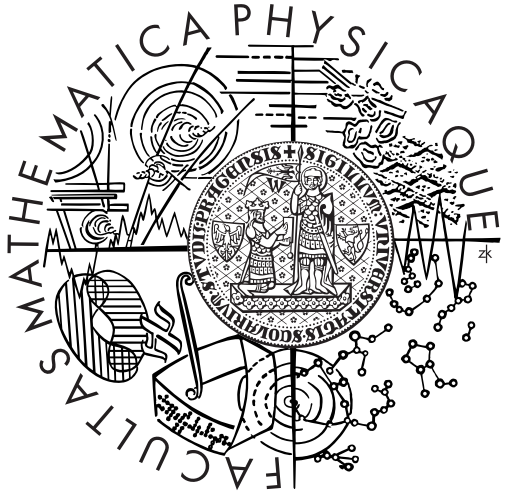
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Ján Zahornadský

Rozpoznávání jazyka na krátkém vzorku textu

Ústav formální a aplikované lingvistiky

Vedoucí bakalářské práce: Mgr. Eduard Bejček

Studijní program: Obecná informatika

2008

Na tomto místě bych rád poděkoval svému vedoucímu, Eduardu Bejčkovi, za jeho obětavou pomoc a rady při prakticky čemkoliv, na co jsem během psaní práce narazil. Taky děkuji Josefu Tomanovi za poskytnutí podnětné bakalářské práce. Děkuji rovněž všem ostatním, bez kterých by tato práce nemohla být dokončena a kteří mě nemalou mírou podpořili.

Prohlašuji, že jsem svou bakalářskou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne 6. srpna 2008 Ján Zahornadský

Obsah

Název práce:	Rozpoznávání jazyka na krátkém vzorku textu
Autor:	Ján Zahornadský
Katedra (ústav):	Ústav formální a aplikované lingvistiky
Vedoucí bakalářské práce:	Mgr. Eduard Bejček
e-mail vedoucího:	bejcek@ufal.mff.cuni.cz

Abstrakt: Práce navazuje na publikaci N-gram Text Categorization pánů Cavnara a Trenkla [2], rozšiřuje studium možnosti aplikace statistických metod na určení jazyka dokumentu jako jednodušší kategorie. Zde navrhovaný program čítá několik předností, mimo jiné modularita přiloženého programu a realizace nad jednou zvolenou univerzální znakovou sadou. Jako vylepšení původní práce je použito automatického filtrování a extrakce textu z internetových stránek pro průběžné zpřesňování naučených dat. Studujeme vývoj přesnosti určování jazyka dokumentu se zaměřením na krátké textové úseky. Tento aspekt jsme nenalezli v dostupné literatuře, jelikož kategorizace textu se ve většině případů zabývá dostatečně dlouhými vstupy. V závěru diskutujeme o možnosti použít naučená data a přístup pro označení a určení případných z jiného jazyka vnesených slov nebo krátkých frází. Konkrétně zkoumáme aplikaci na pražský závislostní korpus [8], kde tyto cizojazyčné fráze nejsou určeny, jen vymezeny.

Klíčová slova: n-gram, jazyk dokumentu, data mining, web

Title:	Language Recognition on Short Text Sample
Author:	Ján Zahornadský
Department:	Institute of Formal and Applied Linguistics
Supervisor:	Mgr. Eduard Bejček
Supervisor's e-mail address:	bejcek@ufal.mff.cuni.cz

Abstract: This paper extends the work of Cavnar and Trenkle N-gram text categorization [2], enhances the study of statistics application on document language recognition as simplier variant of categorization. Proposed program shows qualities like modular design or running on one universal character set. As an enhancement of the original work is presented an automatic text sample filtration algorithm altogether with Internet text extraction and iterative improvement for this purpose. Presented paper studies accuracy development, concentrating on short samples. Similar work was not found in available literature, as categorization (and in corollary language recognition) usually assumes long enough input. In conclusion, a discussion about using the learned data and algorithms created here to mark foreign phrases. To be specific, we study the application on Prague Dependency Treebank [8], where the foreign phrases are not recognized, only their occurences specified.

Keywords: n-gram, document language, data mining, web

Kapitola 1

Úvod

Ačkoli se zdá, že určení nejenom jazyka, ale dokonce i kategorie textu je proveditelné a při dostatečně rozsáhlých trénovacích a vstupních datech i relativně přesné (ukázkou jiných prací budiž třeba [1] nebo [2]), je stále nejisté, jestli je možné vůbec určit totéž na krátké frázi, ne-li slovu. Je taky otázkou, jak rozsáhlý kulturní a textový kontext je potřeba.

Představme si tříznakovou sekvenci písmen *a*, *l* a *e*, „ale“. Zeptat se čtenáře této práce (který je velmi pravděpodobně z českého prostředí), označil by to za českou spojku. Kdežto snad až překvapivě vede první odkaz na vyhledávači Google na stránku o anglickém pivu a druhý na stránky Acquisition Logistics Engineering. Mezi dalšími odkazy nacházíme Atlanta Linux Enthusiasts. Tyto výsledky se dají vysvětlit tím, že jsme začínali na stránkách `www.google.com`. Zkusme tedy českou domovskou stránku této společnosti – první výsledek zůstává stejný, lehce alkoholický nápoj běžný v Belgii nebo ve Spojeném království. Hned druhý však už potvrzuje prvotní domněnku, že jde o české slovo – název stránky „Malé (ale naše) noviny“ rozbíjí případné pochyby o výskytu popisovaného slova v češtině.

Vrátíme se teď k původní otázce – je obtížné poznat krátký vzorek a identifikovat jeho lingvistický původ? Diskutovaný příklad slova „ale“ nás přesvědčuje, že se nejedná o trivialitu. Tato tříznaková sekvence je totiž nejen naprosto přirozená pro jazyk nám vlastní – češtinu, příbuznou slovenštinu, ale určitě kdejaký Angličan by toto slovo bez okolků označil za své. Sám o sobě vzorek poskytuje velmi málo informací a relativně úspěšná analýza, kterou předvádí vyhledávací servery, záleží na dalších (často bez vědomí uživatele zadaných) dodatečných dat. Třeba země, ze které se člověk připojuje, respektive různé historie chování ať už ve formě cookies nebo přímo uložené v uživatelském účtu poskytovaném se službou.

V této práci se tedy pokusíme prozkoumat zaběhnuté praktiky pro kategorizaci textu z pohledu krátkých vstupních dat. Ověříme, zda a nakolik je postačující rozšiřovat možnosti algoritmů (blíže popsaných v kapitole 4) rozšířením vstupních učicích dat.

Hned první problém v popsané úvaze nastává při získávání dostatečně rozsáhlých dat. Rozhodnutí padlo na v současnosti pravděpodobně nejrozsáhlejší volně dostupný zdroj různojazyčných textů – internetové stránky. Je bohužel pravda, že tyto nemůžeme v žádném případě považovat za kvalitní prameny, proto navrhujeme několik úrovní a způsobů filtrování. Navrhneme způsoby, jak pokud možno maximalizovat efektivitu získávání informací a zkusíme natrénovat takový systém. Poté provedeme diskusi, snažíme se zachytit možné problémy a navrhnout jejich řešení. Také zkusíme, do jaké míry je systém vhodný na cizí fráze obsažené v Pražský závislostním korpusu.

Kapitola 2

Stav dosavadního výzkumu

Dříve, než se pustíme do samotného popisování metod, které aplikujeme v této práci, zběžně se zmíníme o modelech a výzkumu, na který navazujeme nebo se na ně odkazujeme.

2.1 N-gram a Markovův řetězec

Velmi často a se značným úspěchem se pro kategorizaci psaného textu používaná n-gramová metoda – značná úspěšnost se dostavuje už při vzorcích cca 300 znaků [1] a na několika stránkách textu dosahují podle [5] až přes 65% přesnosti určení autora (ovšem, údaj je závislý na mnoha dalších okolnostech).

Jedná se o přístup, kdy předpokládáme, že pravděpodobnost výskytu daného písmena (slova) je dána posloupností písmen předcházejících, formálně jde o $P\left(x_i|x_{i-1}x_{i-2}\cdots x_{i-n}\right)$. Je to zároveň jistá aproximace problému předpokládající, že

1. Pravděpodobnost výskytu každé jednotky je úměrná pouze určitému počtu jednotek před ní. Formálně

$$P(X_i = x|X_{i-1}X_{i-2}\cdots X_1) \approx P(X_i = x|X_{i-1}X_{i-2}\cdots X_{i-n})$$

2. Pravděpodobnost výskytu jednotky (slova, znaku, ...) závisí na vzpomínaném seznamu předchozích, ne však již na pozici v analyzovaném textu:

$$P(X_i = x|X_{i-1}X_{i-2}\cdots X_{i-n}) \approx P(X_n = x|X_{n-1}X_{n-2}\cdots X_1)$$

Předpokládejme ale případ, kdy je analyzován jev, který nenastal v trénovacích datech. Je často žádoucí, aby se i takovému jevu přiřadila nějaká (byť velmi malá) pravděpodobnost. Toho dosáhneme tzv. vyhlazováním.

Uvažovaný model dále skýtá mnohá rozšíření, zobecnění nebo změny formulace. Například některé práce ukazují [6], že tyto předpoklady mohou být až příliš striktní a autoři dokonce ukázali, že při zpracovávání delších textů (celých vět apod.) je možné analýzu podstatně zpřesnit, přidávají-li ke každé pravděpodobnosti jeden rozměr navíc. Tento rozměr získají tak, že rozdělí vyšší jednotku (větu) na nastavený počet přihrádek a pro každou mají pravděpodobnostní funkci plně nezávislou, neboli používají model

$$P(X_i = x|X_{i-1}X_{i-2}\cdots X_{i-n}) \approx P(X_n = x|X_{n-1}X_{n-2}\cdots X_1, p)$$

kde p je popsaná kategorie pro pozici ve větě.

Zobecnění popsané v předchozím odstavci je však vzhledem k zaměření práce na kratší vzorky textu (jednotlivá slova nebo fráze) zanedbatelné. Taky vzhledem k relativně velké historii (zvolíme v 4.2.3) je pravděpodobné, že budou celé části slov nebo slova naučena, a pak není nutné poslední uvedenou rovnici uvažovat.[1]

2.2 Alternativní statistický přístup

De facto stejnou matematickou ideu ale můžeme formulovat taky jinak. Z učicích dat získáme všechny souvislé n-gramy délky min až max a seřadíme je dle četnosti, nehledě na jejich délku (tu jsme používali pouze jako filtr, aby se neuvažovaly všechny podřetězce). Označme podřetězec w a jeho pozici $p_W(w)$ v uspořádaném seznamu všech naučených podřetězců W. Pokud však $w \notin W$, položme $p_W(w) = \emptyset$.

