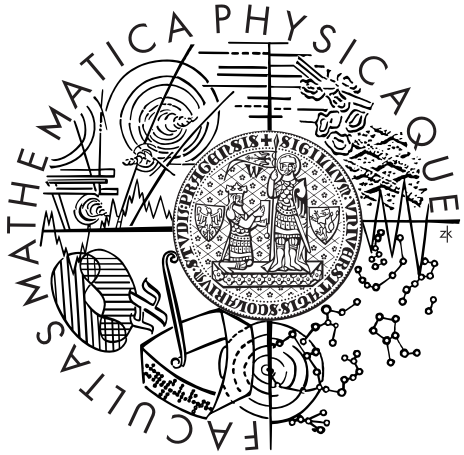
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE



Bc. Ondřej Odcházel

Automatické doporučování ilustračních snímků

Ústav formální a aplikované lingvistiky

Vedoucí diplomové práce: RNDr. Pavel Pecina, Ph.D.
Studijní program: Informatika
Studijní obor: Matematická lingvistika

Praha 2014

děkuji máje, že se nebojí létat

Prohlašuji, že jsem tuto diplomovou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V ..... dne ..... Podpis autora

Název práce: Automatické doporučování ilustračních snímků

Autor: Bc. Ondřej Odcházel

Katedra: Ústav formální a aplikované lingvistiky

Vedoucí diplomové práce: RNDr. Pavel Pecina, Ph.D., Ústav formální a aplikované lingvistiky

Abstrakt:

Klíčová slova: vyhledávání obrazových informací

Title: Automatic suggestion of illustrative images

Author: Bc. Ondřej Odcházel

Department: Institute of Formal and Applied Linguistics

Supervisor: RNDr. Pavel Pecina, Ph.D., Institute of Formal and Applied Linguistics

Abstract:

Keywords: information retrieval, image retrieval

Obsah

1. Úvod

Cílem diplomové práce je implementovat kompletní webovou aplikaci pro doporučování a vyhledávání ilustračních obrázků v textu. Vytořit takovou aplikaci přináší mnoho rozličných úkolů a problémů. Tato kapitola se bude snažit tyto problémy načrtnout. Další kapitola se bude jednotlivými problémy zabývat podrobně.

1.1 Práce s daty

Zadaná data obsahují 20 milionů anotací obrázků. Základním úkolem je být schopen takové množství dat vůbec nahrát do databáze a být schopný obsloužit mnoho požadavků za minutu. Bude zmíněn současný stav vývoje databázového software pro práci s velkými daty zejména s ohledem na snadnost hledání a škálovatelnost.

1.2 Extrakce klíčových slov

Extrakce klíčových slov je důležitý podobor NLP. V práci budou rozebrány algoritmy pro extrakci klíčových slov. Bude kladen zejména důraz na rychlost a nenáročnost na zdroje. Z uživatelských testovaní společnosti Google vychází, že rychlost načtení stránky je jedním z klíčových vlastností pro spokojenost uživatele. Klíčová slova budou mít v aplikaci dvě využití. Pokud uživatel zadá pouze text článku, extrahovaná klíčová slova se použijí na vyhledávání relevantních obrázků. Prvních několik klíčových slov bude navíc použita jako nápověda uživateli, ten pak může tato klíčová slova využít k exaktnímu omezení množiny klíčových slov.

1.3 Překlad do češtiny

Popisky klíčových slov jsou v angličtině. Tato práce řeší překlad množiny klíčových slov do češtiny. Kromě překladu je pro hledání také nutno implementovat algoritmus na stemming. Celá aplikace je navržena tak, aby případný další jazyk mohl být přidán co nejjednodušeji.

1.4 Detekce jazyka

Jednou z drobností, kterou ocení uživatel aplikace je detekce jazyka. Uživatel bude mít možnost zadat jazyk vstupního článku exaktně, ale aplikace bude také jazyk vstupního textu sama detekovat. Budou prozkoumány možnosti detekce jazyka. Opět se nejedná o nějakou klíčovou funkci aplikace. Výstup detekce bude moci být uživatelem měněn (podobně jako funguje Google Translate[1]), důraz bude tedy kladen na rychlost a jednoduchost.

