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

AUTOMATICKÉ NAVRHOVÁNÍ KLÍČOVÝCH SLOV

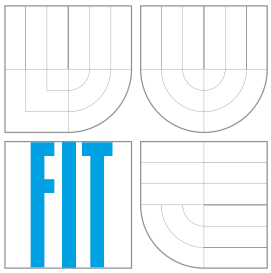
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

SVATOPLUK ŠIMARA

BRNO 2013

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

AUTOMATICKÉ NAVRHOVÁNÍ KLÍČOVÝCH SLOV

AUTOMATIC KEYWORDS SUGGESTION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — SVATOPLUK ŠIMARA
AUTHOR

VEDOUCÍ PRÁCE — Ing. LUBOMÍR OTRUSINA
SUPERVISOR

BRNO 2013

Abstrakt

Tato práce se zabývá automatickým návrhem klíčových slov českým dokumentům. Pro jejich návrh je využito čistě statistických metod. K analýze jsou použity diplomové a jiné závěrečné práce. Statistické metody jsou na vybraných dokumentech podrobně otestovány a vyhodnoceny, a pro konečný návrh klíčových slov jsou vybrány jen ty nejúspěšnější metody. Výsledky návrhu jsou na závěr porovnány s ručně přiřazenými klíčovými slovy.

Abstract

This thesis deals with the automatic keywords suggestion. The suggestion is based only on the statistic methods. For the analysis are used diploma thesis and similar documents. Statistic methods are detailed tested and evaluated by using these documents. For the final keywords suggestion ale chosen only the most successful methods. In the end, the suggested keywords are compared with the manual assigned keywords.

Klíčová slova

klíčová slova, navrhování klíčových slov, víceslovné termíny, vyhodnocení návrhu, statistické metody

Keywords

keywords, keywords suggestion, multiword terms, statistical methods, evaluation of the suggestion

Citace

Svatopluk Šimara: Automatické navrhování klíčových slov, bakalářská práce, Brno, FIT VUT v Brně, 2013

Automatické navrhování klíčových slov

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Lubomíra Otrusiny

.....
Svatopluk Šimara
15. května 2013

Poděkování

Děkuji Ing. Lubomíru Otrusinovi za nesmírně vstřícný přístup a cenné rady při vedení bakalářské práce.

Obsah

Kapitola 1

Úvod

Cílem této práce je vytvořit systém pro automatické navrhování klíčových slov dokumentům. Klíčová slova lze dobře uplatnit při vyhledávání; pokud hledaný dotaz odpovídá klíčovému slovu přiřazenému k dokumentu, bude pravděpodobně tento dokument relevantní k vyhledávanému dotazu.

Ruční přiřazení klíčových slov dokumentu je proces poměrně náročný, je třeba přečíst celý text dokumentu a pak s citem vybrat slova, která tento dokument vystihují, klíčová slova. Občas jsou to slova velice konkrétní, například odborné výrazy, ale někdy je třeba jako klíčové slovo zvolit i slovo obecné. Zároveň je třeba zvážit počet slov, která budou označena jako klíčová. Při použití malého počtu klíčových slov se snižuje pravděpodobnost nalezení dokumentu, naopak při použití velkého množství klíčových slov se zvyšuje riziko, že klíčová slova nebudou dobře vystihovat dokument.

Cílem této práce je automatizace tohoto procesu. Zadání jsem si vybral z čisté zvědavosti. Je vůbec možné proces návrhu klíčových slov automatizovat? A pokud ano, jak se budou výsledky lišit od ručního přiřazení?

Systém vytvořený v rámci této práce bude obsahovat algoritmy, kterými budou jednak vybrána klíčová slova, ale také bude vybrán jistý optimální počet těchto slov, a to v českém jazyce. Protože je čeština jazyk flektivní, bude třeba řešit časté ohýbání slov. To je značná komplikace oproti jiným jazykům, které jsou flektivní méně či dokonce vůbec.

Aby bylo možné jednoduše a hlavně rychle porovnávat výsledky návrhu, bude také vytvořen nástroj pro vyhodnocení vhodnosti automaticky navržených klíčových slov.

Tato práce navazuje na diplomovou práci Tomáše Strachoty z roku 2010, který se zabýval stejným tématem. Ve své práci popsal přístup k řešení problémů a hlavně, jak sám píše, implementoval framework, který by měl být jednoduše rozšiřitelný. V této práci bude jeho framework využíván a dále rozšiřován.

V následující 2. kapitole je uveden určitý teoretický základ, v kapitole 3 je analyzován framework, na který je navazováno. Kapitola 4 se zabývá automatickým vyhodnocením systému navrhující klíčová slova, v kapitole 5 je popsáno, pro jaké dokumenty je systém navržen a jak byly tyto dokumenty získány. Kapitoly 6, 7 a 8 se zabývají jádrem této práce - navržením klíčových slov, vybráním jejich vhodného počtu a vytvořením jazykově správného tvaru výsledných klíčových slov.

Kapitola 2

Použité metody a nástroje

V této kapitole jsou popsány metody a nástroje, které jsou využity při tvorbě této práce. Pomocí PDT z kapitoly 2.2 jsou navržena kandidátní slova, algoritmy z kapitoly 2.3 těmto slovům zjistí míru příslušnosti k dokumentu a v kapitole 2.4 jsou popsány metriky určené k vyhodnocení úspěšnosti systému.

2.1 Klíčová slova a kandidátní slova

Klíčová slova jsou slova a krátká slovní spojení přirozeného jazyka, která vyjadřují sémantický obsah dokumentu [1]. Jednotným číslem bude v této práci označováno i slovní spojení tvořící jedno klíčové slovo. Množné číslo - klíčová slova bude označovat více těchto slov a slovních spojení.

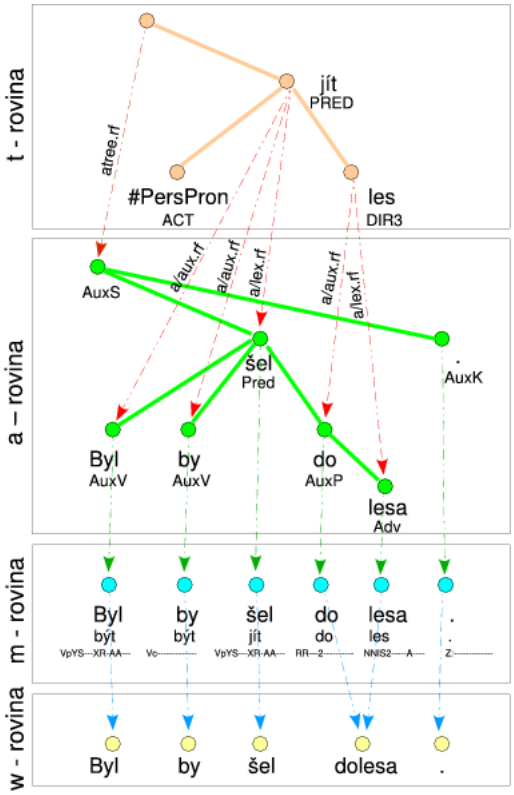
Jako kandidátní slovo bude označeno takové slovo či slovní spojení, u kterého je z nějakého důvodu pravděpodobné, že se může stát slovem klíčovým.

2.2 PDT

Pražský závislostní korpus (PDT) je probíhající projekt pro ruční anotaci velkého množství českých textů bohatou lingvistickou informací, sahající od morfologie přes syntax až po sémantiku/pragmatiku. PDT 2.0 obsahuje anotaci morfologie, povrchovou syntax, hloubkovou syntax a sémantiku, aktuální členění, koreferenci a lexikální sémantiku založenou na valenčním slovníku. Obsahuje texty o rozsahu 2 miliony slov s provázanými anotacemi na úrovni morfologie, 1,5 milionů slov je anotováno na úrovni povrchové syntaxe a 800 tisíc na úrovni hloubkové syntaxe a sémantiky [7].

PDT 2.0 obsahuje softwarové nástroje pro prohledávání korpusu, anotaci dat a jazykovou analýzu. PDT 2.0 slouží především k aplikaci teoretických výsledků Pražské lingvistické školy na velké množství dat a tím ověřit a zachovat teorii funkčně generativního popisu (FGD). Dále PDT umožňuje vytvořit nástroje automatické analýzy a generování jazykových dat.

PDT 2.0 anotuje data ve čtyřech rovinách. Jsou to: slovní rovina (w-rovina), morfologická rovina (m-rovina), analytická rovina (a-rovina), tektogramatická rovina (t-rovina). Vyšší rovina staví vždy na nižší, jejich propojení je zobrazeno na obrázku 2.1.



Obrázek 2.1: Propojení rovin anotace [7]

2.2.1 Morfologická rovina

Posloupnost slovních jednotek slovní roviny je rozdělena do vět, jsou přiřazeny některé atributy, z nichž nejdůležitější jsou morfologické lemma a tag. Lemma je základní tvar slova, lemmatizace je proces převodu slova do jeho základního tvaru [16], tag vyjadřuje slovní druh a hodnoty dalších morfologických kategorií.

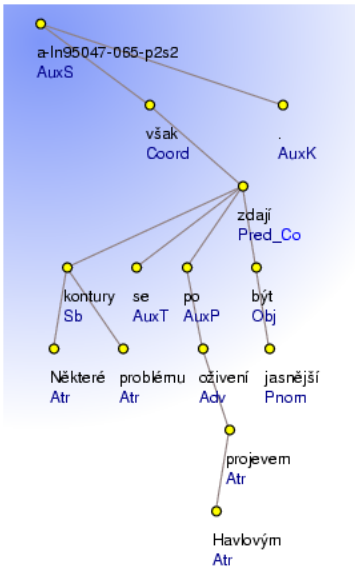
V příloze C je uvedena kompletní morfologická analýza věty „*Některé kontury problému se však po oživení Havlovým projeven zdají být jasnější.*“

Lemma má dvě části. První, povinnou část, tvoří samotné lemma. Může ještě následovat číslem, které rozlišuje různý původ stejného lemma (například slovo *vazba* - spojení nebo vazba obviněného). Druhá, nepovinná část, obsahuje další informace o lemma například sémantickou či odvozenou informaci.

