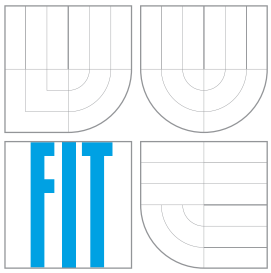
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

SYSTÉM PRO PROVÁZÁNÍ TEXTŮ STÁTNICOVÝCH TÉMAT, STUDIJNÍCH OPOR A DOPLŇKOVÝCH MATERIÁLŮ

System for interlinking texts of state exam topics, learning support and other supplementary materials

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAKUB HRADÍLEK

VEDOUCÍ PRÁCE
SUPERVISOR

doc. RNDr. PAVEL SMRŽ, Ph.D.

BRNO 2016

Abstrakt

Hlavním úkolem této práce je se seznámit s metodami vyhledávání definic odborných pojmů napříč texty. Následně navrhnout a vytvořit systém, který bude schopen propojit texty státnicových témat, studijních opor a doplňkových materiálů. Na závěr vyhodnotit vytvořený systém na materiálech z VUT FIT v Brně a zhodnotit výsledky vzhledem k použitelnosti výstupů pro přípravu studentů k závěrečným zkouškám.

Abstract

The main goal of this thesis is to survey methods which are used for keyword extraction from articles and text documents. After that design and create system, which will be able to interlink texts of state exam topics, learning support and other supplementary materials. Finally step is evaluate the created system to materials from VUT FIT in Brno and appraise results in applicability for preparing students for final exams.

Klíčová slova

klíčová slova, vyhledávání klíčových slov, extrakce, morfologická analýza, rejstřík

Keywords

keywords, keywords search, extraction, morphological analysis, index

Citace

Jakub Hradílek: Systém pro provázání textů státnicových témat, studijních opor a doplňkových materiálů, bakalářská práce, Brno, FIT VUT v Brně, 2016

Systém pro provázání textů státnicových témat, studijních opor a doplňkových materiálů

Prohlášení

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

.....
Jakub Hradílek
17. května 2016

Poděkování

Rád bych poděkoval panu doc. RNDr. Pavlu Smržovi, Ph.D. za poskytnutí odborné pomoci, užitečných rad a za veškerý čas, který mi věnoval.

Obsah

1 Úvod **3**

2 Metody vyhledávání klíčových slov **5**
2.1 Četnost slov . . . 5
2.2 Weirdness . . . 5
2.3 Term Frequency . . . 6
2.4 Term Frequency - Inverse document frequency . . . 6
2.5 Okapi BM25 . . . 7
2.6 C-value . . . 8

3 Určování relevantních slovních spojení **9**
3.1 Pearsonův X^2 test . . . 9

4 Morfologický analyzátor **10**
4.1 MorphoDiTa: Morphological Dictionary and Tagger . . . 10
4.2 Ajka . . . 12

5 Použité nástroje **13**
5.1 PDFMiner a Slate . . . 13
5.2 Python . . . 13

6 Návrh a implementace systému **15**
6.1 Požadavky na systém . . . 15
6.2 Struktura systému . . . 15
6.3 Převod materiálů z PDF do textové podoby . . . 17
6.4 Úprava textu . . . 17
6.5 Výběr a tvorba klíčových slov . . . 18
6.5.1 Parsování otázek . . . 19
6.5.2 Značkování . . . 19
6.5.3 Zpracování n-gramů . . . 19
6.5.4 Stoplist . . . 20
6.5.5 Backgroundový model . . . 20
6.5.6 Odsekávač slov . . . 21
6.5.7 Určení témat a vyhledání klíčových slov . . . 22
6.6 Implementace systému a jeho použití . . . 22
6.7 Výsledný soubor a jeho struktura . . . 24

7 Testování a výsledky **26**
7.1 Kvalita převodu PDF souborů 26
7.2 Slovní druhy 27
7.3 Délka jednotlivých slov 27
7.4 Průměrná délka klíčových slov 28
7.5 Redundance klíčových slov 28
7.6 Praktické použití systému 30

8 Závěr **32**

Literatura **34**

Kapitola 1

Úvod

Odborné studijní materiály na každé škole jsou nejdůležitějším zdrojem informací pro tamější studenty. Všechny tyto materiály obsahují velké množství informací, potřebné pro zvládnutí zkoušek a úplně nakonec i zvládnutí státních závěrečných zkoušek. Pro vyhledávání v takových materiálech by měl sloužit hlavně rejstřík. Bohužel, ne každá publikace obsahuje rejstřík, který by pomohl studentům projít velké množství informací a následně je směrovat pouze ke konkretním částem.

Jedna možnost je vytvořit rejstřík pro každou publikaci, ve které chceme vyhledávat. Tato možnost je příliš pracná, protože pro co nejlepší výsledky je nutné udělat rejstřík ručně. To je ale nadlidský výkon pro jednoho člověka a navíc by tato možnost ne-eliminovala ruční prohledávání rejstříku každé publikace. Druhou možností je vytvořit systém, který by byl schopen zpracovat velké množství informací a vytvořit rejstřík, nebo "rozcestník" pro studenty, kteří hledají konkrétní informaci nebo konkrétní probíranou látku a nemají čas procházet velké množství materiálu. O druhé možnosti bude pojednávat tato bakalářská práce.
Cílem práce je seznámit se s metodami umožňující správné detekování klíčových slov a vytvořit systém, který bude schopen na základě vstupních dat vyhledat klíčové slova napříč zpracovaným textem. Práce se zaměřuje na české texty a kvalita bude měřena na technických textech Fakulty informačních technologií. To neznamená, že systém bude zaměřen pouze na tyto texty, bude ho možné použít i na humanitní obory, které obsahují více textu a méně definic či rovnic.

Protože je čeština flektivní(ohebný) jazyk, je mnohem těžší vyhledat konkrétní informaci napříč textem. Pokud chceme hledat klíčové slovo, nebo víceslovné klíčové slovo, tak nestačí hledat pouze konkrétní napsaný tvar, ale také i vyskloňované verze tohoto slova, abychom dosáhli přesnějších výsledků. Pokud bychom hledali pouze jeden tvar slova, mohlo by nám uniknout spoustu důležitých informací a zmínek v textech, které odkazují na další informace.

Hlavní cílovou skupinou systému jsou odborné texty studijních opor a materiálů na Fakultě informačních technologií VUT v Brně. Nástroj by měl studentům ulehčit práci hlavně s vyhledáváním klíčových slov obsažených ve státnicových tématech napříč všemy odbornými texty na škole a tím studentům umožnit lepší a pohodlnější přípravu na závěrečné zkoušky.

V následující kapitole 2 a 3 jsou rozebrány běžně používané algoritmy pro určování klíčových slov, které byly brány v potaz při návrhu systému. Kapitola 4 popisuje morfologické analyzátory, které slouží pro značkování textu a jsou důležitým kom-

ponentem systému, bez kterého by nebylo možné převádět slova na jejich základní tvary. Následně v kapitole 5 jsou rozebrány použité nástroje. V kapitole 6 je popsán návrh a implementace systému. Tato část znázorňuje obecné schéma systému, jsou zde popsány všechny důležité části a nakonec je zde vysvětleno i ovládání nástroje. Předposlední kapitola 7 se věnuje testování a experimentování. A nakonec v kapitole 8 je popsán závěr a zhodnocení celkové práce.

Kapitola 2

Metody vyhledávání klíčových slov

2.1 Četnost slov

Jedná se o nejjednodušší způsob vyhledávání klíčových slov v textu. Tento způsob určí klíčová slova na základě jejich četnosti výskytu v textu. Metoda funguje dobře pro vyhledávání víceslovných výrazů, kvůli často se opakujícím ustáleným slovním spojením. Naopak k selhání dochází při hledání jednoslovných výrazů z důvodu, že v českém jazyce se nejčastěji vyskytují v textu předložky, spojky a zájmena. Při vyhledávání jednoslovných výrazů je proto dobré použít morfologický analyzátor, který odfiltruje nežádoucí slovní druhy a zůstanou nám pouze podstatná jména a přídavná jména, ze kterých se klíčová slova převážně skládají. Tato metoda není vhodná pro vyhledávání v krátkých textech, ve kterých se nevyskytuje dostatečné množství slov, výsledek by mohl být zkreslený[15].

2.2 Weirdness

Metoda Weirdness[1] je založena na zvláštnostech v textu. Většina "zvláštních" výrazů má tendenci se uskupovovat blíže k sobě než ostatní. Pro to, abychom mohli využít tuto metodu, musíme mít speciální a obecný korpus. Míra zvláštnosti se určí jako rozdíl mezi těmito korpusy. Vzorec pro metodu je:

$$Weirdness = \frac{\frac{Ws}{Ts}}{\frac{Wg}{Tg}} \tag{2.1}$$

kde:

- Ws - frekvence slov ve speciálním korpusu
- Ts - celkový počet slov ve speciálním korpusu
- Wg - frekvence slov v obecném korpusu
- Tg - celkový počet slov v obecném korpusu

2.3 Term Frequency

Term Frequency (dále jen TF) je metoda, která vyjadřuje množství výskytu hledaného slova v korpusu nebo kolekci souborů. Vzhledem k tomu, že každý soubor je jinak dlouhý, a tedy v delším souboru je větší šance výskytu daného slova než v kratším souboru, tak se počet nalezených slov dělí počtem slov z celého dokumentu. TF složku vypočítáme podle vzorce:

$$TF(i) = \frac{f(i)}{\sum_k f(k)} \tag{2.2}$$

TF se nejčastěji využívá pro ohodnocování kandidátů v lingvistickém předzpracování textu[7].