[1] https://translate.google.com/

1.5 Webová aplikace

Všechny předchozí komponenty se spojí v jedné webové aplikaci. Webový vývoj zažívá bouřlivý rozvoj. Na backendu jsou nové zejména způsoby práce s velkým množstvím dat v distribuovaném prostředí. Ve frontendové části probíhá rozvoj pomocí implementace nových technologií, známých pod hlavičkou HTML5, do moderních prohlížečů. Práce bude rozebírat všechny možnosti tvorby moderních webových aplikací.

1.6 Testování

Aplikace bude otestována na několika úrovních. Extrakce klíčových slov bude otestována pomocí korpusu článků a klíčových slov. Bude vytvořena komplexní webová aplikace pro testování doporučených obrázků. Tato aplikace bude vydělena ze samotné webové aplikace a bude používána i nezávisle.

2. Zadání

Většina zpravodajských serverů často opatřuje publikované články tzv. ilustračními snímky, jejichž úkolem je vizuálně dokreslovat obsah článku a upoutat na něj čtenářovu pozornost. Ilustrační snímky většinou pocházejí z rozsáhlých fotografických databází, jsou vybírány autory článku a s obsahem článku souvisejí jen relativně volně. Výběr ilustračních snímků probíhá nejčastěji na základě porovnávání klíčových slov specifikovaných autorem textu a popisků, kterými jsou obrázky v databázi opatřeny (typicky svými autory).

Proces výběru ilustračních snímků (dotazování ve fotografické databázi) je obtížný jednak pro samotný vyhledávací systém (hledání relevantních fotografií na základě uživatelských dotazů), jednak pro autory, kteří musí dotazy vytvářet. Konstrukce dotazů spočívá v několika krocích: uživatel nejdříve musí identifikovat ústřední téma (či témata) článku, které chce ilustrovat vhodnou fotografií, a ta potom popsat vhodnými klíčovými slovy, zvolit a zkombinovat je tak, aby vedla k nalezení vhodného obrázku. Tento proces by mohl být zjednodušen tím, že konstrukce dotazů pro vyhledávání bude prováděna automaticky pouze na základě textu článku.

Cílem diplomové práce je navržení a implementace komfortní webové aplikace pro automatické navrhování ilustračních snímků na základě textu článku, bez nutnosti explicitně konstruovat vyhledávací dotazy. Součástí práce bude i uživatelská evaluace celého systému. Pro experimenty bude použita kolekce ilustračních snímků od společnosti Profimedia.

3. Teorie: Jak najít vhodné obrázky

3.1 Popis datové sady

Datová sada poskytnutá firmou Profimedia obsahuje 20 014 394 oanotovaných obrázků ve formě CSV souboru "profi-text-cleaned.csv". CSV obsahuje sloupce "locator","title","description","keywords". Sloupec locator obsahuje ID obrázku v databázi Profimedia. Sloupec description je prázdný. Sloupce title a keywords obsahují řetězce anglických slov popi sujících obrázek a oddělených mezerou.

Příklad jednoho řádku souboru profi-text-cleaned.csv:

```
"0000000980","hradec kings holy ghost cathedral","",
   "outdoors nobody urban scenes architecture houses
   towers czech czech republic europe buildings build
   history historical churches church fronts holy
   ghost cathedral spirit ceska republika cathedrals
   sv hradec kralove"^M
```

Na příkladu je vidět, že data obsahují fráze jako "holy ghost cathedral", tyto fráze však nejsou strojově čitelně vyznačené. Dalším problémem je špatný překlad dat do angličtiny. Fráze "hradec kings" vznikla evidentně doslovným překladem názvu "hradec králové".

Důležitým aspektem dat je jejich nepodobnost běžnému novinovému textu. Anotované texty neobsahují většinu nejfrekventovanějších anglických slov.

3.2 Teoretické cíle

V teoretické části je hlavním cílem práce nalézt nejvhodnější metodu extrakce klíčových slov textu. Tato klíčová slova pak budou použita při vyhledávání ilustračních obráyků v databázi Profimedie. Celá práce, pokud nebude uvedeno jinak, označuje za slova stemy vstupních slov. Jako stemmer se využívá ??? stemmer.

