VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

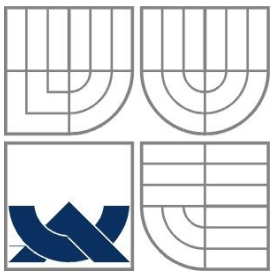
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

STATISTICKÝ STROJOVÝ PŘEKLAD MEZI ČEŠTINOU A SLOVENŠTINOU

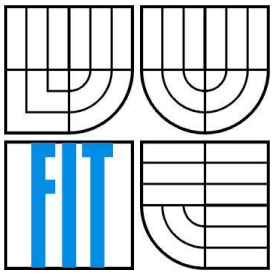
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE LUKÁŠ ASTALOŠ
AUTHOR

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

STATISTICKÝ STROJOVÝ PŘEKLAD MEZI ČEŠTINOU A SLOVENŠTINOU

CZECH-SLOVAK STATISTICAL MACHINE TRANSLATION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

LUKÁŠ ASTALOŠ

VEDOUCÍ PRÁCE
SUPERVISOR

doc. RNDr. PAVEL SMRŽ, Ph.D.

BRNO 2013

Abstrakt

Cílem téhle práce je navrhnout, implementovat a vyhodnotit úspěšnost vytvořeného systému pro překlad českých textů do slovenštiny. Popisuje teoretické základy statistického strojového překladu a pak samotnou fázi návrhu a vývoje systému. Zvolenou strategií bylo použít jeden rozsáhlý paralelní korpus v kombinaci s jazykovým modelem poskytovaným místním jazykovědným ústavem pro vytvoření překladového modelu založeném na frázích. Pro tenhle účel byl použit nástroj Moses. Experimentálně byl otestován také slovníkový překlad neznámých slov použitím stemmování. Úspěšnost systému byla vyhodnocena metrikou BLEU, přičemž dosažené výsledky byly porovnatelné s jinými systémy.

Abstract

The aim of this thesis is to design, implement and evaluate the translation system capable of translating texts from Czech to Slovak language. It describes theoretical foundations of statistical machine translation and then the phase of design and development of system. The chosen strategy was to build phrase-based translation model using one large parallel corpus in combination with language model from local institute of linguistics. The statistical machine translation Moses was used to achieve this goal. The vocabulary translation of unknown words using stemming was proposed and tested. Precision of build system was evaluated with BLEU score and it achieved comparable results with other systems.

Klíčová slova

statistický strojový překlad, zarovnání slov, IBM modely, grow-diag-final-and, fráza, dekodér, jazykový model, paralelný korpus, BLEU, Google Translate, Česílko, Moses, GIZA, slovníkový překlad, MERT

Keywords

statistical machine translation, word alignment, IBM models, grow-diag-final-and, phrase, decoder, language model, parallel corpus, BLEU, Google Translate, Česílko, Moses, GIZA, vocabulary translation, MERT

Citace

Astaloš Lukáš: Statistický strojový překlad mezi češtinou a slovenštinou, bakalářská práce, Brno, FIT VUT v Brně, 2013

Statistický strojový překlad mezi češtinou a slovenštinou

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana doc. RNDr. Pavla Smrže, Ph.D.
Další informace mi poskytl Ing. Jan Kouřil.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Lukáš Astaloš
16. dubna 2013

Poděkování

Ďakujem vedúcemu mojej bakalárskej práce za usmerňovanie a odbornú pomoc pri jej tvorbe.

Obsah

1 Úvod

Možnosti komunikácie s okolitým svetom sa neustále rozširujú, a tak vzniká potreba dorozumievania sa v cudzích jazykoch. Ľudský preklad je však často nedostupný, pomalý a nákladný. Bolo preto prirodzené, že sa začali hľadať spôsoby, ako nahradiť človeka – tlmočníka strojom. V druhej polovici 20. storočia sa tak začína rozvíjať odvetvie strojového prekladu. Ako však naučiť stroj „myslieť“, aby zvládol preložiť text rovnako dobre ako človek? Jednou z možností je využitie štatistického strojového prekladu, ktorý sa „naučí“ prekladať vďaka analýze veľkého množstva rovnakých textov v dvoch jazykoch.

Úlohou tejto práce je navrhnúť, implementovať a vyhodnotiť štatistický prekladový systém, ktorý by dokázal prekladať české texty do slovenčiny. Pre hlbšie pochopenie problematiky bolo nutné naštudovať teoretické pozadie strojového prekladu. To je náplňou kapitoly 2. Predstavené sú problémy, ktoré preklad sprevádzajú spolu s princípmi ich riešenia. Dôraz je kladený prevažne na tie oblasti, ktoré boli v ďalšej práci prakticky využité. Princípy sú vysvetlené pre jazyky všeobecne, bez dôrazu na konkrétny jazyk.

V kapitole 3 už začína hovoriť o konkrétnej dvojici jazykov čeština a slovenčina. Systém štatistického strojového prekladu potrebuje v prvej fáze vývoja veľké množstvo dvojjazyčných textov. Predstavené sú zdroje, z ktorých boli tieto texty čerpané a použité nástroje pri vývoji.

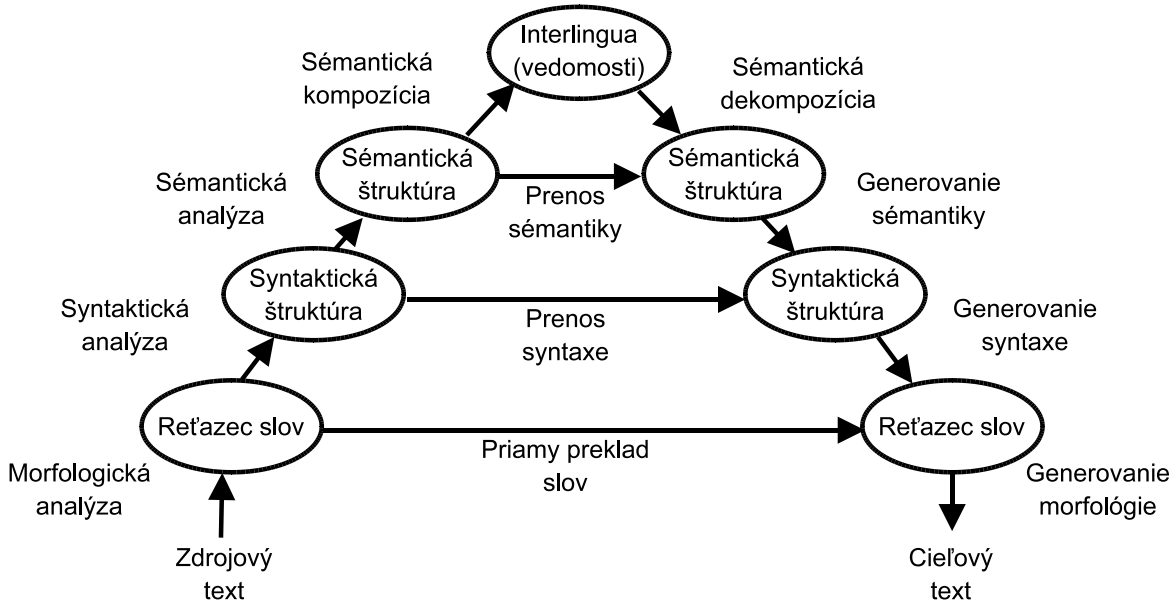
Ďalšiu etapu vývoja popisuje kapitola 4. Obsahuje opis prípravy a spracovania dvojjazyčných textov, vytvorenie prekladového systému a jeho implementované vylepšenia.

Posledným krokom bolo vyhodnotenie vytvoreného systému, ktoré tvorí obsah kapitoly 5. Zároveň popisuje vykonané experimenty, z ktorých vzišli niektoré zaujímavé otázky, hypotézy.

V závere je zrekapitulovaná odvedená práca a nastolené námety pre ďalší vývoj v budúcnosti.

2 Teoretické pozadie strojového prekladu

Strojový preklad je odvetvie výpočtovej lingvistiky, ktoré sa zaoberá automatickým prekladom textu alebo reči z jedného prirodzeného (ľudského) jazyka do iného prirodzeného jazyka bez zásahu človeka. Od obdobia 50. rokov 20. storočia [1], kedy sa s rozvojom počítačov začína písať aj história strojového prekladu, sa vyvinulo niekoľko rôznych prístupov k tejto problematike. Ich vývoj a úrovne, na ktorých pracujú znázorňuje tzv. Vauquoisov[1] trojuholník na obrázku 2.1.



Obrázok 2.1: Vauquoisov trojuholník [2]

Najnižšiu úroveň trojuholníka predstavuje najtriviálnejší prístup - priamy preklad „slovo na slovo“. Pri preklade sa využívajú prekladové slovníky a obvykle je nevyhnutná aj morfologická analýza. Jedným z problémov je, že slovo v zdrojovom jazyku môže mať viacero prekladov v cieľovom jazyku. Ďalším problémom je rozdielny slovosled jazykov [3]. Tento prístup nachádza uplatnenie pri preklade medzi veľmi blízkymi jazykmi, ako sú napr. slovanské jazyky.

Pri druhom prístupe spracujeme zdrojový text do podoby syntaktického stromu cieľového jazyka použitím vhodných pravidiel. Z tohto stromu sme už schopní vygenerovať syntakticky správny preklad, avšak častokrát s pozmenenou sémantikou (významom) [3].

Pri prenose sémantiky berieme pri preklade do úvahy aj pôvodný význam zdrojového textu. Ten zostane zachovaný, avšak v niektorých prípadoch môže byť nejasný až nezrozumiteľný kvôli rozdielnej vetnej skladbe.

Interlingua[2] je abstraktná reprezentácia znalostí o jazykoch, ktorá je nezávislá na konkrétnom jazyku. Jej hlavnou výhodou je, že nie je potrebné vytvárať prekladové modely medzi všetkými pármi jazykov. Namiesto toho stačí vytvoriť prekladový model medzi konkrétnym jazykom a interlinguou. Napriek tomu je však praktická realizácia tohto prístupu zatiaľ náročná [3].

Podľa prístupu, ktorý je využitý, kategorizujeme strojový preklad nasledovne [4]:

[1] Bernard Vauquois - profesor na univerzite v Grenobli, jeden z pionierov strojového prekladu
[2] Interlingua (interlingva) je tiež označenie pre medzinárodný pomocný jazyk ako napr. aj Esperanto

- Preklad založený na pravidlách – známy tiež ako preklad založený na znalostiach, príp. klasický prístup; je založený na lingvistických informáciách získaných z bilingválnych slovníkov a gramatík pokrývajúcich základnú sémantiku, morfológiu a syntax jazyka; patrí sem slovníkový preklad, preklad využívajúci morfologickú a syntaktickú analýzu a preklad využívajúci interlingvu [5].
- Preklad založený na príkladoch – je založený na myšlienke dekompozície prekladaného textu na frázy, preklade týchto fráz a ich zloženia do výsledného prekladu; ako báza znalostí slúži paralelný korpus[3] v ktorom sa vyhľadávajú analógie s prekladanými frázami [6].
- Štatistický preklad – využíva štatistické metódy odvodené z bilingválneho textového korpusu; tento prístup predstavuje ťažisko mojej práce a bude podrobnejšie vysvetlený v nasledujúcej podkapitole.
- Hybridný preklad – kombinovaný, využíva silné stránky štatistického prekladu a prekladu založenom na pravidlách.

2.1 Štatistický strojový preklad

Štatistický strojový preklad je v súčasnosti najviac rozvíjaným odvetvím strojového prekladu. Ako prvý jeho myšlienku predstavil Warren Weaver v roku 1949 [7]. Myšlienka bola znovu uvedená a rozvinutá výskumnými pracovníkmi z Výskumného centra Thomasa J. Watsona firmy IBM [8].
Na rozdiel od vyššie spomenutých prístupov sa nezameriava na proces prekladu, ale na výsledok. Je založený na zbere a analýze dát z paralelného korpusu.

Nasledujúce odseky čerpajú prevažne z [8] a [9]. Pri preklade často nastáva situácia, kedy je možný preklad jednoduchý, ale nepokryje celý pôvodný význam frázy v zdrojovom jazyku. Ak chceme pokryť celý význam, použijeme opisný spôsob, čím sa stáva preklad zložitejším a neprirodzenejším. Musíme preto zvoliť kompromis. O to isté ide aj v štatistickom strojovom preklade – vybudovať pravdepodobnostné modely vernosti a plynulosti prekladu a ich kombináciou nájsť podľa možnosti najpravdepodobnejší (nie nutne najlepší) preklad.

Predpokladajme, že prekladáme reťazec e v zdrojom jazyku. Pri štatistickom preklade považujeme každý reťazec f za možný preklad pre e. Číslo $P(f|e)$ vyjadruje pravdepodobnosť, že prekladač pre dané e vygeneruje preklad f. Úlohou prekladača je nájsť taký reťazec e, ktorý by mal aj ľudský prekladateľ na mysli, keď by vyprodukoval ako preklad reťazec f. Vyberáme preto reťazec $\hat{e}$, pre ktorý je $P(e|f)$ najvyššia. Použitím Bayesovho pravidla môžeme napísať

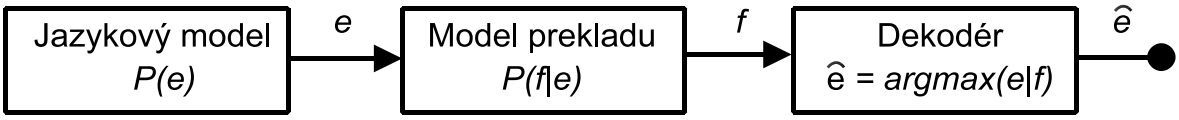
$$\hat{e} = argmax_e P(e|f)$$
$$= argmax_e \frac{P(f|e)P(e)}{P(f)}$$

Menovateľ $P(f)$ je nezávislý na e a teda je konštantný pre všetky e, vďaka čomu ho môžeme zanedbať a dostávame konečný tvar rovnice.

$$\hat{e} = argmax_e P(f|e)P(e) \quad (2.1)$$

Z rovnice vyplýva, že potrebujeme dva komponenty: model prekladu $P(f|e)$ a jazykový model $P(e)$. Tretím komponentom, ktorý potrebujeme je dekodér, ktorý z daného f vyprodukuje najpravdepodobnejšie e. Tieto tri komponenty tvoria tzv. kanál so šumom (noisy channel), ktorý modeluje činnosť štatistického prekladu.

[3] Paralelný korpus – súbor dvojjazyčných, príp. viacjazyčných textov, v ktorom sú zarovnané korešpondujúce vety, príp. iné celky zdrojového a cieľového jazyka



Obrázok 2.2: Kanál so šumom [3]

2.2 Zarovnanie viet

Základným predpokladom štatistického predpokladu je existencia alebo vybudovanie paralelného korpusu. Tejto téme sa venuje kapitola XX. Texty v paralelnom korpuse sú rozdelené do segmentov, ktoré si navzájom korešpondujú, t.j. sú si navzájom prekladmi. Najčastejšou segmentačnou jednotkou býva veta. Zarovnanie medzi párom viet môžeme podľa [8] definovať ako objekt indikujúci pre každé slovo vo vete v zdrojovom jazyku práve to slovo (príp. slová) v cieľovom jazyku, z ktorého vzniklo.