Nyní předpokládejme, že potřebujeme ohodnotit neznámý vzorek. Extrahujeme z něho stejný uspořádaný seznam W'. Vzdáleností d vzorku od naučeného nazveme sumu

$$d = \sum_{\forall w' \in W'} \frac{f\Big(p_W(w'), p_{W'}(w')\Big)}{|W'|}$$

kde funkce f přiřazuje rozdíl pozice řetězce w' v seznamu W' od pozice v seznamu W, případně (pokud w' není prvkem některého ze seznamů nebo by vzdálenost přesáhla zvolené maximum d_{max}) nabývá hodnoty d_{max}:

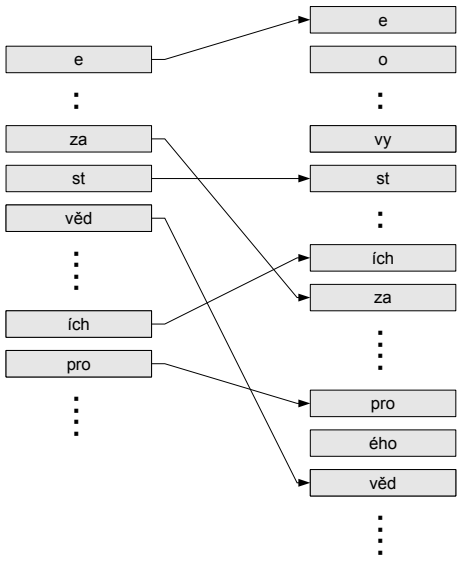
$$f(p_1, p_2) = \begin{cases} |p_1 - p_2| & |p_1 - p_2| \leq d_{max} \\ d_{max} & |p_1 - p_2| > d_{max} \\ d_{max} & p_1 = \emptyset \vee p_2 = \emptyset \end{cases}$$

Tímto přístupem kopírujeme postup popsaný v [2]. Sledujeme jej včetně rozdílů pozic, i přes možnost (při našem zaměření na co do mohutnosti relativně malé W') bez velké újmy volit maximum. Všimněme si třeba, že pokud je W' seznam například pro slovo „příklad“, jsou všechny v něm obsažené podřetězce stejně četné a W' není jednoznačné.

Ačkoli nejhezčí obrázek je vzoreček, lze tyto vzorce pro přehlednost ilustrovat, a tak i uděláme. Celý problém je názorně jednoduchý – z naučených dat získáme všechny podřetězce daných délek (typicky omezené nějak vhodně zvolenými konstantami) a seřadíme je dle četností. Takový (neúplný) seznam může být třeba napravo na obrázku 2.1. Vyčteme z něj například to, že písmena „o“ a „e“ patří mezi nejčastější daného jazyka[2]. O něco méně četné, ale stále typické sekvence znaků jsou například „ích“ nebo „ého“. Jen ještě poznamenáme, že při tomto určování četností nehledíme na délku řetězce – pokud je třeba trigram „při“ v češtině četnější než bigram „čř“ (nebo unigram Ů), není důvod tyto zvýhodňovat.

[1] Přesto lze ale očekávat, že se závěry popsané práce projeví – intuice nám napoví, že čtyřznaková sekvence „ at “ (mezery před a za anglickou předložkou) bude ohodnocena spíš jako anglická než pouhé dva znaky „at“, které jsou typickým bigramem více jazyků.

[2] Ve výsledném modelu uvažujeme i mezery, které jsme zde však nezobrazovali – nejčetnějším znakem českého psaného textu je s velkým odstupem mezera.



Obrázek 2.1: Vizualizace přístupu podle části 2.2
Pozn.: tento obrázek slouží spíš na ilustraci než jako skutečný přehled četností daných n-gramů a nepředstavuje žádný konkrétní vstup.

Stejný podle četnosti seřazený seznam máme připravený i ze vzorku; na obrázku ilustrovaný napravo. Samotné vyjádření příbuznosti je pak už jen prosté vyhledání n-gramů ze vzorku a sečtení vzdáleností. Výsledné číslo nám sice neříká (na rozdíl od metody například uvedené v předchozí kapitole) nějakou pravděpodobnost výskytu slova v jazyce, ale i přesto lze toto získané číslo efektivně porovnávat – vzdálenosti 300 a 1200 mezi vzorkem a dvěma naučenými jazyky jednoznačně mluví v prospěch toho, že se pravděpodobně jedná o první z nich, ten, který získal tímto postupem pouhé tři stovky „trestních bodů“.

2.3 Detekce vneseného bloku

Učicí text není vždy ideální. Internetová stránka může obsahovat úryvky zdrojového kódu v některém programovacím jazyce, úryvek v archaické verzi jazyka nebo přímo několik slov v cizích jazycích. Techniky, jak takovýto blok detekovat a odstranit, jsou popsané například v práci [4]. Postup lze krátce popsat následujícími body:

1. Rozdělit vstupní text na přibližně stejně dlouhé (ale samy o sobě krátké) *trénovací sekce*. Body oddělující sekce nazveme *mezery*.

2. Na každé mezeře spočteme podobnost trénovacích sekcí, které rozděluje. Vzpomínaná práce používá kosinovou míru

$$s(b_1, b_2) = \frac{\sum_{w \in W} w_{b_1} w_{b_2}}{\sqrt{\sum_{w \in W} w_{b_1}^2 \sum_{w \in W} w_{b_2}^2}}$$

kde W je množina všech slov (resp. jiných porovnatelných celků, jak n-gramů) obou bloků, w_{b_1} (w_{b_2}) je váha slova w v bloku b_1 (b_2). Jako váhy lze zvolit četnost výskytů.

Lze ale objevit i jednodušší implementace, například kdy funkce prostě vyjádří množství shodných slov ve dvou po sobě následujících blocích.

3. Podobnosti získané funkcí popsanou v předchozím bodě lze zanášet do grafu. Pro lepší výsledky se graf ještě dodatečně vyhlazuje. Prudké poklesy v této funkci pak indikují přechod z jednoho bloku do druhého.

Přeneseme ideu od segmentace blíže k námi zamýšlené filtraci a každý úsek textu ohodnotíme například postupem zmiňovaným v 2.2. Poté už je jen na uvážení a empirických experimentech se skutečnými učicími daty, jak s informací naložíme. Můžeme například zahazovat vše se vzdáleností vyšší než zvolená konstanta, respektive propustit filtrem jisté procento segmentů, čímž zvyšujeme kvalitu extrahovaných dat.

2.4 Pražský závislostní korpus

Pražský závislostní korpus[3] je obsáhlý soubor ručně anotovaných českých vět, iniciován koncem března 1995. Anotace probíhá na třech úrovních a to na úrovni morfologické (dva miliony slov), analytické (1 500 000) a tektogramatické (cca 800 000). Dále je spolu se samotným projektem vyvíjeno nemalé množství pomocným aplikací.

Data poskytl Český národní korpus a jedná se o anotované a nezkrácené články z následujících zdrojů: Lidové Noviny, Mladá fronta Dnes, Českomoravský Profit a vědecký časopis Vesmír.

Hned jako motivaci uveďme vizualizaci jedné věty anotované na všech vrstvách popisovaného korpusu a obsahující cizí frázi. Obrázek 2.2 ilustruje strukturu pro větu „Skutečné náklady na jednoho žáka English College činí asi čtyři tisíce liber ročně“. Lze si na ni všimnout pomocných uzlů, jejich vzájemné vztahy i toho, jak se cizí slova vyznačují.

2.4.1 Formát dat

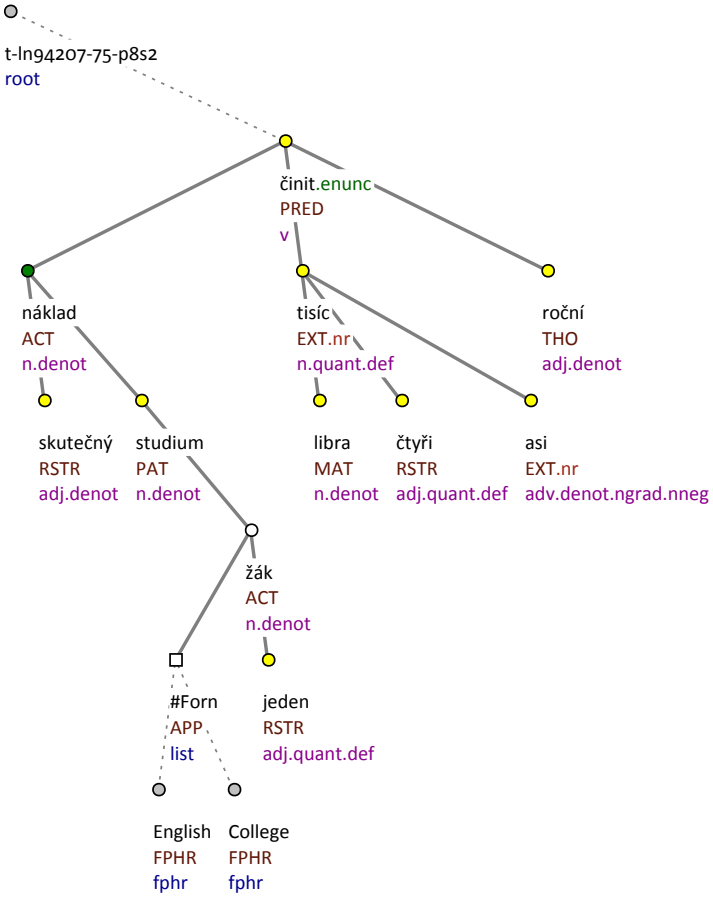
Pražský závislostní korpus verze 2.0 používá primárně formát založený na XML zvaný PML („Prague Markup Language“). Historicky však byly používány ještě další dva formáty, formát FS („Feature Structure“) pro program Graph a formát SGML používaný v první verzi tohoto korpusu. SGML je i nadále v jistém rozsahu používaný jako meziformát v některých starších aplikacích.

Každá vrstva anotace je navíc popsaná v tzv. *PML Schema file*. Tento soubor popisuje elementy, které se na dané úrovni anotace mohou vyskytnout a rovněž jejich vlastnosti jako povolené hodnoty nebo jejich úloha a struktura.

2.4.2 Úrovně anotace

Jak jsme již vzpomenuli, je Pražský závislostní korpus anotován ve třech úrovních. Ve skutečnosti je technicky ještě oddělena čtvrtá úroveň, obsahující čistý text formován do vět a odstavců. Pro označení těchto tří úrovní z praktických důvodů používáme označení *w-layer* (pro

[3] `http://ufal.mff.cuni.cz/pdt2.0/index-cz.html`



Obrázek 2.2: Vizualizace jedné konkrétní věty (výstup z nástroje `TrEd`)

slovní vrstvu), *m-layer* (morfologická anotace), *a-layer* (analytická vrstva) a *t-layer* pro tektogramatickou. Data anotována na dané vrstvě jsou anotována rovněž na všech jednodušších vrstvách.

Pro nás zajímavý fakt je, že cizí slovo resp. frázi lze poznat z tektogramatické úrovně, označuje ho funktor FPHR.

2.4.3 Cizí slova a fráze

Na rozdíl od českých částí textu, neobsahují cizí slova a fráze syntaktickou strukturu (vzhledem k převážně českým anotátorům by se jednalo o nelehký úkol) a pro konzistenci není uvedena ani v případě, kdy je zřejmá (například u slovenských frází).