Vstupní text tedy nejprve rozdělíme na slova. Čísla a interpunkce nás v této úloze nezajímají, jelikož se v datech nenachází. Ze slov pak získáme stemy. Vstupem algoritmu pro nalezení klíčových slov tedy bude množina vhodných stemů. Ke každému stemu si ještě uložíme jednu jeho nestemovou variantu, kterpu pak můžeme zobrazit uživateli.

Nyní můžeme použít některý z algoritmů na extrakci klíčových slov uvedených v další kapitole.

3.3 Rešerše, vhodné algoritmy

Algoritmy na extrakci klíčových slov lze rozdělit do dvou kategorií. V jedné máme k dispozici korpus podobných dokumentů, druhá kategorie tento korpus ke své práci nepotřebuje.

3.3.1 TF-IDF

Technika TF-IDF je známý algoritmus na měření významnosti slov v textu. Využívá korpusu dokumentů D a dvou složek.

$$TFIDF(t, d, n, N) = TF(t, d) \times IDF(n, N) \tag{3.1}$$

Složka TF znamená $TERM\ FREQUENCY$ a pokud t je slovo a $d \in D$ je dokument, je TF

$$TF(t, d) = \begin{cases} 1 & \text{pokud } t \in d \\ 0 & \text{jinak} \end{cases} \tag{3.2}$$

$$TF(t, d) = \sum_{slovo \in d} \begin{matrix} 1 & \text{pokud } slovo = t \\ 0 & \text{jinak} \end{matrix} \tag{3.3}$$

Jedná se tedy o frekvenci slova (stemu) v dokumentu.

Složka IDF, tedy $INVERSE\ DOCUMENT\ FREQUENCY$ vyjadřuje, jak moc daný termín popisuje dokument. Pokud je N počet všech dokumentů v D, tedy $N = |D|$ a n je počet dokumentů, ve kterých se vyskytuje slovo t, je IDF tohoto slova

$$IDF(n, N) = \log\left(\frac{N}{n}\right) \tag{3.4}$$

$$IDF(n, N) = \log\left(\frac{N - n}{n}\right) \tag{3.5}$$

Čím je tedy slovo v korpusu častější, tím více se s logaritmem snižuje jeho informační hodnota. Slova, která jsou velmi běžná většinou klíčovými slovy nejsou.

Výsledný vzorec pak jde shrnout jako:

$$TFIDF(t, d, n, N) = \left(\sum_{slovo \in d} \begin{matrix} 1 & \text{pokud } slovo = t \\ 0 & \text{jinak} \end{matrix} \right) \times \log\left(\frac{N - n}{n}\right) \tag{3.6}$$

Problémem tohoto algoritmu pro je odlišný charakter korpusu a vstupních dat. Vstupní data jsou typicky novinový článek. Pokud bychom jako korpus použily anotované obrázky, získáme špatné výsledky. Běžná anglická slova, jako "the", nebo "a" se v takovém korpusu vyskytují velmi zřídka, jejich IDF tedy bude vysoká. Naopak TF v běžném novinovém textu je vysoké. Takovýto korpus nám pak označuje jako klíčová slova běžná anglická slova.

3.3.2 Extrakce bez korpusu

Dalším druhy algoritmů ke své práci korpus nepotřebují a pracují pouze se vstupním textem.

3.4 Řešení: jaké algoritmy zvoleny, získání tréninkových dat

Jako nejvhodnější řešení byl nakonec zvolen TF-IDF algoritmus. Jako kandidáti jsou odfiltrována slova, která se nenacházejí v datech Profimedie. Jako korpus k měření IDF byla použita data článků z Wikipedie.

Pro rychlé testovací účely bylo oanotováno pár článků z anglických wikinews. V každém článku jsem označil pět klíčových slov. Porovnávání algoritmů na extrakci klíčových slov pak vzalo pět nejpravděpodobnějších klíčových slov podle algoritmu a porovnalo v kolika procentech se algoritmus trefil s anotací.

Zbývá: Zkusit otestovat další metody. Překlad do češtiny.

3.5 Evaluace výsledků

Výsledkem práce by mělo být rozhraní pro anotaci obrázků využívající algoritmus na hledání klíčových slov v textu.