2.4 Term Frequency - Inverse document frequency

IDF metoda na rozdíl od TF reflektuje důležitost slova v dokumentu. Důležitost slova se zvyšuje úměrně k počtu případů, kdy se objeví slovo v dokumentu, ale je kompenzováno četností slova v celkové kolekci souborů. Metoda je často používaná ve vyhledávačích jako ústřední nástroj v bodování relevance dokumentů[7].

TF-IDF se počítá ve dvou krocích, nejprve je důležité spočítat frekvenci výskytu slova v dokumentech podle vzorce TF a následně je důležité vypočítat důležitost výrazu v celé kolekci podle vzorce :

$$idf(i) = \frac{|D|}{|\{j : t_i \epsilon d_j\}|} \tag{2.3}$$

kde $|D|$ je počet dokumentů, ve kterých hledáme a $|\{j : t_i \epsilon d_j\}|$ je počet dokumentů, který obsahuje slovo i .

Výsledná hodnota $TF(i)IDF(i)$ je potom součinem:

$$TF(i) - IDF(i) = TF(i) \times IDF(i) \tag{2.4}$$

kde i je ohodnocovaný termín, TF je metoda Term Frequency a IDF je metoda Inverse document frequency.

2.5 Okapi BM25

BM25 je hodnotící metodou, kterou využívají vyhledávače. Metoda vyhodnocuje dokumenty na základě relevance nalezení vyhledávaného slova.
BM25 je metoda, která hodnotí sadu dokumentů a určuje jejich vhodnost k danému vyhledávanému dotazu. Funkce ignoruje jakýkoliv vztah těchto předávaných dotazů v dokumentu. Předávaný dotaz Q obsahuje klíčová slova q1,. . . .qn.
Hodnocení BM25 se následně vypočítá jako:

$$score(D,Q) = \sum_{i=1}^{n} IDF(q_i).\frac{f(q_i,D).(k_1+1)}{f(g_i,D)+k_1-(1-b+b.\frac{|D|}{avgdl})} \tag{2.5}$$

kde f(qi, D) - qi je frekvence výskytu předávaného dotazu v dokumentu D, $|D|$ je počet dokumentů v korpusu, avgdl je průměrná délka dokumentů v množině k1 a b jsou volné parametry obvykle se volí za k1 (1.2, 2.0) a b se rovná 0.75.

IDF(inverse document frequency) váha dotazu q1 se spočítá pomocí vzorce:

$$IDF(q_i) = \log \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} \tag{2.6}$$

kde N je celkový počet dokumentů v kolekci a n(qi) je počet dokumentů obsahující výraz qi. Existuje více interpretací a variací IDF algoritmu[11]. V původním BM25 je IDF odvozený z BIM(Binary Independence Model).

Modifikace BM25:

BM25F Dokument v této modifikaci je rozdělen na několik částí (jako jsou titulky, hlavní část, . . .) s rozdílnou důležitostí, normalizací délky a relevantní saturaci dotazu.

BM25L Tato modifikace je rozšíření BM25. Modifikace byla vytvořena k rozšíření původního algoritmu, který trpěl nedostatkem znevýhodňující delší dokumenty. Nedostatek způsoboval, že docházelo k nespravedlivému ohodnocování hledaných dotazů v delších dokumentech vzhledem ke kratším. Metoda má jeden volný parametr a to δ , jehož běžná hodnota je 1.0 pokud dojde k absenci trénovaných dat. Popisuje se vzorcem:

$$\sum_{q \epsilon Q \cup D} = \frac{(k_3+1)c(q,Q)}{k_3+c(q,Q)}.f(q,D).\log\frac{N+1}{df(q)+0.5} \tag{2.7}$$

kde $c(q,Q)$ je počet slov q v dotazu Q, N je celkový počet dokumentů, $df(q)$ je počet dokumentů obsahující výraz q a k_3 je parametr.

Následně se použil upravený IDF vzorec aby nedocházelo k negativním IDF hodnotám. Klíčovým komponentem BM25 je sub-lineární Term Frequency (TF) normalizační formula $f(q, D)$, která se vypočítá jako:

$$f(q, D) = \frac{(k_1 + 1)c(q, D)}{k_1(1 - b + b\frac{|D|}{avdl}) + c(q, D)} = \frac{(k_1 + 1)c'(q, D)}{k_1 + c'(q, D)} \tag{2.8}$$

kde $|D|$ reprezentuje délku dokumentu, $avdl$ značí průměrnou délku dokumentu, $c(q, D)$ je četnost slov q v dokumentu D, b a k_1 jsou parametry. $c'(q, D)$ je četnost slova q normalizována délkou dokumentu:

$$c'(q, D) = \frac{c(q, D)}{1 - b + b\frac{|D|}{avdl}} \tag{2.9}$$

Pokud je dokument velmi dlouhý, můžeme vidět, že $c'(q, D)$ může být velice malý a může se blížit hodnotě 0. Následkem toho se i $f(q, D)$ bude blížit hodnotě 0, stejně tak, pokud se hledaný výraz q nebude vyskytovat v dokumentu D. Výsledek tedy může vypadat stejně, jako by se výraz q v dokumentu vůbec nevyskytl. Aby nedocházelo k penalizaci velmi dlouhých dokumentů zavádí se "mezera" mezi $c'(q, D) = 0$ a $c'(q, D) > 0$ a nakonec se upraví normalizační formula:

$$f'(q, D) = \begin{cases} \frac{(k_1+1).[c'(q,D)+\delta]}{k_1+[c'(q,D)+\delta])+c(q,D)}, & \text{if } c'(q, D) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.10}$$

oproti dřívější verzi vzorce je zde parametr δ zajišťující posun dále od nuly[21] [22].

2.6 C-value

C-value[3] je doménově nezávislá metoda speciálně určená pro víceslovná slova. Vstupem je korpus a výstupem metody je potom seznam kandidátů. C-value přístup kombinuje lingvistický a statistický informace. Metoda pracuje s četností slov, jejich délkou, ale také umí vyhledávat i vnořené termíny. Vnořené termíny, které se objeví jako podřetězec a součást dalších termín jsou penalizovány. Vypočítá se takto:

$$C - value(a) = \begin{cases} \log_2 |a|.f(a), & \text{pokud termín není vnořený} \\ \log_2 |a|.(f(a) - \frac{1}{P|Ta|} \sum b \epsilon Taf(b)), & \text{jinak} \end{cases} \tag{2.11}$$

kde a je ohodnocovaný termín, $f()$ je četnost v dokumentu, Ta je množina termínů, které obsahují a, $P|Ta|$ je potom počet těchto termínů.

Kapitola 3

Určování relevantních slovních spojení

3.1 Pearsonův X^2 test

Test dobré shody(taky Pearsonův chí-kvadrát test) je statistická metoda, která je založena na posouzení rozdílu mezi skutečnou četností výskytu ve výběrovém souboru a očekávanou četností výskytu. Test rozhoduje, zda je rozdíl způsoben pouze náhodně a výběrový soubor pochází z populace s normálním rozdělením nebo je rozdíl natolik velký, že výběrový soubor nepochází z populace odpovídající Gaussovu normálnímu rozdělení, ale z nějakého jiného neznámého rozdělení[19].

Postup při testu dobré shody:

1. Obor všech možných hodnot náhodné veličiny se rozdělí na k nepřekrývajících se částí.
2. Pro každou část se stanoví pravděpodobnost p_i, že náhodná veličina nabyde hodnoty z ité části.
3. Provede se N pokusů a zjistí se, kolikrát z těchto pokusů nabyla náhodná veličina hodnoty z 1., 2., ... k-té části. Tyto četnosti se označí X_1, X_2, ..., X_k.
4. Porovnají se očekávané četnosti v jednotlivých částech (Np_i) se skutečnými četnostmi (X_i) pomocí vzorce:

$$\chi^2 = \sum_{i=1}^{k} \frac{(X_i - Np_i)^2}{Np_i} \tag{3.1}$$

Pearsonovo rozdělení o n stupních volnosti, kde X_i představuje pozorované četnosti v jednotlivých třídách výběrového souboru a Np_i teoretické četnosti odvozené výpočtem pomocí tabulek distribučních funkcí normovaného normálního rozdělení. Počet stupňů volnosti $n = m - k - 1$, kde m je počet tříd výběrového souboru a k je počet počet parametrů normálního rozdělení, které neznáme, a musíme je odhadnout z výběrového souboru[17].

Kapitola 4

Morfologický analyzátor

V této kapitole se podíváme na morfologické analyzátory, což jsou základní nástroje pro značkování korpusů, umožňují určit základní tvar slova(lemma) a také gramatickou značku, ve které jsou zakódované informace o slovním druhu a morfologických kategorií.

4.1 MorphoDiTa: Morphological Dictionary and Tagger

MorphoDiTa[16] je nástroj pro morfologickou analýzu textů přirozeného jazyka. Nástroj obsahuje morfologickou analýzu, morfologické generace, značkování, tokenizaci a je distribuován jako samostatný nástroj nebo knihovna společně s trénovanými lingvistickými modely. Pro český jazyk MorphoDiTa dosahuje rychlosti 10-200K slov za sekundu.

Úkolem morfologické analýzy je přiřadit pro každý token(slovo) základní tvar (lemma) a POS značku (part-of-speech). Jako každý jiný nástroj pro lingvistiku, MorphoDiTa potřebuje trénované lingvistické modely. Toto je dosaženo ve dvou krocích:

- Nejdříve vrátí všechny možné páry lemmat a POS značek pro každé slovo, následně je zvolena optimální kombinace lemmat a POS značek pro slova ve větě s použitím algoritmu popsaném v [14].
- V druhém kroku se nástroj snaží odstranit u slov jejich víceznačnost. Více je tento krok rozebrán v knize [4].