Tag je tvořen řetězcem o 15 znacích. V každé pozici je jedním znakem zakódována jedna mluvnická kategorie. Významy jednotlivých pozic jsou popsány v příloze D. Přípustné hodnoty každé pozice a jejich význam je uveden v manuálu pro morfologickou anotaci [8]

2.2.2 Analytická rovina

V analytické rovině ještě zůstávají všechna slova zachována a dostávají svou funkci ve výsledné struktuře. Věta je v této rovině reprezentována stromem, jehož uzly jsou lexikální jednoty a hrany představují vztahy mezi těmito jednotkami, případně jiné vztahy [7]. Grafické znázornění je uvedeno na obrázku 2.2. Každý uzel má dále 12 atributů, které jsou uvedeny v příloze E. Cílem anotace je vytvořit strukturu věty a označit funkce jednotlivých slov ve větě.



Obrázek 2.2: Grafické znázornění stromu analytické roviny [7]

Atributy `lemma` a `tag` jsou popsány již v předchozí kapitole, mohou ale navíc obsahovat dosud nerozlišené hodnoty. Atribut `form` je tvar slova, který byl vstupem do morfologické analýzy. Většinou je shodný s původním slovem, ale může se lišit například pokud analyzátor objevil překlep. Atribut `afun` obsahuje funkci slova ve větě, jde o nejdůležitější atribut získaný z analytické roviny. `lemid` je atribut zamýšlen pro budoucí použití - pro identifikaci víceslovných lexikálních jednotek. `mstag` bude použit jako mezistupeň mezi `tag` a tzv. gramatémy na tektogramatické rovině. Ostatní atributy jsou čistě technického charakteru, pro naše použití nezajímavé (využívají se například při ruční anotaci).

2.2.3 Tektogramatická rovina

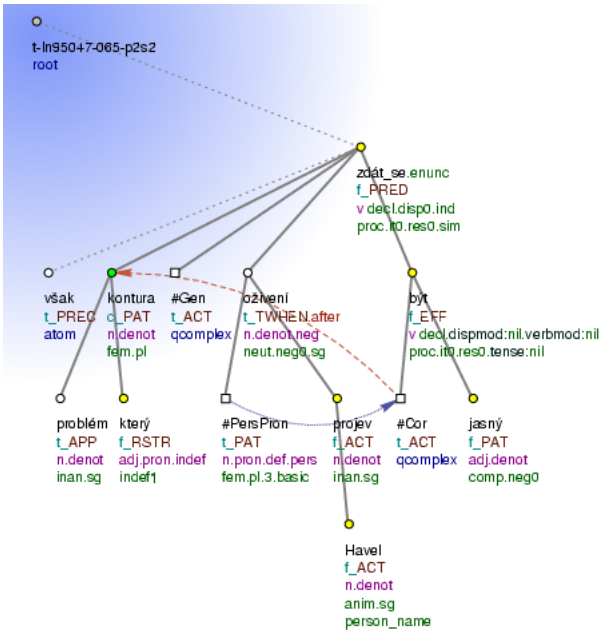
Výsledkem tektogramatické roviny je, stejně jako v analytické rovině, strom. Grafické znázornění stromu je v obrázku 2.3. Oproti analytické rovině je strom orientovaná více sémanticky a obsahuje daleko více lingvistické informace. V tektogramatickém stromu jsou uzly pouze plnovýznamová slova (např. zdají se) a jsou doplněny další uzly (např. nevyjádřený podmět). Tektogramatická rovina se dobře uplatňuje v překladech, protože tektogramatický strom věty a překladu se shodují mnohem více než v případě analytického stromu [7].

2.3 Statistické algoritmy pro ohodnocování kandidátních slov

2.3.1 Názvosloví

V anglické literatuře je pro výraz četnost výskytů používáno slovo *frequency*, případně *term frequency*. Čeština zato dovoluje rozlišit slova *četnost* - veličina udávající počet hodnot daného znaku v souboru dat [11] a *frekvence* - relativní četnost vzhledem k celkovému počtu prvků soboru. Přestože dle Slovníku cizích slov [11] může být frekvence chápána i jako četnost, v této práci budou veličiny frekvence a četnost chápány ve vztahu

$$f = \frac{n}{N}, \tag{2.1}$$



Obrázek 2.3: Grafické znázornění stromu tektogramatické roviny [7]

kde f je frekvence, n je četnost a N je počet prvků v souboru.

2.3.2 Term frequency

Term frequency - zkráceně TF je nejjednodušší a nejrychlejší metoda ohodnocování kandidátních slov. Spočítá se jako:

$$TF(t) = \frac{f(t)}{|N|}, \tag{2.2}$$

kde t je ohodnocovaný termín, *f()* je četnost termínu, N je množina termínů a $|N|$ je počet termínů v množině. Jak je vidět, jde vlastně jen o normalizovanou četnost kandidátního slova [14].

2.3.3 Inverse document frequency

Inverse document frequency - IDF je metoda ohodnocování, která přiřazuje větším váhu kandidátním slovům, které se vyskytují v méně dokumentech a menší váhu slovům, které se vyskytují ve více dokumentech. Její základní podoba je:

$$IDF(t) = log\left(\frac{|D|}{|\{d \in D : t \in d\}|}\right), \tag{2.3}$$

kde t je ohodnocovaný termín, D je množina dokumentů v korpusu, $|D|$ je počet dokumentů v korpusu a $|\{d \in D : t \in d\}|$ je počet dokumentů, které obsahují t [19]. Jak je vidět, tuto metodu je možné použít, pokud ohodnocujeme korpus dokumentů (kde je počet dokumentů větší než 1). Také se dá odhadnou, že čím rozsáhlejší korpus máme, tím hodnotnější budou výsledky.

Podle práce [19] se ukázala IDF jako neobyčejně robustní a Stephen Robertson v této práci dále dodává že je sice možné, že IDF možná v explicitním tvaru v budoucnu vymizí, ale její esence bude žít dále.

2.3.4 Term frequency - inverse document frequency

Term frequency - inverse document frequency, zkráceně TF-IDF je kombinací Term frequency z kapitoly 2.3.2 a Inverse document frequency z kapitoly 2.3.3 [13]. Vypočítá se jako:

$$TF - IDF(t) = TF(t) \cdot IDF(t), \tag{2.4}$$

kde t je ohodnocovaný termín, *TF()* je Term frequency a *IDF()* je Inverse document frequency [19].

2.3.5 Term frequency - inverse paragraph frequency

Term frequency - inverse paragraph frequency, zkráceně TF-IPF je obměna TF-IDF z kapitoly 2.3.4 popsaná v [22]. V části *IPF* se počítají kandidátní slova v odstavcích, ne v dokumentech. Kandidátní slovo, které se vyskytuje v mnoha odstavcích dostane malé ohodnocení, zato kandidátní slovo, které se vyskytuje v málo odstavcích bude ohodnoceno lépe. Pro použití této metody je potřeba mít texty vhodně rozděleny do odstavců. Spočítá se jako:

$$TF - IPF(t) = TF(t) \cdot IPF(t), \tag{2.5}$$

kde t je ohodnocovaný termín, *TF()* je Term frequency a *IPF()* je Inverse paragraph frequency, která se spočítá jako:

$$IPF(t) = log\left(\frac{|P|}{|\{p \in P : t \in p\}|}\right), \tag{2.6}$$

kde t je ohodnocovaný termín, P je množina odstavců v dokumentu, $|P|$ je počet odstavců v dokumentu a $|\{p \in P : t \in p\}|$ je počet odstavců, které obsahují t.

2.3.6 Residual inverse document frequency

Residual inverse document frequency - RIDF je alternativa k IDF z kapitoly 2.3.3. RIDF je definována jako rozdíl mezi IDF a logaritmem IDF předpovězeným Poissonovým rozdělením pravděpodobnosti. Parametr rozdělení pravděpodobnosti $\lambda = \frac{f(t)}{|D|}$, kde t je ohodnocovaný termín, *f(t)* je četnost termínu t v korpusu a $|D|$ je počet dokumentů v korpusu [14]. Vypočte se tedy takto:

$$RIDF(t) = IDF(t) - log\left(\frac{1}{P(Y > 0)}\right) = IDF(t) + log(P(Y > 0)) \tag{2.7}$$

Obecné Poissonovo rozdělení pravděpodobnosti jevu, kdy veličina $Y \in \mathbb{N}^0$ [4]:

$$P(Y = k) = \frac{\lambda^k}{k!} \cdot e^{-\lambda} \tag{2.8}$$

Pravděpodobnost alespoň jednoho výskytu jevu:

$$P(Y > 0) = 1 - P(Y = 0) = 1 - \frac{\lambda^0}{0!} \cdot e^{-\lambda} = 1 - e^{-\lambda} \tag{2.9}$$

Kompletní výpočet RIDF je tedy:

$$RIDF(t) = IDF(t) + log(P(Y > 0)) = IDF(t) + log(1 - e^{-\lambda}) = IDF(t) + log(1 - e^{-\frac{f(t)}{|D|}}) \tag{2.10}$$

2.3.7 Okapi BM25

Okapi BM25 je hodnotící metoda původně určená pro vyhledávače. Určuje míru shody dokumentu k vyhledávacímu dotazu [18]. Pro naše účely můžeme tuto metodu využít tak, že jako dotaz použijeme postupně všechna kandidátní slova. Vypočítá se jako:

$$BM25(t, d) = log\left(\frac{|D| - n(t) + 0.5}{n(t) + 0.5}\right) \cdot \sum_{q \in t} \frac{f(q) \cdot (k_1 + 1)}{k_1((1 - b) + b\frac{|d|}{avglen}) + f(q)}, \tag{2.11}$$

kde t je ohodnocovaný termín, d je dokument pro který se výraz vyhodnocuje. D je množina dokumentů v korpusu, $|D|$ je počet dokumentů v korpusu, $n(t) = |\{d \in D : t \in d\}|$ je počet dokumentů, které obsahují *t*. *f()* je četnost slova v dokumentu d, $|d|$ je délka dokumentu d ve slovech, *avglen* je průměrná délka dokumentu v korpusu měřená ve slovech, q je jedno slovo z kandidátního slova t (suma se provádí přes všechna tato q), k_1 a $b \in < 0; 1 >$ jsou volné parametry, podle [18] volené jako $k_1 = 2$ a $b = 0.75$.