Tento algoritmus se dá uživatelsky testovat několika způsoby. Uživatel vidí text a několik (cca 5) vrácených obrázků algoritmem. Uživatel vybere množinu relevantních obrázků. Další možností je mezi 5 vrácených obráyků vložit jeden náhodný. Úkolem anotátora je pak vybrat ten náhodně vybraný. Přesnost algoritmu pak jde měřit pomocí toho, kolikrát se anotátor trefí do špatného obrázku (potřeba zdrojový článek)

4. Tvorba multijazyčného vyhledávání

Metadata k obrázkům jsou v anglickém jazyce. Tato práce se snaží

5. Praktická část: Implementace moderní webové aplikace

Práce je z velké části implementační. Snažil jsem se tedy vytvořit moderní webovou aplikaci s využitím co nejvíce frontendových novinek v novém standardu HTML5.

5.1 Databáze: jak uložit 20M metadat obrázků

Jak uložit takové množství dat, aby se dalo rychle vyhledávat. Škálovatelnost. Dostupnost knihoven pro práci s databázema. Proč jsem si vybral nakonec ES. SQL vs NoSQL.

5.2 Backend a úprava dat: Komunikace s databází, implementace algoritmů

Proč jsem zkoušel golang a proč jsem nakonec použil Ruby a Ruby on Rails. Základní popis MVC frameworku. Dostupnost knihoven pro získání stemů a prací s databází.

5.3 Frontend: AJAXová aplikace na zobrazování obrázků

Jaké jsou dnešní možnosti vývoje frontendu. Single page aplikace. Možnosti moderních prohlížečů. JavaScriptové knihovny. Proč to nedělám v jQuery, ale používám Google Closure. Google Closure Library, Templates, Compiler.

Návrh rozhraní bez jediného tlačítka. Responzivní webdesign.

5.4 Anotační rozhraní

Jak lze v Ruby on Rails vyrobit jednoduše anotační rozhraní s uživateli a s ukládáním do databáze.

5.5 Návod k použití

Popis prvků. Screenshoty aplikace.

5.6 Preklad

Dva postupy jak pouzit anglicka data v jinych jazycich. Bud je mozne prelozit vzdy zadany cesky dotaz do ciloveho jazyka. Nebo je mozne prelozit vsechna data

u fotek. Pak je nutne pouzit pro kazdy jazyk nejaky lemmatizer, nebo stemmer. Pro cestinu jsme nakonec zvolili druhou variantu.

Jak prelozit 20 milionu popisku? Jednou moznosti je preklad slovo od slova. Pouzit pouze slovnik. Preklad pomoci google je drahy. Stalo to zhruba 1300Kc. Preklad celych frazi by byl lepsi (automaticky objevi fraze), ale pomoci google velmi drahy. Rozchodil jsem tedy prekladovy nastroj Moses s modelem prilozenym ve verzi 2.1 (http://www.statmt.org/moses/RELEASE-2.1/models/en-cs/model/). Po nekolikahodinovem nacitani se model nacetl (i kdyz mam SSD disk). Preklady v prolozenem modelu jsou velmi pomale (jeden segment trva priblizne 3s). Preklad neni idealni a mam spustu —UNK slov.

Google: "0000000003", "malé dítě s úsměvem", "", "dítě děti dítě děti kojenci děti dětství jednotlivé plochy těla nahá nake výrazy obličeje, úsměvu, usměvavý sledování sledování kterým zábava zábavné zábavní pobavený pobavit laškoval frolicing hrát wantoning otevřený" ^{M}

Preklad jineho obrazku: "0000000102","young woman cleaning teeth","","single faces people humans young youth hands indoors interiors woman women females blond fair young adult s girls close view beauty home home dental bathrooms person portrait adult years half length portrait open mouth hygiene teeth dental care years cleaning toothbrush underwear bras"^{M}

Moses: "0000000102","young—UNK—UNK—UNK žena čištění teeth","","single—UNK—U čelí mladí lidé , lidé mládež rukou uvnitř interiors—UNK—UNK—UNK žena žen , žen , blonďák spravedlivé mladé dívky zavřít dospělé s cílem krásy vnitřní vnitřní stomatologické koupelny portrét dlouhé roky polovina dospělé osoby portrét otevřené úst hygienické zuby zubní kartáček prádlo bras"$^{M}|UNK|UNK|UNKpeletitn$