České modely

Pro českou morfologii se využívá nástroj Morflex CZ 160310 český morfologický slovník a český značkovač, které jsou trénovaný na PDT 3.0 .
MorfFlex CZ je český morfologický slovník vytvořený Janem Hajičem jako nástroj pro kontrolu hláskovaní slov a slovník pro lemmatizaci. V současné době obsahuje morfologické informace pro každý pokrytý slovní tvar, stejně tak informace pro derivační, sémantické a jmenné entity. Morflex CZ slovník, nástroj pro odhadování prefixů a nástroj pro statistické odhadování jsou implementovány zvlášť a mohou být volitelně použity při provádění morfologické analýzy.

V českém jazyku, MorphoDiTa využívá český morfologický systém od Jana Hajiče[5].

Český model obsahuje:

- Morfologický slovník(czech-morfflex-160310.dict), využívající systém od Jana Hajiče s PDT značkamy vytvořené nástrojem MorfFlex
- Morfologický značkovač(czech-morfflex-pdt-160310.tagger) trénovaný na PDT 3.0 využívající sadu funkcí neopren. Obsahuje morfologický slovník a dosahuje přesnosti při určování značky 95.57% a přesnost určení lemmat 97.75% a 94.93% celkové přesnosti na PDT 3.0 datech, které jsou přemapovány. Rychlost modelu je přibližně 10k slov ze sekundu.

Oproti knihovně Featurama(knihovny implementující nejrůznější algoritmy pro sekvenční značení) jsou modely 5x rychlejší a 10x menší.

Anglické modely

Anglické modely jsou tvořeny pomocí následujících dat:

- SCOWL (Spell Checker Oriented Word Lists): Tento seznam slov je používán v morfologické generaci k vytvoření všech možných slovních forem vkládaného slova.
- Wall Street Journal, část Penn Treebank 3: Morfologicky anotované texty, které se běžně využívají k trénovaní anglických POS značkovačů.

Morfologická generece je inverzní morfologická analýza, přesněji je to proces konvertující vnitřní reprezentaci slova na jeho vnější podobu. Na základě stavby věty dokáže tato analýza opravit chyby, které mohou být v této větě obsaženy. Například pokud analýza zjistí, že věta obsahuje podstatné jméno jednotného čísla ve větě za nějakou číslovkou, tak umí toto slovo převést do množného čísla a tak se pokusit opravit chybu.

Přestože se v anglické morfologii využívá standard značek z The Penn Treebank Project, tak je struktura lemmat unikátní. Nástroj rozezná negativní prefixy a odstraní je ze základního tvaru slova. Podle pravidel MorphoDiTa je "surové" lemma takové lemma, které neobsahuje negativní prefix. Negativní prefix je často ukládán, aby se následně mohla provést morfologická generace formy slova se stejným negativním prefixem.

Anglický model obsahuje:

- Morfologický slovník(english-morphium-140407.dict), SCOWL seznam slov je automaticky analyzován a lemmatizován a využit jako slovník
- Morfologický značkovač(english-morphium-wsj-140407.tagger), který je trénován na Wall Street Journal (Sekce 0-18). Poslední verze značkovače dosahuje přesnosti značek 97.27% na datech z Wall Street Journal (Sekce 22-24). Rychlost modelu je přibližně 60k slov za sekundu.

4.2 Ajka

Ajka[12] je morfologický analyzátor českého jazyka, který vznikl jako diplomová práce pana Radka Sedláčka na Masarykově Univerzitě. Využívá slovníkový přístup, to znamená, že veškerá data potřebná ke správné funkci morfologického analyzátoru jsou uložena ve strojovém slovníku češtiny. Strojový slovník češtiny byl navrhnut tak, aby byl uživatelsky co možná nejjednodušší a přitom data uložená uvnitř byla možna použít pro lingvistické experimenty. Základním požadavkem bylo, aby data ve slovníku šly jednoduše přemístit. Stavební jednotkou je heslo, která jde sdružovat do sekcí. Heslo má tři části.

- Lexikální část - obsahuje základní tvar(lemma) nebo další tvary, pokud u nich dochází ke změně podoby kmene vzhledem k základnímu tvaru, může se skládát z jednoho nebo více slovních tvarů
- Gramatická část - která obsahuje informace o vzoru, u sloves dále informaci o vidu, reflexivitě a možnosti tvoření negativní formy, u adjektiv o možnosti připojení za číselný prefix a tvoření negativní formy
- Prefixová část - obsahuje prefixy, pokud slovní tvar muže mít prefixy a přidáním prefixu se nezmění gramatická informace ve druhé části hesla

Hlavními zdroji pro analýzu jsou dva binární soubory, první obsahuje definice koncových vzorů a množin, druhý je binární podobou vlastního strojového slovníku a obsahuje základy českých slov. Nejdříve program načte data uložená ve zmíněných binárních souborech, poté se následně program zapíná opakovaně a textové informace se zpracovávají v cyklech. Program využívá algoritmů pro segmentaci slov a identifikaci segmentových prvků, které jsou inspirované z disertační práce viz.[9].

Kapitola 5

Použité nástroje

5.1 PDFMiner a Slate

PDFMiner je nástroj, který slouží pro extrakci dat z PDF dokumentů. Oproti ostatním nástrojům se tento nástroj soustředí pouze na získání a analyzování textových dat. PDFMiner umožňuje získat jak informaci o přesném umístění textu, tak jiné informace o textu, jako například typ písma. Nástroj obsahuje PDF konvertor, který dokáže převést PDF text na texty různých formátů, například HTML. Mezi nevýhody tohoto programu patří pomalý převod textu, který dosahuje až dvacetkrát pomalejší převod oproti konkurenčním programům psaných v C/C++ jako například Xpdf[13].

Slate je knihovna do jazyka Python, která zjednodušuje proces extrakce dat z PDF dokumentů. Slate využívá nástroje PDFMiner, ale je psán s větším důrazem na jednoduchost. Oproti nástroji PDFMiner je nástroj jednodušší na ovládání, nevrací objekt, ale uživatel pracuje pouze s jedinou třídou "PDF", která pracuje s dokumentem jako objektem a vrací veškerý text z dokumentu jako řetězec znaků. Jednotlivé stránky se uloží do seznamu se kterým se dále pracuje[8].

Tyto nástroje jsem se rozhodl použít, protože PDFMiner a Slate jsou oproti Xpdf naprogramovaný v Pythonu, který jsem použil při tvorbě systému. Nemusí se tedy volat a spouštět další binární soubor, pouze se zavolají určité metody. Práce s balíkem PDFMiner respektive Slate je oproti Xpdf opravdu jednoduchá. Systém volá pouze jednu metodu. Výsledky převodu byly u obou testovaných nástrojů podobné, je tedy nutné se rozhodnout podle jiných kritérií. Rychlost nebyla pro mě tak důležitá jako jednoduchá práce s nástrojem.

5.2 Python

Python je vysokoúrovňový programovací jazyk, který v roce 1991 navrhl Guido van Rossum. Python je vyvíjen jako open source projekt, což znamená, že každý může přispět k jeho vývoji. Jazyk je multiplatformní, nabízí instalační balíky pro většinu platforem (Mac OS, Unix, Windows). Nabízí dynamickou kontrolu datových typů a podporuje různá programovací paradigmata, včetně objektově orientovaného, imperativního, procedurálního nebo funkcionálního[18].

Python je velice univerzální, hodí se jak k rychlému prototypování aplikací, tak k vývoji webu nebo jako skriptovací nástroj administrátorů. Jazyk je dobře čitelný a přehledný, jeho veliká přednost je v rychlosti učení a je tak dobrý pro výuku programování. Bývá občas zařazován jako skriptovací jazyk, ale jeho možnosti jsou mnohem větší. Python je hybridní jazyk, to znamená, že umožňuje psát programy nejen pomocí objektově orientovaného paradigmata, ale i procedurálního a v omezené míře i funkcionálního. S vysokou produktivností souvisí dostupnost a snadná použitelnost široké škály knihovních modulů, umožňujících snadné řešení úloh z řady oblastí. Python se snadno vkládá do jiných aplikací (embedding), kde pak slouží jako jejich skriptovací jazyk. Tím lze aplikacím psaným v kompilovaných programovacích jazycích dodávat chybějící pružnost. Jiné aplikace nebo aplikační knihovny mohou naopak implementovat rozhraní, které umožní jejich použití v roli pythonovského modulu[2].

Práce s textem je hlavní náplní tohoto systému, vybral jsem tedy tento jazyk, protože oproti jiným jazykům nabízí mnohem víc nástrojů, které umožňují různou práci s textem. Python bere jednotlivé znakové řetězce jako neměnné sekvence, může pracovat s jednotlivými znaky a odkazovat na ně podobně jako na položky v seznamech nebo si extrahovat výběr znaků pomocí "slice"funkce. Jazyk podporuje použití regulárních výrazů, které jsem také při tvorbě použil. Kód jazyka je čitelný, přehledný a dají se díky němu napsat kratší programy.

Kapitola 6

Návrh a implementace systému

V této kapitole popisuji kompletní vlastní návrh a implementaci systému. Jsou zde popsány kroky celého systému od zpracování vstupních dat, přes určení klíčových slov až po vytvoření finálního souboru. Nakonec je popsána struktura finálního souboru a používání jednotlivých modulů.