Anotace probíhá tak, že je z celé fráze nejdříve vyjmut jeden reprezentant. Volba však silně závisí případ od případu a na volbě anotátora. První volbou je poslední slovo, ale často může být volba i jiná – například je typické zvolení slova, které je na rozdíl od posledního vyskloňované. Opět je ale důležité rozhodnutí anotátora, neboť například tvar slova může být shodný s nominativem a tudíž volba reprezentanta nemusí být zřejmá.

Kapitola 3

Přístup k problému

3.1 Koncepční rozhodnutí

3.1.1 Filtrování internetových stránek

Jelikož jsou data, která používáme pro učení, ze své podstaty nepříliš kvalitní, používáme několik úrovní filtrování. Pouze pokud projde část textu (a následně celek) všemi filtry, použijeme tento vzorek pro učení. V této kapitole si popíšeme navrhovaný systém filtrů. Jelikož je na Internetu dostupné obrovské množství zdrojů, není na škodu někdy zahazovat o něco více dat než je nezbytně nutné. Naopak vzhledem k nízké kvalitě některých stránek je přímo žádoucí nastavit sílu filtru o něco výše pro vyloučení těchto textů z potenciálních učicích dat.

Primárním filtrem jsou informace poskytované webovým serverem a uživatelem o stránce. Norma komunikace HTTP [9] popisuje systém specifikace jazyka dokumentu. Tato informace sice velmi často chybí nebo je chybná (například, ponechal-li autor stránky výchozí nastavení serveru apod.), přesto jsme se rozhodli tuto informaci neignorovat. V případě její absence (nebo prezence za předpokladu, že indikuje jiný jazyk, než je požadovaný) stránku v návaznosti na úvahu o menším množství kvalitnějších dat nepoužijeme.

Rovněž požadujeme, aby (z důvodů popsaných v sekci 3.1.3 o kódování) webový server v hlavičkách posílaných klientovi uváděl kódování. Pokud není uvedeno, stránku nepoužijeme. Pokud je uvedeno nesprávně, je i tak vysoká pravděpodobnost zahození stránky z důvodů popsaných dále.

Nyní předpokládejme, že všechny testy až doposud prošly a máme k dispozici stránku v HTML (resp. podobném XHTML) formátu. Naše koncepční rozhodnutí je uvažovat pouze úseky textu uvozené tagem `<div>`. Tyto úseky textu jsou dále upraveny tak, že jsou z nich odstraněny jakékoli další tagy, je ale ponechán jejich obsah. Bílé znaky (mezera, přechod na nový řádek apod.) normalizujeme – jakoukoli sekvenci po sobě následujících bílých znaků nahradíme jedinou mezerou. Pro ilustraci uveďme příklad: nechť je součástí stránky následující text

```
<h1>Úvod</h1>
<p>Poté, co jsem jednoho dne našel odkaz na
<a href="http://www.mff.cuni.cz/">stránky matematicko-fyzikální
fakulty</a>, nemohl jsem se dočkat až uvidím jejich nádherné sídlo.
A skutečně &mdash; jen se sami podívejte: <img src="./img/matfyz.jpg"/>
```

```
</p>
```

bude popsaným způsobem okleštěn do formy

```
Poté, co jsem jednoho dne našel odkaz na stránky matematicko-fyzikální fakul
ty, nemohl jsem se dočkat až uvidím jejich nádherné sídlo. A skutečně - jen
se sami podívejte:
```

(přechody na nový řádek jsou tentokrát již pouze technické, ne součástí výsledného textu).

Tímto podtrháváme a snažíme se řešit následující problémy:

1. Parsování HTML. Syntaxe HTML se postupem času dostala do stadia, kdy její komplikovanost samotná (pro jednoduchost stále vynecháváme další jazyky a skripty, které mohou HTML tvořit) je dostatečným samostatným tématem pro mnohem obsáhlejší práce. Nehledě na to, stále existuje mnoho stránek, které není snadné naparsovat vzhledem k četným chybám, respektive zastaralým konstrukcím, které jejich autoři použili. Tímto pravidlem, které jsme utvořili (prosté vyhledávání tagu `<p>`) odpadá náročná a nejednoznačná analýza nekorektních stránek.

2. Volba uvážené části textu. Dokument, který zpracováváme je v současném způsobu využívání Internetu primárně určen jako vizuální prezentace a obsahuje mnoho navigačních (často textových) prvků. Tyto prvky po úvaze nepovažujeme za součástí jazyka – jejich započtení do učicích dat nevhodně vychyluje pravděpodobnost některých slov a konstrukcí, které se často používají, například „další stránka“, „návrat na index“, „zpět“ a podobné. Čtenář jistě dá za pravdu, že tyto netvoří typický vzorek psaného českého jazyka.

   Právě do popsaného tagu jsou často vsazeny delší logické úseky textu, které jsou přesně tím, co potřebujeme.

3. Souvislost textu. Mnohokrát je nezbytná a důležitá součást textu vložena ve speciálním formátování, ve formě externího odkazu nebo zvýrazněného textu. Tyto textové součásti chceme zachovat, abychom – dává-li HTML kód souvislou větu – měli možnost tuto souvislou větu získat i naším přístupem. Rozhodnutí padlo na odstranění veškerých tagů *uvnitř* zpracovávaného tagu `<p>`, zčásti tak kopírujeme funkcionalitu prohlížeče, který tyto tagy nezná. Absolutní většina případů při tomto jednoduchém zpracování textu splňuje naše požadavky.

4. Zpracování bílých znaků. HTML dovoluje autorovi dokumentu libovolně zacházet s mezerami, tabulátory a podobnými tzv. bílými znaky. Jelikož z lingvistického hlediska nás toto formátování nezajímá (a správně se má stejně ignorovat), provádíme závěrem popsanou normalizaci.

Shrnutím těchto bodů dostáváme efektivní postup, jak extrahovat za přiměřeného poměru složitosti celého procesu a kvality získaných dat text vhodný jako vstup pro učící se algoritmy.

3.1.2 Filtrování jazyka

Postup popsaný v předchozí kapitole je jenom malá část celkového síta pro vhodná data. Není výjimkou, že stránka je sice v češtině, ale obsahuje úryvky jazyka úplně jiného – ať už programovacího nebo přirozeného. Popišme tedy, jak bychom chtěli předcházet těmto situacím. Navrhované schéma je založeno na dvou filtrech: pro kratší a pro delší texty.

Využijeme přímo faktu, že máme (díky čtení po úsecích ohraničených tagem `<div>`) text rozdělený na víceméně logické celky/odstavce. Toto rozdělení je sice čistě subjektivní a vytvořeno autorem dané internetové stránky, ale vesměs platí, že je na jedné stránce několik takovýchto celků.

Z čistě výkonnostního hlediska je pro nás výhodné, pokud dokážeme rychle odstranit krátké, zjevně nevhodné, vzorky textu a kumulovat si potenciální kandidáty pro uvážení do bufferu, který je později detailněji prozkoumán. Našim cílem bude proto na bázi jednoduchého porovnání vyhodit těch několik odstavců, které nejsou v dostatečné shodě s naučenými daty. Následně spojíme všechny zbylé odstavce do jednolitého textu a provede detailnější statistickou analýzu – pokud vzorek vyhovuje našim požadavkům, je vhodným zdrojem pro učení.

Zaměříme se nyní na detekci cizích bloků. Některé metody, jak tohoto dosáhnout, jsme již shrnuli v 2.3, ale zde můžeme využít dělení do menších celků samotným autorem stránky a přímo se nabízí – udělejme předpoklad, že jeden blok určený již popsaným způsobem extrakce textu tagem `<div>` je možné celý přijmout nebo celý zahodit, tj. je celý v jednom jazyku, psaný jedním a tím samým stylem a nemáme důvod ho dále dělit nebo z něj vyčleňovat menší celky. Analýzu můžeme u jednodušších jazyků provést už samotnou detekcí národní abecedy (to většinou bezpečně rozliší vzdálenější jazyky) nebo menší n-gramovou tabulkou. Není to ale tak snadné jak by se mohlo zdát; stále čelíme drobným koncepčním problémům. Za zmínku stojí:

1. Národní abeceda. Není snadné ji dobře definovat, primárně z důvodu velkého množství převzatých slov – kvůli několika výskytům anglického „café“ nechceme zahazovat celý blok textu, stejně jako je hitem v asijských znakových jazycích používat z času na čas latinku. Na druhou stranu, existují jazyky, které tyto abecedy mají úplně totožné, a to se už vůbec nesnažíme pojmout případ, kdy se rozlišující znak (např. jeden z mála které čeština má a slovenština postrádá) nemusí vůbec vyskytnout.

2. Vhodné naučení programu. Vložení porovnávacích dat (taky v návaznosti na zde pár narychlo poukázaných jazykových problémů) není rovněž snadné a ruční vkládání by bylo příliš pracné. Navíc autorovi práce nebyl znám žádný způsob, jak tento úkol automatizovat, respektive potřebná data extrahovat z volně dostupných tabulek (například tabulky Unicode znaků, kde jsou znaky mezi jazyky mnohokrát sdíleny).

3. Rozpoznání prahu pro přijetí/odmítnutí vzorku. Tento bod je obzvláště náročný, vzhledem k nevyváženým rozdílům mezi jazyky – pokud dosáhneme určité shody dosud naučených dat a neznámého vzorku u specifického izolovaného jazyka, nemusí stejná shoda pro méně specifický indikovat, že se o něj skutečně jedná.

Po uvážení popsaných omezení a potenciálních problémů byl zvolen následující přístup – získáme potřebná data z úvodního „semínka“ (seed), s tím, že nastavíme toleranci podle vzdálenosti předpřipraveného textu v daném jazyce k naučeným datům. Počítáme s tím, že

Tabulka 3.1: Analýza internetových stránek z hlediska poskytovaných informací

veličina	hodnota	procent.
Celkový počet stránek	144 264	100, 00%
s informaci o kódování	137 333	95, 19%
s informaci o jazyce	27 709	19, 20%
s informací o jazyce i kódování	27 709	19, 20%

Stáhnutí bylo provedeno rekurentně UNIXovým programem `wget` s počátečním URL `http://www.seznam.cz/` 13. května 2007 a zastaveno po několika hodinách běhu s nízkou prioritou; data se získávala nejen ze samotných hlaviček, ale i z tagů uvnitř těla stránky; použité skripty, a holá data pro tuto tabulku jsou dostupné na přiloženém DVD.

toleranci nastavíme o něco vyšší než bude tato vzdálenost. Nemůžeme předpokládat, že se v úvodních učicích datech vyskytnou všechny n-gramy jazyka ve svém skutečném poměrném zastoupení. Zároveň se ale pokusíme nastavit práh dostatečně nízký, aby se sekce textu nevhodné pro učení zahazovaly.

3.1.3 Kódování

V průběhu vývoje nutnosti používat počítače dalšími a dalšími neanglickými a i nezápadními uživateli bylo vyvinuto nepřeberné množství různých kódování. Mnoho předchozích prací zabývající se podobným tématem se s touto variabilitou vyrovnává dvěma hlavními způsoby; buď je kódování ignorováno (hlavně jedná-li se o jednodušší znakový jazyk, kupříkladu angličtina) nebo detekováno.

