jeho četnost. Tato metrika měla úspěšnost 32 ze 44 (tj. 72.7 %) v prvním testu a 39 ze 44 (tj. 88.6 %) ve druhém testu. Jak je patrné z grafu, zbylé metriky nedosáhly ani na padesátiprocentní úspěšnost. To může být způsobeno hned několika důvody:

- Překlady většiny webových stránek nejsou příliš přesné a doslovné.
- Může dojít k nepřesnosti v průběhu překládání anglických slov např. pokud hledaný výraz slovník vůbec neobsahuje a tím pádem se nepřeloží nebo v množině překladů není ten tvar, který potřebujeme.
- Také zpracování českých výrazů není vždy stoprocentní a může se stát, že nejsou úspěšně nalezeny všechny základní tvary. Navíc v případě některých slov nelze jejich základní tvar bez ohledu na kontext určit jednoznačně, např. slovo *má* může být tvarem slova *mít* ale také *moje* nebo *můj*. Aplikace v tomto případě pracuje pouze s prvním nalezeným základním tvarem.

5.3 Spárování souborů

Pro finální párování souborů jsme se rozhodli implementovat celkem sedm metod, které různě kombinují výše popsané matice S a L. Navíc se opíráme o výstup nástroje hunalign (Varga et al., 2005), který zkoumanou dvojici textů zkusí spárovat na úrovni vět a určí kvalitu přiřazení. Kvalita párování po větách je samozřejmě velmi dobrým ukazatelem paralelnosti dané dvojice dokumentů, je však ve srovnání s předešlými metodami výrazně výpočetně náročnější.

Jednotlivé kombinace metod jsou pojmenované takto: *str_lex_multi, str_lex_fix, str_lex_full, str_fix, lex_fix, str_full, lex_full.* A zde je základní návod, jak se v nich orientovat:

Pokud je v názvu metody obsaženo **str**, znamená to, že se pracuje s maticí S, která hodnotí dvojice souborů na základě struktury textu ihned po odstranění HTML značek.

Pokud je v názvu metody slovo **lex**, bere se v úvahu matice L, která hodnotí dvojice souborů na základě shodného výskytu významových slov.

pohledu a hledá takové přiřazení, které je dobře hodnocené oběma metodami. Pro každý český soubor se berou v úvahu tři nejlépe hodnocené anglické soubory z obou porovnání. Označme 1.l nejlépe hodnocený anglický soubor v daném řádku matice L, 1.s nejlépe hodnocený anglický soubor v daném řádku matice S atd. Potom se k danému českému souboru hledá nejvhodnější anglický tak, aby byl mezi třemi nejlépe hodnocenými pomocí obou metod. Pokud se takový najde, je s daným českým souborem zpracován hunalignem, který vrátí kvalitu přiřazení. Parametrem lze určit nejnižší možnou hodnotu, aby se mohla dvojice prohlásit za paralelní (defaultně je nastavena na -1). Shodnost anglických souborů obdržených metodou S a L se testuje v tomto sledu:

1) 1.s – 1.l	4) 2.s – 2.l	7) 3.s – 2.l
2) 2.s – 1.l	5) 3.s – 1.l	8) 2.s – 3.l
3) 1.s – 2.l	6) 1.s – 3.l	9) 3.s – 3.l

Dá se tedy říci, že lehce favorizována je metoda L (vysoce hodnocené soubory metodou L se testují dříve), která by z logiky věci měla mít skutečně větší vypovídací hodnotu.

Slovo **multi** v názvu metody znamená, že pro každý český soubor najdu první vyhovující anglický a ten mu přiřadím. První vyhovující anglický soubor je takový, který první ve výše zmíněném pořadí vytváří s daným českým souborem paralelní korpus o minimálně hraniční kvalitě. V této variantě se může stát, že jeden anglický soubor bude přiřazen k více českým souborům.

Pokud je v názvu metody obsaženo **fix**, znamená to, že pro již přiřazené anglické soubory si pamatuji, s jakou kvalitou byly přiřazeny. Pokud je k danému českému souboru nalezen anglický soubor splňující základní podmínky (tj. je mezi třemi nejlepšími v metodě S nebo v metodě L nebo v obou a spárování je nad hraniční kvalitou), ale přiřazení by dosahovalo nižší kvality, než s kterou byl již anglický soubor přiřazen,

Tabulka 6. Ukázka matice S. Tučně jsou vyznačeny tři nejlepší páry každého řádku.