V ideálnom prípade každej vete v zdrojovom jazyku prináleží práve jedna veta v cieľovom jazyku. V praxi je však často súvetie preložené do dvoch viet alebo opačne. V prípade, keď vety v zdrojovom a cieľovom texte nie sú uvedené v rovnakom poradí, hovoríme o krížovej závislosti [3].

Existujú dva základné prístupy k problematike zarovnania viet, v rámci ktorých vzniklo niekoľko rôznych algoritmov. Nasledujúce dve podkapitoly čerpajú prevažne z [3] a [10].

2.2.1 Zarovnanie založené na dĺžke

Základnou myšlienkou tejto metódy je, že krátka veta by po preložení mala zostať krátkou a naopak dlhá veta by mala zostať dlhou. Dĺžku vety tvorí počet slov alebo počet znakov. S prvými algoritmami implementujúcimi tento prístup prišli Gale a Church (ako kritérium využívali počet znakov) a Brown (počet slov). Tieto algoritmy sú rýchle, na druhej strane však nie veľmi presné ak sa v korpuse nachádzajú vynechané alebo vložené vety alebo v prípade zarovnaní N:1 alebo N:N (príklad súvetí).

2.2.2 Lexikálne zarovnanie

Táto metóda využíva pri procese zarovnávania lexikálne informácie, čím sa stáva robustnejšia ako predchádzajúca metóda. Využíva ideu, že ak si slová vo vetnom páre navzájom korešpondujú, tak pravdepodobne tieto vety sú si navzájom prekladom. Touto metódou je možné dosiahnuť lepších výsledkov ako pri porovnávaní dĺžky. Nevýhodou je jej výpočetná náročnosť, čo ju robí pomalšou.

V praxi sa používajú taktiež kombinácie oboch prístupov. Pri problematike zarovnania viet sa môžeme stretnúť s anglickým výrazom *cognates* (slov. *príbuzný*), označujúci slová historicky príbuzných jazykov s podobnou ortografiou a významom. Patrí sem napríklad anglicko-francúzsky slovný pár *parlament/parliament*. Vo vývoji zarovnávacích metód bola snaha využiť podobné dvojice, tá sa však stretla s úspechom len pri určitej skupine jazykov.

Dostupnými nástrojmi na zarovnanie viet sú napr. Hunalign[4] alebo Vanilla.

2.3 Zarovnanie slov

Ďalším problémom, ktorý je súčasťou strojového prekladu je zarovnanie slov. Odvolávajúc sa na definíciu zarovnania v kapitole 2.2 si môžeme zarovnanie prestaviť ako mapovanie medzi slovami v zdrojovom jazyku a ich prekladmi v cieľovom jazyku. Zarovnanie slov je nevyhnutné pre budovanie bilingválnych slovníkov.

Jednoduché zarovnanie slov graficky znázorňuje obrázok 2.3. Čiary medzi slovami nazývame prepojenia [8]. Najjednoduchším prípadom je zarovnanie „slovo na slovo“, kedy slovo v zdrojovom jazyku má práve jeden ekvivalent v cieľovom jazyku. Na obrázku však vidieť aj prípad, kedy

[4] http://mokk.bme.hu/en/resources/hunalign/

prekladom jedného slova sú dve slová, príp. naopak. Špeciálnym prípadom sú slová, ktoré nevznikli ako preklad žiadneho slova a teda neexistuje pre ne prepojenie. Ich úloha je čisto syntaktická, patria sem napr. určité a neurčité členy alebo anglické slovo *„do“*. Pre tento účel vkladáme do zdrojového textu prázdne slovo *NULL*.

Obrázok 2.3: Zarovnanie slov

Označíme množinu všetkých možných zarovnaní *(f|e)* ako *A(e|f)*. Zavedieme nasledovné označenie: reťazce (vety) budú označené hrubými písmenami ***e*** a ***f***, ich slová obyčajnými písmenami *e* a *f*. Ak reťazec ***e*** má dĺžku *l* slov a reťazec ***f*** má dĺžku *m* slov, potom existuje *lm* rôznych prepojení medzi slovami. Zarovnanie je dané prepojeniami, ktoré obsahuje. Keďže podmnožinu možných prepojení je možné zostrojiť 2^{lm} spôsobmi, je veľkosť množiny *A(e|f)* práve 2^{lm} [8]. Tu zavedené označenia budeme používať v nasledujúcom texte bez ďalšieho vysvetľovania.

2.3.1 IBM modely

Táto podkapitola bude vychádzať z práce [8]. Pravdepodobnosť prekladu *P(f|e)* môžeme s využitím zarovnania vyjadriť ako sumu podmienených pravdepodobností *P(f,a|e)*, kde *a* je zarovnanie medzi vetami v zdrojovom a cieľovom jazyku:

$$P(\boldsymbol{f}|\boldsymbol{e}) = \sum_{a} P(\boldsymbol{f}, \boldsymbol{a}|\boldsymbol{e}) \tag{2.2}$$

Suma je nad prvkami množiny ***A(e|f)***. Zarovnanie ***a*** je vyjadrené ako postupnosť a_1^m; *m=0 až l*. Podmienená pravdepodobnosť ***P(f,a|e)*** môže byť vyjadrená sériou podmienených pravdepodobností nasledovne:

$$P(\boldsymbol{f}, \boldsymbol{a}|\boldsymbol{e}) = P(m|\boldsymbol{e}) \prod_{j=1}^{m} P\left(a_j | a_1^{j-1}, f_1^{j-1}, m, \boldsymbol{e}\right) P(f_j | a_1^{j}, f_1^{j-1}, m, \boldsymbol{e}) \tag{2.3}$$

Táto rovnica vyjadruje, že ak generujeme reťazec ***f*** spolu so zarovnaním z reťazca ***e***, môžeme najprv vybrať dĺžku reťazca ***f*** danú našou znalosťou reťazca ***e***. Potom môžeme zvoliť kam pripojiť prvú pozíciu v reťazci ***f*** danú znalosťou reťazca ***e*** a dĺžkou reťazca ***f***. Následne môžeme zvoliť identitu prvého slova v reťazci ***f*** danú znalosťou reťazca ***e***, dĺžky reťazca ***f*** a pozície v reťazci ***e***, ku ktorej je pripojená prvá pozícia v reťazci ***f***. Tak ďalej postupujeme, až kým získame zarovnanie celého reťazec.

Výpočtom podmienenej pravdepodobnosti pre zarovnanie sa zaoberajú IBM modely. Boli vyvinuté vo Výskumnom centre Thomasa J. Watsona firmy IBM. Modelov je celkovo 5 a prvýkrát boli popísané v Brown et al. (1993) (viď [8]). Každý model stavia na modeli predchádzajúcom a zároveň zvyšuje komplexnosť zarovnania. Prvý z nich si predstavíme podrobnejšie, ostatné v skratke.

Model 1

Tento model je najjednoduchší. V modeli 1 aj v modeli 2 vyberáme najprv dĺžku pre reťazec ***f*** za predpokladu, že všetky dĺžky sú rovnako pravdepodobné. Následne pre každú pozíciu v reťazci ***f*** rozhodneme, ako ho prepojiť s reťazcom ***e*** a ktoré slovo z reťazca ***f*** tam umiestnime. Pre model 1

predpokladáme rovnakú pravdepodobnosť všetkých prepojení pre každú pozíciu v reťazci ***f***. Z tohto dôvodu poradie slov v reťazcoch ***e*** a ***f*** nemá vplyv na ***P(f|e)***.

Model 1 predpokladá, že ***P(m|e)*** z rovnice 2.3 označíme ako parameter $\in$ nezávislý na ***e*** a *m*. Uvažujeme $\in$ ako konštantu rovnú nejakému malému číslu. Člen ***P($a_j|a_1^{j-1}, f_1^{j-1}$, m, e)*** je závislý iba na dĺžke *l* a je rovný $(l+1)^{-1}$. Člen ***P($f_j|a_1^{j}, f_1^{j-1}$, m, e)*** je závislý na f_j a e_{a_j}, vyjadriť ho môžeme ako parameter ***t($f_j|e_{a_j}$)***, ktorý nazývame pravdepodobnosť prekladu f_j pre dané e_{a_j}. Rovnica 2.3 teda môže byť prepísaná do nasledovného tvaru:

$$P(\boldsymbol{f}, \boldsymbol{a}|\boldsymbol{e}) = \frac{\in}{(l+1)^m} \prod_{j=1}^{m} t(f_j|e_{a_j}) \tag{2.4}$$

Hodnota a_j pre j v rozsahu 1 až m môže nadobúdať hodnoty v rozsahu 0 až l. Preto môžeme rovnicu 2.2 upraviť nasledovne:

$$P(\boldsymbol{f}|\boldsymbol{e}) = \frac{\in}{(l+1)^m} \sum_{a_1=0}^{l} \dots \sum_{a_m}^{l} \prod_{j=1}^{m} t(f_j|e_{a_j}) \tag{2.5}$$

Pravdepodobnosť prekladu ***t($f_j|e_{a_j}$)*** môžeme vypočítať použitím nasledujúcej rovnice:

$$t(f_i|e_{a_j}) = \lambda_e^{-1} \sum_{s=1}^{S} c(f|e;\, \boldsymbol{f}^{(s)}, \boldsymbol{e}^{(s)}) \tag{2.6}$$

kde:

$$c(f|e;\, \boldsymbol{f}, \boldsymbol{e}) = \frac{t(f|e)}{t(f|e_0) + \cdots + t(f|e_l)} \sum_{j=1}^{m} \delta(f, f_j) \sum_{i=0}^{l} \delta(e, e_i) \tag{2.7}$$

a λ_e označuje, že $t(f_i|e_{a_j})$ musí byť normovaná, aby pre každé slovo *e* platilo:

$$\sum_{f} t(f|e) = 1$$

$$\lambda_e = \sum_{f} \sum_{s=1}^{S} c(f|e;\, \boldsymbol{f}^{(s)}, \boldsymbol{e}^{(s)}) \tag{2.8}$$

Prvá suma v rovnici 2.7 vyjadruje počet slova *f* v reťazci ***f***, druhá suma vyjadruje počet slova *e* v reťazci ***e***. ***δ*** je funkcia Kroneckerovo delta.[5] $c(f|e;\, \boldsymbol{f}, \boldsymbol{e})$ vyjadruje počet prepojení *e* na *f* v preklade ***(f|e)***. *S* vyjadruje počet párov reťazcov v prekladaných dátach a *s* označuje poradie páru.

Využitím uvedených rovníc môžeme odhadnúť parametre ***t(f|e)*** nasledovným postupom:

1. Zvoľ počiatočné hodnoty pre ***t(f|e)***.
2. Využitím rovnice 2.7 vypočítaj počty $c(f|e;\, \boldsymbol{f}^{(s)}, \boldsymbol{e}^{(s)})$ pre každý pár viet.
3. Pre každé *e* ktoré sa objaví aspoň v jednom $\boldsymbol{e}^{(s)}$ vypočítaj λ_e pomocou rovnice 2.8. Pre každé *f* ktoré sa objaví aspoň v jednom $\boldsymbol{f}^{(s)}$ vypočítaj ***t(f|e)*** pomocou rovnice 2.6.
4. Opakuj body 2 a 3 až kým pre hodnoty ***t(f|e)*** nenastane požadovaná konvergencia.

[5] Kroneckerovo delta nadobúda hodnotu 1, ak sa jej dva argumenty navzájom rovnajú, inak nadobúda hodnotu 0

Uvedený postup sa nazýva EM[6] algoritmus. Počiatočná voľba hodnoty pre *t(f|e)* nie je dôležitá, pretože ***P(f|e)*** má pre model 1 práve jedno lokálne maximum. Dôkaz je uvedený v [8].

Model 2

Model 1 je špeciálnym prípadom modelu 2 a teda zdieľajú niekoľko vlastností, ktoré sú uvedené na začiatku popisu modelu 1. Model 1 sa používa na počiatočný odhad parametrov pre model 2. V modeli 2 pravdepodobnosť ***P(f|e)*** závisí na poradí slov v reťazcoch ***e*** a ***f***. Do výpočtu teda vstupuje aj pravdepodobnostný model pre zarovnanie slov.

Člen $P(a_j|a_1^{j-1}, f_1^{j-1}, m, \boldsymbol{e})$ je teraz závislý na j, a_j, m aj l. Pravdepodobnosť zarovnania teda závisí na pozícii oboch slov v reťazci. Vyjadríme ju nasledovne:

$$a(a_j|j,m,l) \equiv P\left(a_j \middle| a_1^{j-1}, f_1^{j-1}, m, l\right) \tag{2.9}$$

Zároveň musí byť splnená podmienka

$$\sum_{i=0}^{l} a(i|j,m,l) = 1$$

pre každú trojicu j, m, l. Rovnicu 2.5 teda upravíme a získame nasledovnú rovnicu:

$$P(\boldsymbol{f}|\boldsymbol{e}) = \epsilon \prod_{j=1}^{m} \sum_{i=0}^{l} t(f_j|e_i) a(i|j,m,l) \tag{2.10}$$

Napriek tomu, že je možné použitím týchto dvoch modelov dosiahnuť zaujímavé korelácie medzi niektorými pármi frekventovaných slov, častokrát vedú k neuspokojivým výsledkom.

Model 3, 4 a 5

Vo zvyšných troch modeloch získavame reťazec ***f*** tak, že najprv pre každé slovo v reťazci ***e*** vyberieme počet slov z reťazca ***f***, ktoré sa s ním prepoja, potom tieto slová identifikujeme a vyberieme pozície v reťazci ***f***, ktoré tieto slová obsadia. Posledný krok určuje prepojenia medzi reťazcom ***e*** a ***f*** a zároveň odlišuje tieto modely od seba.

V modeli 3 závisí pravdepodobnosť prepojenia na pozícii prepojenia a na dĺžke oboch reťazcov. V modeli 4 sa navyše pridáva závislosť na identitách prepojených slov a na pozíciách ostatných slov *f* prepojených na to isté slovo *e*. Model 5 je v mnohom podobný modelu 4, ale odstraňuje dva podstatné nedostatky predchádzajúcich dvoch modelov. Rieši konflikty, kedy by viacero slov obsadzovalo rovnakú pozíciu a keď vhodná pozícia pre slovo je nájdená mimo hraníc reťazca ***f***.

V modeloch 2, 3, 4 a 5 pravdepodobnostné funkcie nemajú práve jedno lokálne maximum. Inicializáciou každého modelu parametrami z predchádzajúceho modelu získame odhad parametrov najvyššieho modelu, ktoré nezávisia na našom počiatočnom odhade parametrov pre model 1.