6.1 Požadavky na systém

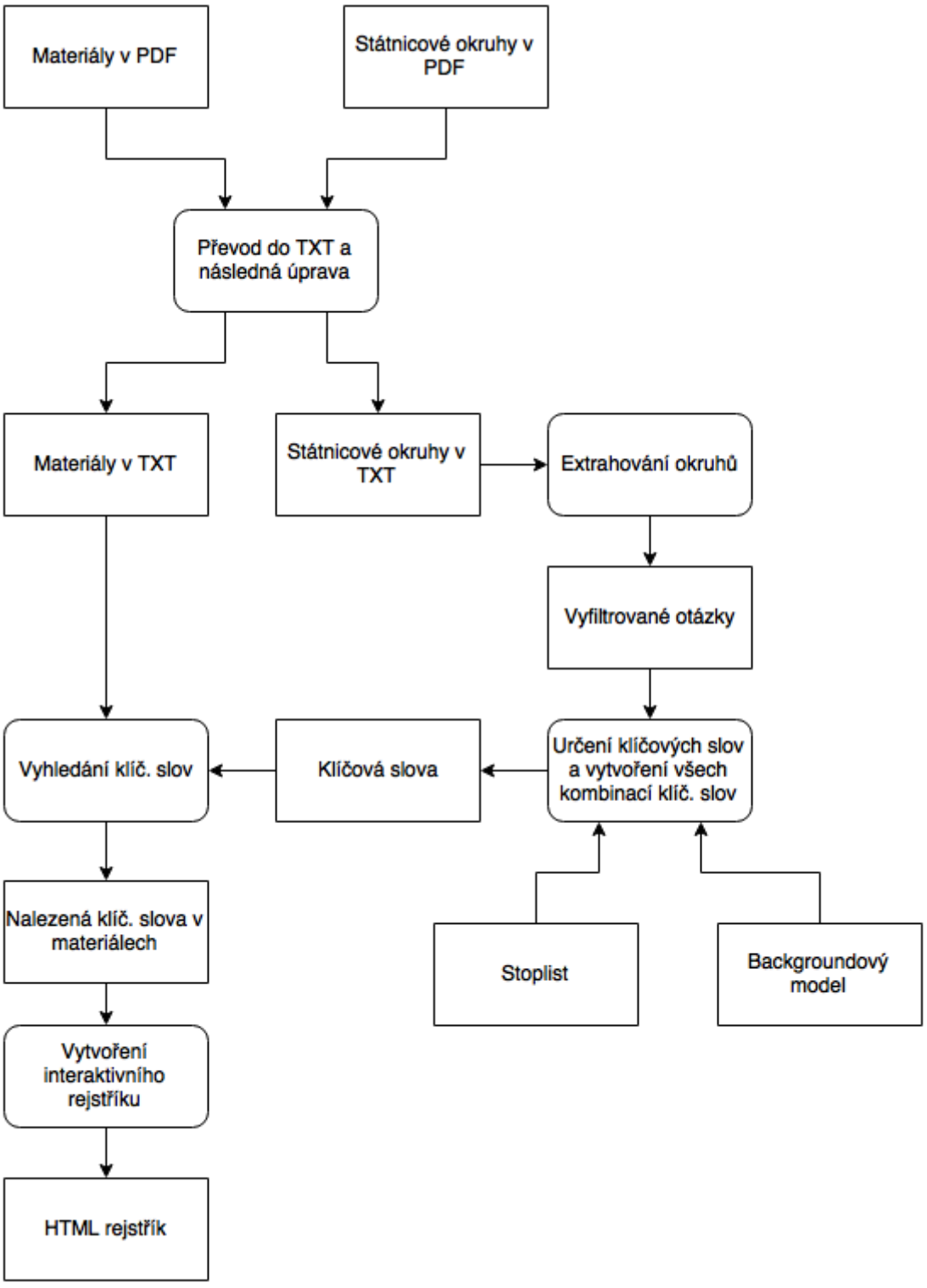
Cílem systému je propojit státnicové okruhy, studijní materiály, jiné materiály a vytvořit rejstřík klíčových slov. Výsledný rejstřík by měl sloužit pro snadnější vyhledání informací napříč velkého množství vstupního textu a tím ušetřit velké množství času, které by bylo potřeba vynaložit pro manuální vyhledávání těchto klíčových slov. Pro ještě větší ušetření času je důležité, aby rejstřík byl interaktivní a obsahoval odkazy na nalezená klíčová slova nebo přímo do konkrétních materiálů, které by mohly čtenáře zajímat.

6.2 Struktura systému

Celý systém je rozdělen do několika na sebe navazujících kroků:

1. Převod materiálů do textové podoby
2. Úprava a předzpracování textu
3. Určení klíčových slov
4. Vyhledání klíčových slov
5. Vytvoření interaktivního rejstříku

Systém je zobrazen na obrázku 6.1. V následujících kapitolách budou podrobněji popsané jednotlivé kroky.



Obrázek 6.1: Struktura systému

6.3 Převod materiálů z PDF do textové podoby

Většina školních nebo studijních materiálů se vyskytuje v PDF formě nebo v menší míře ve formátu DOC, používaný u oblíbeného softwarového balíku Microsoft Office. Rozhodl jsem se proto systém přizpůsobit převodu z PDF dokumentů, které převládaly.

Protože je PDF spíše grafický než textový formát, jde s tímto formátem velice špatně pracovat, proto je nutné nejdříve materiály převést do přijatelnější čistě textové podoby. Pro převedení PDF formátu jsem si zvolil nástroje PDFMiner a Slate, které jsou popsány v kapitole 5.1.

6.4 Úprava textu

Studijní materiály

Dalším krokem je úprava převedených textů. Převedení PDF formátů se u větších dokumentů nikdy neobejde bez chyb, proto je nutné výstup upravit do přijatelnější podoby a sjednotit ho. Sjednocení textů pomůže značně při vyhledávání, ale poté nemusí docházet ke správnému vyhledání. Soubor se převede do textové podoby a následně se prochází jednotlivé stránky převedeného dokumentu a upravuje se výstup:

- Prvním krokem dochází k odstranění bílých znaků a jejich nahrazení za mezery.
- Text se převede na malé znaky.
- Upraví se špatně převedené znaky s diakritikou.
- Znaky, které nepatří do abecedy, se odstraní. Například různé typy pomlček.
- Odstraní samostatně se vyskytující čísla.

Po této úpravě se už může v daném textu vyhledávat s větší přesností. Upravený text se uloží mezi tagy do souboru, který obsahuje veškerý přeložený a upravený obsah všech materiálů v dané složce. Každá složka zpracovávaného předmětu obsahuje právě jeden takový soubor. Vytvořený soubor redukuje prohledávání velkého množství souborů pouze na jeden soubor. Tyto soubory obsahují tagy, které znázorňují začátek, konec každé stránky a každého souboru. Díky tomuto značení se jednoduše dohledá, která strana, kterého dokumentu obsahuje dané klíčové slovo.

Soubor má formát:

```
FILE_START     file_name=název_souboru.pdf

    PAGE_START
            ...text strany1...
      PAGE_END
...
    PAGE_START
            ...text strany N...
      PAGE_END

FILE_END
```

Tématické okruhy a státnicové otázky

Státnicové otázky se vyskytují většinou ve formě 1-2 stránkového dokumentu, který obsahuje pouze okruhy, případně obsahuje i doplňující informace. Pro tento typ dokumentů se využívá regulární výraz, který extrahuje okruhy vyskytujícící se ve formě:

číslo. Téma okruhu.

Tato forma se dodržuje na Fakultě informačních technologií, ale i na jiných fakultách. Následně se text upraví podobně jako u studijních materiálů s menšími změnami. Text se převede na malá písmena a upraví se problémy s diakritickými znaky, ale neodstraňují se neznámé znaky, protože většina státnicových okruhů obsahuje pouze čistý text, který může obsahovat znaky jako závorka, pomlčka, tečka nebo dvojtečka. Takto upravené otázky se uloží do textového souboru zvlášť, mimo převedené materiály.

6.5 Výběr a tvorba klíčových slov

Určování klíčových slov je složitá disciplína, u které se využívá mnoho algoritmů a způsobů spojování slov, některé jsou zmíněny v kapitole 2. Protože máme k dispozici státnicové otázky, které z větší části obsahují pouze klíčové slova, tak je tato část značně zjednodušená, přesto je nutné daná klíčová slova upravit pomocí morfologického analyzátoru a vytvořit další možné tvary klíčového slova nebo spojení klíčových slov, aby při vyhledávání systému neuniklo pokud možno žádné slovo. Jádro systému pro tvorbu klíčových slov je rozděleno na tyto části:

1. Parsování otázek
2. Značkování
3. Zpracování n-gramů a vytvoření lemmat pro všechny slova.

4. Použití stoplistu, korpusu pro filtraci unigramů a odsekávače slov pro zkrácení jmen

6.5.1 Parsování otázek

Parser zpracovává jednotlivé otázky. Otázka se rozdělí na dvě části, hlavní část a pak její podčást, pokud otázka obsahuje závorku s doplňujícími informacemi. Tato seskupení slov se dále kontrolují na přítomnost spojky "a", která slouží jako spojka spojující obě strany např. "bipolární a unipolární tranzistor", ale také jako rozdělovač, který rozděluje obě strany do dvou různých klíčových slov "HTML a Javascript". Pokud parser narazí na spojku, rozhodne se, zda daný výraz rozdělí nebo roznásobí. Například u spojení "protokoly tcp a udp" vytvoří dvě víceslovná klíčová slova a to "protokol tcp" a "protokol udp".