Google Translate: "0000000102", "Mladá žena čištění zubů", "", "jednotlivé tváře lidí, lidé younge mládeží ruce interiéry ženě ženám ženy ženskému blond fair mladý dospělý s dívek close view krása domov domácí zubní koupelny osoba portrét dospělý let poloviční délka portrét otevřená ústa hygienické zubů zubní péče roky čistící kartáček na zuby spodní prádlo podprsenky "^{M}

Detekovane fraze: http://mufin.fi.muni.cz/ xbatko/keyword-clean-phrase-export.csv.gz mail: https://mail.google.com/mail/u/0/search/pecina+

export frazi bez omezeni: wc phrases$_{l}ist.txt$11435135151831371315 75$phrases_{l}ist.txt-takesenedaprelozitrozumnevgoogle$

omezeni na maximalni delku 4 slova: wc phrases$_{l}ist.txt$8992442204616156770 87$phrases_{l}ist.t$

extrakce frazi: http://stackoverflow.com/questions/1643616/algorithms-to-detect-phrases-and-keywords-from-text

detektor jazyka: trigramy, nefunguje na frazi "hello world", chce to dlouha data, ale pro bezne clanky to funguje dobře

5.7 Poznamky

Pouziji data z wiki dumpu.

Cesky dump: http://dumps.wikimedia.org/cswiki/20140612/ a cswiki-20140612-pages-articles-multistream.xml.bz2 /data/wiki$_{d}ump_{c}s.xmlrakees : extract_{w}ords_{f}rom_{w}iki[cs]e$ $60MBtextucestinaobsahujenazacatkuhodnekratkychvygenerovanychclankutypu >$ $1.leden <$

Anglicky dump: torrent z piratebay http://thepiratebay.se/torrent/8114722/Wikipedia$_{2}013$ $extract_{w}ords_{f}rom_{w}iki[en]extrakcedoid10000 > 70MBtextu$

application.rb obsahuje konstanty aplikace

vyrobim nejcastejsi trigramy pro cestinu a anglictinu z wiki dat pomoci: rake es:extract$_{m}ost_{f}requent_{t}rigrams$

Language data jsem stahnul z: http://dumps.wikimedia.org/cswiki/latest/

Ceska data jsou z: /net/seznamdata/profiset/profi-text-cleaned.csv

prekladova data: moses - http://www.statmt.org/moses/RELEASE-2.1/binaries/macosx-mavericks/bin/

prekladovy model: http://www.statmt.org/moses/RELEASE-2.1/models/en-cs/

6. Možnosti tvorby moderního webového frontendu

Možnosti tvorby webových aplikací se posledních několik let rapidně zvětšují. Prohlížeče implementují stále nové technologie, které royšiřují možnosti.

6.1 Možnosti programování frontendu

Programátor webového frontendu si dnes může vybírat z několika paradigmat tvorby webové stránky. Standardem je dnes programovací jazyk Javascript. Spíše historicky bylo výhodné místo Javascriptu používat pro frontendový vývoj různé pluginy. Nejznámější je Adobe Flash, nebo Microsoft Silverlight. Programovat pro tyto pluginy mělo velké výhody. Například snadné přehrávání videa, zobrazení stránky v celoobrazovkovém režimu, nebo přístup k uživatelově webkameře. V době, kdy byla Javascriptová API velmi chudá a rozdílně implementovaná mezi prohlížeči, nabízel Flash zejména tvůrcům webových her konzistenci mezi všemi platformami.

Velkou nevýhodou těchto pluginů byla jejich proprietálnost. Ostatní firmy se bály pustit cizí plugin do svých výrobků. Zlomový moment pro ústup Flashe ze slávy bylo uvedení telefonu iPhone, který podporu pro Flash nenabízel. Softwareoví giganti Microsoft, Apple a Google začali místo proprietálních řešení tlačit otevřenou specifikaci, která se později nazvala HTML5. HTML5 je zastřešující termín pro spoustu technologií, které snaží webová API specifikovat a vytvářet nové.

6.2 JavaScriptové frameworky

Druhým hnacím motorem moderního Javascriptu jsou frameworky. Ještě před několika lety byla práce na interaktivních webech velmi náročná. Prohlížeče se zásadně lišily v implementaci práce s eventy a DOMem. Nové frameworky do jisté míry odstínily programátora od odlišných implementací Javascriptu v prohlížečích.