V problematike rozpoznávania reči je rozšírené používanie tzv. skrytých markovovských modelov (HMM)[7]. Tento prístup bol uplatnený aj pre ďalší model zarovnávania slov. V HMM nepoznáme postupnosť stavov, ktorými model prechádza, poznáme však pravdepodobnosti prechodov medzi nimi. Myšlienkou HMM modelu zarovnávania slov je závislosť pravdepodobnosti zarovnania na relatívnej pozícii zarovnania slov, namiesto absolútnej pozície [11].

Okrem piatich IBM modelov bol mimo IBM vyvinutý aj model 6 [12].

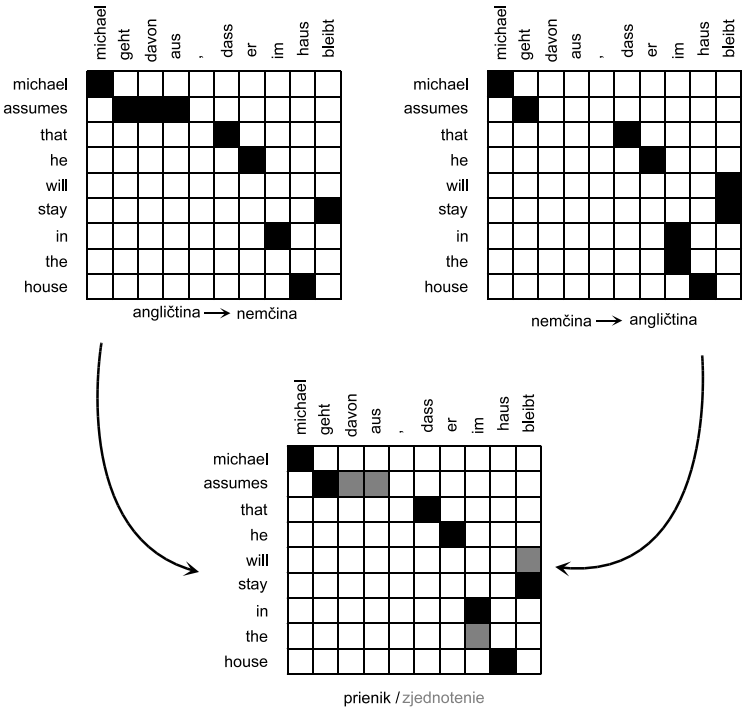
[6] EM = Expectation Maximization
[7] Hidden Markov Models

2.3.2 Symetrizačné zarovnanie slov

Na grafické znázornenie slov sa obvykle používa matica zarovnania. Riadky predstavujú slová v zdrojovom jazyku a stĺpce slová v cieľovom jazyku. Zvýraznené prvky matice predstavujú možné zarovnania medzi slovami (body zarovnania). Toto bude ilustrované na obrázku 2.4.

Ako je vidieť na obrázku 2.3, z jedného slova z zdrojovom jazyku môže v preklade vzniknúť viac slov a taktiež viac slov v zdrojovom jazyku môže byť preložené jedným slovom. Vyššie uvedené modely umožňujú iba zarovnanie 1:N, druhý prípad teda nepodporujú.

Jednoduchým riešením tohto problému je vytvorenie zarovnania slov v oboch smeroch a ich spojenie pomocou symetrizačných heuristík. Ak by sme spravili prienik vzniknutých matíc, získali by sme s vysokou pravdepodobnosťou správne zarovnania, avšak nie všetky. Zjednotenie obsahuje všetky správne zarovnania, ale okrem nich aj niektoré nesprávne.



Obrázok 2.4: Grafické znázornenie zarovnania a jeho symetrizácia [13]

Symetrizačné heuristiky

Vhodným kompromisom medzi prienikom a zjednotením sú symetrizačné heuristiky. Jednou z používaných heuristík je algoritmus grow-diag-final alebo jej striktnejší variant grow-diag-final-and. Do množiny konečných zarovnaní pridá najprv všetky body zarovnania z prieniku. Pokračuje pridávaním bodov zo zjednotenia, ktoré susedia s niektorým z bodov z prieniku. Susediacimi bodmi môžu byť body vľavo, vpravo, hore a dole (variant grow), príp. všetky body 8-okolia (variant grow-diag). V poslednom kroku sa pridávajú zvyšné body zarovnania medzi slovami zo zjednotenia. Pridajú sa tie body, ktorých aspoň jedno slovo nebolo doteraz zarovnané (variant grow(-diag)-final), príp. ani jedno nebolo doteraz zarovnané (variant grow(-diag)-final-and). Pseudokód posledného variantu znázorňuje obrázok 2.5 [14].

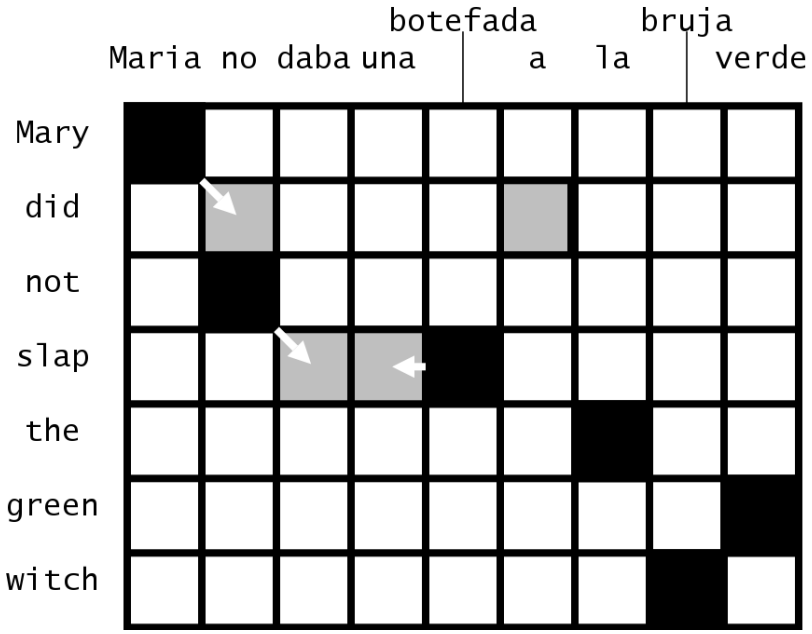
```
GROW-DIAG-FINAL(e2f,f2e):
  susedne_body = ((-1,0),(0,-1),(1,0),(0,1),(-1,-1),
                  (-1,1),(1,-1),(1,1))
  zarovnanie = prienik(e2f,f2e);
  GROW-DIAG(); FINAL-AND(e2f); FINAL-AND(f2e);

GROW-DIAG():
  iteruj kým je možné pridávať nové body
    pre slovo v zdrojovom jazyku e = 0 .. en
      pre slovo v cieľovom jazyku f = 0 .. fn
        ak (e je zarovnané s f)
          pre každý bod zarovnania (e_a, f_a)
            ak ((e_a nebolo zarovnané alebo f_a nebolo zarovnané)
                 a súčasne (e_a, f_a) patrí do zjednotenia (e2f,f2e))
                pridaj bod zarovnania (e2f,f2e)

FINAL-AND(a):
  pre slovo v zdrojovom jazyku e_a = 0 .. en
    pre slovo v cieľovom jazyku f_a = 0 .. fn
      ak ((e_a nebolo zarovnané a súčasne f_a nebolo zarovnané)
           a súčasne (e_a, f_a) patrí do zarovnania a)
          pridaj bod zarovnania (e_a, f_a)
```

Obrázok 2.5: Pseudokód symetrizačnej heuristiky grow-diag-final-and

Činnosť tejto heuristiky si vysvetlíme na príklade podľa obrázku 2.6. Do množiny bodov zarovnania sa najprv pridajú všetky body prieniku, ktoré sú vyznačené čiernou farbou. Druhým krokom je pridávanie susedných bodov zo zjednotenia vyznačených šedou farbou. Bod zarovnania medzi slovami *did* a *a* sa v prípade heuristiky grow-diag-final-and nepridá, pretože slovo *did* bolo už zarovnané so slovom *no* v druhom kroku pri pridávaní susedných bodov. Ak by sme však použili menej striktnú heuristiku grow-diag-final, tento bol by bol tiež pridaný [14].



Obrázok 2.6: Príklad činnosti symetrizačnej heuristiky grow-diag-final(-and) [14]

2.4 Model prekladu založený na frázach

Doteraz sme uvažovali, že veta sa skladá zo slov a na rovnakej úrovni sme vykonávali aj preklad. Oveľa výhodnejšie sa však ukázalo rozdeliť vetu na menšie sekvencie slov nazývaných *frázy*. Prekladový systém vytvorený popri tejto práci využíva práve preklad založený na frázach, preto si podrobnejšie vysvetlíme jeho pozadie. Obsah podkapitoly bol prevzatý z [15].

Počas prekladu je veta $\boldsymbol{f}$ rozdelená na I fráz $\bar{f}_1^I$. Pravdepodobnosť všetkých možných rozdelení je rovnaká. Každá fráza v cieľovom jazyku $\bar{f}_i$ je preložená na frázu $\bar{e}_i$ v zdrojovom jazyku. Frázy v zdrojovom jazyku môžu byť preusporiadané. Pravdepodobnoť prekladu frázy budeme označovať $\phi(\bar{f}_i|\bar{e}_i)$. Povšimnime si, že smer prekladu je kvôli Bayesovmu pravidlu obrátený.

Preusporiadanie fráz modelujeme ako distribúciu pravdepodobnosti zmeny poriadia fráz $d(a_i - b_{i-1})$, kde a_i označuje počiatočnú pozíciu frázy v cieľovom jazyku, ktorá bola preložená na i-tú frázu v zdrojovom jazyku a b_{i-1} označuje koncovú pozíciu frázy v cieľovom jazyku preloženú na *(i-1)*. frázu v zdrojovom jazyku. Jednoduchý model zmeny poradia fráz bude mať nasledujúci tvar:

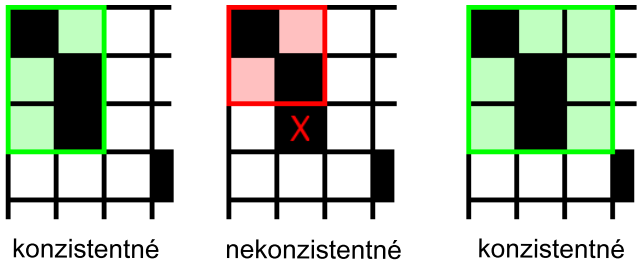
$$d(a_i - b_{i-1}) = \alpha^{|a_i - b_{i-1} - 1|} \tag{2.11}$$

Model prekladu nadobudne výsledný tvar:

$$P(\bar{f}_1^I | \bar{e}_1^I) = \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i)\, d(a_i - b_{i-1}) \tag{2.12}$$

Pre frázový preklad potrebujeme zhotoviť tabuľku fráz. To je možné urobiť viacerými navrhnutými spôsobmi. Frázovú tabuľku môžeme vyextrahovať zo zarovnania slov.

Zo zarovnania slov vyberieme tie páry fráz, ktoré sú konzistentné so zarovnaním slov. Konzistentné páry fráz sú tie, ktorých slová sú zarovnané výlučne medzi sebou a nie so slovami mimo daného páru fráz. Na obrázku 2.7 si môžeme všimnúť, že súčasťou konzistentnej frázy môžu byť aj nezarovnané slová.



Obrázok 2.7: Konzistencia frázových párov [14]

Relatívna frekvencia výskytu frázy je daná vzťahom:

$$\phi(\bar{f}|\bar{e}) = \frac{počet(\bar{f}, \bar{e})}{\sum_{\bar{f}} počet(\bar{f}, \bar{e})} \tag{2.13}$$

kde $počet(\bar{f}, \bar{e})$ vyjadruje, koľkokrát je fráza $\bar{f}$ zarovnaná s frázou $\bar{e}$ v paralelnom korpuse.

Iným prístupom k vytváraniu tabuľky fráz je obmedzenie výberu iba na tie frázy, ktoré sú syntakticky správne. Marcu a Wong zase navrhli odvodiť korešpondencie medzi frázami priamo z paralelného korpusu. Tento prístup je však výpočtovo náročný.

Jedným spôsobom, ako overiť kvalitu prekladu frázy je skontrolovať, ako si vzájomne korešpondujú preklady medzi jednotlivými slovami frázy. Zavedieme preto distribúciu

pravdepodobnosti lexikálneho prekladu, ktorú určíme z rovnakého zarovnania slov, z ktorého sme extrahovali frázy.

$$w(f|e) = \frac{počet(f,e)}{\sum_{f'} počet(f',e)} \tag{2.14}$$

Pre daný pár fráz $\bar{f}, \bar{e}$ a zarovnanie slov *a* medzi pozíciami slov v cieľovom jazyku *i = 1, ..., n* a pozíciami slov v cieľovom jazyku *j = 0, 1, ..., m* určíme lexikálnu váhu nasledovným vzťahom:

$$p_w(\bar{f}, \bar{e}, a) = \prod_{i=1}^{n} \frac{1}{|\{j|(i,j)\epsilon a\}|} \sum_{\forall(i,j)\epsilon a} w(f_i|e_j) \tag{2.15}$$

Lexikálnu váhu pridáme do modelu prekladu rozšírením rovnice 2.12

$$P(\bar{f}_1^I|\bar{e}_1^I) = \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i)\, d(a_i - b_{i-1}) p_w(\bar{f}_i|\bar{e}_i, a)^{\lambda} \tag{2.16}$$

kde $\boldsymbol{\lambda}$ určuje silu lexikálnej váhy. Vhodnými hodnotami pre tento parameter sú čísla okolo 0,25.

2.4.1 Dekodér

Úlohou dekodéra je zhotoviť najlepší preklad využitím jazykového modelu a modelu prekladu. Pri preklade sa veta rozloží na všetky možné frázy, čím vznikne viacero možností prekladu. Počet týchto možností exponenciálne rastie s dĺžkou prekladanej vety, čím sa zväčšuje náročnosť dekódovania. Na prehľadávanie stavového priestoru možností prekladu sa využívajú informované metódy vyhľadávania, napr. algoritmus Best First Search, jeho variant Beam Search, príp. A*. Popri tom sú využívané rôzne heuristiky, ktoré zmenšujú stavový priestor vyhľadávania vyraďovaním niektorých hypotéz. Na tento účel sa často používa práve algoritmus Beam Search.

Prekladanie je založené na tvorbe a hodnotení hypotéz. Začína sa s prázdnou hypotézou, preložením frázy táto hypotéza expanduje na nové hypotézy. Pokračuje sa, kým sa nevyčerpajú všetky možnosti prekladu. Vznikajúce hypotézy aktualizujú svoje ohodnotenie. Finálna hypotéza bez nepreložených slov s najlepším ohodnotením sa stáva výstupom hľadania. Niekedy o nej hovoríme ako o najlacnejšom riešení.