6.5.2 Značkování

Systém následně zavolá morfologický analyzátor MorphoDiTa, popsaný v kapitole 4.1. U analyzátoru se využívá pouze první vrstva - Morfologická rovina, která nám v tomto systému postačuje. Analyzátor nám určí základní tvar každého slova a pomocí tagu nám řekne velké množství důležitých informací o slově. Tato značka má podobu patnácti znaků. Pro naše účely nám postačuje prvních 5 znaků:

1. první znak - určuje slovní druh(nejdůležitější jsou: N - podstatné jméno, A - přídavné jméno, V - sloveso)
2. druhý znak - určuje podkategorii slovního druhu(archaické slovo, infinitiv, kolikátý,...)
3. třetí znak - určuje rod(střední - N, ženský - F, mužský životný - M, mužský neživotný - I)
4. čtvrtý znak - určuje zda jde o jednotné nebo množné číslo
5. pátý znak - určuje vzor rodu

6.5.3 Zpracování n-gramů

Největší zastoupení klíčových slov v textu jsou jednoslovné výrazy, neboli unigramy. Pro vytvoření kombinací slov unigramů stačí určit základní tvar, ze kterého se určí kořen a poté dle určeného slovního druhu, rodu a jeho vzoru přidat všechny možné přípony, například pro slovo "vrstva" se vytvoří další kombinace jako "vrstvy", "vrstvami", "vrstvě" a další. U víceslovných výrazů je tato metoda komplikovanější. Musí se určit slovní lemma u každého slova, poté ke kořenům slov připojit přípony a nakonec zkombinovat všechny slova. Z testování jsem ale došel k závěru, že efektivnější řešení bude převést celý text včetně státnicových okruhů na základní tvar a tento tvar použít i pro vyhledávání. Tímto se eliminuje nutnost skloňovat slova a výrazně se sníží počet redundantních tvarů slov. Pokryjí se všechny tvary slova, protože v tomto případě je tvar jen jeden a to základní.

Čím delší slovní spojení, tím náročnější. Následně víceslovné výrazy systém rozbíjí na menší části. Pro takový trigram, nebo-li slovní spojení o třech slovech se výraz rozbije na 3 slova, zopakují se postupy zmíněné výše a následně se budou slova kombinovat podle pořadí. Ve výsledku se trigram rozbije na kombinace bigramů a ty se nakonec rozbijí na unigramy, pokud se dané výrazy nevyskytují ve stoplistu nebo korpusu, který takový výrazy filtrují.

Filtraci, která je volitelná, jsem aplikoval až na unigramy, protože mají v textu největší zastoupení. Takové slovo "vrstva" je poměrně obecné, nic neříkající, ale se slovem "aplikační" už vyhledávání směruje k předmětům týkající se sítí v informačních technologií. Smazáním tohoto slova bychom mohli přijít o důležité slovní spojení, proto se ve výsledku filtruje pouze unigram, který vznikl rozbitím víceslovných výrazů, ale výceslovný výrazy se to pokusí najít v celku.

6.5.4 Stoplist

Dalším způsobem redukce slov nebo slovních spojení je použití stoplistu. Stoplist je seznam slov, které díky své vysoké frekvenci ztrácejí ve vyhledávání význam. Tyto seznamy jsou tvořeny převážně ze spojek a předložek, které se vyskytují nejčastěji. Můžou je ale tvořit i podstatná nebo přídavná jména, která se systém snaží ve studijních textech vyhledat. Pro co nejpřesnější výsledky v analýze textu je potřebat odstranit obecná slova, která se v textech vyskytují.

Pro vytvoření stoplistu systém při konvertování textu z pdf do textové podoby může tvořit současně i frekvenční seznam, obsahující frekvenci všech slov ve zkoumaných dokumentech. Tento seznam je možné použít pro tvorbu vlastního stoplistu, ale ideální je si stoplist vytvořit z co největšího vzorku dat.

Vytvořil jsem si stoplist tak, že jsem si určil hranici určující, která slova se uloží do stoplistu. Stoplist tvoří přibližně prvních 400-500 slov z celkových 87 000 unikátních slov ve zkoumaných dokumentech. Nutné je ale zdůraznit, že určení hranice pro výběr jednotlivých slov je čistě subjektivní. Existuje globálně mnoho slov, které se vyskytují jen zřídka, ale přesto nemají pro analýzu žádný význam. Hranice určující, která slova do stoplistu zařadit není a nikdy nebude přesně určená, proto dochází k různě zkresleným výsledkům. Velikou roli hraje, z jakých korpusů je daný stoplist vytvářen, tedy zda je korpus reprezentativní vzhledem k analyzovanému tématu. Nemělo by moc smysl vytvářet stoplist z korpusů vytvořeného z lékařských údajů a tento stoplist následně použít pro technické obory. Systém podporuje použití vlastního stoplistu a použití stoplistu není povinné.

6.5.5 Backgroundový model

Backgroundový model je rozsáhlá kolekce textových dat v elektronické podobě, která převážně obsahuje seznam slov a jejich četnost. Mohou se vytvářet i korpusy pro speciální účely a mohou obsahovat slova v několika jazycích, mluvená slova nebo chybová slova. Korpus je vytvořen tak, aby v něm bylo možné hledat různé jazykové jevy, hlavně slova a slovní spojení. O kvalitě korpusu vypovídá množství slov, ze kterého je korpus složený. Zpravidla platí, čím větší množství slov, tím kvalitnější korpus. Kvalitní korpusy obsahují milióny slov. Počet slov ale není jediným kritériem kvality korpusu, abychom mohly výsledky získané z korpusu vztáhnout na celý text

je nutné zajistit, aby byl korpus reprezentativní. Backgroundové modely se dělí na dvě skupiny:

- doménový (specializovaný)
- korpusový (obecný)

Obecný model je tvořen z obecných textů a používá se na práci s obecnými nespecializovanými texty, pro specializované a vědecké texty je dobré použít doménové korpusy reprezentující daný obor. Pokud například pro práci s dokumenty o autech využijeme korpus specializovaný na tuto problematiku, tak se nám nestane, aby se slovo "auto" vybralo jako jedno z klíčových slov, protože toto slovo je pro daný text moc obecné[10].
V rámci systému je možné použít backgroundový korpusy. Korpus se skládá z velkého množství vyextrahovaných slov. Systém porovnává jednotlivá klíčová slova s korpusem a snaží se odfiltrovat slova na základě jeho pokrytí v textu. Odfiltruje slova, které se vyskytují ve zpracovávaných souborech přibližně stejně často jako v korpusu, tedy ve větším souboru dat. Touto metodou se smažou slova, které jsou pro zkoumaný soubor moc obecná a mohou zkreslovat statistiky.
Vytvořil jsem si pro testovací data doménový korpus s tématikou informačních technologií čítající přes 250 tisíc unikátních slov, tvořený z 800 dokumentů, které jsem shromáždil a vypočetl jejich zastoupení v textu. V systému je možné použít vlastní korpus, je ale dobré si ho předzpracovat, aby byl systém co nejpřesnější.

Tvorba korpusu spočívá ve shromáždění co největšího počtu dokumentů se zkoumanou tematikou a následnou úpravou těchto textů. Důležité je vyčistit text od zbytečných znaků a rušivých elementů, které by mohly zkreslovat výsledek. Po vypočítání četnosti všech slov je dobré pro větší efektivitu tento seznam pročistit od slovních druhů, které nás nezajímají, jako například spojky. Tím se vyhledávání v tomto seznamu značně urychlí a tím se urychlí i celkový běh systému.

6.5.6 Odsekávač slov

Systém spolupracuje především s morfologickým analyzátorem. Na tento nástroj není ale vždy spolehnutí, je proto potřeba zohlednit případné chyby v podobě špatně určeného slovního druhu, případně jeho neurčení nebo špatně převedeného slova na základní tvar. Tento problém se vyskytuje především u jmen a z nich odvozených přídavných jmen jako "boolovský" nebo "Karnaughova" a další. Proto jsem do systému implementoval jednoduchý odsekáváč slov, který dokáže delší výrazy zkrátit přibližně o 50% , čímž docílíme přibližného určení základního tvaru slova a rozšíříme pokrytí daného shluku klíčových slov. Například přídavné jméno "boolovský" se zkrátí na "bool" a takové slovo se v textech s tématem informačních technologií vyskytuje častěji. Systém pracuje s lemmatizovaným textem, přesto se najdou slova, která morfologický analyzátor špatně určí a nepřevede na požadovaný tvar. Odsekávač slov může pomoci, ale také může zbytečně krátit slova, u kterých to nepotřebujeme a proto jsem se rozhodl ho do systému přidat jako volitelnou součást.

Podobně jako u stoplistu i u této metody úpravy klíčových výrazů neexistuje správné pravidlo nebo hranice, které určí o kolik je nutné zkrátit dané slovo. Experimentoval jsem a došel jsem k závěru, že ideální je delší slovo zkrátit na přibližně

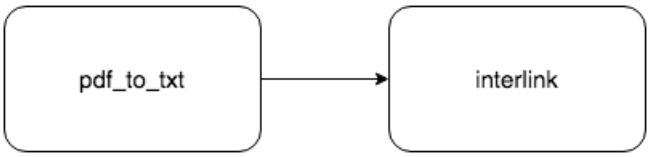
40-50% délky původního slova a abychom pokryli i další slova, tak jsem určil další hranici a to přibližně 60-70% délky původního slova.