První případ není pro naše účely vhodný, snažíme-li se pojmout co nejvíce světových jazyků. Druhý je sice již lepší, ale stále narážíme na to, že budujeme nástroj silnější, než je nezbytně nutné a ve výsledku i méně přesný (rozdíl mezi kombinací čeština/ISO-8859-2 a čeština/windows-1250 je jen v šesti nepříliš častých znacích – ty mohou samy o sobě tvořit menší rozdíl než specifika jednotlivých autorů).

Druhý případ, kdy detekujeme kódování, je sice zajímavý (a vhodný třeba pro použití u dokumentů bez známého původu a kódování), ale domníváme se, že za předpokladu, že data získáváme z Internetu, není až tak atraktivní.

Krátká automatická rekurentní analýza (shrnutí výsledků viz tabulka 3.1) ukázala, že zhruba 95% stránek definuje své kódování, což je mnohem víc, než procento stránek definující svůj jazyk – pouhých cca 19%. Je důležité si uvědomit, že v souladu s popsaným prvním filtrem stránky nedefinující svůj jazyk zahazujeme. Navíc v celém úvodním testu (čítajícím přes 140 tisíc stránek) se nevyskytl jediný případ, kdy by (za předpokladu, že je ve hlavičkách uveden jazyk) stránka nedefinovala své kódování. Proto padlo rozhodnutí, že znakovou sadu budeme vždy považovat za známou a převádět ji do nějaké vnitřně univerzální. Nabízí se přímo Unicode, které nejenže pokrývá většinu znaků současných živých jazyků, ale se navíc nedávno podle statistik Google [10] stalo dominantní volbou při psaní webových stránek.

Dalším aspektem, který může hrát roli při volbě této možnosti je fakt, že takto nedělíme síly – není našim cílem utvořit data popisující češtinu třikrát (ne-li ještě vícekrát), preferujeme jediný, ale kvalitnější popis.

3.1.4 Programovací jazyk pro doprovodnou aplikaci

Zvoleným programovacím jazykem je Java. Byla vybrána po zvážení následujících bodů:

1. Dostupné knihovny. Pro práci s počítačovými řetězci, stahování souborů z Internetu nebo změnu kódování má Java hned několik knihoven. Možná sice ne až tak rychlých, ale velmi silných co do možností. Vzhledem k faktu, že samotná doba odezvy serveru bývá řádově delší než zpracování HTML stránky, ustupuje argument o nižší rychlosti oproti některým jazykům nižší úrovně do pozadí. Knihovny nám tedy usnadní samotné psaní programu a můžeme přesunout pozornost na zajímavější aspekty práce.

2. Platformová nezávislost. Z dostupných nástrojů je Java jeden z mála neinterpretovaných jazyků, u kterých máme zaručený běh na všech majoritních operačních systémech a architekturách. Navíc dosahuje relativně dobrých výsledků co se výkonu týče – kód může být virtuální mašinou optimalizován za běhu (třeba i za využití informací z profileru).

3. Práce s vlákny. Jak jsme již diskutovali, je prodlení kvůli reakční době serveru mnohokrát významnější faktor než samotné zpracování stránky. Nabízí se proto vícevláknová operace, kdy by se toto prodlení v jednom vlákně mohlo snadno využít ve vlákně jiném ke zpracování nezávislých dat (typicky dat pro jiný jazyk). Java poskytuje komfortní práci s vlákny, více než postačující pro dané účely.

3.2 Diskuze dalších záležitostí

Následující položky nebyly sice do závěrečné práce zahrnuty, byla ale provedena jejich analýza a budou v dalším sledovány. Případné předchozí studie jsou zběžně popsány.

3.2.1 Opakované stahování stejného obsahu

Abychom nestahovali stejný text, program je navržen tak, že si pamatuje nejen seznam adres, na kterých se pravděpodobně nachází kvalitní vzorky učeného jazyka (to se pozná například tak, že odkaz na stránku byl nalezen na jiné stránce, která lehce prošla filtry – pravděpodobně tedy byla sama kvalitní), ale i seznam již navštívených URL. Na již navštívené URL se program nikdy nevrátí, aby předešel ukládání do své databáze stejného textu.

Přichází ale v úvahu ve výsledku podobný jev; a to situace, kdy existuje text učeného jazyka, který se mnohokrát cituje, respektive opakuje. Příkladem může být licenční ujednání (například kopie GPL se rozhodně na Internetu nevyskytuje řídce) nebo jiný oficiální často citovaný dokument.

3.2.2 Postupný přechod do jiného jazyka

O málo vážnější problém je postupný přechod do jiného jazyka. Příkladem může být například technický text v japonštině. Jelikož není latinka pro japonské texty úplně cizí a některé často používané převzaté tvary mohou v textu dominovat, je možné, že se statisticky popíší některá písmena abecedy jako pro japonštinu typická.

Ilustrujeme to uměle vytvořeným příkladem:[1] narazí-li náš program na stránku obsahující například (text je převzat z japonské Wikipedie)

> その後NT系のOSはWindows 2000、Windows XPへと進化していく。また、Windows XPの発売によってMicorosoftは9x系のOSの開発を停止し、Windowsの開発はNT系へと一本化されていく。現在、パソコン市場において最も広く使用されているWindowsはWindows XPである。[2]

Je pravděpodobně už zřejmé, že se u jistých nastavení filtrů text vyhodnotí jako japonský (žádný jiný jazyk neobsahuje hiraganu v takovém množství). Rovněž se ale může naučit, že znaky latinky tvořící sekvenci „Windows“ jsou v japonštině časté. Není již těžké si představit, že program narazí na jiný odstaveček, kde je popsaná znaková sekvence rovněž dominantní a postupně se naučit jinou znakovou sadu (latinku) jako japonskou – až do takové míry, že bude typičtější texty původního jazyka odmítat.

Bližší případ je alternativa, kdy se přejde se stahováním českých textů z Internetu na stránky věnované například historickým nebo básnickým formám češtiny. Pokud projde takový úsek textu filtrem a program na stránkách zůstane dostatečně dlouhou dobu, může mluvenou češtinu začít označovat jako cizí jazyk (například slovenštinu).

3.2.3 Ekvivalentní přepisy textů, romanizace

Jedním z problémů ztěžujících n-gramovou statistickou analýzu je záměnnost jistých (jednotlivých nebo sekvencí) znaků za jiné. Například následující tři odstavce jsou v mluveném jazyce (Korejštině) ekvivalentní, liší se pouze použitými znaky pro zápis[3]:

1. Modeun Ingan-eun Tae-eonal ttaebuteo Jayuroumyeo Geu Jon-eomgwa Gwonrie Isseo Dongdeunghada. Ingan-eun Cheonbujeog-euro Iseong-gwa Yangsim-eul Bu-yeobadass-eumyeo Seoro Hyungje-ae-ui Jeongsin-euro Haengdongha-yeo-yahanda.

2. 모든 인간은 태어날 때부터 자유로우며 그 존엄과 권리에 있어 동등하다. 인간은 천부적으로 이성과 양심을 부여받았으며 서로 형제애의 정신으로 행동하여야 한다.

3. 모든 人間은 태어날 때부터 自由로우며 그 尊嚴과 權利에 있어 同等하다. 人間은 天賦的으로 理性과 良心을 賦與받았으며 서로 兄弟愛의 精神으로 行動하여야 한다.

Na příkladu vidíme, že je diametrální rozdíl mezi druhou a třetí ukázkou i přesto, že oba zápisy jsou pro daný jazyk nativní (druhý však kvůli vyšším nárokům na memorizaci hanji ustupuje do pozadí). Třetí ukázka čítá 38 (27 různých) znaků hangul a 28 (25 různých) znaků hanji. Je tedy zřejmé, že text jen těžko může projít i prvním jazykovým filtrem popisovaným v 3.1.2, je-li například program naučen na druhém vzorku a analyzuje vzorek třetí.

Ponecháme-li stranou občasné používání znaků národa Han místo ekvivalentních fonetických zápisů v asijských jazycích, stále nám zůstává romanizace (neboli přepis do latinky).

[1] Bylo by však vhodné podotknout, že není úplně cizí – slovo „Windows“ se v jednom testu s nižším prahem pro filtry skutečně naučilo jako typicky japonské

[2] text je beze změny převzat z japonské Wikipedie

[3] zdroj: Všeobecná deklarace lidských práv, článek 1.

Není zvykem psát například v německé tiskovině korejské znaky, spíše se používá některého fonetického přepisu (první odstavec). Není možné správně rozpoznávat fonetický přepis jazyka, naučí-li se program rozeznávat nativní abecedu. Navíc, ve většině případů, kdy narážíme na tento problém, nemáme k dispozici dostatečné množství učicích dat pro fonetický přepis – rodilí mluvčí jej nepoužívají a v případě použití transkripce obvykle chybí psaný tvar v původním jazyce.

Kapitola 4

Popis použitých nástrojů a nastavení

Spolu s touto prací byl jako ročníkový projekt vyvinut program podle rozhodnutí popsaných a zdůvodněných v předcházející kapitole. Nebudeme se zabývat detailně jeho konfigurací – pro zájemce existuje uživatelská resp. programátorská dokumentace [7], rovněž konfigurační soubory a skripty vytvořené podle popisu v této práci jsou přiložené na doprovodném DVD. Zato si popíšeme jak byl program inicializován, které jazyky byly zvoleny a jak jsme nastavili filtry pro obsah webových stránek.

4.1 Software

Celý systém byl vyvinut kolem ročníkového projektu a sestává téměř výhradně z pomocných bashových skriptů kolem samotného programu. Ten je už více méně samostatný a na základě své konfigurace udržuje jak seznam internetových stránek, které navštívil nebo plánuje navštívit, tak veškerá jazyková data. Rovněž i podle potřeby spouští a ukončuje jednotlivá vlákna, je-li ukončen uživatelem, nebo vlákno dokončí svou činnost (typicky získá předepsané množství nových dat).

Co ale potřebujeme připravit ručně, jsou mimo jiné inicializační semínka pro statistický přístup popsaný v části 2.2. Dále potřebujeme vytvořit konfigurační soubor s jistou množinou jazyků, které se bude program učit a každému modulu pro vyjádření vzdálenosti vzorku od naučených dat (tzv. „LanguageCheck“) zadat prahovou hodnotu, nade kterou nebude vstupní data zahrnovat do učicích (viz části o filtrování jazyka 3.1.2 a vyjádření podobnosti dvou textů jako v 2.2).

Pro pozdější potřeby rovněž použijeme nástroje vyvinuté pro práci s pražským závislostním korpusem, jmenovitě stromečkový editor `TrEd` a jeho dávkovou verzi `btred`. Úkol bude ale velmi snadný – pouze vyhledáme uzly označené jako cizí fráze pro potřeby závěrečného experimentu.

4.2 Jazyková data

4.2.1 Počáteční učicí data

Ještě před započetím automatizované extrakce učicích dat z internetu potřebuje připravený program (ručně) připravené vzorky textů v různých jazycích. Jako zdroj těchto textů byl zvolen text Všeobecné deklarace lidských práv, který je volně dostupný na stránkách [11] v mnoha světových jazycích. Jako další zdroj dat byla zvolena Wikipedie – hlavně z důvodu, že obsahuje mnoho referencí na stránky mimo, velmi pravděpodobně psané v jazyce původního dokumentu. Oba tyto zdroje však skýtají několik problémů, které prodiskutujeme později.