Ohodnotenie hypotézy pozostáva z kombinácie súčasného a potenciálneho budúceho ohodnotenia. Súčasné ohodnotenie je celková pravdepodobnosť doteraz preložených fráz v hypotéze, ktorá vznikla kombináciou informácií z modelu prekladu, jazyka a preusporiadania. Ak by sa použilo len súčasné ohodnotenie, vyhľadávanie by malo tendenciu označovať za najlepší výsledok hypotézu, ktorá mala na začiatku vysoko ohodnotené slová. Pre potenciálne budúce ohodnotenie sa neuvažuje model preusporiadania [9]. Pre rôzne dlhé frázy sa z jazykového modelu používa pravdepodobnosť unigramu pre prvé slovo, bigramu pre druhé slovo, trigramu[8] pre tretie a zvyšné slová [15].

Dĺžku výstupu dekodéra môžeme ovplyvniť pridaním parametra penalizácie ω do rovnice 2.1 pre každé generované slovo v zdrojovom jazyku. V závislosti na jeho hodnote sú preferované dlhšie alebo kratšie výstupy. Model prekladu $P(\boldsymbol{f}|\boldsymbol{e})$ možno potom rozpísať pomocou rovnice 2.16.

$$\hat{e} = argmax_e P(\boldsymbol{f}|\boldsymbol{e}) P(\boldsymbol{e}) \omega^{dĺžka(\boldsymbol{e})} \tag{2.17}$$

[8] Pojmy budú vysvetlené v podkapitole 2.5

2.5 Jazykový model

Pri skladaní fráz do výsledného prekladu využívame taktiež predikciu slov. Na to slúži jazykový model, ktorý obsahuje informácie o tom, s akou pravdepodobnosťou sa v jazyku vyskytuje určitá postupnosť slov. Jazykový model sa používa iba pre cieľový jazyk.

Odhadovaná postupnosť slov však nemôže mať neobmedzenú dĺžku. Preto sa pri vytváraní modelu vychádza z markovského predpokladu, že nasledujúce slovo ovplyvňuje iba niekoľko predchádzajúcich slov. Modely sú prevažne založené na *n-gramoch*, kde za *n* môžeme dosadiť ľubovoľné kladné celé číslo. Nasledujúce slovo ovplyvňuje *n-1* slov. V praxi sa pre *n* používajú najčastejšie hodnoty 2 a 3, s čím analogicky súvisia pojmy *bigram* a *trigram* [3].

Pri vytváraní jazykového modelu sa často stretneme s postupnosťou slov, ktorá doteraz nebola zaznamenaná. Aby jej nebola priradená nulová pravdepodobnosť, využíva sa niekoľko techník vyhladzovania modelu (napr. Good-Turing, Written-Bell).

Empiricky bolo zistené, že veľkosť jazykového modelu priamo vplýva na výslednú kvalitu prekladu [16]. Model môžeme vybudovať z trénovacích dát v cieľovom jazyku alebo z dostatočne rozsiahlych jednojazyčných korpusov, ktoré zvyčajne spravujú národné jazykovedné ústavy. Využiť k tomu môžeme voľne dostupné nástroje (napr. RandLM, KenLM, SRILM, IRSTLM).

2.6 Metódy hodnotenia kvality strojového prekladu

Po tom, ako získame preklad nejakého textu nás prirodzene zaujíma, ako kvalitný je tento preklad. Hodnotenie kvality prekladu nie je triviálny problém, pretože kvalita prekladu nie je jednoznačný pojem. Často existuje viacero správnych prekladov a pri rozhodovaní, ktorý je kvalitnejší, často do procesu vstupuje subjektívny názor. Preto bolo dôležité zaviesť metriky, ktoré tento problém rozhodnú.

2.6.1 Ručné hodnotenie

Pri manuálnom hodnotení kvality prekladu nás zaujíma názor človeka – prekladateľa alebo pre ktorého je cieľový jazyk jazykom materinským. Hodnotitelia rozhodujú o presnosti a plynulosti výstupného textu prekladača. Využívajú sa dotazníky a známkovanie rôznymi stupnicami. Taktiež môžeme hodnotiteľa požiadať, aby strojový preklad opravil a my zisťujeme, aké úpravy boli vykonané. Na zaistenie kvality ručného hodnotenia potrebujeme zaistiť viacero hodnotiteľov. Ručné hodnotenie je teda menej dostupné, pomalé a teda nevhodné pre fázu vývoja prekladového systému.

2.6.2 Automatické hodnotenie

Odstrániť nevýhody ručného hodnotenia je úlohou automatických metód hodnotenia. Takéto vyhodnotenie je rýchle, lacné a opakovateľne použité. Preklad z jedného prekladového systému ohodnotí vždy rovnako. Vyžaduje však existenciu aspoň jedného, ideálne však viac človekom vytvorených referenčných prekladov. Hodnotenie danej metriky by navyše malo korelovať s ľudským hodnotením.

BLEU

Najpoužívanejšou metrikou pre hodnotenie kvality strojového prekladu je BLEU[9] vyvinutá firmou IBM. Jej podrobný opis nájdeme v správe [17], z ktorej čerpá táto podkapitola. Vychádza z myšlienky, že čím podobnejší je strojový preklad s profesionálnym ľudským prekladom, tým je lepší. Na porovnávanie medzi prekladmi dokáže použiť viacero referenčných prekladov. Metrika porovnáva zhody n-gramov medzi strojovým a referenčným prekladom a na základe ich počtu prideľuje skóre. Tieto zhody sú navyše nezávislé na pozícii vo vete.

Vyhodnotiť počet zhôd môžeme jednoducho tak, že spočítame počet slov (unigramov) v kandidátnom (strojovom) preklade, ktoré sa zároveň objavujú aj v niektorom z referenčných prekladov. Tento počet následne vydelíme celkovým počtom slov v kandidátnom preklade. Pri tejto

[9] BLEU = BiLingual Evaluation Understudy, číta sa [blu:]

jednoduchej úvahe však dosiahne nezmyslený preklad s vhodnými opakujúcimi sa slovami vysoké skóre, iba vďaka tomu, že tieto slová sa vyskytli v referenčnom preklade. Navyše strojový preklad často generuje väčší počet slov, ako je v referenčnom preklade. To je samozrejme nežiaduce, preto metrika BLEU používa tzv. modifikovanú unigramovú presnosť. Spočíta sa najprv počet výskytov slov v jednotlivých referenčných prekladoch a vyberie sa maximum. Toto maximum nám určuje najvyšší možný počet prepojení slova z kandidátneho textu so zhodným slovom z referenčného textu. Ak sa teda nájde v referenčnom preklade zhodné slovo, toto už nemôže byť použité pri ďalšej nájdenej zhode a dané slovo sa v kandidátnom texte už môže vyskytnúť najviac maximum-1 krát. Počet prepojení sa nakoniec vydelí celkovým počtom kandidátnych slov. Podobným spôsobom sa pracuje aj pre väčšie celky slov (n-gramy). Pri použití unigramov sa zameriavame na adekvátnosť použitých slov, zatiaľ čo väčšie celky slov skúmajú plynulosť prekladu. Experimentálne bolo zistené, že pre najlepšiu koreláciu s ľudským hodnotením je najvhodnejšie použitie 4-gramov. Používajú sa však v kombinácii s unigramami, bigramami a trigramami, pričom využívame ich geometrický priemer.

Pre celý text najprv určíme zhody n-gramov pre každú vetu zvlášť. Potom zistíme počet prepojených n-gramov a vydelíme ho celkovým počtom kandidátnych n-gramov.

$$p_n = \frac{\sum_{K \epsilon \{Kandidáti\}} \sum_{n-gram\ \epsilon K} počet_{prepojených}(n-gram)}{\sum_{K' \epsilon \{Kandidáti\}} \sum_{n-gram' \epsilon K'} počet(n-gram')} \tag{2.18}$$

Metrika BLEU berie do úvahy aj dĺžku prekladu. Preklady, ktoré sú dlhšie ako referenčné, penalizuje modifikovaná n-gramová presnosť. Pre kratšie preklady bola zavedená penalizácia stručnosti *BP*. V prípade viacerých referenčných prekladov vyberáme tú dĺžku, ktorá sa najviac blíži dĺžke kandidátneho prekladu (najbližšia dĺžka). Pre zhodné dĺžky má penalizácia hodnotu 1. Pre výpočet penalizácie celého prekladu určíme najprv efektívnu dĺžku referenčného prekladu *r* súčtom najbližších dĺžok všetkých kandidátnych viet. Ako *c* označíme celkovú dĺžku kandidátneho prekladu.

$$BP = \begin{cases} 1 & ak\ c > r \\ e^{(1-r/c)} & ak\ c \leq r \end{cases} \tag{2.19}$$

Výsledné skóre sa potom počíta tak, geometrický priemer modifikovanej n-gramovej presnosti vynásobíme penalizáciou:

$$BLEU = BP \times exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{2.20}$$

$$\log BLEU = min\left(1 - \frac{r}{c}, 0\right) + \sum_{n=1}^{N} w_n \log p_n \tag{2.21}$$

Odporúčaná hodnota pre uniformné váhy $w_n = 1/N, N = 4$. Získame tak číslo v rozsahu 0 až 1 (resp. vynásobené číslom 100), kde vyššie číslo znamená lepší preklad. Výsledné skóre závisí na viacerých faktoroch, medzi inými aj na počte referenčných prekladov.

Metrika BLEU má však aj niekoľko nevýhod. Za zmienku stojí samotné využívanie n-gramov, čím sa znevýhodňujú krátke vety. V práci [18] bolo poukázané na niekoľko zásadných problémov. Hlavným cieľom bolo v rozpore s tvrdeniami autorov metriky [17] ukázať, že metrika nemusí vždy korelovať s ľudským ohodnotením. Ďalej bolo ukázané, že metrika sa nehodí na porovnávanie prekladových systémov založených na rozdielnych prekladových stratégiách. Taktiež zlepšenie skóre nemusí nevyhnutne znamenať zlepšenie kvality prekladu.

NIST

Túto metriku vyvinula z poverenia agentúry DARPA rovnomenná organizácia NIST[10]. Je založená na predošlej práci spoločnosti IBM na metrike BLEU. Odstraňuje niektoré vyššie spomenuté nevýhody metriky BLEU. Pre výpočet skóre sa používa aritmetický priemer namiesto geometrického, čím sa odstraňuje znevýhodnenie kratších viet. Ďalším rozdielom je priraďovanie väčšej váhy tým n-gramom, ktoré majú vyššiu informačnú hodnotu. Čím menej frekventovaný je n-gram, tým má vyššiu hodnotu. Iný prístup bol zvolený aj pre určovanie penalizácie stručnosti. Bol tak minimalizovaný vplyv malých rozdielov medzi dĺžkami prekladov [19].

METEOR

Rekciou na predošlé dve metriky bol vznik novej, ktorej autori uvádzajú lepšie dosiahnuté výsledky. Metrika dokáže identifikovať správnosť prekladu aj pri použití synoným či parafráz. Nie je teda až tak závislá na počte referenčných prekladov, aj keď ich samozrejme vyžaduje. Je založená na porovnávaní unigramov, pre ktoré vytvára zarovnanie. Taktiež využíva stemmovanie[11] [20]. Nevýhodou metriky METEOR je jej malé pokrytie na jazykoch, v súčasnosti slovenčina, na rozdiel od češtiny, stále nie je podporovaná.

Ďalšími používanými metrikami je Word Error Rate (WER), založený na editačnej vzdialenosti na úrovni slov. Ide o jednu z prvých používaných a veľmi jednoduchých metrík. Medzi ďalšie metriky patrí NEVA, WAFT a TER/HTER.

2.7 Existujúce riešenia

Na trhu je možné nájsť viacero voľne dostupných aj komerčných softvérových riešení strojového prekladu. Najjednoduchšie z nich využívajú priamy preklad pomocou rozsiahleho slovníka. Dokážu užívateľovi poskytnúť približný, nie veľmi kvalitný preklad. Pre pochopenie kontextu prekladu je často nutná aspoň znalosť základov daného cieľového jazyka. Ako príklad môžeme uviesť PC Translator. Medzi sofistikovanejšie systémy, ktoré sa objavili na česko-slovenskom trhu môžeme zaradiť Skik, Transen či Eurotran.

Vyvinúť kvalitný prekladový systém je stále náročnou úlohou, počet úspešných projektov nie je vysoký. Bolo však preukázané, že preklad medzi dvojicami príbuzných jazykov dosahuje viditeľne lepšie výsledky v porovnaní s jazykmi z rozdielnych jazykových skupín [21].

Jedným z takýchto systémov bol projekt RUSLAN vyvinutý v Ústave formálnej a aplikovanej lingvistiky na Univerzite Karlovej v Prahe v 80. rokoch. Slúžil najmä na preklad dokumentácie z češtiny do ruštiny. Systém bol založený na pravidlách, vykonával úplnú morfologickú a syntaktickú analýzu češtiny, prenos a syntaktické a morfologické generovanie ruštiny. Spočiatku sa predpokladalo, že prenos syntaxe a sémantiky nebude kvôli podobnosti jazykov vôbec potrebný. Toto sa však v priebehu vývoja zmenilo, bolo využitých niekoľko originálnych myšlienok. Vyhodnotenie výsledkov ukázalo, že približne 40% prekladu je správne, 40% vyžaduje drobnú opravu človekom a zvyšných 20% predstavovalo nesprávny preklad vyžadujúci opravy [22].

Skúsenosti nadobudnuté pri vývoji systému RUSLAN boli tým istým ústavom využité pri ďalšom prekladovom systéme ČESÍLKO. Jeho primárnym cieľom bola opäť lokalizácia prevažne technických textov. Systém v súčasnosti prekladá do slovenčiny a experimentálne aj do poľštiny a litovčiny. Základnou myšlienkou systému bolo maximálne ťažiť z podobnosti týchto jazykov, preto bola zvolený priamy preklad „slovo na slovo“. Najväčším problémom, s ktorým sa systém stretával pre túto dvojicu jazykov bola morfologická nejednoznačnosť jednotlivých slovných foriem. Riešením bola aplikácia morfologickej analýzy a značkovača. Systém dosahuje približne 90% úspešnosť v porovnaní s ľudským prekladom [22].

[10] NIST = National Institute of Standards and Technology

[11] Stemmovanie je proces, ktorým sa pre dané slovo vytvorí jeho koreň

Medzi najpoužívanejšie voľne prístupné online prekladače patria Google Translate od spoločnosti Google a Bing Translator od spoločnosti Microsoft. Oba nástroje predstavujú zástupcov štatistického strojového prekladu. Google Translate v súčasnosti ponúka preklad medzi 66 jazykmi, Bing Translator medzi 41 jazykmi. Je zrejmé, že kvalita prekladu je silno závislá na množstve dostupných paralelných textov. Pre dvojicu čeština - slovenčina dosahuje na prvý pohľad lepšie výsledky Google Translate. Použitie oboch nástrojov je obmedzené určitými limitmi, preto sa nehodia na jednorazové preklady obsiahlejších textov.