6.5.7 Určení témat a vyhledání klíčových slov

Předposlední krok systému je vyhledání jednotlivých klíčových slov nebo slovních spojení. Po dlouhém pracném převádění souborů a následně skloňování slov nebo tvoření víceslovných klíčů je vyhledání těchto slov už ta jednodušší část. Systém prochází všechny strany dokumentů a snaží se najít jednotlivá slova. Původně systém vyhledával v textu víceslovná klíčová slova v celém textu a jednotlivé unigramy se systém pokoušel najít i jako podřetězec v jednotlivých slovech, tímto ale docházelo ke zkresleným výsledkům. Po změně na lemmatizovaný text už dochází k přesnějšímu vyhledávání. U tohoto vyhledávání občas dojde ke zkresleným měření a to pouze když dojde k dříve zmiňovanému problému převádění pdf dokumentů na textové.
V tomto kroku se naleznou veškerá místa, kde se daná klíčová slova nachází, uloží si název dokumentu, stránku v dokumentu a celkový počet vyskytů v daném dokumentu. Dané metriky se využijí pro určení témat.
Výsledný soubor, který má podobu rejstříku, obsahuje mnoho řádků. Tyto řádky obsahují frekvence jednotlivých klíčových slov pro každý dokument, ve kterém se vyskytuje. Aby se redukoval značný počet těchto řádků a umožnil uživatelům výběr kvalitnějších nálezů tak jsem na základně výše změřených metrik určil nejdůležitější předměty, kde se dané slovo vyskytuje. Určil jsem pro každé slovo tři nejdůležitější témata podle frekvence výskytu klíčových slov, která by mohla obsahovat důležité informace spojené s tímto slovem. Po vlastním testování na materiálech z Fakulty informačních technologií jsem došel k závěru, že jednotlivá klíčová slova jsou obsažena vždy ve 2-3 hlavních předmětech, které zpravidla na sebe i navazují. Například předmět "Počítačové sítě a komunikace" a předmět "Síťové aplikace", který na předešlý předmět navazuje. Tyto předměty se doplňují, případně i trochu opakují látku obsaženou v předešlém dokumentu. Zvolit jenom jeden z nich by sice zpřehlednilo výsledný rejstřík, ale obsahoval by neúplné informace a čtenáři by mohlo uniknout spousty důležitých informací. Na druhou stranu je nutné zvolit rozumnou hranici maximálního počtu témat, které nejspíš s daným klíčovým slovem souvisí. Pokud bychom nezvolili nějakou hranici, mohlo by se stát, že systém zavalí uživatele spoustou redundantních informací, například takové klíčové slovo "email", který je rozebírán právě v předmětu "Síťové aplikace". Pokud by nebyla zvolena hranice, je pravděpodobné, že by systém našel odkazy ve všech dokumentech, protože takový email se zpravidla píše do všech slajdů nebo studijních opor jako kontaktní údaj na osobu, která daný materiál napsala.

6.6 Implementace systému a jeho použití

Finální systém je napsán v programovacím jazyku python a rozdělil jsem ho do dvou samostatných velkých modulů.



Obrázek 6.2: Hlavní části

První modul slouží, jak už název napovídá, k převodu pdf souborů do textové podoby. Před tímto krokem je důležitá příprava souborů, které máme v úmyslu analyzovat. Očekává se od uživatele vytvoření složky, která bude systémem zpracována a v této složce je nutné vytvořit podsložky pro každý předmět nebo téma, ve kterém má v úmyslu uživatel vyhledávat.

Modul se spouští s následujícímy parametry:

pdf_to_text.py -documents_path=PATH -topics_path=PATH -corpus -background=PATH

- documents_path - cesta složky, která obsahuje podsložky s materiály
- topics_path - cesta k souboru s otázkami, tématy, které se vyextrahují
- corpus - přepínač, pro vytvoření jednoduchého frekvenčního seznamu
- background - cesta k backgroundovým datům, skript se snaží vyhledat slova ze státnicových okruhů ve vytvořeném seznamu a v seznamu, který se zadává parametrem a odstraní ty, které se vyskytují v přibližném zastoupení

Všechny parametry jsou nepovinné a je zcela na uživately, zda bude chtít přeložit pouze soubor s otázkami, převedení materiálů do textové podoby, vytvoření frekvenčního seznamu nebo všechny tyto činnosti.
Po převedení souborů do textové podoby budou jednotlivé složky obsahovat po jednom textovém souboru s názvem dané složky, který bude obsahovat veškerý text ze všech pdf souborů v dané složce. Modul současně vytváří i frekvenční seznam ze všech materiálů, který se může použít pro různé statistiky. Druhý modul navazuje na první a využívá textové soubory pro vyhledávání.

Modul spouštíme s následujícímy parametry:

interlink.py -documents_path=PATH -topics_path=PATH -stoplist=PATH -c

- documents_path(povinný) - cesta složky, která obsahuje zpracované materiály
- topics_path(povinný) - cesta souboru s otázkami, nebo klíčovými slovy, které nás zajímají

- stoplist - cesta ke stoplist souboru, který chcete použít
- c - zapnutí odsekávače slov

Parametr documents_path odkazuje na složku se zpracovanými daty. Pokud se zadá cesta k souboru neobsahující tagy, které systém využívá pro vyhledávání, modul nebude schopen provést operaci vyhledávání nad těmito soubory a skončí.

6.7 Výsledný soubor a jeho struktura

Posledním krokem je vytvoření souboru, se kterým bude uživatel pracovat. Pro tvorbu souboru je využito HTML, což je značkovací jazyk používaný pro tvorbu internetových stránek.
Hlavní důraz byl kladen na to, aby práce se souborem byla pohodlná, rychlá a uživatel našel veškeré informace, které potřebuje. Proto má soubor strukturu podobnou rejstříku.
Rejstřík je důležitý nástroj, který se využívá k organizaci informací a jejich vyhledání v dokumentu. Jde o dokument, který obsahuje odkazy na umístění daného pojmu v konkrétním dokumentu. Typickou vlastností rejstříku je uvedení lokačních údajů u každé lexikální jednotky. Může se přitom jednat o umístění dané informace, nebo o umístění dokumentu pojednávajícím o daném tématu. Informace se třídí podle určitého pravidla. Dokument obsahuje tedy dvě složky, klíč a odkaz. Klíčem je lexikální jednotka, která může být klíčové slovo, nebo slovní spojení. Toto klíčové slovo by nemělo obsahovat běžné slova, ale pouze specifické slova, které se vyskytují v minimální míře a je proto těžší je najít. Odkaz je adresa, která ukazuje na konkrétní umístění informace v primárním dokumentu, ke které se klíč vztahuje[6][20].
Rejstříky se dělí podle různých kritérií:

Dle způsobu řazení:

- abecední
- chronologické
- systematické

Dle způsobu obsahu:

- autorské
- předmětové
- slovníkové
- názvové

Vytvoření obyčejného rejstříku by nemuselo práci urychlit a usnadnit natolik, kolik se očekává. Z tohoto důvodu jsem vytvořil interaktivní rejstřík, který obsahuje u jednotlivých klíčů odkazy na konkrétní stránky v materiálech. Tato vlastnost umožňuje pohodlné vyhledávání napříč velkým množství textu a šetří čas uživatele, který by jinak ztratil při neustálém ručním otevírání souborů a listováním na požadované stránky.
Soubor obsahuje v hlavičce dokumentu přehled státnicových okruhů. Jednotlivá slova v těchto okruzích představují odkazy, které přesměrují uživatele do části dokumentu věnující se tomuto klíčovému slovu. Jednotlivé části jsou v dokumentu seřazeny tak, jak je systém zpracovával, tedy ve frontě. První část se věnuje prvnímu klíčovému slovu a úplně poslední se věnuje poslednímu klíčovému slovu. Části mají podobu rejstříku, tedy jsou ve formě klíč a odkaz. Jednotlivé výrazy daného klíčového slova jsou řazeny dle délky nalezeného výrazu, na prvních místech jsou výrazy, které více vystihují klíčové slovo a následně jsou řazeny dle frekvence jejich výskytu v dokumentech. První jsou vypsány soubory s největším výskytem, které mohou být pro uživatele zajímavější než soubory s nižším výskytem. Řadí se tedy vždy vzhledem k danému klíčovému slovu a ne napříč celým dokumentem nebo tématickým okruhem. Jednotlivé nalezené stránky u každého klíčového slova fungují jako odkazy, které přesměrují uživatele přímo do dokumentu na konkrétní stranu.
Frekvence výskytu klíčového slova se mi během testování osvědčila jako důležitá statistika pro určování důležitosti dokumentu. Materiály popisující danou problematiku zpravidla obsahují větší frekvenci klíčových slov spojené s tímto tématem, zatímco materiály, které se na tuto látku odkazují obsahují zanedbatelné množství těchto slov.

Forma dokumentu vypadá následovně:

Klíčové slovo

předmět výraz soubor počet výskytů strany(odkazy na strany)

Pro demonstraci si ukážeme formu dokumentu pro klíčové slovo ”Jazyk uml”:

Jazyk UML

předmět: IPP výraz: jazyk uml soubor: opora_IPP-II.pdf počet výskytů: 5x strana: 50 52 53 55 67

Kapitola 7

Testování a výsledky

Cílem v této kapitole je zhodnotit kvalitu a efektivitu systému. Hlavní testy probíhaly na materiálech z VUT FIT. Nejdříve jsem otestoval převádění pdf souborů do textové podoby, protože tyto soubory jsou důležité pro vyhledávání informací. Následně jsem analyzoval státnicové otázky. Testový vzorek obsahuje přes 8000 slov obsažených ve státnicových okruzích z více než 20 fakult vysokých škol. Testy se zaměřují na analyzování průměrné délky slov, určení slovních druhů, na redundanci vytvořených klíčových slov, zastoupení jednotlivých slovních druhů a nakonec jsem tento systém testoval na dobrovolnících.

K testování jsem využil stoplist a doménový background, který jsem si vytvořil ze shromážděných dat. Doménový background, který jsem vytvořil obsahuje přes 250 tisíc unikátních slov a je složený z více než 800 dokumentů zaměřující se na tematiku informačních technologií.

7.1 Kvalita převodu PDF souborů

Abychom mohli vyhledávat informace je důležité mít podklady v dobré kvalitě. Čím čistější text, tím přesnější výsledky. Měřil jsem proto kvalitu převodu jednotlivých dokumentů. Pro otestování jsem použil přibližně 2000 studijních materiálů. Tyto materiály tvořily studijní opory, slajdy, elektronické knihy a studentské výpisky v pdf formátu.