2.3.8 BM25L

Autoři článku [12] odhalili, že *Okapi BM25* má tendenci penalizovat dlouhé dokumenty. Navrhují také vylepšení této metody právě pro dlouhé dokumenty. V jejich článku je použit mírně odlišný způsob výpočtu oproti 2.11, proto zde bude uveden v plném rozsahu.

$$\sum_{q \in Q \cap D} \frac{(k_3 + 1)c(q, Q)}{k_3 + c(q, Q)} \cdot f(q, D) \cdot log\frac{N + 1}{df(q) + 0.5}, \tag{2.12}$$

kde *c(q,Q)* je počet slov v dotazu Q, N je celkový počet dokumentů, *df(q)* je počet dokumentů, ve kterých se vyskytuje slovo q, k_3 je parametr. Funkce *f(q,D)* se vypočítá jako:

$$f(q, D) = \frac{(k_1 + 1)c(q, D)}{k_1(1 - b + b\frac{|d|}{avglen}) + c(q, D)} = \frac{(k_1 + 1)c'(q, D)}{k_1 + c'(q, D)}, \tag{2.13}$$

kde $|d|$ je délka dokumentu, *avglen* je průměrná délka dokumentu v korpusu měřená ve slovech, *c(q,D)* je četnost slova q v dokumentu D, b a k_1 jsou parametry. $c'(q, D)$ je četnost slova q normalizována délkou dokumentu následujícím způsobem:

$$c'(q, D) = \frac{c(q, D)}{1 - b + b\frac{|d|}{avglen}} \tag{2.14}$$

Pokud je dokument velice dlouhý, tak je $c'(q, D)$ velice malé a začíná se blížit 0 [12], díky tomu se i $f(q, D)$ začne blížit 0. Výsledek tedy vypadá stejně, jako by se q v dokumentu D vůbec nevyskytlo. V článku [12] je popsán způsob, jak odlišit stav, kdy se q v D nenalézá a

kdy je pouze dokument velice dlouhý. Zavádí „mezeru“ mezi skóre 0 a nízké skóre způsobené délkou dokumentu. Jde o úpravu funkce $f(q, D)$ následujícím způsobem:

$$f(q,D) = \begin{cases} \frac{(k_1+1)\cdot[c'(q,D)+\delta]}{k_1+[c'(q,D)+\delta]} & \text{pro } c'(q,D) > 0 \\ 0 & \text{jindy} \end{cases}, \tag{2.15}$$

kde δ je parametr zajišťující posun výsledného skóre dále od nuly.

2.3.9 Lexical cohesion

Metoda lexical cohesion byla vytvořena pouze pro víceslovné termíny. Měří, jak moc jednotlivá slova termínu spolu souvisejí [17]. Toho dociluje poměrem výskytu celého termínu a součtu výskytů jednotlivých slov. Předpokládá se, že termín t se skládá ze slov $w_1, w_2, \ldots$ Výpočet má tvar:

$$\text{Lexical coehsion}(t) = \frac{|t| \cdot log_{10}(f(t)) \cdot f(t)}{\sum_{w \in t} f(w)}, \tag{2.16}$$

kde t je ohodnocovaný termín, $|t|$ je počet slov tvořící termín, *f()* je četnost v dokumentu, w je slovo z termínu t.

2.3.10 Weirdness

Doslovně přeloženo je metoda Weirdness založena na „zvláštnostech“. Míra zvláštnosti je definována jako rozdíl rozložení slova ve specializovaném korpusu oproti obecnému korpusu. Zvláštnější slovo bude pravděpodobně reprezentovat dokument lépe než slovo obecné [2]. Vypočítá se jako:

$$\text{Weirdness}(t) = \frac{fs(t)}{fg(t)}, \tag{2.17}$$

kde t je ohodnocovaný termín, *fs(t)* je frekvence termínu v dokumentu a *fg(t)* je frekvence termínu v obecném korpusu. Podmínkou pro použití této metody je znalost rozložení slov v obecném korpusu daného jazyka.

2.3.11 C-value

C-value je metoda ohodnocování navržena pouze pro víceslovné termíny [5]. Krom četnosti slova a jeho délky při výpočtu jako jediná zkoumaná metoda pracuje i s vnořenými termíny. Metoda penalizuje termíny, které jsou obsaženy jako součást delších termínů (tím vlastně přiřazuje vyšší skóre delším termínům). Vypočítá se jako:

$$C - value(t) = \begin{cases} log_2|t| \cdot f(t) & \text{pokud } t \text{ není vnořený termín} \\ log_2|t| \cdot (f(t) - \frac{1}{|L|}\sum_{b \in L} f(b)) & \text{jindy} \end{cases}, \tag{2.18}$$

kde t je ohodnocovaný termín, *f()* je četnost v dokumentu, L je množina kandidátních termínů, které obsahují t, $|L|$ je počet těchto termínů. Pokud je kandidátní slovo pouze jednoslovné, pak $log_2|t| = log_2(1) = 0$ a tím je celkové skóre přiřazené tomuto jednoslovnému termínu 0.

2.3.12 Kombinace algoritmů

Při kombinování algoritmů je problém s různým rozsahem hodnot. Při kombinaci algoritmů, u kterých není omezen rozsah hodnot nelze použít jednoduchou normalizaci.

V práci [23] je tento problém vyřešen. Klíčová slova jsou každým algoritmem seřazena a jejich ohodnocení vyplývá z jejich umístění. Pro každý algoritmus musí být také přiřazena váha a tím vzniká celkové ohodnocení:

$$\text{rank}(t) = \sum_{i=1}^{k} \frac{1}{R(t_i)} w_i, \tag{2.19}$$

kde t je ohodnocovaný termín, k je počet algoritmů, w_i je váha algoritmu a $R(t_i)$ je pořadí termínu t ve výsledcích algoritmu i. Váha w_i je vypočítána jako:

$$w_i = \frac{P_i}{\sum_{i=1}^{k} P_i}, \tag{2.20}$$

kde w_i je váha algoritmu a P_i je přesnost algoritmu i.

2.4 Metriky pro vyhodnocení úspěšnosti systému

Metriky popsané v této kapitole budou použity pro ohodnocení úspěšnosti systému navrženého v rámci této práce. V této části jsou tyto metriky popsány jen obecně, samotné použití těchto metrik pro ohodnocení systému bude popsáno v dalších kapitolách.

2.4.1 Typy chyb

Při vyhodnocování může nastat jedna ze čtyř variant uvedených v tabulce 2.1 [9].

Výsledek	**Ve skutečnosti je**	
	Pravda	**Nepravda**
Pravda	**True positive (Skutečně pozitivní)**	False positive (Falešně pozitivní)
Nepravda	False negative (Falešně negativní)	**True negative (Skutečně negativní)**

Tabulka 2.1: Možné výsledky při vyhodnocování

Tučně jsou v tabulce 2.1 uvedeny možnosti, kdy je výsledek shodný se skutečností. Jsou to takzvané *true positive* - výsledek měl být kladný a také byl a *true negative* - výsledek byl záporný a také záporný být měl.

Pokud je výsledek kladný, ale ve skutečnosti kladný být neměl, jde o chybu zvanou *false positive*, můžeme se také setkat s názvem *false alarm*.

Pokud je výsledek záporný, ale ve skutečnosti měl být kladný, jde o chybu nazývanou *false negative*.

2.4.2 Přesnost a pokrytí

Přesnost, anglicky precision, je míra vyjadřující, jak velká část výsledků je relevantních [21].

$$\text{Přesnost} = \frac{true\ positive}{true\ positive + false\ positive} \tag{2.21}$$

Pokrytí, anglicky recall, je míra vyjadřující, jak velká část výsledků ze všech možných, byla nalezena [21].

$$\text{Pokrytí} = \frac{true\ positive}{true\ positive + false\ negative} \tag{2.22}$$

2.4.3 F-measure

Přesnost a pokrytí lze zkombinovat do jedné míry nazývané *F-measure*. Vypočítá se jako [20]:

$$\text{F-measure} = \frac{1}{\alpha\frac{1}{p} + (1-\alpha)\frac{1}{r}}, \tag{2.23}$$

kde p přesnost (precision), r je pokrytí (recall) a $\alpha \in < 0; 1 >$ je váhový parametr určující důležitost přesnosti či pokrytí. Čím vyšší je α, tím vyšší význam má přesnost na únor pokrytí a naopak.

Obvykle je zvolena hodnota $\alpha = 0.5$, což znamená, že přesnost a pokrytí mají stejnou váhu. Tím se výpočet zjednodušuje na:

$$\text{F-measure} = \frac{2pr}{p + r}, \tag{2.24}$$

což je vlastně harmonický průměr přesnosti a pokrytí. Při takto zvolené váze $F_{\alpha=0.5}$ se metrika nazývá také F_1*-measure* [20].

2.4.4 Seskupování výsledků

Existují dva základní způsoby počítání úspěšnosti systémů zpracovávající přirozený jazyk založené na přesnosti a pokrytí. Jsou to *mikro průměr* a *makro průměr*. Rozdíl mezi mikro a makro průměrem je, krom výpočtu, že mikro průměr přiřazuje stejnou váhu všem průměrovaným prvkům, kdežto makro průměr přiřazuje stejnou váhu všem kategoriím [3].

$$\text{Přesnost}_{mikro} = \frac{\sum_{d \in C} tp(d)}{\sum_{d \in C}(tp(d) + fp(d))} \tag{2.25}$$

$$\text{Pokrytí}_{mikro} = \frac{\sum_{d \in C} tp(d)}{\sum_{d \in C}(tp(d) + fn(d))} \tag{2.26}$$

$$\text{Přesnost}_{makro} = \frac{1}{|C|}\sum_{d \in C}\frac{tp(d)}{tp(d) + fp(d)} \tag{2.27}$$

$$\text{Pokrytí}_{makro} = \frac{1}{|C|}\sum_{d \in C}\frac{tp(d)}{tp(d) + fn(d)}, \tag{2.28}$$

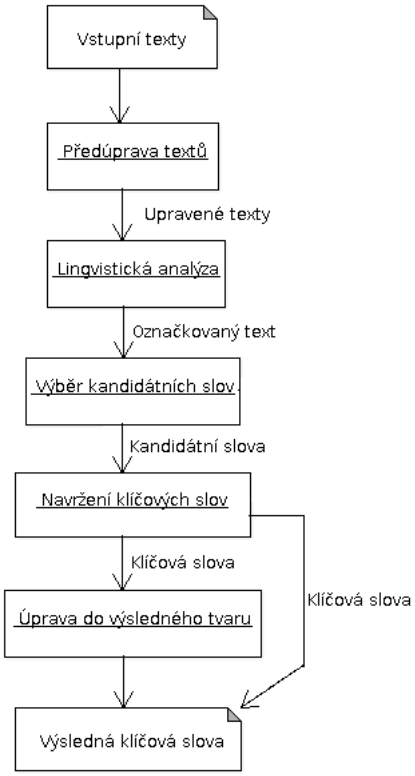
kde C je množina (dokumentů), pro kterou vyhodnocujeme metriky, $|C|$ je počet prvků množiny C, *tp(d)* je počet true positive, *fp(d)* je počet false positive a *fn(d)* je počet false negative.