6.2.1 jQuery

Nejpopulárnějším frameworkem současnosti je jQuery. Ten nabízí jednoduché rozhraní a pro velkou většinu menších webových aplikací je zcela dostačující. Čím je ale aplikace větší, tím začíná být její vývoj s pomocí jQuery náročnější. Zkusme ukázat, jak by v jQuery vypadalo přidání CSS třídy red oknu s id okno:

```
$("#okno").addClass("red");
```

Kód má několik problémů. Pokud neexistuje žádné okno s id okno, jQuery nevrátí žádnou chybu. Stačí malý překlep a chyba v kódu se hledá dost obtížně. jQuery nenabízí žádnou funkci typu vratObjektPodleId. Pokud by tyto funkce nabízel, velikost jeho kódu by se zvětšila. Pokud chce programátor použít knihovnu jQuery, musí si ji uživatel stránky celou stáhnout. Verze XX má XX bajtů.

6.2.2 Google Closure

Jiný přístup k vývoji frontendových aplikací přináší Google. Pro svou první webovou aplikaci Gmail vyvinul sadu nástrojů, kterou později vydal jako open source pod názvem Google Closure. Kromě Gmailu ji Google využívá v Google Vyhledávání, Google Mapách, nebo Google Dokumentech. Skládá se ze tří částí - Closure Compiler, Closure Library a Closure Templates. Closure Library je obsáhlá knihovna funkcí pro práci s DOMem, Eventy, matematickými výpočty a spoustou dalších věcí, které webový programátor může využít.

Closure Compiler je inteligentní minifikátor Javascriptového kódu. Odstraňuje funkce, které nejsou volány, přejmenovává všechny názvy funkcí a proměnných na co nejkratší řetězce a v ADVANCED módu se snaží i o pokročilejší optimalizace kódu (například kód funkcí, které jsou volány pouze jednou, je vložen inline). Nejlepích výsledků dosahuje s použitím speciálních anotací, které například vynucují typ proměnné a jsou schopny udělat z Javascriptu typovaný jazyk. Closure Compiler tyto anotace vyhodnocuje a při chybném přiřazení hodnoty vrátí chybu. To umožňuje tvořit více bezpečný javascriptová kód.

Třetí částí Google Closure jsou Closure Templates, šablonovací systém pro Javascript a Javu. Pomocí Closure Templates se snadno vytváří zanořené HTML šablony. Všechny uživatelské vstupy jsou escapované což zabraňuje sniffing útokům.

Částí Templates a Compiler jdou použít odděleně v jakémkoliv Javascriptovém projektu. Používat Closure Library be Compileru nedává příliš smysl, uživatel by při návštěvě webu musel stahovat ohromné množství zbytečných dat.

Tato práce na frontendu používá všechny části knihovny Google Closure. Díky tomu si uživatel při první návštěvě webu musí stáhnout pouze jediný soubor, který má pouze XX Kb.

6.3 Stylování uživatelského rozhraní a CSS

Specifikace HTML5 rozšiřuje i možnosti vizualizace pomocí CSS stylů. Nejviditelnějšími novinkami je podpora kulatých rohů, stínování, nebo barevných přechodů. Webový programátor nyní může ke stránce načíst i vlastní font. Všechny tyto možnosti velmi rozšířily možnosti webovým grafikům.

7. Instalace a zprovoznění

Celé anotační rozhraní je webová aplikace napsaná v jazyce Ruby a frameworku Ruby on Rails. Je k dispozici pod svobodnou licencí MIT. K jejímu spuštění potřebujete ruby verze alepoň 2.0 (nižší verze nejsou otestované), javu a ke stažení zdrojového kódu git. Program jde spustit na Linuxu a Macu.

7.1 Instalace

Zdrojový kód je volně dostupný na webu GitHubu[1]. Stáhnou tedy lze příkazem

```
git clone https://github.com/hypertornado/diplomka
```

Tento příkaz vytvoří adresář diplomka. Závislosti aplikace nainstalujete pomocí bundleru:

```
bundle install
```

Instalace může vyžadovat přístup administrátora. Dále je potřeba stáhnout knihovnu elasticsearch[2] do adresáře bin/elasticsearch. Stačí verze 1.0 a vyšší. Ve verzi 1.2.1 jsme objevili menší chybu[3], která je způsobena chybou v Javě a jde obejít nastavením delšího hostname počítače.