4.2.2 Zvolená množina jazyků

Laťku, co se týče množiny jazyků, které chceme zpracovávat, si nastavíme tak, aby daný jazyk měl nejen dostatečné zastoupení na internetu, ale požadujeme i vhodná počáteční data. Jako ukazatel zastoupení na internetu, ač ne nutně ideální a přesný, zvolíme svobodnou encyklopedii Wikipedia.

První experimentální nastavení je založeno na Všeobecné deklaraci lidských práv podle překladů Organizace spojených národů. Pro jazyky, které v době psaní této práce měly na Wikipedii alespoň 10 000 článků vybereme vhodně odpovídající text ze stránek OSN. Vzorky pro několik jazyků ale tímto způsobem nelze získat – důvody by se daly shrnout do několika kategorií:

1. OSN nemá jazyk z Wikipedie ve svém repozitáři. Typicky jde o lokální mutace jazyků, resp. jazyky s nižším počtem mluvčích (za zmínku však stojí, že i přesto patří co do obsahu mezi majoritní jazyky Wikipedie).

   Seznam jazyků: বিষ্ণুপ্রিয়া মণিপুরী (Bishnupriya Manipuri, Indoárijský jazyk s přibližně 450 000 mluvčími), नेपाल भाषा (Nepal Bhasa, jeden z jazyků Nepálu), Nnapulitano (italské nářečí z okolí Neapole), Piemontèis (románský jazyk severozápadní části Itálie, Piemontu), Plattdüütsch (dolnoněmčina), Sicilianu (jazyk Sicílie), Simple English (mutace angličtiny omezená na menší slovní zásobu), Srpskohrvatski / Српскохрватски (srbochorvatština) a Volapük (umělý jazyk).

2. OSN poskytuje pouze obrazový nebo jinak obtížně použithttp://www.unhchr.ch/udhr/elný materiál. Některé jazyky nemají digitální podobu textu Všeobecné deklarace lidských práv, pouze například naskenovaný materiál, respektive materiál ve formátu PDF. Jelikož nebyl nalezen zdroj, který poskytuje celý text jako text Unicode, nebylo možné tyto jazyky v této fázi natrénovat. Typicky tento výčet zahrnuje sice početné jazyky, ale s relativně speciálním písmem.

   Mezi tyto diskutované jazyky patří: Azərbaycan (ázerbájdžánština), فارسی (perština), עברית (hebrejština), indické jazyky मराठी (maráthština), தமிழ் (tamilština) a తెలుగు (telugština), ไทย (thajština) a Tiếng Việt (vietnamština).

3. Jazyky předposlední skupiny jsou ty, které jsou buď na jednom nebo druhém zdroji omezené. Například Wikipedie uvádí dvě jazykové variace běloruštiny: Акадэмічная

a Тарашкевіца. Organizace spojených národů však ve svém repozitáři má jenom moderní Branislawem Tarashkyevichem standardizovanou verzi. Druhým případem je bosenština, kterou natrénujeme jenom pro latinku, její cyrilikou psaná verze, ač přítomna na stránkách OSN, není jazykem Wikipedie.

4. Závěrem popišme případ makedonštiny a pár dalších jazyků psaných cyrilikou, kde sice nebyla data poskytnutá Spojenými národy v pořádku, jednalo se ale o lehce napravitelný nedostatek. Text Všeobecné deklarace lidských práv je v tomto jazyku mixem latinky a cyriliky (v homografických případech znaků jako „o“ nebo „e“ je preferována latinka). Naproti tomu nebyla nalezena na Internetu stránka, kde by se podobné praktiky uplatňovaly. Proto byla tyto učicí data pro makedonštinu upravena nahrazením popsaných znaků za jejich ekvivalenty v cyrilice.

I přes tyto nedostatky jsme získali vstup pro 59 různých jazyků (respektive 63 vzorků celkem, uvážíme-li po dvou získaných variantách čínštiny, latiny a norštiny).

4.2.3 Velikost učené informace

Vzorek, který nyní máme k dispozici, je sice velmi specifický (jen málo podobný většině internetových textů), je ale pečlivě kontrolovaný co do stejného významu. Fakt, že francouzská variace říká téměř to samé jako variace čínská, se nám hodí pro porovnání jazyků po znakové stránce. Různé způsoby zápisu ukládají do jednoho znaku různé množství informace. Tak jako potřebujeme pět znaků na zapsání slova „forêt“ (les) ve francouzštině, tak nám postačuje znak jediný v čínštině: 林.[1] Ve smyslu této úvahy jsme vytvořili tabulku 4.1.

K číslům v tabulce poznamenejme, že počty unikátních znaků zahrnují i číslice použité k popsání bodů nebo článku a ty se řídí jazykovými zvyklostmi. Například v češtině píšeme jeden a ten samý znak „3“ v „Článek 3“ i ve výčtu „3.“ („za třetí“). V latině naopak vidíme v obou případech „III“, tedy ne nový znak *3*, ale již existující znak velkého *I*. Čínština a japonština používají podle typografických konvencí jedno z „3“, „３“, „三“ nebo „㊂“ apod. Rovněž se v počtu unikátních znaků pro němčinu vyskytly téměř všechna velká písmena (podstatná jména se píší velkým počátečním písmenem), ale v českém jen část.

Tabulka ukazuje, že ačkoli se délka většina překladů pohybuje v rozsahu $9\,000 - 13\,000$ znaků, je citelný silný propad počtu znaků u čínštiny, japonštiny, korejštiny a snad i arabštiny. Nižší počet znaků u arabštiny není až tak výrazný a je pravděpodobně důsledkem toho, že je zvykem v psané formě tohoto jazyka vynechávat krátké samohlásky. Zbylé tři případy jsou však zřejmé – jeden znak nese mnohem větší informaci než ve většině dalších studovaných jazyků.

V případě třech hlavních znakových jazyků (čínštiny, japonštiny a korejštiny) vidíme jasnou nepřímou úměru mezi počtem unikátních znaků a délkou textu ve znacích. Ačkoli je sice počet unikátních znaků v japonštině téměř shodný s jejich počtem v čínštině, dovolujeme si vyslovit myšlenku (z omezené středoevropské znalosti těchto jazyků), že u delších textů bude rozdíl patrnější. Předně se ve vzorku vyskytla již téměř celá hiragana (cca padesát znaků) a co se týče znaků kanji, ty používají Japonci v omezenější míře než Číňané. Naopak vzorek je rela-

[1] respektive dva znaky (森林); v každém případě to snad postačuje pro ilustraci situace, ve které se nachází skript zapisující větší celky než hlásku, jakými jsou morfémy nebo slabiky

Tabulka 4.1: Rozdělení počtu znaků a unikátních znaků v různých překladech

znaků celkem	unikátních	N_1	N_2	jazyk
2 858	506	3	2	mandarínská čínština (tradiční)
2 858	505	3	2	mandarínská čínština (zjednodušená)
4 131	502	3	2	japonština
4 781	317	4	2	korejština
7 625	57	5	2	arabština
9 188	61	5	3	kurdština
9 400	60	5	3	ido
9 404	70	5	3	srbština (cyrilice)
9 519	69	6	3	bengálština
9 782	70	6	3	čeština
9 829	44	6	3	latina (druhá verze)
9 865	63	6	3	bosenština
9 871	54	6	3	chorvatština
9 879	60	6	3	esperanto
9 912	48	6	3	latina (první verze)
9 965	58	6	3	norština (nová, Nynorsk)
10 055	77	6	3	slovenština
10 095	52	6	3	velština
10 177	58	6	3	norština (knižní, Bokmål)
10 186	63	6	3	islandština
10 217	64	6	3	asturština
10 243	58	6	3	turečtina
10 335	56	6	3	afrikánština
10 423	53	6	3	slovinština
10 426	66	6	3	lotyština
10 586	64	6	3	kreolština (haitská)
10 594	69	6	3	bretonština
10 596	58	6	3	angličtina
10 657	63	6	3	makedónština
10 753	70	6	3	ukrajinština
10 786	75	6	3	hindština
10 842	61	6	3	dánština
10 855	54	6	3	estonština
10 856	71	6	3	litevština
10 912	70	6	3	katalánština
10 915	54	6	3	švédština
10 959	56	6	3	baskitština
11 042	65	6	3	albánština
11 061	55	6	3	finština
11 191	55	6	3	galicijština
11 260	65	6	3	portugalština
11 311	69	6	3	běloruština
11 334	73	6	3	bulharština
11 471	59	6	3	valonština (minoritní románský jazyk Belgie a Francie)
11 549	73	6	3	polština
11 576	65	6	3	okcitánština (provensálština)
11 677	65	6	3	ruština
11 819	59	6	3	španělština
11 850	61	6	3	francouzština
11 855	70	6	3	němčina
11 864	62	6	3	rumunština
11 894	59	6	3	italština
11 917	49	6	3	gruzínština
11 993	58	6	3	maďarština
12 149	53	7	3	cebuano (regionální jazyk Filipín)
12 306	78	7	3	lucemburština
12 326	50	7	3	jazyk Tagalog (jazyk Filipínské republiky)
12 336	59	7	3	sundaština (jazyk malajského souostroví Sunda)
12 364	66	7	3	řečtina
12 543	57	7	3	malajština
12 565	54	7	3	indonéština
12 724	59	7	3	holandština
13 647	63	7	3	javánština

tivně krátký a úzce zaměřený k tomu, aby se v čínském textu vyskytlo všech několik (desítek) tisíc různých ideogramů.

Je zajímavé, že se na druhou stranu neprojevila žádná zřejmější korelace délky textu a počtu unikátních znaků u dalších jazyků. Částečně uspokojivým může být rozdílná délka slov (angličtina může mít až pětinu textu tvořenou mezislovními mezerami), množství kontextuální výslovnosti (rozdílné „c“ ve slovech „cap“ a „ace“), zastoupení nevyslovovaných hlásek apod.

Povzbudivé pozorování je, že některé blízké jazykové páry, jako čeština a slovenština, švédština a finština, valonština a francouzština nebo angličtina a na angličtině založená haitská kreolština jsou blízko i v této tabulce. Lze i vysledovat, že třeba nová norština je oproti své knižní verzi o něco kratší. Můžeme se tedy dohadovat, že postupem evoluce tento jazyk přestal zapisovat některé zvuky, respektive se spřežky nahradily samostatnými znaky.