Z takto zprůměrovaných výsledků se celková *F-measure* opět vypočítá harmonickým průměrem hodnot Přesnost$_{mikro}$ a Pokrytí$_{mikro}$ nebo Přesnost$_{makro}$ a Pokrytí$_{makro}$.

Kapitola 3

Analýza aktuálního systému

Systém pro navrhování klíčových slov, na jehož základech staví tato práce, je rozdělen do 6 relativně oddělených částí [22]. Jeho schéma je na obrázku 3.1



Obrázek 3.1: Schéma systému

3.1 Předúprava vstupních textů

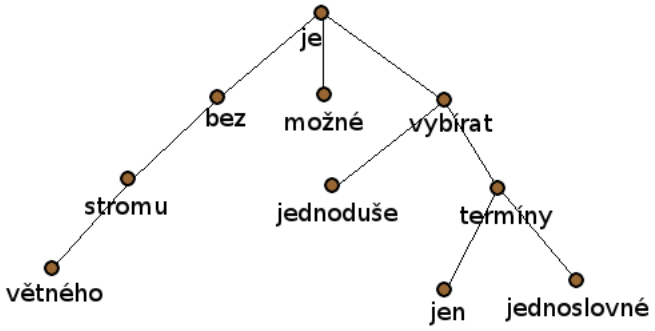
Vstupní texty je nejprve třeba upravit tak, aby si s nimi poradil lingvistický analyzátor. Každá věta musí přijít na samostatný řádek, toto je minimální požadavek na předúpravu. V předchozí práci [22] se navíc v textech odstraňují spojovníky na koncích řádků a odstraňují se rovnice.

3.2 Lingvistická analýza

Je využito nástroje PDT, který je popsán v kapitole 2.2. Je využíván až do analytické roviny, to znamená, že každé slovo je označeno mluvnickými kategoriemi, je mu přiřazeno lemma, je vytvořen strom reprezentující větu a každému slovu ve větě je přiřazena jeho funkce.

Jelikož se v češtině velice často používá skloňování a časování, je nejdůležitější ze všech těchto operací přiřazení lemma. Bez této operace by statistické metody pro počítání četností selhaly, protože by nebyly schopny odhalit stejná slova, byť v jiném tvaru.

Druhou významnou operací je sestavení stromu věty. Bez větného stromu je možné jednoduše vybírat jen jednoslovné termíny. Pokud bychom chtěli i víceslovné termíny, museli bychom brát v úvahu všechny kombinace slov. To by například pro větu „*Bez větného stromu je možné jednoduše vybírat jen jednoslovné termíny.*“ znamenalo 45 dvouslovných a 120 trojslovných termínů.



Obrázek 3.2: Příklad vytvořeného větného stromu

Protože se můžeme pohybovat pouze pro větvích stromu, tak tento větný strom, znázorněný v diagramu 3.2, redukuje počet možností na 9 dvouslovných a 11 trojslovných termínů.

Mluvnické kategorie a funkce ve větě dále usnadňují vybírání termínů. Odfiltrováním kombinací, které nepřináší žádné klíčové slovo, je možné opět redukovat počet vybraných kandidátů.

PDT pracuje s kódováním ISO-8859-2, kdežto zbytek systému pracuje s kódováním UTF-8. Pro převod kódování do ISO-8859-2 i z něj bylo použito nástroje *konwert* [10].

3.3 Výběr kandidátních slov

Jakmile jsou vytvořeny větné stromy, může začít proces návrhu kandidátních slov.

Jednoslovná kandidátní slova není třeba vybírat pomocí stromu věty. Jsou vybírána na základě svého slovního druhu či funkce ve větě.

Pro výběr víceslovných kandidátních slov jsou sestavena složitější pravidla, každé pravidlo jde zapsat pomocí vlastního stromu. Ve stromu věty jsou vyhledávány podstromy a pokud podstrom věty odpovídá pravidlům, je tento podstrom vybrán jako kandidátní slovo.

3.3.1 Pravidla

Každé pravidlo v sobě zahrnuje informaci o tvaru podstromu a pro každý uzel může volitelně definovat pravidla pro lemma, slovní druh a funkci ve větě. Struktura pravidla:

```
l|t|f
```

l je lemma, *t* je značka (tag) a *f* je funkce. Každou část pravidla lze vynechat zapsáním symbolu *?* místo části, která není relevantní. Konkrétní pravidlo pro uzel může vypadat:

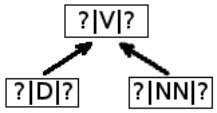
```
?|AU|Obj
```

Takovému pravidlu vyhovují všechna přivlastňovací přídavná jména, která ve větě plní funkci předmětu.

Pravidla zahrnující více slov vytváří malý strom, přesto se stále zapisují na jeden řádek. Příklad pravidla:

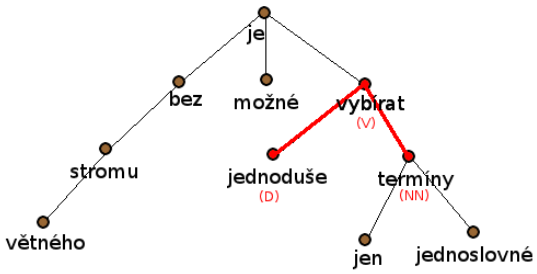
```
[?|V|?]([?|NN|?][?|D|?]) 2|3|1
```

První část určuje, jakou podobu má podstrom a druhá část výsledné pořadí. Překreslení pravidla do grafické podoby je na obrázku 3.3. Kvůli pořadí `2|3|1` bude mít kandidátní slovo na prvním místě příslovce *(D)*, na druhém místě sloveso *(V)* a na třetím místě podstatné jméno *(NN)*.



Obrázek 3.3: Grafické znázornění příkladu pravidla

Pro naši zkoumanou větu „*Bez větného stromu je možné jednoduše vybírat jen jednoslovné termíny.*“ toto pravidlo odpovídá podstromu *vybírat termíny jednoduše*, tento nález je zobrazen na obrázku 3.4.



Obrázek 3.4: Nalezení podstromu podle pravidla

Při seřazení slov podle druhé části pravidla - `2|3|1` - je z interního tvaru *vybírat termíny jednoduše* vytvořen výsledný tvar *jednoduše vybírat termíny*.

3.4 Navržení klíčových slov

Kandidátní slova nejsou seřazena. V tomto kroku je potřeba jim přiřadit skóre, seřadit je, vybrat několik prvních a ta označit za klíčová. Pro každé kandidátní slovo v každém dokumentu se musí vypočítat hodnoty metod z kapitoly 2.3. Hodnoty různých algoritmů pro každé kandidátní slovo jsou zkombinovány a tím je vytvořeno konečné skóre pro kandidátní slovo v dokumentu.

Ze seřazených kandidátních slov lze vybrat buď několik prvních nebo ta kandidátní slova, jejichž skóre překročilo určitý práh. Záleží na použití klíčových slov.

Výstupem jsou pouze lemma klíčových slov, to ale často nevadí, protože je první pád či infinitiv často vhodný.

3.5 Úprava do výsledného tvaru

Hlavně víceslovná klíčová slova není někdy vhodné ponechat v základním tvaru (lemma). Například slovní spojení *strom věty* vypadá v základní tvaru jako *strom věta*.

Úprava do výsledného tvaru probíhá ve více krocích. Nejprve jsou ze všech výskytů klíčového slova extrahovány značky (tagy) a ty jsou agregovány do společné značky.

Pokud je společná značka v textu nalezena, tvar tohoto slova se použije a můžeme si být téměř jistí, že je výsledek jazykově správný. Bohužel se stále můžou stát chyby, které neovlivníme, protože se spoléháme na autory textů a na automatickou morfologickou analýzu.

Pokud tvar klíčového slova v požadované značce nelze najít, je třeba jej vytvořit. K tomu je použit nástroj *Czech „Free“ Morphology* [6]. Pokud se nástroji nepodaří vytvořit výsledný tvar slova, je použito lemma.

3.6 Zhodnocení aktuálního systému

Při studiu současného systému se ukázalo, že je dobře navržený a hlavně také přehledně implementovaný. Přestože je zdrojový kód dokumentován pouze diplomovou prací, je i bez dalších komentářů většinou srozumitelný.

Systém byl otestován na několika dokumentech a bylo příjemné vidět, že systém pracuje a poskytuje přijatelné výsledky. Pro tyto dva důvody bylo zvoleno pokračovat v současné implementaci.

Kapitola 4

Vyhodnocení systému

Do současného systému je postupně provedeno spoustu zásahů. Pokud ovšem chceme znát výsledky modifikací, je velice vhodné určit si metriky, podle kterých se bude úspěšnost systému posuzovat. Ještě lepší je vytvořit nástroj, který tyto metriky bude měřit a bude dávat nezkreslený pohled na prováděné modifikace.

Předpokládá se, že ke každému dokumentu jsou známa klíčová slova, která danému dokumentu byla předem přiřazena, například autorem dokumentu, jako referenční klíčová slova. S těmito referenčními klíčovými slovy budou porovnávána automaticky navržená klíčová slova.

4.1 Zkoumané metriky

Systém bude vyhodnocován metrikami *přesnost*, *pokrytí* a *F-measure*. U všech třech metrik platí, že jejich obor hodnot $y \in <0;1>$ a také, že čím vyšší hodnota, tím je výsledek lepší. Díky těmto vlastnostem lze všechny 3 metriky zobrazovat dohromady beze ztráty čitelnosti.

Výsledky z více dokumentů budou seskupovány pomocí *mikro průměru*, který oproti *makro průměru* dává stejnou váhu každému dokumentu [3], což je pro naše experimenty žádoucí.

4.2 Grafy metrik

Při vyhodnocování známe pro každý dokument klíčová slova seřazená od nejlépe po nejhůře ohodnocené. Budou postupně brány skupiny klíčových slov, nejdříve první klíčové slovo, pak první dvě, první tři, atd. až budou brána všechna klíčová slova. Pro každou tuto skupinu budou vypočítány metriky a ty budou zaneseny do grafu.

Máme například dokument, pro který byla navržena slova uvedené v tabulce 4.1, tučně zvýrazněná jsou klíčová, ostatní jsou falešně klíčová (false positive).

V obrázku 4.1 je zobrazena metodika sestavování grafů. Z navržených klíčových slov se vezme prvních x položek a pro tyto položky je vypočítána hodnota metriky a promítnuta na osu y.