3 Návrh systému

Počnúc touto kapitolou bude za zdrojový jazyk považovaná čeština a za cieľový jazyk slovenčina. Obsahom kapitoly bude popis použitých textových zdrojov a nástrojov.

3.1 Paralelný korpus

Základným predpokladom vytvorenia štatistického prekladového systému je existencia alebo vybudovanie paralelného korpusu. Paralelný korpus (bitext) je súbor dvojjazyčných textov v strojovo čitateľnom formáte, pričom jeden text je prekladom druhého. Korpusom však nie je možné pokryť celú škálu prirodzeného jazyka, vždy bude predstavovať iba vzorku. Preto našou snahou je vybudovať čo najväčší korpus.

Ako jeden z najstarších paralelných textov môžeme uviesť staroegyptskú Rosettskú dosku, ktorá obsahuje jediný text v troch rôznych prekladoch. Jej objav výrazne prispel k rozlúšteniu hieroglyfického písma. Paralelné korpusy vznikajú najčastejšie ako zbierka písomných dokumentov štátov a organizácií, ktoré majú viac ako jeden úradný jazyk. Paralelný korpus vytvorený zo zápisov z rokovania kanadského parlamentu, v ktorom sa používa angličtina a francúzština, sú známe pod názvom Hansard. Podobná situácia je napr. aj v Hong Kongu či Švajčiarsku. Ďalším zaujímavým zdrojom pre vybudovanie paralelného korpusu sú dokumenty Organizácie spojených národov či Európskej únie.

Pre dvojicu jazykov čeština a slovenčina je situácia mierne komplikovanejšia. Je to zapríčinené ich veľkou blízkosťou a podobnosťou. Veľká väčšina populácie oboch národov rozumie bez väčších problémov druhému jazyku. Najmä však slovenská populácia využíva česky lokalizované písomné dokumenty (knihy, filmové titulky a pod.), preto je produkcia slovenskej lokalizácie menej frekventovaná.

Existujúci paralelný slovensko-český korpus spravuje Jazykovedný ústav Ľ. Štúra Slovenskej akadémie vied. Jeho online verzia však umožňuje iba vyhľadávanie. Preto bolo nutné vytvoriť vlastný paralelný korpus.

3.1.1 Použité zdroje

V tejto podkapitole si predstavíme zdroje, z ktorých som čerpal pri budovaní paralelného korpusu.

JRC-ACQUIS Multilingual Parallel Corpus

Acquis communautaire je súhrnný názov pre legislatívne texty prijaté Európskou úniou (EÚ) a jej príbuznými organizáciami od 50. rokov po súčasnosť. Ich prijatie je vstupnou podmienkou pre kandidátov na vstup do EÚ. Tieto texty sú preložené do všetkých oficiálnych jazykov EÚ, predstavujú preto výborný zdroj pre paralelné česko-slovenské texty. Dáta sú voľne dostupné ako samostatné XML súbory pre každý jazyk. Pre každý jazykový pár je dostupný zarovnávací súbor vytvorený pomocou nástroja Hunalign alebo Vanilla, ktorý užívateľovi umožňuje zo získaných XML súborov vytvoriť vlastný paralelný korpus [23].

OPUS

OPUS[1] je projekt, ktorý vytvára neustále rastúcu zbierku prekladov textov dostupných na internete. Dáta sú voľne poskytované ako samostatný TMX súbor pre každý dostupný jazykový pár. Texty sú v ňom už teda zarovnané, avšak bez manuálnej korekcie. Z dostupných paralelných textov pre dvojicu čeština - slovenčina som využil dokumentáciu Európskej centrálnej banky, dokumenty Európskej medicínskej agentúry (EMEA), text Európskej ústavy, lokalizačné texty pracovného

[1] http://opus.lingfil.uu.se/

prostredia KDE4 a filmové titulky zo servera OpenSubtitles[2]. Dostupný bol aj korpus z manuálu k PHP, avšak tento som po zbežnej vizuálnej kontrole vyhodnotil ako nekvalitný [24].

Europarl
Europarl[3] je korpus podporovaný projektom EuroMatrixPlus, zameraným na využitie strojového prekladu jazykov EÚ. Je vybudovaný zo zápisníc rokovaní Európskeho parlamentu v rokoch 1996-2011. V aktuálnej verzii 7 je v korpuse obsiahnutých 21 jazykov. Nevýhodou tohto korpusu však je, že zarovnania sú dostupné pre každý jazyk iba s angličtinou [21].

Prekladový slovník
Do korpusu som sa rozhodol zaradiť aj heslá z poskytnutého prekladového slovníka[4] z neznámeho zdroja. Tento som navyše rozšíril aj o heslá z českého synonymického slovníka[5].

Knihy
Pre potreby tohto projektu som zostrojil aj korpus kníh. Zdrojom boli elektronické verzie kníh, najmä svetové bestsellery, povinná literatúra na školách, Biblia. Podarilo sa mi zozbierať 51 kníh dostupných v oboch jazykoch v elektronickej podobe.

Tento korpus som sa však rozhodol nepoužiť z dvoch hlavných dôvodov. Preklady literárnych diel do češtiny a slovenčiny vo väčšine prípadov vychádzali z iného cudzieho jazyka (angličtina, nemčina) a často sa prejavovala fantázia a voľnosť prekladateľa, kvôli čomu mali jednotlivé preklady od seba viditeľne ďaleko. Druhým dôvodom je, že väčšina literárnych diel nepatrí medzi voľne šíriteľné a tak je ich použitie otázne z právneho hľadiska.

3.2 Jazykový model

Na vytvorenie jazykového modelu som v prvej fáze vývoja použil slovenskú časť vytvoreného paralelného korpusu. Model bol pomenovaný join.clean.low.sk. Ako už bolo spomenuté v kapitole 2.5, veľkosť modelu má vplyv na výslednú kvalitu prekladu. Výsledky v práci [16] ukazujú, že väčšie modely dosahujú lepšie výsledky v porovnaní s menšími modelmi. Preto som sa ešte pred začiatkom tréningovej fázy rozhodol použiť iný model.

Na stránkach Jazykovedného ústavu Ľ. Štúra Slovenskej akadémie[6] vied je voľne k dispozícii korpus s označením prim-5.0. Súčasťou tohto korpusu sú aj dostupné jazykové modely. Pre vývoj systému som zvolil model s označením prim-5.0-sane-lowercase, čo znamená, že model je vytvorený z vyčisteného korpusu bez textov nezodpovedajúcich niektorým kritériám (správna diakritika, súčasný spisovný jazyk, nelingvistické texty) a z tokenov[7] konvertovaných na malé písmená. Model bol vytvorený pomocou nástroja IRSTLM s vyhladzovaním Written-Bell a je vo formáte iARPA. Model obsahuje n-gramy do veľkosti 3 v nasledovných počtoch:

Veľkosť n-gramu	Počet	
	prim-5.0-sane-lowercase	join.clean.low.sk
1	3562288	881389
2	88773704	5382778
3	308220646	3244695

Tabuľka 3.1: Veľkosti jazykových modelov

[2] http://www.opensubtitles.org/
[3] http://www.statmt.org/europarl/
[4] súbor *sloces*, /mnt/minerva1/nlp/projects/mt_sk2
[5] /mnt/minerva1/nlp/dicts/others/synonyma/1998
[6] http://korpus.juls.savba.sk/prim(2d)5(2e)0.html
[7] Za token sa považuje slovo, číslo, interpunkčné znamienko, operátor a pod.

Pre porovnanie, veľkosť modelu prim-5.0-sane-lowercase je 12,1 GB a modelu join.clean.low.sk 0,27 GB.

3.3 Použité nástroje

Čitateľovi, ktorý nevynechal kapitolu 2, príp. sa orientuje v teórii štatistického strojového prekladu, je jasné, že vytvorenie prekladového systému nie je triviálna záležitosť. Implementácia vlastných nástrojov vytvárajúcich všetky potrebné modely pre preklad by bola nad rámec tejto práce. V tejto oblasti však už bolo vyvinutých niekoľko kvalitných nástrojov, ktoré som sa rozhodol použiť. V tejto podkapitole si ich predstavíme.

3.3.1 MOSES

Moses[8] je neustále vyvíjaný open source toolkit (súbor nástrojov) vydaný pod licenciou GNU LGPL[9], slúžiaci na vývoj štatistického prekladového systému. Umožňuje vytvárať prekladové modely pre ľubovoľný jazykový pár.

Jadrom Mosesa je dekodér, ostatná činnosť (tréningová fáza) je zabezpečená nástrojmi vyvinutými tretími stranami. Pre tréningový proces je vyžadovaný paralelný korpus so zarovnaním po vetách. Za jednu z výhod Mosesa je považované to, že prekladový systém môžeme vybudovať na mieru dátam, ktoré budeme prekladať. Ak teda paralelný korpus obsahuje texty z podobnej domény, akú budeme prekladať, dosiahneme lepšie výsledky. Ďalšou výhodou je, že je dobre zdokumentovaný. Moses dokáže vytvoriť prekladové modely založené na frázach, hierarchické modely založené na frázach a syntaktické modely. Pre účely budovaného prekladového systémy použijeme modely založené na frázach.

Proces tréningu zabezpečujú externé nástroje, väčšinou implementované v jazyku Perl alebo C++. Tréning obsluhuje skript train-model.perl, proces má 9 krokov a príslušnými parametrami môžeme zvoliť, ktorým krokom začať, príp. skončiť. Štandardne tréning prebieha v nasledujúcich krokoch:

1. Príprava dát (korpusu)
2. Beh GIZA++ – proces prebieha v oboch smeroch zo zdrojového jazyka na cieľový a naopak; časovo najnáročnejší krok
3. Zarovnanie slov – kombinácia výsledkov predošlého kroku pomocou zvolenej symetrizačnej heuristiky
4. Vytvorenie lexikálnej prekladovej tabuľky
5. Extrakcia fráz
6. Ohodnotenie fráz
7. Vytvorenie modelu preusporiadania
8. Vytvorenie generovacieho modelu
9. Vytvorenie konfiguračného súboru

3.3.2 GIZA++

Na zarovnávanie slov Moses využíva nástroj GIZA++[10]. Ide o rozšírenie pôvodného programu GIZA, ktorý bol súčasťou projektu EGYPT[11]. Ten bol vyvinutý počas letného workshopu v roku 1999 v Centre pre spracovanie reči a jazyka na Univerzite Johna Hopkinsa.

[8] http://www.statmt.org/moses/
[9] http://www.gnu.org/copyleft/lesser.html
[10] https://code.google.com/p/giza-pp/
[11] http://old-site.clsp.jhu.edu/ws99/projects/mt/toolkit/

GIZA++ navrhol a vytvoril F. J. Och [12]. Program implementuje IBM modely 1 až 5 a model HMM. Dokáže sa naučiť štatistické strojové modely, Moses ju však využíva iba na zarovnanie slov. Úplný zoznam vylepšení oproti pôvodnému programu je obsiahnutý v dokumentácii.

Program vyžaduje na vstupe špeciálny formát súborov. Ten je možné vytvoriť pomocou priložených programov. Prvým krokom je spustenie programu plain2snt.out, ktorý pretransformuje čistý text z korpusu na požadovaný formát. Výstupom je slovníkový súbor VCB a bitextový súbor SNT. Slovníkový súbor obsahuje riadky vo formáte „ID reťazec počet_výskytov". V bitextovom súbore sú pre každú dvojicu viet určené tri riadky. Na prvom riadku je zapísaný počet výskytov tohto vetného páru. Na druhom riadku sa nachádza veta v zdrojovom jazyku a na treťom riadku veta v cieľovom jazyku. Všetky tokeny v oboch vetách sú nahradené unikátnym číselným identifikátorom. Dôvodom je menšia výpočetná náročnosť operácií nad číslami oproti reťazcom.

Proces zarovnávania slov je nielen časovo, ale aj pamäťovo náročný. Preto je pre veľké korpusy vhodné spustiť program snt2cooc.out, ktorý vytvorí súbor obsahujúci miery spoločného lexikálneho výskytu v korpuse. Taktiež pre IBM modely 4 a vyššie je vhodné rozdeliť slová do tried. Na to slúži program mkcls. Všetky doteraz uvedené činnosti boli súčasťou kroku 1 z vyššie uvedenej postupnosti.

Výstupom zarovnania slov nástrojom GIZA++ je viacero súborov, nás však bude zaujímať len súbor zarovnania, ktorý je pomenovaný *.A3.final. Ako už bolo spomenuté, proces zarovnávania slov prebieha v oboch smeroch, preto tieto súbory sú dva a prefixom ich názvu je označenie jazykov vyjadrujúce zároveň smer zarovnania (cs-sk, sk-cs). Informácia o zarovnaní je pre každý vetný pár v súbore na troch riadkoch. Prvý riadok obsahuje poradové číslo vety v korpuse, dĺžky zdrojovej a cieľovej vety a pravdepodobnosť ich zarovnania. Druhý riadok obsahuje cieľovú vetu. Tretí riadok predstavuje zdrojovú vetu, kde za každým tokenom je množina čísel, ktoré znamenajú pozície tokenov v cieľovej vete, na ktoré sú tokeny v zdrojovej vete zarovnané. Formát súboru znázorňuje obrázok 3.1.

```
# Sentence pair (3) source length 7 target length 7 alignment score : 0.00401359
komise evropského společenství pro atomovou energii ,
NULL ({ }) komisia ({ 1 }) európskeho ({ 2 }) spoločenstva ({ 3 }) pre ({ 4 })
atómovú ({ 5 }) energiu ({ 6 }) , ({ 7 })
...
...
# Sentence pair (7) source length 3 target length 4 alignment score : 3.00085e-09
přijala toto rozhodnutí :
NULL ({ }) rozhodla ({ 1 2 }) takto ({ 3 }) : ({ 4 })
```

Obrázok 3.1: Príklad súboru zarovnania

Pre urýchlenie procesu zarovnávania slov je možné využiť alternatívy k nástroju GIZA++. Prvou je modifikácia MGIZA++[12], ktorá umožňuje viacvláknové spracovanie. Jej použitie je vhodné na počítačoch s viacjadrovým procesorom (napr. athena1, 2, 3). Druhou možnosťou je spúšťať tento proces paralelne na počítačovom clusteri nástrojom PGIZA++[13].

3.3.3 Ostatné nástroje

Pre vytvorenie pôvodného jazykového modelu bol použitý nástroj SRILM[14], ktorý je po súhlase s licenčnými podmienkami možné zadarmo získať na výskumné účely.

Na spracovanie textových dát a prípravu paralelného korpusu bolo využitých niekoľko skriptov. Niektoré boli vydané v rámci korpusu Europarl a sú tiež dodávané ako súčasť Mosesa. Ďalej som použil niekoľko vlastných skriptov, ktoré však väčšinou slúžili na jeden účel a nie sú podrobnejšie zdokumentované. Výnimku tvorí konzolová aplikácia, ktorej dokumentácia je v prílohe.