Opusťme na chvilku tabulku a řekněme si něco o volbě velikosti databáze n-gramů. Ta byla provedena na základě této tabulky a vyhodnocení délky (co do významu) stejného textu. Množství analyzovaných dat (neboli délku maximálního n-gramu) zvolíme na počtu znaků vzorku. Je-li l délka textu pro jazyk, označíme celé číslo určující délku nejdelšího znakového n-gramu, který budeme ve strukturách programu uchovávat jako funkci dvou proměnných s a d následovně

$$N(s,d) = 1 + \left\lceil \frac{l+s}{d} \right\rceil$$

a v návaznosti na tento zvolený princip definujeme základní velikosti databáze pro kompletní analýzu[2]

$$N_1 = N(500, 2500) = 1 + \left\lceil \frac{l+500}{2500} \right\rceil$$

a pro rychlé porovnání krátkých bloků textu (viz druhy filtrování, kapitola 3.1.2)

$$N_2 = N(-4000, 5000) = 1 + \left\lceil \frac{l-4000}{5000} \right\rceil$$

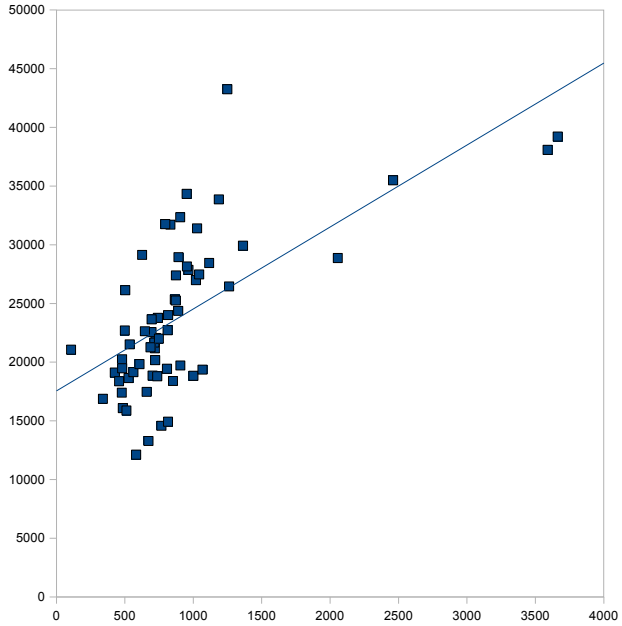
Tyto hodnoty jsou pro přehlednost rovněž uvedeny v tabulce 4.1.

4.2.4 Nastavení prahů pro filtry

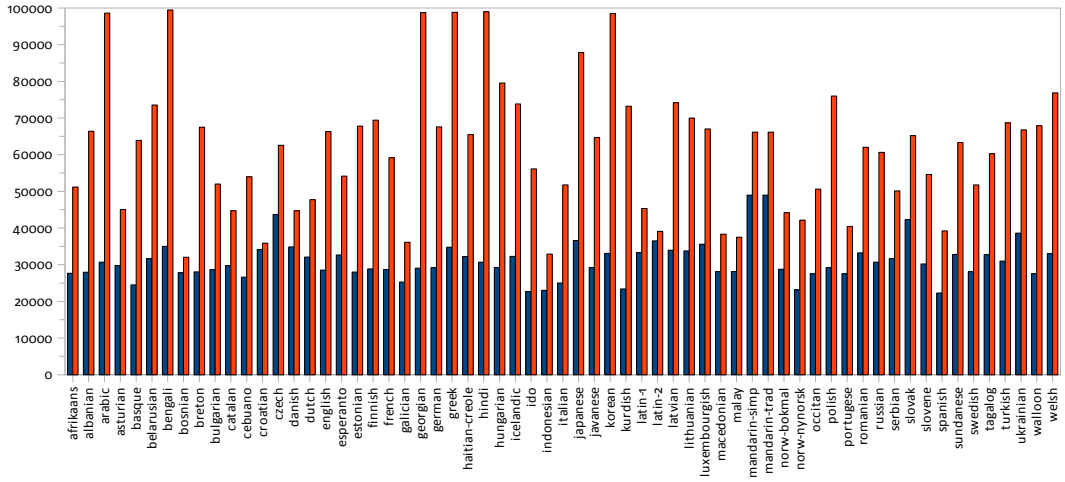
Nejdříve uděláme menší výzkum, co můžeme očekávat od vztahů popsaných v 2.2 a výsledky naneseme do grafů. Zajímáme se o vzdálenost posledního, třicátého, článku Všeobecné deklarace lidských práv, máme-li naučený celý předchozí text (pochopitelně bez tohoto článku). Samotný nadpis „Článek 30.“ rovněž není obsažen v testovacích datech.[3] Jelikož jsme popsali dva filtry (jednodušší a komplexnější), provedeme tyto měření pro oba z nich. Výsledky korelace vracených hodnot obou filtrů lze vidět na obrázku 4.1 a absolutní hodnoty pro komplexní filtr jsou na obrázku 4.2.

[2] s a d jsou voleny tak, aby jazyky prostřední části tabulky (tj. bengálština až italština) byly ohodnoceny stejně; druhé ohodnocení je poloviční a s posunem zvoleným tak, aby končil na trigramech

[3] vložení tohoto nadpisu do testu by mohlo podobnost posunout až příliš nízko – po naučení těch několik odstavců, kde vlastně každý začíná tímto slovem se dá podobné chování čekat; ostatně už takto se styl projevu i některá slova z předchozího textu opakují



Obrázek 4.1: Přibližná shoda mezi výsledky dvou filtrů: obrázek naznačuje, že pravděpodobně existuje korelace mezi vracenými vzdálenostmi od vzorku. Jednodušší analýza je naznačena na vodorovné ose, komplexní na svislé.



Obrázek 4.2: Porovnání vzdálenosti vzorku jazyka a druhého nejpravděpodobnějšího kandidáta.

Volby pro maximální rozdíl d_{max} byly provedeny zkusmo a ustáleny na hodnotách $10\,000$ a $100\,000$ pro odstavcový (resp. komplexní) filtr jazyka. Na základě těchto hodnot byl nastaven i limit pro délku seznamů W_1 a W_2 na dvojnásobek této hodnoty (i když vůbec nepředpokládáme, že například seznam vytvořený ze zkoumaného vzorku v našem případě může narůst do takové délky).

Z prvního obrázku vidíme, že už rozdíl jazyků na základě dvou navrhovaných systémů porovnávání velmi příjemně koreluje. Aplikujeme-li tento poznatek na náš záměr navrhnout vhodně navrhnout práh zahazování neznámého textu, můžeme téměř bez obav navrhnout jeden postup a ten aplikovat na oba filtry. Nadále, zaměříme-li se na druhý obrázek, vidíme, v jakých mezích se pohybují diference prvních dvou tipů na jazyk vzorku[4]. Nelze sice určit univerzální mez, protože jak lze nahlédnout, podobnost textů a vzorků jednoho jazyka je v některých případech až o řád posunuta vůči jiným jazykům.

Zavedeme proto relativní práh: hladinou zamítání (tj. vyjádří-li se vzdálenost na stejnou nebo vyšší hodnotu) bude dvojnásobek hodnoty podobnosti třicátého článku Deklarace a zbytku textu před tímto článkem. Je-li však dvojnásobek menší než $2\,500$ (resp. $25\,000$ u komplexního filtru), bude uvažována tato hodnota. Na obranu toho, že uvažujeme až dvojnásobný rozdíl, uveďme, že aritmetický průměr poměru prvního a druhého kandidáta v seznamu tipů programu je u jednoduššího filtru vysoko nad tímto číslem, přibližně $3,81$. Co se týče komplexnější analýzy, jsou rozdíly menší (částečně ale díky nedostatku informací o delších n-gramech jazyka), ale i tak se pohybují něco málo nad číslem 2. Druhá část volby spodní meze vyplynula z experimentálního provozu programu, kdy několik vláken zpracovávajících jazyky, kterých se to týče, aktivně zahazovaly téměř veškerý text extrahovaný z internetových stránek.

Tím můžeme kapitolu o přípravných nastaveních doprovodného softwarového díla uzavřít a dostáváme se k analýzám výsledků v následující kapitole.

[4] ověřili jsme, že nejbližší jazyk je vždy skutečný jazyk vzorku; druhý tip programu byl povětšinou některý blízce příbuzný jazyk

Kapitola 5

Získaná data

Zde popíšeme data získaná spuštěním celého systému dávkových skriptů a analyzujeme graficky získané numerické výsledky.

5.1 Pozorování vývoje v průběhu učení

Toto pozorování je kritické co se týče efektivity filtrů. Učení probíhalo ve dvanácti cyklech po deseti tisíci znacích, tudíž (při odhadu na šedesát trénovaných jazyků) v každém cyklu jsme přibrali cca 600 kB nových dat. Jak se tato data projevovala po každé iteraci na ohodnocení původního vzorku (třicáté kapitoly Deklarace) je naznačeno graficky na obrázku 5.1.

Křivky jsou založeny na průměru pro všechny trénované jazyky (vzhledem k jejich velkému počtu není praktické ani přehledné je všechny detailněji rozebírat). Analýza byla rovněž provedena na třech úrovních: testovaný vzorek byl buď celý článek 30 Deklarace (*part-1*), první polovinou (*part-2*), resp. pouhou čtvrtinou už tak relativně krátkého textu (*part-4*).

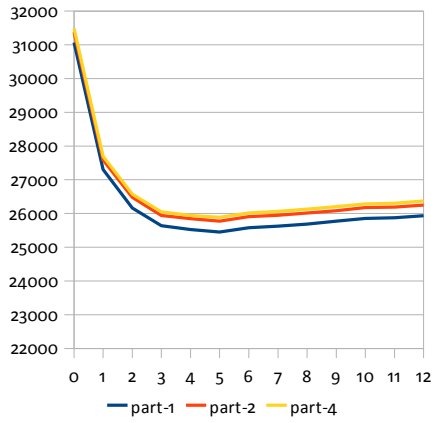
Pozorujeme, že úvodní data jsou velmi rychle vylepšena a dále systém stagnuje. Dokonce je možné zahlédnout i jistou regresi po několika iteracích. Tu vysvětlujeme tím, že se vlákno poté, co obsáhne základní tvary jazyka, postupně naučí i nějaké množství dalších tvarů a obratů jazyka, které ale už pro změnu nejsou do takové míry použity v původním učicím textu.

5.2 Test na krátkém náhodném vzorku

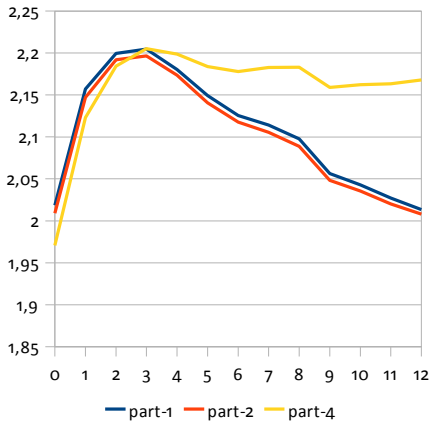
Výsledky předchozí části se dají nazvat uspokojivé, ale to, aby náš systém poznal, v jakém jazyce je napsaný třicátý článek Všeobecné deklarace lidských práv není naší (hlavní) prioritou. Proto provedeme další test na náhodně zvolených větách několika známých jazyků a budeme sledovat přesnost, s jakou je určuje.

Pro celkem 21 jazyků jsme zvolili cca 15 vět pro každý jazyk. Celkem se nasbíralo 296 vět tvořených z cca 36 500 znaků, což dává průměr kolem 120 znaků na větu. Jednalo se vesměs o informativní texty z Wikipedie z náhodně volených článků.