	EN1	EN2	EN3	EN4	EN5	EN6
CZ1	**.99**	**.98**	**.97**	.81	.89	.80
CZ2	**.97**	**.98**	.81	.69	**.94**	.72
CZ3	**.96**	.90	**.98**	**.99**	.67	.78
CZ4	.90	**.93**	**.95**	**.98**	.81	.80
CZ5	.90	**.98**	.82	.87	**.97**	**.96**
CZ6	.76	**.93**	.87	.90	**.95**	**.98**

Tabulka 7. Ukázka matice L.

	EN1	EN2	EN3	EN4	EN5	EN6
CZ1	**18**	**20**	4	**15**	5	9
CZ2	**15**	**17**	3	3	**13**	4
CZ3	2	5	**18**	**19**	**10**	4
CZ4	**12**	**13**	6	**22**	3	6
CZ5	9	**18**	4	7	**14**	**17**
CZ6	5	**14**	4	8	**18**	**17**

potom se toto přiřazení neprovede a hledání pokračuje dál. Naopak, pokud nastane situace, že by anglický soubor mohl být přiřazen s vyšší kvalitou, toto přiřazení se provede a v předešlém běhu programu vytvořená dvojice souborů se rozváže. Pro český soubor z rozvázané dvojice se hledá vhodný anglický protějšek znovu. V této metodě je tedy každý soubor přiřazen nejvýše jednou.

Poslední možnou zkratkou obsaženou v názvu metody je **full**. V tomto případě se postupuje podobně jako v tom předešlém, ale je snahou nacházet opravdu nejlepší přiřazení. To znamená, že pro každý český soubor se otestují všechny anglické soubory, které přicházejí v úvahu a z nich se vybere ten nejlepší (hledání je ovšem hladové a nezaručuje nejlepší párování ze všech možných). Tato varianta je tedy výpočetně nejnáročnější, protože se testuje největší počet přiřazení, měla by však vést ke kvalitnějšímu výsledku. Také v této metodě je každý soubor přiřazen nejvýše jednou.

Pro lepší pochopení v obrázku 4 uvádíme práci metody str_lex_fix nad množinou českých textů CZ1 až CZ6 a množinou anglických textů EN1 až EN6, s maticemi S a L uvedenými v tabulkách 6 a 7.

5.4 Složitost a vyhodnocení

Označme n_c počet českých a n_e počet anglických souborů ve zpracovávaném webu. Potom časová náročnost vytvoření matic L a S je $O(n_c * n_e)$. Časová náročnost hledání paralelních textů je $O(n_c * 3h)$, kde h je časová náročnost spárování dvou souborů pomocí nástroje hunalign (každý český soubor zkoušíme spárovat s maximálně třemi anglickými v každé z implementovaných metod).

Všechny implementované metody jsme podrobili testu. Experiment proběhl na množině 74 dvojic čes-