[12] http://www.kyloo.net/software/doku.php/mgiza:overview
[13] http://www.cs.cmu.edu/~qing/giza/
[14] http://www.speech.sri.com/projects/srilm/download.html

4 Realizácia systému

Obsahom tejto kapitoly bude popis zostavenia prekladového systému využitím predstavených nástrojov od vybudovania paralelného korpusu až po záverečné vyhodnotenie zvolenými metrikami.

4.1 Tvorba paralelného korpusu

Takmer všetky zdroje, z ktorých som čerpal pri zostavovaní korpusu boli dodávané vo formáte XML. Prvým krokom teda bolo odstránenie XML značiek z jednotlivých súborov. Pri vizuálnej kontrole bolo zistené, že korpus z filmových titulkov nie je v niektorých svojich častiach validný. V niektorých úsekoch si repliky vôbec nekorešpondovali, z čoho bol vyvodený záver, že v danom úseku boli k sebe priradené repliky z dvoch rôznych filmov. Manuálnou korekciou boli tieto úseky odstránené.

Ako ďalší som spracoval prekladový slovník. Poskytnutý súbor bol vo formáte, kde na prvom riadku bolo heslo v slovenčine a na druhom a treťom riadku to isté heslo v češtine. Súbor som rozdelil na dva, zvlášť s českými a zvlášť slovenskými heslami. Následne som tieto súbory rozšíril použitím českého synonymického slovníka. Pre každé české heslo bolo nájdené jedno alebo viac synoným, tieto boli doplnené medzi české heslá a k nim bol medzi slovenské heslá priradený preklad pôvodného českého hesla. Obidva súbory sú zarovnané po riadkoch. Pôvodný slovník zväčšil svoj objem po rozšírení z 149 879 hesiel na 258 015 hesiel.

Najnáročnejším pri tvorbe korpusu bolo spracovanie korpusu Europarl. Tento korpus je zarovnaný po vetách, pre každý jazykový pár v dvoch súboroch. Ako už bolo spomenuté, pre tento korpus sú dostupné len zarovnania s angličtinou. Jednotlivé zarovnania medzi sebou navyše veľkosťou nezodpovedajú. Pre konkrétny prípad češtiny a slovenčiny obsahoval česko-anglický korpus viac riadkov ako slovensko-anglický. Rozdiel bol spôsobený predovšetkým absenciou niektorých pasáží v slovensko-anglickom korpuse, ale aj tým, že kým v českej lokalizácii bola dlhá veta z angličtiny preložená ako súvetie, v slovenskej lokalizácii to boli dve vety na dvoch riadkoch. Experimentálne bolo zistené, že v prípade, ak sa existuje preklad anglickej vety do češtiny aj slovenčiny, rozdiel ich umiestnení v súboroch je maximálne 7000 riadkov v oboch smeroch. Preto bol navrhnutý algoritmus, ktorý na základe tohto kritéria nájde korešpondujúce dvojice viet a uloží ich do súboru so zarovnaním po riadkoch. Predtým však bolo potrebné zo súborov odstrániť často sa opakujúce vety, ktoré znižovali úspešnosť algoritmu (napr. zaužívané formuly pre otváranie zasadania a pod.). Proces zarovnania bol kvôli rozsahu súborov, počtu iterácií a operáciám nad reťazcami pomerne časovo náročný. Možnou, ale neimplementovanou heuristikou by bolo vyhľadávanie od pozície posledného úspešne zarovnaného páru. Je pravdepodobné, že po úspešnom zarovnaní bude nasledovať séria ďalších úspešných zarovnaní, až kým sa opäť nenarazí na absenciu, príp. rozdelenie na dve vety. Implementovaný algoritmus však splnil svoj účel.

Pri vytváraní korpusu z literatúry bolo nutné jednotlivé knihy previesť na jednotný textový formát. Ďalej bolo potrebné z kníh odstrániť niektoré údaje, ako napr. úvodné strany, doslovy prekladateľov či tiráž. Z dôvodov uvedených v kapitole 3.1.1 som však tento korpus nepoužil. Nie je však vylúčené jeho použitie v budúcnosti.

Jednotlivé vytvorené korpusy z rôznych domén som sa rozhodol spojiť do jedného veľkého korpusu rozdeleného na českú a slovenskú časť do dvoch súborov. Veľkosť výsledného korpusu je 4 766 281 zarovnaných riadkov, súbory obsahujú 51 309 358 českých tokenov a 51 381 335 slovenských tokenov[1].

[1] Merané príkazom wc -w na merlin.fit.vutbr.cz; podľa athena1 je to 51 484 931 (cs) a 51 545 850 (sk).

4.2 Príprava paralelného korpusu

Pred spustením tréningovej fázy bolo tiež nevyhnutné vykonať určité prípravné práce na korpuse. Tie spočívali v nasledovných činnostiach, ktoré boli vykonané pomocou hotových skriptov dodávaných v rámci Mosesa.

Prvým a dôležitým krokom bolo rozdeliť všetky vety v korpuse na tokeny. To znamená vložiť biely znak (medzeru) medzi slová a interpunkčné znamienka, príp. operátory. Dôležitosť tohto kroku spočíva v tom, že napr. interpunkčné znamienko na konci vety by bolo nesprávne považované za súčasť posledného slova.

Ďalším krokom je vyčistenie korpusu príliš dlhých a krátkych viet. Ako hranicu som zvolil minimálnu dĺžku 2 tokeny a maximálnu dĺžku 50 tokenov. Skript vykonávajúci čistenie odstraňuje aj prázdne riadky, tie by sa však v tejto fáze nemali už v korpuse nachádzať. Dôvodom odstránenia príliš dlhých viet (riadkov) je časová náročnosť ich spracovania programom GIZA++. Celkovo bolo odstránených 358 962 riadkov. Neskôr sa ukázalo, že minimálna dĺžka bola zvolená nevhodne, pretože sa tým z korpusu odstránili všetky jednoslovné heslá pridané do korpusu zo slovníka. Preto bol implementovaný nástroj, ktorý doplní v prípade potreby výstup štatistického prekladu o preklad slovníkový. Jeho popis bude uvedený v kapitole 4.6.

Ako nasledujúci krok je nutné vykonať jednu z nasledovných operácií. Prvou možnosťou je zmena všetkých písmen v korpuse na malé písmená, tzv. lowercasing. Druhou možnosťou je tzv. truecasing. Pri ňom sa najprv musia vytvoriť štatistiky z nespracovaného textu. Následne je každému slovu na základe štatistiky pridelená jeho pravdepodobná veľkosť písmen. Veta „Študent Peter DNES navštívil Bratislavu.“ by po aplikovaní truecasingu vyzerala takto: študent Peter dnes navštívil Bratislavu. Keďže použitý jazykový model bol vytvorený zo slov s malými písmenami, pre tento krok som zvolil lowercasing.

Na textových dátach už neboli vykonané ďalšie operácie ako napr. lemmatizácia alebo morfologická analýza (part-of-speech tagging). Jedným z dôvodov bola aj nedostupnosť plnohodnotných nástrojov pre slovenčinu.

Na záver bol korpus rozdelený na tri časti, 90% tvorí tréningovú sadu, 5% tvorí optimalizačnú sadu a 5% testovaciu sadu. Aby sa do každej sady dostali texty zo všetkých použitých korpusov, rozdelenie nie je spojité. Korpus sa teda delil takým spôsobom, že 90 riadkov išlo do tréningovej sady, nasledujúcich 5 riadkov do optimalizačnej sady a ďalších 5 riadkov do testovacej sady atď.

4.3 Trénovanie prekladového systému

Po vykonaní predošlých krokov môžeme pristúpiť k fáze tréningu. Ten obsluhuje Moses, konkrétne skript train-model.perl. Jeho činnosť je možné ovplyvniť širokou škálou parametrov. Ako symetrizačnú heuristiku som nastavil grow-diag-final-and.

Proces tréningu je časovo náročný, v mojom prípade trval približne 30 hodín na stroji athena3. Produktom tohto procesu je viacero súborov, najdôležitejšie sú však tri: tabuľka fráz, tabuľka preusporiadania a konfiguračný súbor pre dekodér. Konfiguračný súbor obsahuje cesty k týmto tabuľkám a k jazykovému modelu a váhy pre jednotlivé modely.

V tejto fáze už máme funkčný prekladový systém. Spustenie dekodéra však trvá pár minút. Preto je vhodné previesť jazykový model, tabuľku fráz aj tabuľku preusporiadania do binárnej podoby. Ich načítanie sa tak výrazne urýchli. Pre ilustráciu, veľkosť jazykového modelu sa z pôvodných 12,1 GB zmenšila na 7,27 GB. Pre jednorazový preklad veľkého objemu dát je tiež kvôli urýchleniu možné tabuľky vyfiltrovať, aby obsahovali iba dáta potrebné pre konkrétny preklad.

4.4 Optimalizácia váh

O správnosti prekladu rozhodujú pravdepodobnosti, ktoré sú prideľované štyrmi modelmi: tabuľkou fráz, tabuľkou preusporiadania, jazykovým modelom a penalizáciou dĺžky výstupu (viď rovnice 2.16 a 2.17). Dekodér pri zostavovaní prekladu používa váhy, ktoré určujú dôležitosť týchto modelov. Vygenerovaný konfiguračný súbor však obsahuje preddefinované hodnoty týchto váh.

V tejto fáze sa teda pokúsime nájsť optimálne váhy pre spomenuté modely, čo môže zlepšiť kvalitu prekladu. Použijeme na to pripravenú optimalizačnú sadu. Opäť využijeme Moses, ktorý pre optimalizáciu (tuning) podporuje viacero algoritmov. Preddefinovanou a mnou zvolenou metódou bol MERT[2]. Optimalizácia spočíva v niekoľkých iteráciách prekladu. Pri každom preklade sa pomocou vybranej metriky (BLEU) ohodnotí aktuálny preklad a nastavia sa nové váhy pre ďalšiu iteráciu. Proces sa opakuje, kým nenastane konvergencia, kedy už nie je možné dosiahnuť lepšie skóre. Optimalizácia nám však dosiahnutie lepšieho skóre negarantuje, taktiež jej vykonanie na rovnakých dátach nemusí skončiť rovnakým výsledkom.

Proces optimalizácie váh bol jednoznačne časovo najnáročnejší. Po spustení s 8 vláknami na stroji athena3 trval 23 dní. Celkovo bolo vykonaných 7 iterácií. Produktom bol nový konfiguračný súbor s optimalizovanými váhami. Ideálnejšie by bolo zvoliť menšiu optimalizačnú sadu, pretože aj 5% dát predstavuje veľké množstvo, čo malo značný vplyv na dĺžku procesu. Aký vplyv by to však malo na úspešnosť optimalizovaných váh je otázne.

4.5 Spracovanie výstupu dekodéra

Keďže sme dekodéru poskytli na vstup textové dáta s malými písmenami, je vhodné, aby ich veľkosť vrátila podľa možnosti do rovnakého alebo aspoň správneho tvaru. Tento proces nazývame recasing. Najprv je nutné podobne ako v prípade truecasingu natrénovať určité štatistiky z neupraveného (ale tokenizovaného) korpusu. Vytvorí sa tak nový model, ktorý sa potom použije v príslušnom skripte na zmenu veľkosti písmen. V prípade, že pôvodné slovo pred lowercasingom bolo v podobe, kedy obsahovalo všetky písmená kapitály, po aplikovaní recasingu sa vo väčšine prípadov nezachová, pretože to nie je najpravdepodobnejšia podoba slova.

Posledná úprava, ktorú je vhodné vykonať pred prezentáciou výstupu užívateľovi, je detokenizácia. Interpunkčné znamienka sa vrátia do prirodzenej podoby, kedy nie sú od slov oddelené medzerami. Detokenizácia, ale aj jej opačný proces, môže mať pre niektoré jazyky osobité výnimky, ktoré príslušný skript zohľadňuje, ak mu dáme informáciu o jazyku. Pre slovenský aj český jazyk je zoznam týchto výnimiek k dispozícii.

4.6 Slovníkový preklad neznámych slov

Výstupom dekodéra je okrem samotného prekladu aj niekoľko informácií, ktorých množstvo môžeme ovplyvniť príslušnými parametrami dekodéra. Medzi nimi nájdeme aj informáciu, ktoré slová dekodér nedokázal preložiť. Najčastejšie ide o slová, s ktorými sa stretol prvý krát, pretože sa nevyskytli v tréningovej sade. Taktiež však môže ísť o slová v rôznych pádoch, tvaroch a časoch, ktoré síce v inom tvare dekodér dokáže preložiť, ale nie je na toľko inteligentný, aby dokázal rozpoznať iný tvar natrénovaného slova. K tomu by bolo potrebné aplikovať na dáta ešte pred tréningom morfologickú analýzu, čo sme však neurobili.

Preddefinované správanie dekodéra pri nakladaní s neznámymi slovami je, že ich ponechá v nepreloženom tvare. Pre blízke jazyky ako čeština a slovenčina to nepredstavuje príliš veľký problém, niekedy je slovenský preklad dokonca rovnaký ako české slovo. Pre zaobchádzanie s neznámymi slovami však bol implementovaný skript translator.py, ktorý si popíšeme.

Skript bol napísaný v jazyku Python verzie 3. Okrem slovníkového prekladu vykonáva všetky potrebné úkony na preklad textu. Na vstupe očakáva text, ktorý sa bude prekladať. Skript ďalej spúšťa skripty tretích strán na tokenizáciu a lowercasing. Následne spustí dekodér, ktorý spracovaný text preloží. Voliteľným parametrom je možné určiť, či budeme požadovať aj slovníkový preklad.

Pri slovníkovom preklade sa najprv pomocou informácií od dekodéra identifikujú neznáme slová. Implementácia počíta aj s negovanými tvarmi slov, kedy prefix „ne“ pri vyhľadávaní neuvažuje. V prvej fáze sa pracuje so slovníkmi, ktoré boli pôvodne zaradené aj do korpusu. Nájdu sa všetky možné preklady neznámeho slova. Ak je takýchto možností viac, treba rozhodnúť, ktorý z nich

[2] Minimum Error Rate Training

je najpravdepodobnejší a zrejme aj najsprávnejší. To sa deje na základe viacerých kritérií. Ak je niektorý z možných prekladov zhodný ako české slovo, je tento preklad hneď vyhlásený za správny. V opačnom prípade je jednotlivým možnostiam prideľované skóre na základe ich podobnosti s českým slovom. Rozhodujúca je predovšetkým podobnosť dĺžky a podobnosť prvého a posledného písmena. Každá z podobností je ohodnotená iným skóre. Ak niektoré možné preklady dosiahnu rovnaké a zároveň najvyššie skóre, rozhodujúcim faktorom sa stáva ich frekvencia v slovenskom jazyku. Na to je využitá frekvenčná tabuľka slov, v ktorej sú slová zoradené podľa početnosti ich výskytu spolu s údajom o ich počte. Táto tabuľka je poskytovaná v rámci korpusu prim-5.0, ktorý sme spomínali v kapitole 3.2.