Test probíhá tak, že necháme naučený program přiřadit jazyk ke každé větě; pokud je správný jazyk hned prvním kandidátem, započteme bod. Přičítáme půl bodu, pokud je druhým kandidátem, a v případě, že není ani druhým kandidátem, započteme nulu. Úspěšností



(a) vzdálenost testovacího vzorku a naučených dat



(b) poměr správného jazyka a dalšího nejlepšího kandidáta

Obrázek 5.1: Změna vlastností systému po každé iteraci (průměrné hodnoty pro všechny jazyky, pro komplexní filtr)

Tabulka 5.1: Test na zkracujících se náhodných vzorcích 21 jazyků

délka	1	1/2	1/3	1/4	1/5	1/6
1. iterace úspěšnost	95, 93%	92, 88%	88, 98%	85, 08%	78, 64%	77, 12%
1. iterace shoda	92, 88%	90, 17%	84, 41%	80, 34%	74, 24%	70, 85%
4. iterace úspěšnost	95, 25%	94, 75%	93, 39%	92, 20%	87, 63%	86, 10%
4. iterace shoda	91, 86%	91, 53%	89, 83%	89, 49%	84, 41%	83, 73%
konečná úspěšnost	98, 47%	96, 61%	95, 08%	92, 54%	90, 68%	89, 32%
konečná shoda	96, 95%	94, 24%	91, 86%	89, 15%	87, 46%	85, 42%

rozumíme procento získaných bodů z maximálních možných. Jako shodu chápeme procento těch případů, ve kterých dostal program plný bod (tj. analýza byla perfektní).

Celý postup několikrát opakujeme, pro celé věty, první polovinu větu, první třetinu až pouhou šestinu věty (v tomto případě se už dostáváme k pouhému jednomu až dvěma slovům). Výsledky jsou naznačeny v tabulce 5.1.

Vidíme, že přesnost se zkracujícím se vzorkem sice klesá, ale i přesto dosahuje velmi slušných hodnot. Tabulka ukazuje úspěšnosti v první, čtvrté a poslední iteraci. Z předchozí části se může zdát, že systém by měl dosahovat nejlepších výsledků kolem čtvrté iterace (pak se relativní vzdálenost jazyků snižuje a rovněž se mírně vzdaluje testovací vzorek), tabulka však ukazuje, že to není úplně ten případ. Jak jsme si určitě už řekli, testovací vzorek je relativně specifický.

Tabulka 5.2: Porovnání zastoupení tradičních a zjednodušených znaků čínštiny před a po procesu učení

vlákno/vzorek	sdílené	tradiční	zjednodušené	ostatní
tradiční vzorek	67, 24%	22, 07%	0, 03%	10 63%
zjednodušený vzorek	68, 05%	0, 00%	21, 30%	10 63%
mandarin-traditional	55, 08%	15, 47%	9, 45%	19, 18%
mandarin-simplified	55, 06%	12, 40%	12, 47%	20, 05%

5.3 Některé specifické jevy

5.3.1 Dvě verze čínštiny a latiny

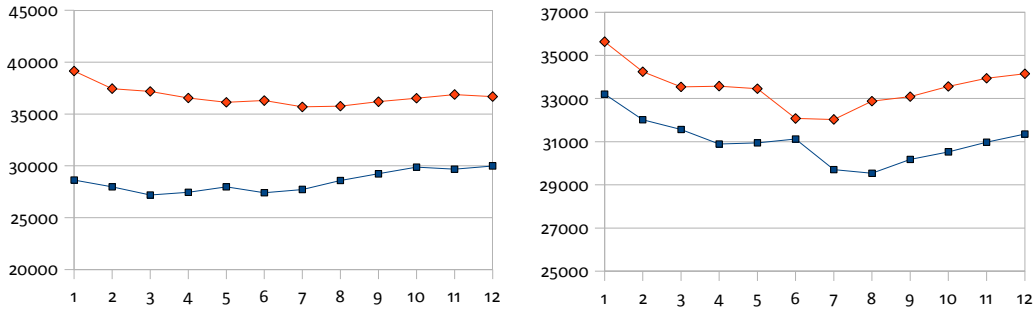
Toto specifikum samo o sobě tvořilo zajímavý problém. Moderní čínština má dvě znakové sady – tradiční a zjednodušenou, kde druhá vznikla v Čínské lidové republice ve snaze zvýšit gramotnost obyvatelstva. Systém psaní se sice nezměnil, ale některé komplexní tvary znaků byly nahrazeny jednoduššími (oblíbený příklad je dvojice 讓 a 让).

Po stránce této práce je však zajímavé, že čínská Wikipedia nerozděluje tyto znakové sady a tudíž není nezvyklé, že jeden a ten samý článek je složen z několika odstavců, střídavě psaných jednou a druhou formou. Rovněž z HTTP hlaviček dalších internetových stránek nelze spolehlivě poznat kódování, protože jsou tyto dva způsoby psaní často sjednocovány pod jediným označením jazyka – čínština.

Tento jev je relativně snadný na pozorování – existují seznamy znaků, které jsou oficiální jenom v jedné, respektive druhé variantě písma. Připravíme si tedy tři množiny: množinu znaků, které jsou shodné v obou variantách, znaky typické pro zjednodušenou variantu a nakonec znaky typické pro tradiční formu. Rozdělíme znaky textů (vzorek, na kterém se program učil, tj. Všeobecnou deklaraci lidských práv a celý text, který byl po poslední iteraci získán z Internetu) do těchto tří skupin a porovnáme jejich velikosti. Výsledek srovnání je shrnut v tabulce 5.2.

Varianty jsou od sebe nadále do jisté míry odlišitelné (filtry přeci jen částečně pracují), ale je zřejmé, že se obě vlákna postupem času naučila nejen další čínské znaky své sady, ale i nemalé množství znaků ze sady druhé. Tento jev se nám pouhým nastavením filtrů nepodařilo odstranit. Vlastně už tabulka 4.1 ukazuje, že vzorek obsahuje pouhých pět set unikátních znaků (porovnáváme s cca třemi tisíci požadovanými pro gramotnost v tomto jazyce). Je pochopitelné, že aby filtr přijal běžný text z Internetu, bude muset být nastaven dostatečně benevolentně, aby povolil výskyt nemalého množství zatím nenaučených znaků. Vzhledem k relativně malému počtu zjednodušených novotvarů je ale bude přijímat také. Fakt, že tyto dvě varianty navíc nejsou dostatečně označeny nenahrává automatickému učení.

Podobně publikují Spojené národy dva různé překlady Deklarace do latiny. Obě vlákna jsme inicializovali stejným počátečním URL `http://la.wikipedia.org/`. Protože ale nyní už nemůžeme tak snadno sčítat znaky nepřirozené pro jednu verzi jazyka, použijeme graf znázorňující vzdálenost vzorku (třicáté kapitoly Deklarace) jednotlivých verzí latiny od naučených dat na konci každé iterace. Vzdálenost pro první variantu latiny je na obrázku 5.2(a) a



(a) latin-1 (b) latin-2

Obrázek 5.2: Vývoj podobnosti vzorku dvou variant latiny a naučených dat na konci každé iterace

pro druhou 5.2(b).

Ačkoli se na obou projevují jisté fluktuace, je rozdíl relativně dobře patrný jak v počáteční fázi, tak i na konci. Fluktuace by bylo možné ještě více omezit tak, že upravíme prahy filtrů, ale tím bychom mohli vytvořit příliš konzervativní model, který příliš mnoho nových informací z dostupných dat nezíská – jakoukoli novou informaci zahodí. To ale rozhodně není náš záměr, pro který bylo filtrování textu vyvinuto. Naopak můžeme říct, že se na základě těchto informací zdá, že filtr funguje relativně dobře na rozdíl od čínštiny. Možná právě proto, že absolutní většinu znaků jazyka už obsáhl a jen se přesněji učí části slov.

5.3.2 Bosenština a chorvatština

Vzhledem k relativně velkému množství jazyků není překvapující, že některé jazyky jsou si velmi podobné. Jazyky často sdílejí celá slova nebo fráze se žádnými nebo tak malými rozdíly, že vedle nich se jeví slovenština a čeština jako úplně jiné jazyky. Netrénovali jsme sice případy jako britskou a americkou angličtinu, ale i tak nemusíme pro příklad příliš daleko.

Tři jazyky – bosenština, chorvatština a srbština jsou si navzájem velmi blízké a mluvčí těchto jazyků nemají větší problémy s porozuměním. A i když vypustíme srbštinu (pro tu uvažujeme cyriliku, ne latinku, což ji dělá relativně snadno odlišitelnou), jsou to jazyky velmi podobné a stojí za to je zkoumat.

Na rozdíl od předchozích případů, kdy jsme byli nuceni klást důraz na filtrování učicích dat, tady je situace o poznání lehčí – nesdílejí jeden jazykový kód (bosenština je dle ISO-639 buď `bs` nebo `bos`, zatímco chorvatština je označovaná `hr`, resp. `hrv`). Tedy až na případ, kdy se intenzivně citují fráze jednoho jazyka v textu druhého se nemusíme obávat podobných problémů jako jsme popsali v předchozí kapitole. Na druhou stranu si může program tyto jazyky plést: drobné rozdíly mezi *tačka* a *točka* nebo i delší fráze *Šta je rekajo?* proti *Što je rekajo?* může být statisticky zanedbatelná a znesnadňuje analýzu.

Použijeme stejného způsobu testování jako v případě části 5.2; výsledky si lze prohlédnout v tabulce 5.3. Vidíme, že už samotná přesnost u poznávání těchto jazyků je znatelně snížená a u kratších vzorků podstatně klesá. U těchto je totiž větší pravděpodobnost, že celý vstupní text je nerozeznatelný, je stejný jak v chorvatštině, tak v bosenštině. Proti takovým případům by byla potřebná analýza, jak blízko sebe jsou tipované jazyky – pokud například dostane bosen-

Tabulka 5.3: Test na zkracujících se náhodných vzorcích bosenštiny a chorvatštiny

délka	1	1/2	1/3	1/4	1/5	1/6
úspěšnost	73, 11%	68, 49%	63, 87%	57, 14%	54, 62%	55, 88%
shoda	47, 90%	42, 02%	43, 70%	34, 45%	35, 29%	38, 66%

ština ohodnocení 14 577 a chorvatština 15, 281, je velmi pravděpodobné, že jsou oba jazyky správnou odpovědí.

Kapitola 6

Diskuze: aplikace na cizí fráze Pražského závislostního korpusu

V této kapitole se pokusíme aplikovat program spolu s automaticky získanými daty na krátké fráze z Pražského závislostního korpusu. Nejdříve byly z prvních dvou trénovacích souborů dat extrahovány tyto fráze a pak ručně označeny jazykem, případně jazyky (pokud je možné označit spojení více způsoby – například „Koh-i-noor Hardthmuth“ je spojení výrazu hindštiny nebo bengálštiny pro horu světla a německé příjmení zakladatele). Několik zajímavých údajů už z tohoto procesu lze vidět v tabulce 6.1.