1. stav na zásobníku: (CZ1, CZ2, CZ3, CZ4, CZ5, CZ26)
   - na vrcholu zásobníku je prvek CZ1
   - tři nejlepší kandidáti matice S: EN1, EN2, EN3
   - tři nejlepší kandidáti matice L: EN2, EN1, EN4
   - jako první zkusíme spárovat CZ1 s EN2, nechť je kvalita spárování h(CZ1,EN2)=0.7
   - soubor EN2 zatím nebyl spárován, tedy dvojice CZ1,EN2 vyhovuje
2. stav na zásobníku: (CZ2, CZ3, CZ4, CZ5, CZ6)
   - na vrcholu zásobníku je prvek CZ2
   - tři nejlepší kandidáti matice S: EN2, EN1, EN5
   - tři nejlepší kandidáti matice L: EN2, EN1, EN5
   - h(CZ2,EN2)=0.9 > 0.7
   - zruší se dvojice CZ1,EN2 a vytvoří se nová dvojice CZ2,EN2
3. stav na zásobníku: (CZ1, CZ3, CZ4, CZ5, CZ6)
   - S: EN1, EN2, EN3; L: EN2, EN1, EN4
   - h(CZ1,EN2)=0.7 < 0.9, h(CZ1,EN1)=0.8
   - vytvoří se nová dvojice CZ1,EN1
4. stav na zásobníku: (CZ3, CZ4, CZ5, CZ6)
   - S: EN4, EN3, EN1; L: EN4, EN3, EN5
   - h(CZ3,EN4)=0.5 ⇒ nová dvojice CZ3,EN4
5. stav na zásobníku: (CZ4, CZ5, CZ6)
   - S: EN4, EN3, EN2; L: EN4, EN2, EN1
   - h(CZ4,EN4)=0.6 > 0.5 ⇒ nová dvojice CZ4,EN4
6. stav na zásobníku: (CZ3, CZ5, CZ6)
   - S: EN4, EN3, EN1; L: EN4, EN3, EN5
   - h(CZ3,EN4)=0.5 < 0.6
   - h(CZ3,EN3)=0.8 ⇒ nová dvojice CZ3,EN3
7. stav na zásobníku: (CZ5, CZ6)
   - S: EN2, EN5, EN6; L: EN2, EN6, EN5
   - h(CZ5,EN2)=0.3 < 0.9
   - h(CZ5,EN6)=0.6 ⇒ nová dvojice CZ5,EN6
8. stav na zásobníku: (CZ6)
   - S: EN6, EN5, EN2; L: EN5, EN6, EN2
   - h(CZ6,EN5)=0.6 ⇒ nová dvojice CZ6,EN5

Obrázek 4. Postup metody **str_lex_fix** nad maticí S a L z tabulek 6 a 7.

kých a anglických textů (celkem tedy 148 souborů). Postupně jsme zkusili soubory spárovat pomocí všech sedmi metod. Ke zpracování výsledků jsme se rozhodli použít metodu precision-recall. V tomto konkrétním případě true positive=správně vybraná dvojice textů, true negative=správně nevybraná dvojice, false positive=špatně vybraná dvojice a konečně false negative=špatně nevybraná dvojice (metody vybírají ty dvojice, které považují za vzájemné překlady). Jelikož na vstupu má metoda 74 českých a 74 anglických souborů, existuje celkem 74 x 74 = 5 476 dvojic souborů. Z toho 74 dvojic tvoří překlady a zbylých 5 402 dvojic k sobě nepatří.

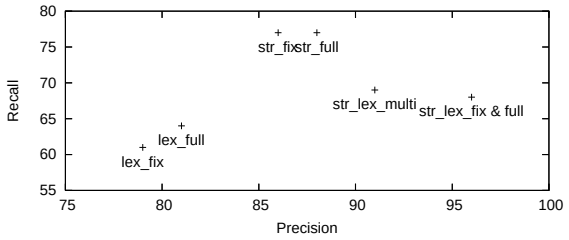
První testovanou metodou byla str_lex_multi. Tato metoda našla 51 správných dvojic a 5 špatných. Hledané hodnoty precision a recall se tedy spočítají takto:

$$precision = \frac{tp}{tp + fp} = \frac{51}{51 + 5} = 0.91 \qquad (4)$$

$$recall = \frac{tp}{tp + fn} = \frac{51}{51 + 23} = 0.69 \qquad (5)$$

Stejný výpočet probíhá i s výsledky zbylých metod. Obdržené hodnoty zobrazuje graf v obrázku 5.

Na základě grafu lze udělat tato pozorování:



Obrázek 5. Výsledek testu metod párování souborů

- Kombinace porovnání souborů na základě struktury a nejčastějších slov přinesla nejpřesnější výsledky, ovšem ne nejvyšší recall.
- Dá se říci, že metody full dosahují v průměru lepších výsledků než metody fix.
- Z našeho testu vyšla metoda str lépe než metoda lex. Tento výsledek se ale odvíjí od charakteru vstupních webů, a proto ho dle našeho názoru nelze příliš zobecňovat.
- Metoda str může být zajímavá v případě, kdy klademe důraz na časovou optimalizaci úlohy, neboť je v porovnání s metodou lex výrazně rychlejší.