Do druhej fázy sa slovníkový preklad dostane v prípade, ak pre daný tvar slova nebolo v základnej verzii slovníka nájdené žiadne heslo. Na toto slovo je aplikovaný tzv. stemming. Ide o operáciu podobnú lemmatizácii, avšak nezískame základný tvar slova, ale jeho koreň. Pre každý jazyk je obvykle nutné vytvoriť vlastnú sadu pravidiel. Na toto je využitá portovaná verzia českého stemmera do Pythonu z Javy, ktorej autorom je Luís Gomes. Stemmovanie slova sa vykoná v dvoch režimoch (light a aggressive). Slovo sa najprv hľadá vo vopred pripravenom ostemmovanom slovníku v režime light, pri neúspechu prejde na agresívny režim. Z nájdených možných prekladov je najpravdepodobnejší vybratý rovnakým spôsobom ako v prvej fáze.

V poslednej fáze sú neznáme slová nahradené ich slovníkovými prekladmi. Parametrom je možné nastaviť, aby sa slovníkový preklad, ktorý vznikol pomocou stemmovania obalil do zátvoriek hneď za neznámym slovom. V prípade, že ani jedna fáza pri hľadaní slovníkového prekladu neuspeje, je vo vete ponechané neznáme slovo v nepreloženom tvare. Samotné stemmovanie je tiež možné parametrom zakázať.

Po skončení, príp. vynechaní slovníkového prekladu je ďalšou voliteľnou operáciou recasing. Na záver je preklad detokenizovaný a zapísaný do zvoleného súboru alebo na štandardný výstup. Skript počas svojej činnosti vytvára niekoľko dočasných súborov, ktoré však priebežne vymazáva.

Návod na použitie skriptu translator.py je súčasťou prílohy.

5 Vyhodnotenie

Pre objektívne posúdenie kvality prekladu, ktorý produkuje náš systém, použijeme metriku BLEU. Vyhodnotenie prebehlo na niekoľkých testovacích vzorkách. Prvou bola testovacia sada vyčlenená z testovacieho korpusu. Ďalšími boli text Varšavskej zmluvy[1], jeden z protokolov Lisabonskej zmluvy[2] a novinový článok[3]. Pre všetky vzorky okrem novinového článku boli k dispozícii referenčné preklady. Pre novinový článok bol vytvorený vlastný referenčný preklad.

Ideálnejším prípadom by bolo, ak by sme mali k dispozícii viacej referenčných prekladov. Tieto preklady by však nemali byť strojové, preto je ich získanie častokrát náročné. Pre porovnanie však budú uvedené aj výsledky, ktoré dosiahol iný štatistický prekladový systém, konkrétne Google Translate a Česílko. V dostupnom online prekladovom demo systéme Česílko[4] je limit na prekladaný text 5000 znakov, preto na testovaciu sadu z korpusu nebude použitá. Uvedené výsledky slúžia výlučne na porovnanie, nie na vyhodnotenie, ktorý systém je lepší, nakoľko metrika BLEU nie je vhodná pre porovnávanie systémov s odlišnou stratégiou prekladu.

Texty, ktoré boli predkladané prekladovým systémom boli najprv prekonvertované na malé písmená a tokenizované. Na ich výstup bol aplikovaný recasing použitím natrénovaného modelu. Vyhodnotenie prebiehalo na nemodifikovanom výstupe aj na výstupe s aplikovaným recasingom.

Vyhodnotenia budú zhrnuté v tabuľkách s komentárom použitých modelov a dosiahnutých výsledkov. Na meranie BLEU skóre bol použitý skript multi-bleu.perl dodávaný k Mosesu.

Prekladový systém	**Testovacie vzorky**			
	Testovacia sada	Varšavská zm.	Lisabonská zm.	Novinový čl.
Moses	0,4427	0,5326	0,4400	0,4138
Moses -lc	0,4530	0,6239	0,4583	0,4239
Google Translate	0,4601	0,7012	0,7228	0,8014
Google Translate -lc	0,4685	0,7523	0,7361	0,8014
Česílko	–	0,5469	0,3087	0,6321
Česílko -lc	–	0,6508	0,3228	0,6657

Tabuľka 4.1: Porovnanie prekladových systémov

Označenie -lc znamená, že dáta boli vyhodnocované s aplikovaným lowercasingom. V prípade dekodéru Moses boli použité preddefinované neoptimalizované váhy. Použitý bol nebinarizovaný jazykový model prim-5.0-sane-lowercase.

Rovnaké vyhodnotenie prebehlo aj s tým istým jazykovým modelom v binárnej podobe. Výsledky sú prekvapivo odlišné, vo všetkých prípadoch okrem jedného horšie. Rozdiely sú však pomerne malé, daňou za rýchlejší preklad je teda mierne zhoršenie skóre.

Prekladový systém	**Testovacie vzorky**			
	Testovacia sada	Varšavská zm.	Lisabonská zm.	Novinový čl.
Moses	0,4422	0,5328	0,4368	0,4083
Moses -lc	0,4526	0,6257	0,4564	0,4163

Tabuľka 4.2: Dosiahnuté skóre s jazykovým modelom v binárnej podobe

[1] http://smsjm.vse.cz/wp-content/uploads/2008/10/sp29.pdf (cs), http://www.zakonypreludi.sk/zz/1955-45 (sk)
[2] http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:C:2007:306:0148:0148:CS:PDF (cs), http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:C:2007:306:0148:0148:SK:PDF (sk)
[3] http://zpravy.idnes.cz/kimovy-rude-gardy-vymenily-pusky-za-lopaty-a-na-polich-sbiraji-mrvu-11j-
[4] http://quest.ms.mff.cuni.cz/cesilko/index.php

Pre zaujímavosť uvádzam aj dosiahnuté NIST skóre na testovacej sade z korpusu. Pre účely merania bolo nutné všetky vstupné dáta prekonvertovať do formátu SGML. Nástroj mteval-v12.pl meria taktiež BLEU skóre, ale mierne odlišným spôsobom, čo spôsobuje malý rozdiel v skóre. Testovacia sada dosiahla NIST skóre 11.4704 a BLEU skóre 0.4486.

Ďalšie meranie spočívalo v tom, že dekodér pri preklade tentoraz použil menší jazykový model zostavený zo slovenskej časti paralelného korpusu. Cieľovom bolo overiť tvrdenie, že kvalita prekladu závisí na veľkosti jazykového modelu [16]. Pri meraní nebola použitá testovacia sada.

Prekladový systém	**Testovacie vzorky**		
	Varšavská zm.	Lisabonská zm.	Novinový čl.
Moses	0,4049	0,8132	0,5475
Moses -lc	0,4760	0,8287	0,5675

Tabuľka 4.3: Dosiahnuté skóre s jazykovým modelom join.clean.low.sk

Toto tvrdenie prekvapujúco nebolo potvrdené. Preklad prvej testovacej vzorky dosiahol síce horšie skóre, zvyšné dva však napriek očakávaniam majú skóre lepšie. V prípade Lisabonskej zmluvy ide o výrazný nárast. Ako možné vysvetlenie sa ponúka hypotéza, že jazykový model bol zostrojený aj z dát, ktoré sú z príbuznej domény ako Lisabonská zmluva.

Doteraz boli pre všetky merania použité preddefinované váhy. V nasledujúcom meraní vyhodnotíme úspešnosť optimalizácie váh. Vrátime sa k jazykovému modelu prim-5.0-sane-lowercase v binárnej podobe.

Prekladový systém	**Testovacie vzorky**			
	Testovacia sada	Varšavská zm.	Lisabonská zm.	Novinový čl.
Moses	0,7855	0,4845	0,8157	0,6250
Moses -lc	0,7989	0,5734	0,8354	0,6537

Tabuľka 4.4: Dosiahnuté skóre po optimalizácii váh

V troch prípadoch je možné pozorovať výrazné zlepšenie skóre. V prípade Varšavskej zmluvy sme však zaznamenali pokles. Ide už o tretí experiment, kedy sa skóre tejto vzorky nejakým spôsobom odchyľovalo od zvyšných vzoriek. Pravdepodobne to môžeme prisúdiť špecifickosti domény, z ktorej pochádza. Pôvodný predpoklad však bol, že Varšavská a Lisabonská zmluva budú dosahovať podobné výsledky.

Ďalším experimentom nadviažeme na pokus s použitím jazykového modelu join.clean.low.sk. Pôvodná preddefinovaná váha, ktorá určovala dôležitosť tohto modelu pri preklade bola 0.5000, po optimalizácii má hodnotu 0.00631481. Aký vplyv to malo na výsledné skóre, znázorňuje tabuľka 4.5.

Prekladový systém	**Testovacie vzorky**		
	Varšavská zm.	Lisabonská zm.	Novinový čl.
Moses	0,4661	0,8240	0,5906
Moses -lc	0,5394	0,8420	0,6181

Tabuľka 4.5: Dosiahnuté skóre po optimalizácii vás s jazykovým modelom join.clean.low.sk

Optimalizované váhy opäť zvýšili dosiahnuté skóre. Pri porovnaní hodnôt z tabuliek 4.4 a 4.5 však nemôžeme konštatovať, že kombinácia optimalizovaných váh a jazykového modelu join.clean.low.sk je najideálnejšia. Výsledné skóre bude vždy závisieť na doméne prekladaného textu.

Posledný experiment mal za úlohu overiť úspešnosť doplnkového slovníkového prekladu. Použitý bol model prim-5.0-sane-lowercase, otestované boli režimy s povoleným aj zakázaným

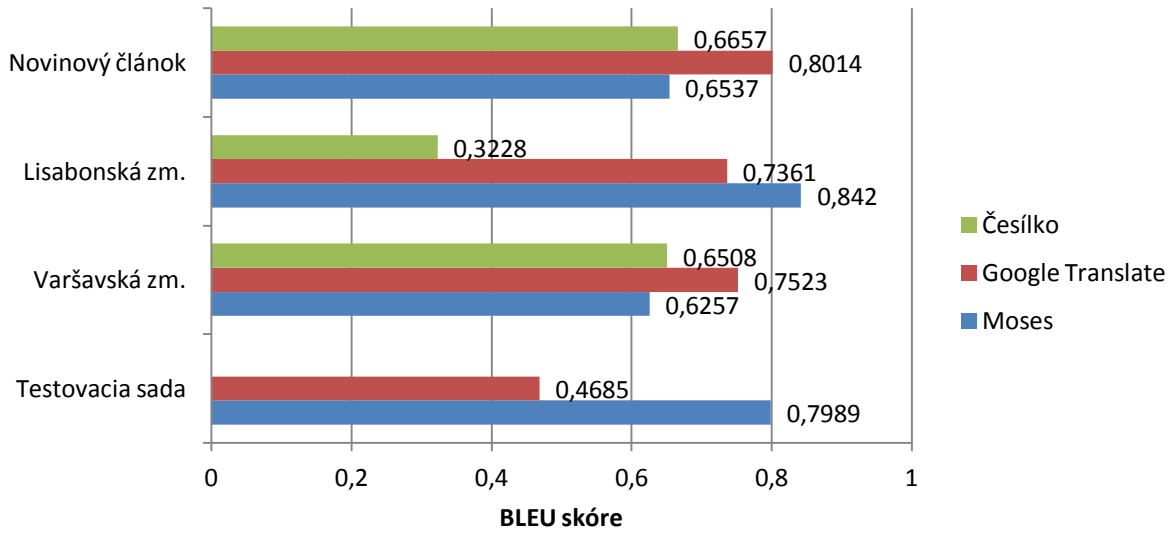
stemmingom. Testovacie sady boli iba vo verzii s malými písmenami. Označenie -voc znamená aplikáciu slovníkového prekladu, -ns znamená zakázaný stemming.

Prekladový systém	**Testovacie vzorky**			
	Testovacia sada	Varšavská zm.	Lisabonská zm.	Novinový čl.
Moses -lc -voc	0,4523	0,6236	0,4583	0,4022
Moses -lc -voc -ns	0,4523	0,6239	0,4583	0,4163
Moses -lc	0,4530	0,6239	0,4583	0,4239

Tabuľka 4.6: Dosiahnuté skóre po aplikácii slovníkového prekladu

Z tabuľky vyplýva, že stemming ma negatívny vplyv na výsledné skóre. Dôvod je pravdepodobne ten, že pri úspešnom preklade je dosadené slovo v základnom tvare. Celkovo aplikácia slovníkového prekladu vykazuje zhoršenie skóre, rozdiel však nie je veľký. Je však možné, že zvolené testovacie vzorky neodhalili jeho potenciál, malé alebo žiadne rozdiely v skóre by tomu mohli nasvedčovať.

Nasledovný graf znázorňuje súhrn najlepšieho dosiahnutého skóre pre porovnávané systémy.



Graf 4.1: Prehľad najlepšieho dosiahnutého skóre

6 Záver

V rámci tejto bakalárskej práce bol navrhnutý a implementovaný systém pre preklad českých textov do slovenčiny.

V prvej fáze som sa zoznámil s teoretickým pozadím a technikami štatistického strojového prekladu. Oblasť strojového prekladu sa neustále vyvíja a vznikajú nové prístupy k problémom, ktoré strojový preklad sprevádzajú. V tejto práci je venovaný priestor predovšetkým tým technikám, ktoré boli neskôr využité v ďalšej práci pri implementácii. Pre ďalší vývoj som sa rozhodol použiť prístup založený na vytvorení frázových tabuliek.

Nevyhnutným predpokladom vývoja štatistického prekladového systému bolo vybudovanie paralelného korpusu. Zoznámil som sa s existujúcimi korpusmi, analyzoval vhodnosť ich spracovania a taktiež som zozbieral ďalšie textové dáta pre vytvorenie nových korpusov. Budovanie korpusu si vyžadovalo operácie ako rozdelenie viet na tokeny, konverzia na malé písmená a pod. V ostatných prácach som sa stretol s postupom, kedy štatistické prekladové modely boli trénované zvlášť na menších korpusoch z rôznych domén. Ja som sa rozhodol všetky získané korpusy spojiť a vybudovať jeden veľký korpus. Taktiež som sa rozhodol použiť rozsiahly jazykový model z miestneho jazykovedného ústavu, vychádzajúc z tvrdenia že veľkosť jazykového modelu má pozitívny vplyv na výslednú kvalitu prekladu.

Nasledovala fáza implementácie. Najdôležitejším využitým nástrojom bol systém Moses, pomocou ktorého je možné zostaviť prekladový systém pre ktorúkoľvek dvojicu jazykov. Zhotovený prekladový systém dosahoval pomerne dobré výsledky, avšak používal preddefinované váhy jednotlivých modelov. Preto ďalším krokom bola optimalizácia váh. Tento proces bol časovo najnáročnejší. Čas som preto využil na testovanie dekodéra s neoptimalizovanými váhami. Taktiež bol implementovaný doplnkový slovníkový preklad neznámych slov. Pre automatizáciu potrebných úkonov na preklad textu som navrhol a napísal konzolovú aplikáciu, ktorá s využitím iných skriptov postupne pripraví textové dáta na preklad, preloží ich a vykoná ďalšie úpravy pred tým, ako bude preklad prezentovaný užívateľovi.