4.2.1 Pokrytí

Pokrytí je metrika odpovídající na otázku *„Kolik referenčních klíčových slov bylo nalezeno?“* a její vyhodnocení je zobrazeno v grafech 4.2. Na ose x je zobrazen počet klíčových

Pozice	Klíčové slovo
1	**lidový píseň**
2	lidový kultura
3	**dechový kapela**
4	hudební folklór
5	lidový tanec

Tabulka 4.1: Příklad navržených klíčových slov

slov, které byly zařazeny do výpočtu pokrytí. Na ose y je pokrytí odpovídající vybraným klíčovým slovům. Pokud dosáhne hodnoty 1, znamená to, že jsou vybrána všechna správná klíčová slova. „Schody“ v grafu jsou místa, kde se vyskytlo správně nalezené klíčové slovo (true positive).

4.2.2 Přesnost

Přesnost je metrika odpovídající na otázku *„Jaký je podíl správně nalezených klíčových slov ve výsledku?“* a její vyhodnocení je zobrazeno v grafech 4.3. Na ose x je, stejně jako v případě pokrytí, zobrazen počet klíčových slov, které byly zařazeny do výpočtu přesnosti. Na ose y je přesnost odpovídající vybraným klíčovým slovům. Pokud by přesnost dosáhla hodnoty 1, znamenalo by to, že všechna vybraná klíčová slova odpovídají některým slovům z referenční množiny. Pokud je hodnota 0, znamená to, že se ani jedno z navržených klíčových slov nenalézá v referenční množině. „Zuby“ v grafu jsou místa, kde se vyskytlo správně nalezené klíčové slovo (true positive).

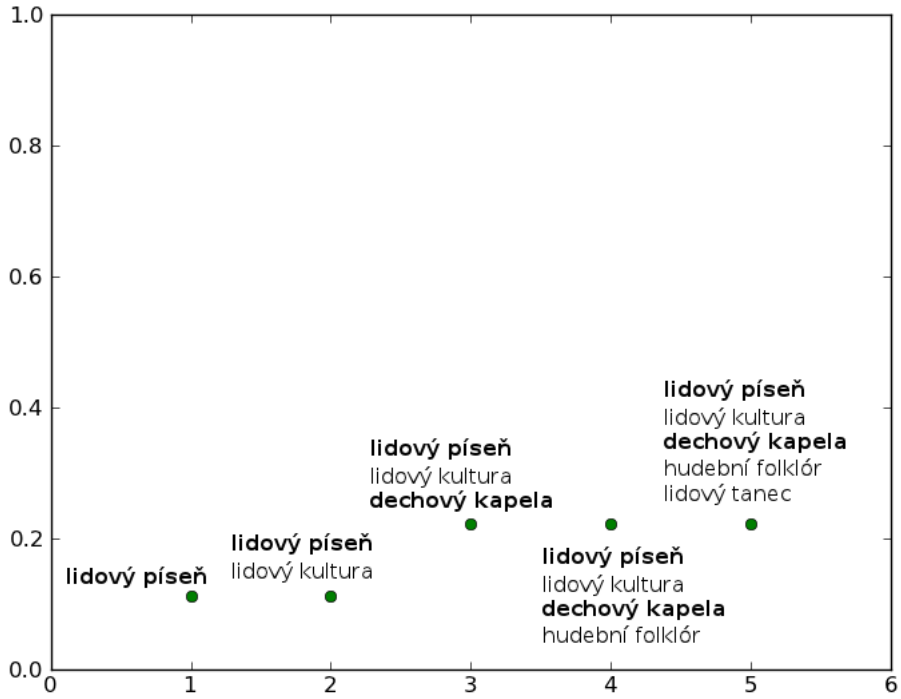
4.2.3 F-measure

Pro výpočet F-measure je třeba určit, jak je důležitá přesnost a pokrytí. Protože ale tato informace není známá, je oběma metrikám dána stejná váha. Jde tedy o harmonický průměr mezi oběma metrikami a je použit zjednodušený výpočet uvedený v 2.24. Vyhodnocení F-measure pro jeden dokument je zobrazeno v grafech 4.4. Na ose x je, stejně jako v předchozích případech, zobrazen počet klíčových slov, které byly zařazeny do výpočtu přesnosti. Na ose y je hodnota F-measure odpovídající vybraným klíčovým slovům. Pokud je hodnota 0, znamená to, že přesnost či pokrytí dosahuje také hodnoty 0. Pokud by hodnota byla 1, znamenalo by to optimální stav, kdy systémem vybraná klíčová slova zcela odpovídají referenčním klíčovým slovům.

4.2.4 Souhrnný graf

Všechny tři výše uvedené metriky lze přehledně zanést do souhrnného grafu 4.5. Pořád platí, že na ose x je zobrazen počet klíčových slov, které byly zařazeny do výpočtu přesnosti. Na ose y je zobrazena hodnota metriky odpovídající vybraným klíčovým slovům. Čím je hodnota vyšší, tím je výsledek lepší.

V dalších částech práce bude používán většinou graf zobrazující hodnoty pro prvních 100 klíčových slov. Volba zobrazovat pouze prvních 100 výsledků je dána především tím, že přesnost se ve vyšších hodnotách x nijak zajímavě nemění, a to má přímý dopad na F-measure, která se chová podobně.



Obrázek 4.1: Pokrytí pro skupiny klíčových slov

4.3 Navržený evaluátor

Jde o nástroj pro automatické vyhodnocování navržených klíčových slov, který vznikl v rámci této práce. Je navržen pro těsnou spolupráci se systémem automatického navrhování klíčových slov. Bez potřeby dalších úprav zpracovává výstup z vytvořeného systému.

Krom přesného shody klíčového slova s referenčním také zvládne porovnat lemma klíčového slova, a to i bez ohledu na výsledné pořadí slov v klíčovém slově.

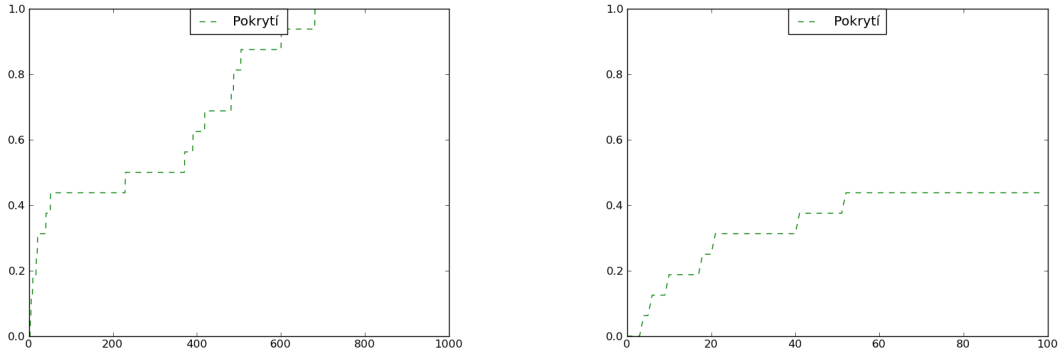
4.3.1 Referenční klíčová slova

Je třeba znát referenční klíčová slova, oproti kterým bude probíhat vyhodnocení. Dále je třeba znát lemmata referenčních klíčových slov, protože evaluátor není nijak propojený s PDT či jiným nástrojem na morfologickou analýzu.

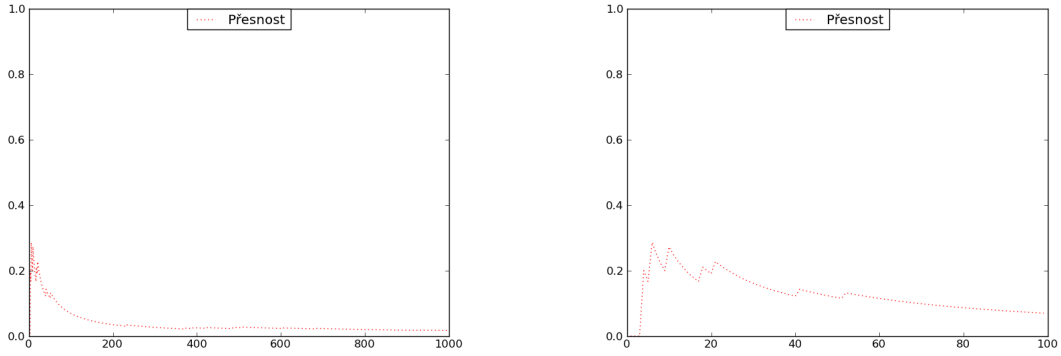
4.3.2 Porovnání různých systémů

Při vyhodnocování systému je velice vhodné mít jedinou hodnotu, která určí úspěšnost systému. Tuto hodnotu bude představovat maximum hodnot *F-measure*. V grafech 4.5, tak je maximální hodnota F-measure *0.263* a této hodnoty se dosahuje při použití *21* klíčových slov.

Pokud budou různé systémy posuzovány pouze podle této jedné hodnoty, bude prostým porovnáním ihned jasné, který systém je úspěšnější. Bohužel tato jedna hodnota zcela ztrácí



Obrázek 4.2: Příklad vyhodnocení pokrytí na jednom dokumentu pro prvních 1000 klíčových slov a detail pro prvních 100 klíčových slov



Obrázek 4.3: Příklad vyhodnocení přesnosti na jednom dokumentu pro prvních 1000 klíčových slov a detail pro prvních 100 klíčových slov

informaci o tom, v jaké oblasti je systém lepší oproti druhému, ale to není ani účel této hodnoty.

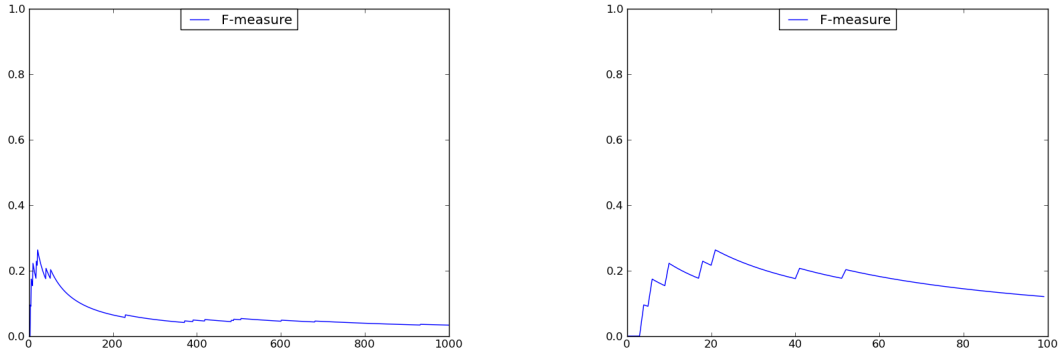
4.3.3 Průběh evaluace

Ve výsledku automatického navrhování klíčových slov je více dokumentů a pro každý dokument je uvedeno více klíčových slov, které jsou seřazeny podle skóre.