6 Celkové vyhodnocení

Na závěr jsme se rozhodli otestovat, nakolik úspěšné bude nalezení vhodných párů dokumentů na náhodně vybraných 5 odkazech s nejvyšším hodnocením – „p“ pro stránku i doménu. Počáteční odhodlání testovat několik náhodně vybraných domén z každé skupiny hodnocení jsme museli opustit kvůli časové i prostorové náročností stahovaní celých domén. Nechali jsme si tedy stáhnout 5 náhodně vybraných domén a na nich pustit párovací algoritmus s výchozími možnostmi nastavení párování, jak je uvedeno výše.

Na daném testovacím vzorku bylo správně spárováno 69 stránek ze 75 nalezených párů, co představuje přesnost (precision) 92 %. Malý korpus vytvořený z těchto 5 domén obsahuje 2 400 paralelních vět, 51 700 českých a 55 200 anglických slov. Jak se tedy ukázalo podle kvality získaných URL a úspěšnosti párování, výsledný korpus je vytvořen ze zhruba 10 % nalezených URL, přičemž každá z nich přispěje průměrně 12 paralelními stránkami. Je však nutno podotknout, že tato statistika byla vytvořena jen na malých datech, a proto ji není možno považovat za zcela spolehlivou. Rozsáhlejší testování získaných dat bude předmětem dalších pokusů.

7 Závěr

Práce shrnuje metody automatického sběru paralelních korpusů z webu a testuje je na páru jazyků čeština-angličtina. Navržené metody automatického rozpoznání jazyků a párování stažených stránek dosahují uspokojivých výsledků, i když další možnosti zlepšení lze uvažovat, např. s použitím metod strojového učení.

Ukazuje se, že pro získaní paralelního korpusu je dnes limitujícím faktorem spíše schopnost nalézt dostatečné množství paralelních webů. Vyhodnocení námi použitých metod ukázalo, že přibližně deset procent nalezených URL adres je použitelných pro další zpracování. Dalším prohledáním domén získaných z URL se dá i při relativně malém množství počátečních adres nalézt velké množství paralelních dokumentů. Problémem však často zůstávají vestavěná omezení ve webových vyhledávačích, o něž se naše metoda opírá. V další práci se tedy zaměříme zejména na vylepšení vyhledávání paralelních webů.

Literatura

Ondřej Bojar, Miroslav Janíček, Zdeněk Žabokrtský, Pavel Češka, and Peter Beňa. CzEng 0.7: Parallel Corpus with Community-Supplied Translations. In *Proc. of LREC*, Marrakech, Morocco, May 2008. ELRA.

Jiang Chen and Jian-Yun Nie. Automatic construction of parallel English-Chinese corpus for cross-language information retrieval. In *Proc. of ANLP*, pages 21–28, Seattle, Washington, 2000.

Jisong Chen, Rowena Chau, and Chung-Hsing Yeh. Discovering Parallel Text from the World Wide Web. In *The Australasian Workshop on Data Mining and Web Intelligence (DMWI-2004)*, Dunedin, New Zealand, 2003.

Ted Dunning. Statistical identification of language. Computing Research Laboratory Technical Memo MCCS 94-273, New Mexico State University, Las Cruces, New Mexico, 1994.

Jan Hajič. *Disambiguation of Rich Inflection - Computational Morphology of Czech*, volume I. Prague Karolinum, Charles University Press, 2001. 334 pp.

Hana Klempová. Nástroj pro sběr paralelních textů z webu. Bakalářská práce, Matematicko-fyzikální faktulka, Univerzita Karlova v Praze, 2009.

Philip Resnik and Noah A. Smith. The Web as a Parallel Corpus. In *Computational Linguistics, Volume 29, Issue 3*, pages 349–380, September 2003.

Hervé Saint-Amand. Gathering a Parallel Corpus from the Web. Master's thesis, Universität des Saarlandes, Saarbrücken, Germany, September 2008.

Dániel Varga, László Németh, Péter Halácsy, András Kornai, Viktor Trón, and Viktor Nagy. Parallel corpora for medium density languages. In *Proc. of RANLP*, pages 590–596, Borovets, Bulgaria, 2005.

Christopher C. Yang and Kar Wing Li. Automatic construction of English/Chinese parallel corpora. In *Journal of the American Society for Information Science and Technology*, pages 730–742, 2003.