Kvalita převodu	Počet souborů	Počet souborů v %
90-100%	1548	79,46%
80-90%	326	16,51%
50-80%	88	4,45%
méně než 50%	12	0,6%

Tabulka 7.1: Kvalita převodu

Jak můžeme vidět, většina materiálů se převedla do uspokojivé kvality. Téměř vždy docházelo k drobným problémům. U převodu tabulek, grafů, zdrojových kódů a obrázků docházelo k převedení textů, kde převedený text tvořil nejčastěji popisky. Podobně i texty obsažené v záhlaví nebo zápatí se převedly a následně se uložily náhodně do textu. Stávalo se, že uprostřed některé věty se uložil název kapitoly, popisek

obrázku nebo grafu. Vyhledávat se v takových textech dá a například vyhledávání unigramů není tímto efektem poznamenané, ale vyhledávání víceslovných kalokací už může na tento problém narážet a tím zkreslovat výsledky. Některé texty naopak byly nečitelné celé. Docházelo k tomu převážně jen u textů, které obsahovaly více grafických elementů, než samotného textu. Vzácně docházelo k problému s kódováním, které se projevilo na převedeném textu.

7.2 Slovní druhy

Další z testů sledoval zastoupení jednotlivých slovních druhů. Tyto testy byly prováděny na testovacích datech, které obsahovaly více než 8000 slov. Slova měla zastoupení jednotlivých slovních druhů:

Slovní druh	Počet slov	Počet slov v %
Podstatná jména	5187	61,26%
Přídavná jména	1827	21,57%
Slovesa	115	1,36%
Předložky	445	5,25%
Spojky	675	7,97%
Ostatní	217	2,56%

Tabulka 7.2: Zastoupení slovních druhů

Z tabulky můžeme vidět, že státnicové okruhy jsou nejvíce skládány z podstaných a přídavných jmen, které tvoří celkově více než 80% všech slovních druhů. Předložky a spojky se sice v menší míře vyskytují, přesto k určení klíčových slov jsou nepodstatné a mohou se zcela vypustit. Nejméně zastoupeny jsou slovesa a ostatní slovní druhy, které obsahují kromě ostatních slovních druhů i chybně určené nebo neznámé slovní druhy. K tomuto jevu docházelo téměř vždy u názvů a jmen.

7.3 Délka jednotlivých slov

Tento test byl vytvořen za účelem zjištění, jak dlouhá klíčová slova se v průměru v textu vyskytují. Z těchto údajů jsem poté vytvořil odsekávač slov, který zkracuje dlouhá slova na jejich kratší verze za účelem vytvoření kořene a následně k většímu pokrytí ve vyhledávání.

Nejčastěji se v textu vyskytují slova o délce 6-9 znaků. Slova o délce 1-3 znaky tvoří hlavně spojky a předložky. Rozhodl jsem se proto odfiltrovat spojky, protože slova o délce 2-3 znaků se občas v textu vyskytují v podobě zkratek. V prvním sloupci najdete délku slova o daném počtu písmen, druhý sloupec znázorňuje počet nalezených slov této délky a poslední sloupec značí procentuální zastoupení slov v testovaném textu.

Délka slova	Počet slov	Počet slov v %
4 znaky	420	4.95
5 znaků	432	5.1
6 znaků	1182	13.98
7 znaků	1092	12.89
8 znaků	1244	14.64
9 znaků	1140	13.46
10 znaků	460	5.43
11 znaků	404	4.77
12 znaků	176	2.07
13 znaků	125	1.47
14 znaků	68	0.08

Tabulka 7.3: Délka jednotlivých slov

7.4 Průměrná délka klíčových slov

Státnicové okruhy v dokumentech se skládaly převážně z víceslovných výrazů. Provedl jsem proto měření a zjistil jsem, že nejpočetnější zastoupení v textu mají bigramy, neboli dvouslovné výrazy. Testované materiály obsahovali téměř polovinu těchto slovních spojení. Na více informací se můžeme podívat v tabulce:

Slovní spojení	Počet slovních spojení	Počet slovních spojení v %
1-gram	2397	28,34%
2-gram	4139	48,72%
3-gram	1769	20,82%
4-gram	269	1,75%
5-gram	35	0,32%
6-gram a víc	7	0,05%

Tabulka 7.4: Průměrná délka slov

Nejvíce rozšířené klíčové slova obsahují 2 slova, za nimy jsou unigramy a nakonec trigramy. Slovní spojení o 4 a víc slovech tvoří zanedbatelné procento textu. Navíc tyto slovní spojení obsahují obecné slova, které prodlužují vyhledávání. Při provádění vyhledávání víceslovných výrazu o délce 4 a více slov se téměř nikdy nenalezne celé klíčové slovo. Vytváření tedy kombinací těchto slov a vyhledávání pouze zatěžuje systém.

7.5 Redundance klíčových slov

Tvoření klíčových slov spočívá hlavně v ohýbání a tvoření všech možných variant každého klíčového slova. Pomocí morfologického analyzátoru se určí základní tvar slova a toto slovo se ohýbá, dle rodu. U víceslovných výrazů probíhá jejich roznásobení, aby došlo k pokrytí každého tvaru. U této metody dochází k vytvoření velkého

množství redundantních slov jak můžeme vidět v tabulce. První sloupec značí, kolik slov tvoří vyhledávaný výraz, druhý sloupec znázorňuje, kolik se průměrně vytvoří slovních tvarů pro daný n-gram, následně třetí sloupec znázorňuje průměrný počet využití těchto slovních tvarů. Využitím je myšleno, kolik systém průměrně najde z těchto slov nebo slovních spojení v dokumentu. Poslední sloupec znázorňuje procentuální využití klíčových slov k vytvořeným variantám tohoto slova. U tohoto testu jsem nepoužil žádný stoplist, korpus nebo jiný filtr, který by odfiltroval obecná slova. Jak vidíme, tak u 4-gramů dochází k veliké redundanci a obrovskému zpomalení systému, protože systém hledá všechny vytvořené kalokace. I když státnicové okruhu tvoří i slovní spojení o více než 4 slovech, tak jsem tyto klíčové slova vzhledem k jejich vzácnému výskytu v textu nebral v úvahu.

Slovní spojení	Počet vytvořených slov	Počet využitých slov	Využití slov v %
1-gram	3	2	66,6
2-gram	14	5	36
3-gram	50	10	20
4-gram	279	7	2,5

Tabulka 7.5: Klíčové slova bez filtrace

Následující tabulka obsahuje průměrný počet klíčových slov po jejich filtraci. K filtraci se použil stoplist a backgroundový model. Jak vidíme, dochází k velkému optimalizování. Filtrace se projevila nejvíce u 4-gramů, kde se odfiltrovalo průměrně 87,4% nepotřebných slovních spojení. U 3-gramů se dostavilo vyfiltrování přibližně 36%, bigramů 14,2%, unigramy filtrace nepoznamenala. Použití filtrace způsobylo celkové zrychlení systému o 44% a ušetřila minuty vyhledávání.

Slovní spojení	Počet vytvořených slov	Počet využitých slov	Využití slov v %
1-gram	3	2	66,6
2-gram	12	5	42
3-gram	32	8	25
4-gram	35	9	26

Tabulka 7.6: Klíčové slova s filtrací

Tato metoda jak vidíme je pomalá, složitá a nepokryje všechny tvary kvůli ne vždy přesnému určení kořene slova. Proto jsem dospěl k názoru využít morfologický analyzátor pro převedení všech slov v textu na základní tvar. Lemmatizování obchází problém při skloňování a umožňuje rychlé vyhledání informací. V následující tabulce můžeme vidět, o kolik méně se vytvořilo variant klíčových slov. Celkově se zvýšilo využití vytvořených odkazů, nejvíce je to znát na 4-gramech a 2-gramech. Přestože je nutné převádět veškerý vstupní text a tím se příprava dokumentů prodlužuje, tak se nám veškerý čas vrátí při rychlém vyhledávání. Celkově se systém zrychlil o 48% oproti předchozímu řešení s filtrací.

Slovní spojení	Počet vytvořených slov	Počet využitých slov	Využití slov v %
1-gram	2	1	50%
2-gram	5	3	60%
3-gram	11	4	36%
4-gram	19	8	42%

Tabulka 7.7: Lemmatizované klíčové slova

7.6 Praktické použití systému

Vytvořený systém, respektive vytvořený interaktivní rejstřík pomocí tohoto systému jsem nechal otestovat dobrovolníky. Testovací skupinu tvořilo 10 dobrovolníků, kteří testovali systém nezávisle na sobě. Na těchto dobrovolnících jsem testoval, které informace jsou pro ně prioritní, které odkazy je zajímají více a jaké méně. Rejstřík byl tvořen pro studenty VUT FIT v Brně bakalářského studia. Z výsledků testovaní jsem zjistil, čemu uživatelé dávají přednost při vyhledávání. V následují tabulce můžeme vidět priority studentů seřazené sestupně.