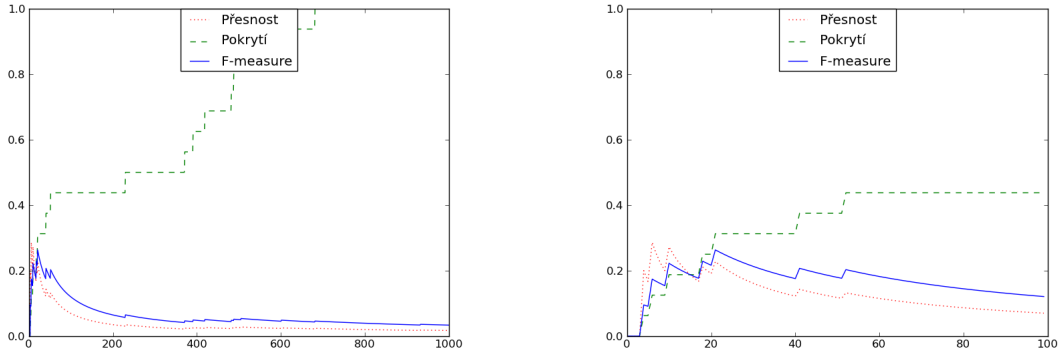
Každé navržené klíčové slovo je porovnáno s referenčními klíčovými slovy a i lemma klíčového slova je porovnáno s referenčním lemma. Při porovnávání se nebere ohled na pořadí slov v rámci klíčového slova.

Pro každý dokument jsou postupně brány skupiny klíčových slov o počtu jedno slovo, dvě slova, atd., stejně jak je uvedeno v kapitole 4.2. V rámci skupiny jsou navržená klíčová slova brána jako rovnocenná a pro každou skupinu je vyhodnocen počet *true positive*, *true negative*, *false positive* a *false negative* a také sledované metriky *přesnost*, *pokrytí* a *F-measure*.

Při agregaci výsledků se vždy použijí všechny vyhodnocované dokumenty a agregují se dohromady skupiny o stejné velikosti. Tím jsou zjištěny průměry sledovaných metrik pokud



Obrázek 4.4: Příklad vyhodnocení F-measure na jednom dokumentu pro prvních 1000 klíčových slov a detail pro prvních 100 klíčových slov



Obrázek 4.5: Příklad vyhodnocení sledovaných metrik na jednom dokumentu pro prvních 1000 klíčových slov a detail pro prvních 100 klíčových slov

by bylo vybráno první klíčové slovo jako výsledek, první dvě klíčová slova, první tři atd.

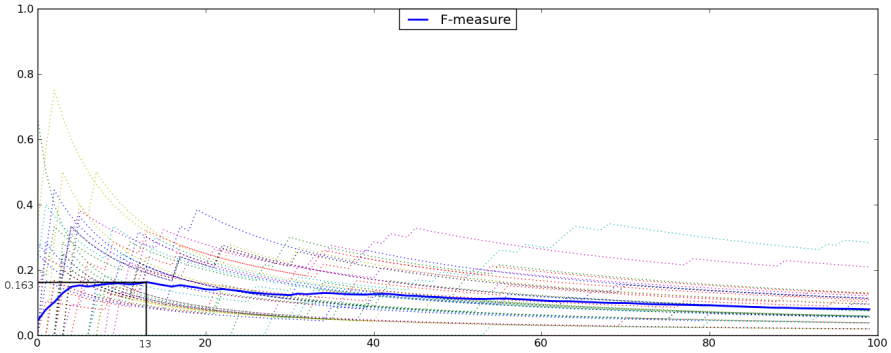
Z agregovaných výsledků jsou vykresleny grafy, je zjištěna maximální hodnota F-measure a také při kolika vybraných klíčových slovech je této hodnoty dosaženo.

V grafu 4.6 jsou zaznamenány výsledky F-measure pro 40 náhodně vybraných dokumentů (tečkovaně) a výsledný průměr F-measure (tučně). Smyslem tohoto grafu je ukázat, že vyhodnocování každého dokumentu zvlášť by bylo velice náročné, kdežto získat informaci z průměrných hodnot je poměrně snadné. Maximální průměrné hodnoty *0.163* je dosaženo při použití *13* klíčových slov.

4.3.4 Implementace

Pro implementaci byl zvolen jazyk Python 2.7, grafy jsou vykreslovány pomocí PyLab.

Soubory obsahující referenční klíčová slova jsou umístěny ve složce `answers`. Pro každý dokument existují dva soubory, jeden obsahující referenční klíčová slova a druhý obsahující jejich lemmata. Soubory mají název odpovídající názvu dokumentu, kterému byla navržena klíčová slova. Například byl-li dokument označkovaný pomocí PDT nazván `10109.csts`, očekává se, že soubory obsahující referenční klíčová slova budou pojmenovány `10109.kw` a



Obrázek 4.6: Vyhodnocení F-measure pro 40 dokumentů

`10109.kw.lemma`. Klíčová slova jsou uvnitř souborů zapsána po jednom na každém řádku. Spuštění evaluátoru je popsáno v příloze F.

Kapitola 5

Zdrojová data

Data, na kterých je systém vyvíjen a testován, jej ovlivňují, a také určují jeho reálné použití. Pokud je systém vyvíjen pro vybírání velkého množství klíčových slov z krátkých textů, například novinových článků, bude se nejspíš chovat nevhodně při vybírání malého množství kandidátních slov z rozsáhlých textů, například knih.

Jinou možností je vytvořit systém univerzální, takový ale nebude předmětem této práce.

5.1 Databáze závěrečných prací MUNI

Databáze závěrečných prací Masarykovy univerzity [15] je veřejná databáze obsahující práce z různých fakult Masarykovy univerzity, za něž byly uděleny většinou tituly Bc., Mgr., Ing. a v menším množství také tituly JUDr., PhDr., RNDr a Ph.D. Protože se jedná o závěrečné práce, dá se očekávat, že půjde o práce dobré kvality a většího rozsahu. Výhodné je, že jde o závěrečné práce z více fakult, systém tedy nebude natrénován pro specifickou doménu dokumentů. Ke každé práci jsou ve zvláštním souboru uvedena klíčová slova, která práci přiřadil její autor.

Z těchto důvodů byla pro vývoj systému vybrána právě databáze závěrečných prací MUNI.

5.2 Postup stahování

Vyhledávání v závěrečných pracích funguje spolehlivě, ale protože MUNI nepodporuje hromadné stahování závěrečných prací, vznikla sada skriptů pro toto stahování. Jde vlastně jen o specifický web crawler (program, který prochází web a stahuje jeho obsah). Tímto způsobem se podařilo stáhnout stránky obsahující detail práce se zadáním, anotací, posudkem a klíčovými slovy. Příklad takové stránky: `http://is.muni.cz/th/359420/fi_b/`.

Na detailu stránky si lze vybrat vždy z formátu *PDF* `.pdf`, textové verze v kódování *UTF-8* `.txt` a občas i z *Microsoft Word* verze `.doc`. Volba padla na přímé stahování textové verze, protože není třeba dalšího zpracování. Textová verze neobsahuje obrázky a podobné netextové informace, ale ty stejně systém pro návrh klíčových slov neanalyzuje.

Název souboru obsahující výslednou práci nemá standardizovaný formát, je zvolen pravděpodobně autorem práce. Proto byly při stahování vzaty v potaz všechny soubory s příponou `.txt` a z nich byl následně vybrán největší. Tato metoda výběru se ukázala jako spolehlivá, a to i přesto, že je vybíráno mezi posudky a přílohami.

Soubor obsahující klíčová slova se vždy jmenoval `keywords.txt`, proto s jeho identifikací nebyl žádný problém.

Tato práce se zabývá pouze navrhováním klíčových slov v češtině, proto musely být vyřazeny závěrečné práce psané v jiných jazycích (nejčastěji slovenština, angličtina, němčina, francouzština, ale podařilo se například najít i závěrečnou práci v řečtině). Na stránce s detailem práce je tato informace uvedena, proto nebylo třeba v textech rozpoznávat jazyk.

5.3 Předzpracování dokumentů

Před samotným uložením dokumentu byl jeho text rozdělen na jednotlivé věty kvůli dalšímu zpracování PDT. Konec věty byl identifikován pomocí tečky s výjimkou následujících případů:

- Před tečkou je číslo (jde pravděpodobně o řadovou číslovku nebo číslo kapitoly)
- Před tečkou je akademický titul
- Před tečkou je zkratka. Jsou detekovány pouze zkratky *str.*, *s.*, *Sb.*

Jistě by šlo použít sofistikovanější způsoby rozdělení, nicméně tento jednoduchý a rychlý způsob se osvědčil a nebylo třeba hledat jiné řešení.

5.4 Úprava klíčových slov

Klíčová slova jsou v souborech `keywords.txt` většinou uvedena nejprve v češtině a poté je uveden jejich anglický překlad. Jsou obvykle oddělena čárkou nebo středníkem. Ovšem tento formát je pravděpodobně pouze doporučený a autoři závěrečných prací jej občas nedodržují.

Původně byla použita automatická metoda, při které byla klíčová slova rozdělena na poloviny a slova z první poloviny byla označena jako česká referenční klíčová slova. Tento způsob se ovšem neosvědčil. Většina autorů uvádí různý počet českých a anglických klíčových slov, někteří je uvádí po párech české a anglické, někdy byla česká a anglická slova úplně promíchána, někteří autoři používali k oddělení klíčových slov pouze mezeru a někdy nebyla anglická klíčová slova vůbec uvedena.

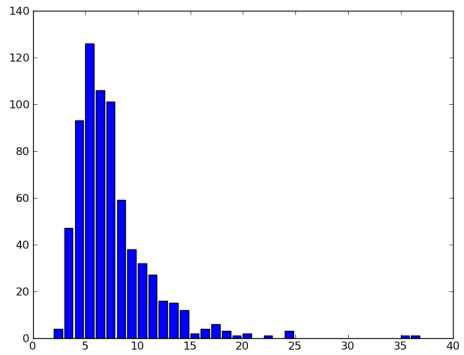
Referenční klíčová slova uvedená autorem práce musela být z těchto důvodů procházena ručně. Takto bylo zpracováno 700 dokumentů.