Posledným krokom bolo vyhodnotenie úspešnosti prekladu. Za smerodajnú som zvolil metriku BLEU. Implementovaný prekladový systém dosahoval pomerne vysoké skóre porovnateľné s inými systémami. Optimalizácia váh taktiež priniesla očakávaný výsledok v podobe zvýšenia skóre. Pri vyhodnocovaní a experimentovaní som však narazil na viacero paradoxov, ktoré som sa v práci snažil odôvodniť. Napríklad sa nepreukázalo, že by veľkosť modelu výrazne vplývala na kvalitu prekladu.

Už počas návrhu a vývoja som sa niekoľkokrát rozhodoval medzi použitím odlišných prístupov. Tieto tak predstavujú potenciál do budúceho vývoja. Jednou z možností je namiesto fráz využiť modely založené na stromoch. Ďalšou možnou variáciou by mohlo byť použitie inej symetrizačnej heuristiky. Taktiež výber použiteľných nástrojov je širší, niektoré z nich by mohli prinajmenšom znížiť časovú náročnosť niektorých operácií. Sľubne, avšak náročne vyzerá aplikácia morfologickej analýzy. V čase písania tejto práce už je k dispozícii aj nový, rozsiahlejší slovenský korpus prim-6.0. Potenciál má aj rozširovanie paralelného korpusu, v menších množstvách sa paralelné texty dajú nájsť v rôznych zdrojoch. V neposlednom rade zostáva nezodpovedaná otázka, aký vplyv okrem zníženia časovej náročnosti by malo použitie menšej sady pri optimalizácii váh dekodéra.

Literatúra

[1] History of machine translation. [online]. 9. 4. 2013 [cit. 2013-04-20]. Dostupné z: http://en.wikipedia.org/wiki/History_of_machine_translation

[2] DORR, B.; HOVY, E.; LEVIN, L. Machine Translation: Interlingual Methods. In: *Encyclopedia of Language & Linguistics (Second Edition)*. BROWN, K. ed. Oxford: 2006, s. 383–94.

[3] MANNING, C. D.; SCHÜTZE, H. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999. ISBN 0-262-13360-1.

[4] Machine translation. [online]. 10. 4. 2013 [cit. 2013-04-20]. Dostupné z: http://en.wikipedia.org/wiki/Machine_translation

[5] Rule-based machine translation. [online]. 3. 3. 2013 [cit. 2013-20-04]. Dostupné z: http://en.wikipedia.org/wiki/Rule-based_machine_translation

[6] Example-based machine translation. [online]. 7. 3. 2013 [cit. 2013-04-20]. Dostupné z: http://en.wikipedia.org/wiki/Example-based_machine_translation

[7] WEAVER, W. Translation (1949). In: *Machine Translation of Languages*. Cambridge, MA: MIT Press, 1955.

[8] BROWN, P. et al. The mathematics of statistical machine translation: parameter estimation. In: *Computational Linguistics*, sv. Vol. 19, No. 2. 1993, s. 263-311.

[9] JURAFSKY, D.; MARTIN, J. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational …*. Second Edition. Prentice Hall, 2008. ISBN 978-0-13-187321-6.

[10] TÓTH, K.; FARKAS, R.; KOSCOR, A. Sentence alignment of Hungarian-English parallel corpora using a hybrid algorithm. *Acta Cybernetica*. 2008, č. Vol. 18, No. 3. s. 463-78.

[11] VOGEL, S.; NEY, H.; TILLMANN, C. HMM-based word alignment in statistical translation. In: *COLING '96 Proceedings of the 16th conference on Computational linguistics*, sv. Volume 2. 1996, s. 836-41.

[12] OCH, F. J.; NEY, H. A systematic comparison of various statistical alignment models. Cambridge, MA, USA: MIT Press, 2003, č. Volume 29 Issue 1, s. 19-51.

[13] KOEHN, P. In: *Statistical Machine Translation System. User Manual and Code Guide* [online]. [cit. 2013-04-24]. Dostupné z: http://www.statmt.org/moses/manual/manual.pdf

[14] KOEHN, P. et al. Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation. In: *Proceeding of International Workshop on Spoken Language Translation*. Pittsburgh, PA: 2005.

[15] KOEHN, P.; OCH, F. J.; MARCU, D. Statistical phrase-based translation. In: *NAACL '03 Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, sv. Volume 1. s. 48-54.

[16] BRANTS, T. et al. Large Language Models in Machine Translation. In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Praha: 2007, s. 858–67.

[17] PAPINENI, K. et al. BLEU: a method for automatic evaluation of machine translation. In: *ACL '02 Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. 2002, s. 311-18.

[18] CALLISON-BURCH, C.; OSBORNE, M.; KOEHN, P. Re-evaluation the Role of Bleu in Machine Translation Research. In: *Proceedings of EACL*. 2006, s. 249-56.

[19] DODDINGTON, G. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In: *HLT '02 Proceedings of the second international conference on Human Language Technology Research*. 2002, s. 138-45.

[20] BANERJEE, S.; LAVIE, A. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In: *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. 2005, s. 65–72.

[21] KOEHN, P. *Europarl: A Multilingual Corpus for Evaluation of Machine Translation*. Marina del Rey, CA, USA: 2002. University of Southern California, Information Sciences Institute.

[22] HAJIČ, J.; HRIC, J.; KUBOŇ, V. Machine Translation of Very Close Languages. In: *6th ANLP Conference / 1st NAACL Meeting. Proceedings*. Seattle, Washington: 2000, s. 7-12.

[23] STEINBERG, R. et al. The JRC-Acquis: A Multilingual Aligned Parallel Corpus with 20+ Languages. In: *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'2006)*. Janov: 2006, s. 2142-47.

[24] TIEDEMANN, J. News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces. In: NICOLOV, N. et al. *Recent Advances in Natural Language Processing V*. Amsterdam/Philadelphia: John Benjamins, 2009, s. 237-48.

Prílohy

A Manuál konzolovej aplikácie

Konzolová aplikácia je tvorená skriptom translator.py napísaným v jazyku Python verzie 3. Vykonáva všetky potrebné činnosti pre preklad textu: príprava textu (lowercasing, tokenizácia), spustenie dekodéra, slovníkový preklad, úprava výstupu (recasing, detokenizácia).

Popis argumentov príkazového riadku

- `-h, --help` – zobrazenie nápovedy, nápoveda sa zobrazí aj pri nesprávnej kombinácii parametrov alebo pri spustení aplikácie bez parametrov
- `--config` – povinný parameter, jeho argumentom je cesta ku konfiguračnému súboru pre dekodér
- `--norec` – voliteľný prepínač, ak nie je zadaný, výstup je automaticky recasovaný
- `--voc` – voliteľný prepínač, aktivuje slovníkový preklad pre neznáme slová
- `--ns` – voliteľný prepínač, viazaný na `--voc`, zakazuje stemming
- `--br` – voliteľný prepínač, viazaný na `--voc`, slovníkový preklad obalí zo zátvoriek
- `-o, --output` – voliteľný parameter, jeho argumentom je názov výstupného súboru; ak nie je zadaný, výstup je presmerovaný na štandardný výstup
- `--tmp` – cesta k dočasným súborom, vytváraným počas prekladu (vhodné pre spustenie z DVD)

Požiadavky

Program pre svoje spustenie vyžaduje nainštalovaný Python verzie 3. Predpokladá sa existencia jazykového modelu, tabuľky fráz a preusporiadania, na ktoré správne odkazuje konfiguračný súbor dekodéra.

Program je odporúčané spúšťať z jeho aktuálneho umiestnenia. Program je dodávaný s nasledovnými skriptami, ktoré kvôli funkčnosti musia byť umiestnené v rovnakej zložke ako translator.py:

- moses – spustiteľný súbor dekodéra
- tokenizer.perl (© Pidong Wang)
- detokenizer.perl (© Josh Schroeder)
- zložka nonbreaking_prefixes s príslušnými súbormi pre činnosť prvých dvoch skriptov
- recase.perl – štandardná súčasť Mosesa
- lowercase.perl – štandardná súčasť Mosesa
- dict.cs, dict.sk, dict.stm.cs, dict.stm.agr.cs – slovníky v zložke dicts
- czech_stemmer.py (© 2010 Luís Gomes)

Príklady spustenia

```
python3 translator.py --config model/config.ini --voc --br <
textovy_subor

echo "Ahoj, prekladač!" | python3 translator.py --config
model/config-tuned.ini -o decode.out --norec
```

B Ukážky prekladu[1]

Veta na vstupe	Preklad	Referenčný preklad
Předseda sboru rozhodců stanoví den a hodinu prvního slyšení.	Predseda arbitrážneho senátu stanoví dátum a čas prvého vypočutí.	Predseda arbitrážneho senátu stanoví dátum a čas prvého pojednávania.
Na zadních sedadlech udělal malý incident, ale už jsem to vyčistil.	Na zadných sedadlách spôsobil malý incident, ale už som to vyčistil.	Na zadných sedadlách spôsobil malý incident, ale už som to vyčistil.
Pragmatismus našich činů musí jít ruku v ruce s vědomím , že budujeme Unii pro lidi a zásluhou lidí.	Pragmatizmus našich činov musí ísť ruka v ruke s vedomím, že budujeme Úniu pre ľudí a vďaka ľudí.	Pragmatizmus našich činností sa musí spájať s pochopením, že budujeme Úniu pre ľudí a vďaka ľuďom.
S jadernou apokalypsou to zřejmě nebude zas tak žhavé, jak Pchjongjang v posledních dnech hrozí.	S jadrovou apokalypsou to nebude zas vám dlho netrvalo, ako Pchjongjang hrozí. V posledných dňoch.	S jadrovou apokalypsou to zrejme nebude zase také horúce, ako Pchjongjang v posledných dňoch hrozí.
Čche byl povolán do parašutistického oddílu v roce 1968, když Severní Korea zadržela americkou loď USS Pueblo.	Čche bol zavolaný do parašutistického časti v roku 1968, keď Severná Kórea zadržala americkú loď USS Pueblo.	Čhce bol povolaný do parašutistického oddielu v roku 1968, keď Severná Kórea zadržala americkú loď USS Pueblo.
Výjimky jsou možné v případě naléhavosti, přičemž se důvody pro ně uvedou v aktu nebo postoji Rady.	Výnimky sú prípustné v naliehavých prípadoch, ktorých dôvody sa uvedú v akte alebo v pozícii Rady.	Výnimky sú prípustné v naliehavých prípadoch , ktorých dôvody sa uvedú v akte alebo v pozícii Rady.
Dojde-li v Evropě k ozbrojenému útoku proti jednomu státu nebo několika státům zúčastněným na Smlouvě se strany kteréhokoli státu nebo skupiny států, každý stát zúčastněný na Smlouvě na základě práva na individuální nebo kolektivní sebeobranu, v souhlase s článkem 51 Charty Organisace Spojených národů, poskytne státu nebo státům, které byly takto napadeny, okamžitou pomoc, individuálně i v dohodě s ostatními státy zúčastněnými na Smlouvě a to všemi prostředky, které považuje za nutné, včetně použití ozbrojené síly.	Ak sa v Európe k ozbrojenému útoku proti jednému štátu alebo niekoľkých štátom zaangažované v zmluve sa strany ktoréhokoľvek štátu alebo skupiny štátov, každý štát zúčastnený na zmluve na základe práva na individuálne a kolektívne sebaobranu, v súlade s článkom 51 Charty organisace Spojených národov, poskytne štátu alebo štátom, ktoré boli takto napadnuté, okamžitú pomoc, zhodnotil aj v dohode s ostatnými štátmi zúčastnenými na zmluve a to všetkými prostriedkami, ktoré považuje za potrebné vrátane použitia ozbrojené sily.	Ak dôjde v Európe k ozbrojenému útoku proti jednému štátu alebo niekoľkým štátom zúčastneným na Zmluve zo strany ktoréhokoľvek štátu alebo skupiny štátov , každý štát zúčastnený na Zmluve na základe práva na individuálnu alebo kolektívnu sebaobranu , v súhlase s článkom 51 Charty Organizácie spojených národov , poskytne štátu alebo štátom , ktoré boli takto napadnuté , okamžitú pomoc , individuálne i po dohode s ostatnými štátmi zúčastnenými na Zmluve , a to všetkými prostriedkami , ktoré považuje za nutné , včítane použitia ozbrojenej sily
President Československé republiky - VILIAMA ŠIROKÉHO	President Česko-slovenskej republiky - Viliama širokospektrálneho	Prezident Československej republiky - VILIAMA ŠIROKÉHO

[1] Použitý jazykový model prim-5.0-sane-lowercase, dekodér s optimalizovanými váhami

C Popis obsahu DVD

Na priloženom médiu sa nachádza text tejto práce a vytvorený systém vrátane všetkých potrebných súčastí. V koreňovom adresári sa tiež nachádza súbor README, ktorý obsahuje všetky potrebné informácie pre zaobchádzanie s týmto médiom a jeho obsahom.

Adresár thesis obsahuje text tejto práce vo formáte PDF, zdrojový súbor textu vo formáte DOCX a plagát prezentujúci prácu vo formáte PDF.

Adresár translator obsahuje vytvorený systém. Pre jeho obsluhu je vhodné preštudovať súbor README v koreňovom adresári, príp. prílohu A vyššie. Kvôli funkčnosti systému je dôležité nemeniť vytvorenú hierarchiu súborov. Taktiež sa odporúča systém spúšťať z jeho aktuálneho umiestnenia.

Systém na priloženom médiu má niektoré obmedzenia vyplývajúce z obmedzenej kapacity nosiča DVD. Nebolo možné priložiť modely v binárnej podobe, čo má výrazne negatívny vplyv na rýchlosť dekodéra, ktorého činnosť trvá neúmerne dlho kvôli načítavaniu potrebných dát. Plnohodnotná verzia systému vrátane binárnych modelov a pripravených konfiguračných súborov sa nachádza v /mnt/minerva1/nlp/projects/mt_sk2/translator. V prípade spúšťania z DVD je vhodné venovať pozornosť parametru --tmp.

Okrem nevyhnutných súčastí vytvoreného systému sa v adresári translator nachádza vytvorený paralelný korpus rozdelený na tri časti v pomere 90:5:5. Ďalej je tu možné nájsť testovacie vzorky textov, vrátane výstupov vytvoreného systému ale aj porovnávaných systémov a tiež referenčné preklady. V adresároch training a tuning sa nachádzajú záznamy fázy trénovania a optimalizácie váh.

Doteraz nespomenuté adresáre a súbory sú súčasťou prekladového systému.