Už při anotaci však bylo podchyceno několik jevů, které naznačují horší výsledek:

1. Již v sekci 3.2.3 popsaný problém romanizace. Ačkoli se v závěru nasčítalo deset frází označených jako čínských, ani jedna samozřejmě nebyla v nativní znakové sadě. To sice nelze od českého periodika ani čekat, ale na druhou stranu se tyto případy špatně strojově cvičí a jsou mimo náš původní záměr použít data na Internetu pro automatické učení. Pokud bychom chtěli podchytit tyto případy, je nutné vyřešit hned několik dílčích problémů: například to, že se v různých kontextech používá jiné transkripce (např. jde-li o vlastní jméno, zeměpisný název apod.).

Tabulka 6.1: Statistické údaje z procesu anotace uzlů označených FPHR

celkem unikátních frází	247
průměrný počet uzlů na frázi	2, 87
průměrná délka fráze ve znacích	16, 77
nejkratší fráze	3 znaky[a]
nejdelší fráze	14 uzlů, 12 slov[b]
označeno jako angličtina	168
označeno jako němčina	19
označeno jako slovenština	6
neurčeno	4

[a] Bad

[b] Židia sa významne pričinili o založenie iného, rovnako hrozného totalitného molocha – komunizmu

Tabulka 6.2: Shrnutí výsledků aplikace metod a naučených dat na PDT

celkem frází	247	100,00%
správný jazyk byl jeden ze tří hlavních kandidátů[a]	158	63,97%
označeno přímo správným jazykem	115	46,56%
jazyk byl správně určen z alternativního zápisu	21	8,50%
jazyk nebyl nebyl natrénován nebo určen	7	2,83%
nebylo možné správně určit jazyk	73	29,55%

[a] respektive některý z očekávaných jazyků

2. Relativně velká (vzhledem k množství dat) četnost chyb. Právě cizí fráze jsou těžko kontrolovatelné, ať už gramaticky nebo faktograficky. Uveďme dva příklady za všechny:

   (a) věta „Tri dny ma naháňali“ kombinuje slovenskou slovní zásobu s českou gramatikou; slovo *deň* má v nominativu množného čísla v slovenštině tvar *dni*;

   (b) že jde o nepozornost, která vyústila ve faktografickou chybu, se domníváme u věty „Dohodu za USA podepsala Charlene Barshefská, která stojí v čele americké delegace, a za čínskou stranu náměstek ministra obchodu Sun Žen-žu.“ Z dostupných zdrojů se nepodařilo vypátrat žádného náměstka ministra obchodu, kterého jméno by se přepisovalo do češtiny právě tímto způsobem. Zato však existuje novinový článek [3] pravděpodobně popisující tu samou událost, pouze s tím rozdílem, že tam zmiňované jméno by podle české transkripce prof. Švarného znělo *Sun Čen-jü* (Sun Zhenyu).

3. Množství obchodních a na konkrétním jazyce nezávisle vytvořených názvů. Ještě je rozumné požadovat, aby se některá anglická příjmení označovala jako angličtina, ale pokud narazíme na slova jako Coca-Cola[1], SOS[2] nebo Penske Ilmor[3], nelze očekávat jejich úplně korektní zvládnutí.

I přes tyto problémy byl nakonec vytvořen test a výsledky jsou shrnuty v tabulce 6.2. Vidíme, že téměř třetina dat nebyla správně určena. Na druhou stranu, alternativní zápis textu (který jsme navíc uvedli jen k několika frázím) značně pomohl určování. Pro ilustraci dat, na které se dá v Pražském závislostním korpusu narazit, ještě přikládáme tabulku 6.3.

Výsledky vzhledem k problémům, které jsme popsali nejsou vůbec špatné, ostatně původní práce počítá s minimálním vzorkem 300 znaků, kterou jsme s průměrem kolem šestnácti znaků na cizí frázi několikanásobně snížili.

[1] pojmenovaná po dvou hlavních ingrediencích: listy kokainovníku (Erythroxylon coca) a afrických oříšků kola (Cola)

[2] SOS a fráze jako „Save Our Souls“ vznikly až zpětně po domluvě na signálu nouze jako mnemotechnická pomůcka; na rozdíl od tří znaků Morseovy abecedy se vysílá bez přestávek mezi nimi. A ačkoli „Save Our Sails“ se za angličtinu považovat dá, je podobnost SOS s kterýmkoli jazykem spíš náhodná.

[3] pojmenováno po trojici Američanů: Roger Penske, Mario Illien a Paul Morgan

Tabulka 6.3: Ukázka ručně anotovaných cizích frází Pražského závislostního korpusu a výsledků jejich analýzy

Data jsou z pochopitelných prostorových důvodů neúplná, kompletní soubor lze najít na přiloženém DVD. Vyjádření vzdálenosti od jednotlivých jazyků je zaokrouhleno na celá čísla. Hvězdička u očekávaného výsledku značí, že jazyk nebyl natrénován, resp. není vzorek žádným přirozeným jazykem. Výsledek „almost“ značí, že je očekávání mezi třemi kandidáty, „ok“ pokud je to přímo první kandidát a „alter“ pokud systém poznal ručně doplněný alternativní text (v tom případě se ale testuje pouze první kandidát na očekávání).

vzorek	alt. text	očekávání	kandidáti	analýza
de facto		latin	latin-1 10650 latin-2 12807 welsh 13258	ok
Les Marines de Gassin		french	french 31887 catalan 37695 occitan 38319	ok
know-how		english	english 49222 polish 56117 finnish 56724	ok
Šin Bet	שב"כ	hebrew*	german 32196 basque 40238 icelandic 42029	fail
Čchin Š' Chuang-tim[a]	秦始皇帝	mandarin	slovak 58379 tagalog 61527 indonesian 64028	alter
Buenos Aires		spanish	english 9698 spanish 13691 ido 26875	almost
tovaryšč Michajlov	товарищ Михайлов	russian	slovene 57779 slovak 58126 lithuanian 59094	fail
Taffy Tolu[b]		english	danish 52075 welsh 54169 english 54835	almost
Bocca della veritá	Bocca della Verità	italian	italian 26597 asturian 38730 catalan 40762	ok
Open GIS Ä[c]	OpenGIS®	english	swedish 48728 german 54488 nor-bokmal 54789	fail
4 PLEX PX - 43 CH[d]		neutral*	asturian 73717 bulgarian 75261 estonian 75582	fail
Pikes Peak		english	english 28251 spanish 42979 esperanto 45016	ok

[a] jedná se o čínského panovníka z let 247-210 př. n. l.; ovšem správně je v nominativu uvČchin Š' Chuang-ti (bez posledního „m“

[b] obchodní název pro první žvýkačky

[c] Ä je pravděpodobně chyba kódování dat

[d] technické označení CD-ROM mechaniky značky Plextor

Kapitola 7

Závěr

V této práci jsme postupně popsali vytvořené softwarové dílo a metody, které k němu vedly. Mimo jiné, strategie prohledávání internetových stránek a extrakce textu z nich. Navrhli jsme dvojúrovňový způsob filtrování, čímž jsme v kombinaci s předpokladem rozdělení stránek na sekce podle tagu `<p>` efektivně nahradili nutnost vyhledávat cizí bloky (úseky textu obsahující nejazyková nebo jinak nevhodná data: zdrojový kód na české stránce, úryvek ze Shakespearovy hry v originálu na stránce francouzské).

Způsob filtrování a extrakce učících dat jsme podchytili statisticky a provedli diskuzi. Zdá se, že pro většinu jazyků je vhodný, třeba dvě rozdílné verze latiny, i přes to, že čerpaly učicí data ze stejného zdroje (latinské Wikipedie), si udržely odstup a zůstaly rozeznatelné.

Naopak, filtr se ukázal jako nepříliš vhodný u jazyka jako je čínština, který obsahuje velké množství znaků (a zdaleka ne všechny z nich přítomny v ručně připraveném učicím textu). Dvě pozorované verze, jedna psána tradičními a druhá zjednodušenými znaky, sice zůstaly po skončení testu částečně rozlišitelné, ale obě obsahovaly velké množství znaků druhé varianty. V tomto případě by pomohla delší učicí data; ta ale typicky, pokud potřebujeme automatizovaně procházet Internet pro jejich získávání, nemáme.

Natrénovali jsme program na něco přes šedesát jazyků a po několika iteracích učení jsme provedli testy. Ty zahrnovaly jak vývoj dat v čase, tak i schopnost a úspěšnost určování jazyka ručně připravených vzorků. Ta dosahovala až cca $98,5\%$ u jedné věty. Sice se zkracujícím se vzorkem přesnost mírně klesla, u jedné šestiny věty se stále drží jen mírně pod hranicí 90%. To je, porovnáme-li tyto údaje se srovnatelnými čísly před započetím učení z Internetu, zlepšení v průměru o $7,35$ procentního bodu a u krátkých vzorků dokonce o $12,2\%$.

Provedli jsme diskusi, zda lze tyto výsledky aplikovat na cizí fráze („foreign phrases“, FPHR) Pražského závislostního korpusu. Sice jsme dosáhli značné přesnosti u slov, které jsou součástí přirozeného jazyka, pozorovali jsme však, že příliš velké množství těchto frází není v nativním písmu jazyka nebo se jedná o obchodní značky a jazykově neutrální data (technické označení součástek apod.).

A jak to dopadlo se slovíčkem „ale“ zmiňovaným v úvodu? Až mi někdo příště řekne, že se samozřejmě jedná o českou spojku, asi budu pochybovat. Z natrénovaných jazyků vede s přehledem rumunština.

Literatura

[1] Beesley Kenneth R. (1988): *N-Gram Based Text Categorization*,
Proceedings of the 29th Annual Conference of the American Translators Association

[2] Cavnar William B., Trenkle John M. (1944): *N-Gram Based Text Categorization*,
Environmental research institute of Michigan

[3] Deseret News: *Trade Peace! U.S., China Sign Copyright Act*
27. února 1995

[4] Hearst M. (1997): *TextTiling: Segmenting Text into Multi-Paragraph Subtopic Passages*,
Computational Linguistics, 23 (1), str. 33-64

[5] Toman Josef (2007): *Statistické rozpoznávání jazyka*,
bakalářská práce, Ústav formální a aplikované lingvistiky

[6] Xiao Jinghui, Liu Bingquan a Wang Xiaolong: *An Empirical Study of Non-Stationary Ngram Model and its Smoothing Techniques*,
Computational Linguistics and Chinese Language Processing,
Vol. 12, No. 2, červen 2007, str. 127-154

[7] Zahornadský Ján: *Crawler, uživatelská a programátorská dokumentace k ročníkovému projektu*

Online zdroje

[8] *Anotace pražského závislostního korpusu na tektogramatické rovině*
http://ufal.mff.cuni.cz/pdt2.0/doc/manuals/cz/t-layer/html/index.html
http://ufal.mff.cuni.cz/pdt2.0/doc/manuals/cz/t-layer/pdf/t-man-cz.pdf

[9] *HTTP/1.1 Header Fields Definitions*
http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html

[10] *Official Google Blog: Moving to Unicode 5.1*
http://googleblog.blogspot.com/2008/05/moving-to-unicode-51.html

[11] *Official UN Declaration of Human Rights Home Page*
http://www.unhchr.ch/udhr/

Uvedené internetové zdroje jsou všechny ke dni 6. července 2008.