5.5 Statistiky klíčových slov

Přestože bylo staženo řádově více dokumentů, k dalšímu zpracování bylo použito pouze 700 dokumentů, kterým byla vybrána česká referenční klíčová slova.

- Počet dokumentů: `700`
- Průměrná délka dokumentu ve znacích: `146816`
- Počet českých klíčových slov: `4952`
- Počet jednotlivých slov v klíčových slovech: `9256`
- Průměrný počet klíčových slov v dokumentu: `7.07`

V grafu 5.1 je uvedeno rozložení počtu klíčových slov v rámci dokumentů. Přestože je k dokumentu přiřazeno průměrně přibližně 7 klíčových slov, je jich nejčastěji 3 - 8 se středem v 5.



Obrázek 5.1: Histogram počtu klíčových slov v dokumentech

Systém bude testován na velice dlouhých dokumentech (kolem 150 000 znaků). Úkolem bude nalézt nejčastěji 3-8 klíčových slov. Z tohoto nepoměru vyplývá, že zřejmě půjde o náročný úkol, kdy správné nalezení byť jednoho klíčového slova bude úspěch.

5.6 Výběr trénovacích dat

Systém navrhující klíčová slova [22] je paměťově poměrně náročný. Při pokusu o zpracování všech 700 dokumentů, v použitém vývojovém prostředí, došla dostupná operační paměť. Proto z těchto dokumentů bylo náhodně vybráno pouze 100 dokumentů, jejichž zpracování použité vývojové prostředí zvládne. Při výběru bylo zachováno rozložení počtu klíčových slov tak, jak je uvedeno v grafu 5.1.

Kapitola 6

Navržení kandidátních slov

Z dokumentů jsou nejprve vybrána slovní spojení, která mají naději stát se klíčovými slovy. Tato slovní spojení jsou nazývána kandidátní slova a tato kapitola se zabývá způsobem jejich návrhu.

6.1 Analýza referenčních klíčových slov

Při procesu návrhu kandidátních slov jsou využívána pravidla popsaná v sekci 3.3. Dokument je pomocí těchto pravidel prohledáván a když je nalezena shoda, je skupina slov odpovídající pravidlu vybrána jako kandidátní slovo.

Aby pravidla pro tvorbu klíčových slov odpovídala datům a ne pouze představám o jejich správnosti, byla pravidla automaticky vytvořena z referenčních klíčových slov. Pro vytvoření pravidel jsou využita referenční kandidátní slova z kapitoly 5, je využito všech klíčových slov, které byly přiřazeny 700 dokumentům. Klíčová slova byla zpracována nástrojem *PDT*, a postupně byly stromy vět a mluvnické kategorie zobecňovány až do té míry, že mohly tvořit obecná pravidla.

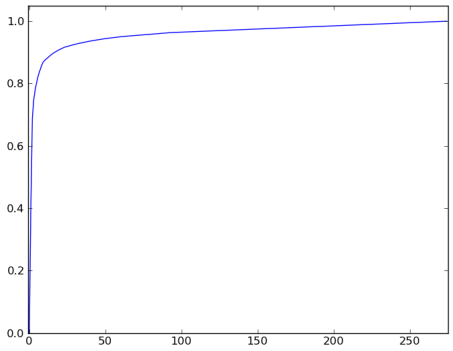
Z údajů poskytovaných *PDT* je možné do pravidel zařadit tvar stromu, všechny mluvnické kategorie a funkci ve větě. Čím více informací je zahrnuto, tím více specifických pravidel je vytvořeno. Na druhou stranu, pokud by do pravidla zařazen pouze tvar stromu, vzniklo by pouze několik, avšak velice obecných pravidel.

Je vhodné zvolit kompromis mezi počtem pravidel a jejich obecností. Pravidla jsou proto tvořena z tvaru větného stromu a slovního druhu jednotlivých slov, ostatní mluvnické kategorie a funkce ve větě jsou ignorovány. Tímto bylo vytvořeno 274 pravidel. Je to mnoho, ale pravidla pokrývají všechna referenční klíčová slova. Každé pravidlo pokrývá určitou část referenčních klíčových slov, některá jsou obecná, jiná jsou velice konkrétní a jsou vytvořena pouze pro jedno klíčové slovo. V tabulce 6.1 jsou uvedeny příklady klíčových slov a k nim vytvořených pravidel.

Klíčové slovo	**Pravidlo**
byrokracie	`N`
bytová situace	`N(A)`
SNMP proxy	`N(N)`
časově rozlišená optická emisní spektroskopie	`N(AAA(D))`

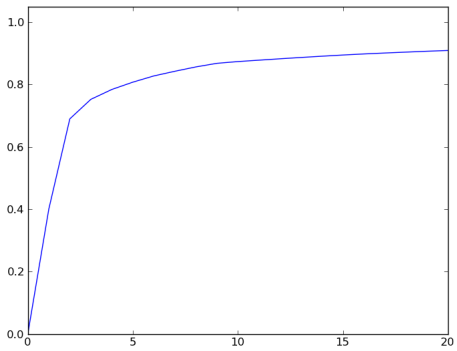
Tabulka 6.1: Příklady klíčových slov a k nim vytvořených pravidel

První tři pravidla budou nejspíš vyhovovat velkému množství klíčových slov, zato poslední pravidlo `N(AAA(D))` bude nejspíš vyhovovat pouze tomuto klíčovému slovu. Proto byl vytvořen graf pokrytí 6.1. Je v něm vyznačena závislost pokrytí referenčních klíčových slov na počtu vzorů.



Obrázek 6.1: Závislost pokrytí na počtu vzorů

Ukazuje se, že malé množství vzorů pokrývá většinu klíčových slov. Od určitého bodu roste již pokrytí takřka lineárně a je třeba velkého množství vzorů byť jen k malému zvýšení pokrytí. Do 20 vzorů každý vzor přidává 10 nebo více klíčových slov, od 21. vzoru už je to 9 nebo méně. Proto byla vybrána hranice 20 vzorů, která zajišťuje pokrytí 93.2 %. Tímto je zavedena do výsledků chyba, 6.8 % klíčových slov nebude nalezeno, protože nebudou ani navržena.



Obrázek 6.2: Detail závislosti pokrytí na počtu vzorů

V grafu 6.2 je znázorněn detail závislosti pokrytí na počtu vybraných vzorů. Nejvíce přispívají k pokrytí první 2 vzory, a to osamocené podstatné jméno, např. *právo*, a přídavné jméno závisející na podstatném jméně, např. *komunikační schopnost*. Další vzory již tak velký přínos nemají, avšak jejich přínos je stále jistý a navíc jsou díky nim výsledky obohaceny o zajímavá, dlouhá, klíčová slova.

Vzor	Příklad klíčového slova	Pokrytí [%]
N	1-dithioláty	40.06
N(A)	řečová výchova	28.94
N(N)	ředitelé škol	9.43
N(AA)	akutní lymfoblastická leukémie	2.34
N(N(A))	rekodifikace trestního řádu	2
N(AN)	rozvojové cíle tisíciletí	1.49
N(R(N))	rezistence k antibiotikům	1.39
A	rozšiřitelný	1.15
N(NN)	básník Zdeněk Rotrekl	0.65
N(R(N(A)))	referendum o Lisabonské smlouvě	0.57
N(AN(A))	klíčový faktor úspěšnosti reintegrace	0.44
N(N(N))	stupeň závislosti osoby	0.42
?(NN)	Jan Komenský	0.4
C	1945	0.28
N(A(D))	individuálně přivlastňovací adjektivum	0.28
N(R(N(N)))	právo na ochranu osobnosti	0.24
N(J(AA))	etnická nebo jiná skupina	0.24
N(C)	první třída	0.24
N(AR(N))	rámcový program pro vzdělávání	0.22
N(NR(N))	hodnocení účinku na žáky	0.22

Tabulka 6.2: Přehled výsledných vzorů

6.2 Výsledné vzory

V tabulce 6.2 je přehled výsledných vzorů. N je podstatné jméno, A je přídavné jméno, C je číslovka, V je sloveso, D je příslovce a R je předložka a J je spojka. Krom zájmen, částic a citoslovcí se v klíčových slovech vyskytují všechny slovní druhy. V pravidlech je brán ohled pouze slovní druh, protože se klíčové slovo může nacházet ve větách v jakémkoliv tvaru, případně i negovaně. Dokonce není podstatný ani slovní poddruh, který by případně mohl i uškodit. Například přivlastňovací přídavné jméno má vlastní poddruh AU a není obecným AA vybráno, proto musí být pravidlem pouze souhrnné A.

6.3 Odhalené chyby

Při používání PDT se vyskytl zajímavý problém. Převádí písmena řecké abecedy na písmena latinské abecedy. Například pro klíčové slovo *β-glukuronidáza* je přiřazeno lemma *b-glukuronidáza*. Protože se jedná o zcela minoritní problém, pouze 2 z 4952 klíčových slov obsahují řecké písmeno, padlo rozhodnutí jej neřešit.

V originální implementaci je mezi stop slova zařazen i regulární výraz . (znak tečky). To znamená, že jakýkoliv jednopísmenný výraz je vyřazen, to ovšem eliminuje, krom jiného, i jednopísmenné předložky a spojky. Chyba byla objevena při testování klíčového slova „*referendum o Lisabonské smlouvě*“ a původně vypadala, jakoby měl systém problém se zpracováním delších pravidel. Při debuggování bylo ovšem odhaleno, že zpracování pravidel pracuje správně, ale slovo „*o*“ bylo vybráno jako stop slovo, tím bylo prohledávání části

věty zastaveno a klíčové slovo nemohlo být nalezeno.

6.4 Srovnání s originální implementací

V originální implementaci [22] je využito 22 pravidel pokrývajících 87.7 % referenčních klíčových slov, pravidla jsou vytvořena maximálně pro tříslovná klíčová slova.

V implementaci vytvořené v rámci této práce je využito 20 pravidel pokrývajících 93.2 % referenčních klíčových slov. Jsou zařazena i pravidla pro čtyřslovná klíčová slova pokrývající dohromady 2.13 % referenčních klíčových slov.

Kapitola 7

Ohodnocení kandidátních slov

V této kapitole budou zkoumány statistické algoritmy pro ohodnocování kandidátních slov z kapitoly 2.3 z praktického hlediska. Každý algoritmus bude implementován, otestován na 100 dokumentech a výsledky návrhu klíčových slov budou porovnány s referenčními klíčovými slovy pomocí evaluátoru z kapitoly 4.3. V této kapitole budou porovnávána pouze lemma, a to bez ohledu na pořadí slov.