Nyní je možné celý projekt spustit. Nejprve se spustí elasticsearch databáze pomocí příkazu `rake es:start`, poté je možné spustit samotnou aplikaci příkazem `rails server`. Po spuštění severu je uživatelské rozhraní dostupné ve webovém prohlížeči na adrese `http://localhost:3000`. Po načtení stránky se zobrazí uživatelské rozhraní, ale veškeré AJAXové dotazy skončí chybou. V databázi nejsou importována data.

7.2 Práce s metadaty k obrázkům

Metadata k obrázkům a obrázky samotné jsou poskytovány firmou Profimedia a nejsou volně dostupné. Ke zprovoznění aplikace je nutné vložit CSV soubor `keyword-cleaned-phrase-export.csv` do adresáře data.

7.3 Překlad metadat

Soubor obsahuje metadata k obrázkům v angličtině. Jedním z úkolů této práce je poskytnout doporučování obrázků i v jiných jazycích, primárně v českém jazyce. Bylo tedy nutné metadata přeložit. Pokoušeli jsme se použít volný nástroj na překlad Moses. Ve verzi 2.1[4] nabízí volně dostupné modely pro překlad z češtiny do angličtiny. I na SSD disku trvá několik hodin, než se překladový model načte do paměti. Překlad jednoho segmentu s tímto modelem byl poměrně pomalý (překlad metadat k jednomu obrázku trval zhruba 3 sekundy) a také dosti nepřesný. Například řádek

[1] `https://github.com/hypertornado/diplomka`
[2] `http://www.elasticsearch.org/downloads/1-0-3/`
[3] `https://github.com/elasticsearch/elasticsearch/issues/6611`
[4] `http://www.statmt.org/moses/RELEASE-2.1/models/en-cs/model/`

```
"0000000003","little baby smiling","","child children
  baby babies infants kids childhood single faces
  body naked nake facial expressions smile smiling
  viewing watching laying fun amusing amusement
  amused amuse dallying frolicing playing wantoning
  open"^M
```

byl do čestiny přeložen takto:

```
"0000000003","little|UNK|UNK|UNK d t
  smiling","","child|UNK|UNK|UNK d t , d tsk
  d ti kojence d ti d tstv jednotn ho el
  org n nah nake|UNK|UNK|UNK po dili
  vyj d en usm vat usm v od vodn n m ,
  kter z bavn sledovat z bavn z bavn ch i
  pobavena t m amuse|UNK|UNK|UNK
  dallying|UNK|UNK|UNK frolicing|UNK|UNK|UNK hr t
  wantoning|UNK|UNK|UNK open"^M|UNK|UNK|UNK
```

Je vidět poměrně velké množství nepřeložených slov (konconky `|UNK`) a překlad je relativně nepřesný. Je možné

Nejprve příkazem `rake data:export_profimedia_words_for_translation` vyexportujeme do souboru `data/word_list.txt` seznam všech slov použitých v metadatech k obrázkům. Tento příkaz běží několik hodin i na moderním počítači s SSD diskem. Z Profimedia dat získáme seznam 352862 slov. Tento soubor je nutné přeložit z angličtiny do dalších podporovaných jazyků, v našem případě češtiny.

8. Evaluace výsledků

8.1 Metodika

Jak budu měřit. Porovnání naivního a nejlepšího algoritmu. Nejlepší algoritmus vybrán pomocí oanotovaných klíčových slov ve Wikinews článcích. Pak anotátoři se budou snažit najít nejméně vhodný obrázek z nabízených hodnot.

Evaluace jenom pro angličtinu, nebo i pro češtinu?

8.2 Výsledky

Grafy, tabulky.

8.3 Možná zlepšení

Jak bychom kmohli mít lepší data a co omezuje použitý algoritmus.

9. Závěr

Seznam použité literatury

[1] Lamport, Leslie. *LaTeX: A Document Preparation System*. 2. vydání. Massachusetts: Addison Wesley, 1994. ISBN 0-201-52983-1.

Seznam tabulek

Seznam použitých zkratek

Přílohy