1. četnost výrazů vzhledem k tématu
2. relevance výrazu vzhledem k tématu
3. četnost jednotlivých výrazů
4. klíčový výraz vzhledem k tématu

Všichni uživatelé se jednotně shodli, že je pro ně nejdůležitější informací četnost výrazů na jednotlivé témata, podle které se převážně orientují.Tato četnost by měla směrovat pozornost uživatele na určité předměty nebo témata. Stejně důležitá je i relevance jednotlivých vyhledávaných výrazů vzhledem k otázce a tématu. Na dalším místě se dostala četnost jednotlivých výrazů. Nejméně dobrovolníky zajímal nalezený klíčový výraz objevující se v určitém předmětu. Uživatel nejdřív zjistil, v jakém předmětu se konkrétní klíčové slovo objevuje nejvíce a v jakých předmětech se dále objevuje ve vysokém počtu. Z pozorovaného chování usuzuji, že "vysoký počet" shod na předmět musí překročit hranici 100. Následně se uživatel orientoval podle četnosti nalezení jednotlivých klíčových slov a v poslední kroku se uživatel orientoval podle četnosti jednotlivých výrazů vzhledem k předmětu, ve kterém bylo dané slovo nalezeno.

Počet použitých odkazů, který rejstřík obsahuje se odlišuje dle délky klíčového slova. Přisuzuji to tomu, že delší výrazy vytváří víc odkazů na výrazy, které mohou být více obecné a nemusí tolik reflektovat význam celého klíčového slova. První sloupec obsahuje o jaký n-gram se jedná a druhý sloupec znázorňuje přibližné využití vytvořených odkazů u daného slova.

Slovní spojení	Využití odkazů
1-gram	90-100%
2-gram	60-80%
3-gram	40-50%
4-gram	20-30%

Tabulka 7.8: Využití vytvořených odkazů

Jak můžeme vidět, čím delší výraz, tím si méně uživatelé prohlíželi jednotlivé odkazy. Docházelo k tomu hlavně z důvodu velkého množství redundantních výrazů. Zatímco jednoslovné výrazy obsahují méně odkazů a jsou výstižnější, u delších výrazů je to přesně naopak.

Kapitola 8

Závěr

Cílem práce bylo prozkoumat základní metody využívané pro detekování a extrakci klíčových slov v textu. Na základě těchto informací vytvořit systém umožňující propojit texty státnicových okruhů, studijních opor a doplňkových materiálů. Nakonec systém otestovat na studentech a zhodnotit, jak moc takový systém pomůže studentům VUT FIT s přípravou na státní závěrečnou zkoušku.
Před začátkem návrhu systému bylo nutné shromáždit veškerý studijní materiál na Fakultě informačních technologií, který se využil k testování. Tento balík tvořil přibližně 110 materiálů pro bakalářské studium a více než 500 materiálů pro magisterské studium, které jsem spolu s dalšímy daty využil pro tvoření korpusu. Následně jsem shromáždil státnicové okruhy z více než 20 fakult v České republice čítající více než 8000 slov, které jsem využil pro testování extrakce. V pozdější fázi testování přibyl balík zaměřující se na ekonomické fakulty. Tento balík jsem využil pro testování obecného využití systému.
Celý systém je implementován v jazyce Python, je zaměřen na české texty a lze ho obecně využít při vyhledávání v různých oborech. Testy prokázaly, že využití stoplistu a korpusu jsou nutností pro zrychlení vyhledávání, a to znamená vytvoření vlastního stoplistu pro daný obor. Systém využívá morfologický analyzátor, který je důležitou součástí, bez které by tento systém nemohl existovat.
Testování se věnuji v kapitole 7 a došel jsem k zajímavým výsledkům. Nejvíce redundantních klíčových slov bylo vytvořeno ze slovních spojení o 4 a více slov. Lemmatizací se systém značně urychlil, přesto je dle mě tvořeno veliké množství redundantních výrazů. Tomuto by se dalo předejít redukcí některých slovních kombinací a nesnažit se vždy tvořit všechny možné kombinace slov. Určitě by při redukci pomohl i backgroundový model, z většího množství textů, než jsem byl schopen shromáždit a pokusit se zkombinovat české i anglické, protože spoustu výrazů se v oboru informatiky napříč jazyky nemění, jako například názvy algoritmů. Dále by mohlo pomoci úplné vypuštění odsekávače slov, přestože se takový odsekávač slov v určitých momentech může hodit. Tím by se vytvořilo méně redundantních klíčových slov a systém by pracoval mnohem rychleji. Zajímavé by také bylo, jednotlivé klíčové výrazy vyhledávat na internetových zdrojích a nabídnout tak čtenáři i alternativní zdroje k vlastním materiálům. Například vytvoření sdílených dokumentů, kde by studenti popisovali podrobnější informace k daným okruhům a na těchto stránkách či dokumentech by následně systém mohl také vyhledávat. V současnosti existuje spoustu diskuzních portálů pro studenty, většina ale působí chaoticky,neorganizovaně a stále

ubývá aktivních uživatelů na těchto portálech, v dnešní době všichni sdílí informace na sociálních sítí. Takový sdílený soubor by mohl obsahovat i vlastní poznámky ke každému předmětu, či dokonce přepisy přednášek, protože největší problém vidím v tom, že ne každý předmět má kvalitní materiály, ve kterých lze vyhledávat. Některé předměty nemají přístupné materiály a některé dokonce nemají pro změnu materiály skoro žádné. Vyhledávání je jen jedna část z celého systému, pro úspěšné vyhledání informací je také potřeba, aby systém měl co prohledávat. V neposlední řadě bych zapracoval na zvýraznění nalezených klíčových slov v dokumentech pro rychlejší orientaci uživatele.

Výsledný rejstřík jsem otestoval i na menším počtu dobrovolníků. Podle jejich reakcí systém splnil většinu očekávání a dle zpětné vazby z ankety by systém uvítalo hodně studentů i mimo Fakultu informačních technologií. Přeci jen každá pomůcka, která umožní pohodlnější průchod studiem a zdolání zkoušek je vítaná. Většina nalezených informací v rejstříku pomohlo najít uživatelům klíčová slova, které hledali, nebo jim alespoň pomohla nasměrovat pozornost požadovaným směrem.
Samotný rejstřík není bezchybný, ale ve výsledku plní funkci rozcestníku, který dokáže do určité míry uživateli pomoci nalézt klíčové slova napříč velkým množství textu.

Literatura

[1] Ahmad, K.; Gillam, L.; Tostevin, L.: Weirdness Indexing for Logical Document Extrapolation and Retrieval (WILDER). *TREC - Text REtrieval Conference*, 1999.

[2] Foundation, T. P. S.: Python. `https://www.python.org/ [online]`, 2001 [cit. 2016-4-18].

[3] Frantzi, K.; Ananiadou, S.; Mima, H.: Automatic Recognition of Multi-Word Terms: the C-value/NC-value Method. *International Journal on Digital Libraries*, ročník 3, 2000: str. 115–130.

[4] Hajič, J.: *Disambiguation of Rich Inflection: Computational Morphology of Czech*. Karolinum, 2004, ISBN 8024602822.

[5] Hajič J., a., Hajičová E.: Průvodce PDT 2.0. 2006.

[6] ČSN ISO 999: Informace a dokumentace, Zásady zpracování, uspořádání a grafické úpravy rejstříků. 1998.

[7] Manning, C. D.; Raghavan, P.; Schütze, H.: *Introduction to Information Retrieval*. Cambridge University Press., 2008, ISBN 0521865719.

[8] McNamara, T.: Slate 0.5.2. `https://pypi.python.org/pypi/slate [online]`, 2007[cit. 2016-4-18].

[9] Osolsobě, K.: *Algoritmický popis české formální morfologie a strojový slovník češtiny*. Diplomová práce, Masarykova univerzita, Brno, 1996.

[10] Pořízka, P.: *Tvorba korpusů a vytěžování jazykových dat: metody, modely, nástroje*. Vydavatelství Filozofické fakulty Univerzity Palackého v Olomouci, 2014, ISBN 978-80-87895-17-7.

[11] Robertson, S.: Understanding Inverse Document Frequency: On theoretical arguments for IDF. *Journal of Documentation*, ročník 5, 2004: str. 503–520, ISSN 0022-0418.

[12] Sedláček, R.: *Morfologický analyzátor češtiny*. Diplomová práce, Masarykova univerzita, Brno, 1999.

[13] Shinyama, Y.: PDFMiner. `http://www.unixuser.org/ euske/python/pdfminer/ [online]`, 2004[cit. 2016-4-20].

[14] Spoustova, D.; Hajič, J.; Raab, J.; aj.: Semi-supervised Training for the Averaged Perceptron POS Tagger. *Conference of the European Chapter of the ACL*, ročník 12, 2009: str. 763–771.

[15] Strachota, T.: *Automatická tvorba tejstříku publikace*. Diplomová práce, VUT v Brně, FIT, 2008.

[16] Straková, J.; Straka, M.; Hajič, J.: Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Baltimore, Maryland: Association for Computational Linguistics, June 2014, s. 13–18.
URL http://www.aclweb.org/anthology/P/P14/P14-5003.pdf

[17] Veterinární a farmaceutická univerzita, F. v. B.: Přednáška 3: Testování hypotéz ve statistice. Testování normality.
http://cit.vfu.cz/statpotr/POTR/Teorie/Predn3/chi2test.htm [online], [cit. 2016-5-4].

[18] Wikipedia.org: Wikipedia: Python. `https://cs.wikipedia.org/wiki/Python [online]`, [cit. 2016-4-18].

[19] Wikipedia.org: Wikipedia: Test dobré shody.
https://cs.wikipedia.org/wiki/Test_dobré_shody [online], [cit. 2016-5-4].

[20] Wikipedia.org: Wikipedia: Rejstřík (organizace informací).
https://cs.wikipedia.org/wiki/Rejstřík_(organizace_informací) [online], [cit. 2016-5-6].

[21] Yuanhua, L.; ChengXiang, Z.: When Documents Are Very Long, BM25 Fails! *Sigir'11*, 2011: s. 1103–1104.

[22] Šimara, S.: *Automatické navrhování klíčových slov*. Diplomová práce, VUT v Brně, FIT, 2013.