7.1 Originální počítání výsledného skóre

Implementace v originálním systému [22] se lehce liší od zpracované dokumentace, ale tato odlišnost má zásadní dopad na výsledky. Počet výskytů kandidátního slova je počítán pro celý korpus a ne pro jednotlivé dokumenty.

Pokud se kandidátní slovo vyskytuje ve více dokumentech, je jeho skóre ve všech dokumentech stejné. Vyskytuje-li se kandidátní slovo v jednom dokumentu velice často, dostane správně vysoké ohodnocení a je označeno jako klíčové. V ostatních dokumentech, kde se vyskytuje zcela náhodně, dostane úplně stejné ohodnocení a bude opět, tentokrát chybně, označeno jako klíčové.

Autor originální implementace si nemusel této vlastnosti všimnout, protože při testování používal texty norem. Ke každé normě bylo přiřazeno přes 200 referenčních klíčových slov a referenční klíčová slova se mohla překrývat. Navíc mohla být použita jiné metodika vyhodnocování systému. Protože jde pouze o dohady, netroufnu si označit odlišný způsob počítání za chybu, ale budu ji považovat za vlastnost systému.

Zajímavostí je, že pokud je původní systém pouštěn na každém dokumentu zvlášť, jeho výsledky jsou smysluplné, avšak selhávají algoritmy, které porovnávají výskyt slova v dokumentu a v celém korpusu.

Část systému, která je zodpovědná za spouštění samotných statistických algoritmů byla kompletně přepsána tak, aby byla pro každý dokument vytvořena nezávislá sada výsledných klíčových slov. Pro každý dokument jsou zvlášť vyhodnocena kandidátní slova, takže jejich skóre není ovlivněno výskytem v jiných dokumentech. Navíc musely být v tomto duchu přepsány i implementace všech statistických algoritmů.

7.2 Odhalené chyby implementace

Při zkoumání současného systému a také v situacích, kdy systém dával zvláštní výsledky, byly objeveny některé chyby, a tyto chyby byly také opraveny.

Nebyla správně počítána průměrná délka dokumentu, tato chyba se projevila při počítání výsledků metody Okapi BM25.

Při počítání RIDF bylo otočeno znaménko v části, kde se porovnává IDF a frekvence předpovězená Poissonovým rozložením. V rovnici 2.10 autor použil místo znaménka + znaménko -, jak v dokumentaci, tak v implementaci. Díky této záměně jsou výsledky originální implementace RIDF nepoužitelné.

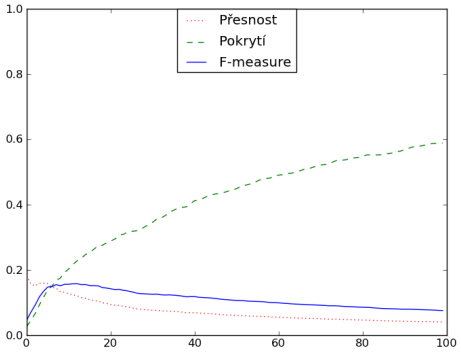
7.3 Implementace statistických algoritmů

Každý algoritmus byl implementován a systém byl poté spuštěn pouze pro tento algoritmus. Navržená klíčová slova jsou porovnána s referenčními klíčovými slovy a pro každý algoritmus je takto na trénovacích datech zjištěna úspěšnost.

Jako metrika pro výsledky a srovnávání je brána F-measure, pokud není upřesněno jinak.

7.3.1 Term frequency

Jedná se o nejjednodušší algoritmus, který počítá jen frekvenci kandidátního slova v dokumentu.



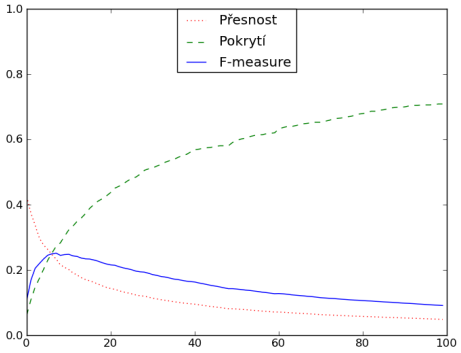
Obrázek 7.1: Úspěšnost implementace term frequency

Je zajímavé, že i takto jednoduchý výpočet vede k nějakým výsledkům, které jsou zobrazeny v grafu 7.1. Nejvyšší hodnoty 0.158 dosahuje algoritmus při 13 vybraných klíčových slovech. Přesnost je přibližně konstantní od jednoho po pět vybraných klíčových slov, pak postupně klesá.

7.3.2 Term frequency - inverse document frequency

Jde o jednoduchou kombinaci základního term frequency a inverse document frequency. Zvýhodňuje kandidátní slova, která se vyskytují v malém počtu dokumentů.

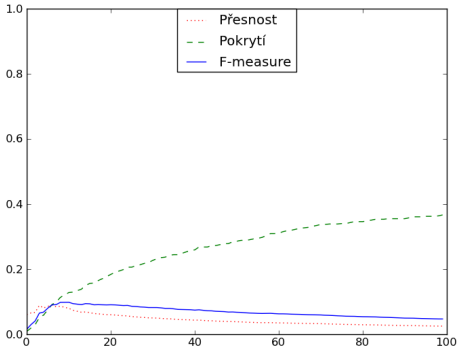
Jednoduchá myšlenka algoritmu ukázala v praxi svůj potenciál, výsledky jsou zobrazeny v grafu 7.2. Dosahuje hodnoty 0.252 při použití 8 klíčových slov. Přesnost dosahuje při použití jednoho kandidátního slova hodnoty vyšší než 0.4. Bohužel přesnost rychle klesá a tím klesá i maximální možné F-measure. Proti přesnosti ale velice rychle roste pokrytí. Tento algoritmus prokázal svou robustnost.



Obrázek 7.2: Úspěšnost implementace term frequency - inverse document frequency

7.3.3 Term frequency - inverse paragraph frequency

Jde o implementaci pokusu měřit inverse frequency v odstavcích místo dokumentů. Předpokladem pro úspěšnost tohoto algoritmu je správné rozdělení dokumentů do odstavců.



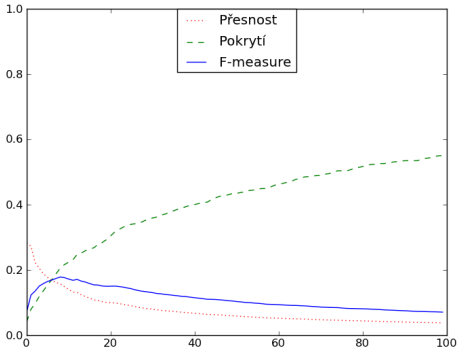
Obrázek 7.3: Úspěšnost implementace term frequency - inverse paragraph frequency

Jak je vidět v grafu 7.3, výsledky jsou velice špatné. Maximální hodnoty 0.098 dosahuje při použití 11 klíčových slov. Jde o horší výsledek než při použití samotného term frequency. Ani jedna metrika nemá lepší průběh než samotné term frequency. Na vině je možná nevhodné rozdělení do odstavců. Tato metoda jistě nebude zahrnuta do výsledného systému.

7.3.4 Residual inverse document frequency

Další metoda využívající inverse document frequency. Metoda porovnává předpovězený výskyt kandidátního slova oproti skutečnému.

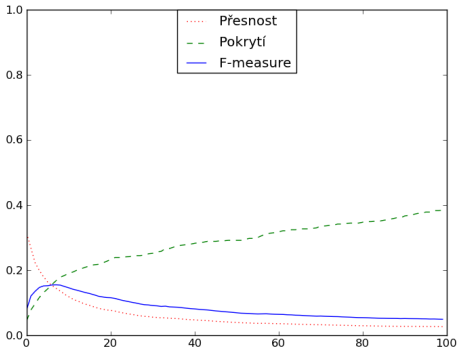
Tato, do značné míry již sofistikovanější metoda, dosahuje výsledku 0.178 při použití 9 klíčových slov, jak zle vyčíst z grafu 7.4. Maximální přesnost je téměř 0.3, ovšem stejně jako v případě TF-IDF také rychle klesá. Oproti tomu pokrytí nestoupá tak rychle a je zajímavé, že je pokrytí při použití 100 klíčových slov nižší než u algoritmu term frequency. Tato metoda nenaplnila očekávání, která do ní byla vložena, ovšem její výsledky jsou použitelné.



Obrázek 7.4: Úspěšnost implementace residual inverse document frequency

7.3.5 Okapi BM25

Algoritmus, který byl vyvinut původně pro vyhledávače a poprvé byl zveřejněn ve stejnojmenném frameworku.



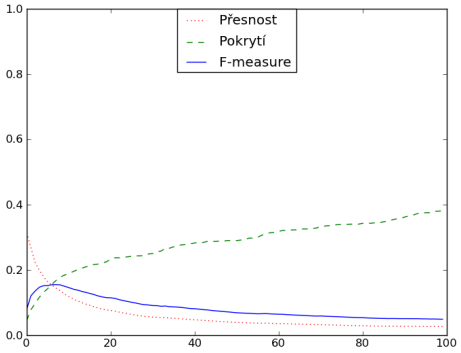
Obrázek 7.5: Úspěšnost implementace Okapi BM25

V grafu 7.5 jsou uvedeny výsledky, maximální hodnoty 0.155 dosahuje při použití 8 klíčových slov, ve výsledném hodnocení je dokonce lehce horší než term frequency. Maximální přesnost je lehce vyšší než 0.3, celkové skóre by mohlo být vzhledem k vyšší přesnosti také vyšší, ale problém je s pokrytím. Ani při použití prvních 100 klíčových slov nedosahuje pokrytí 0.4, kdežto term frequency dosahuje při 100 klíčových slovech téměř hodnoty 0.6.

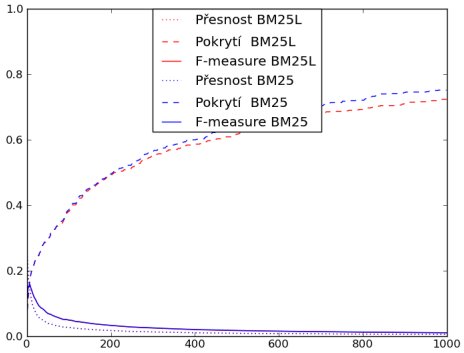
7.3.6 BM25L

Jedná se o adaptaci Okapi BM25 pro dlouhé dokumenty a dlouhými dokumenty se zabývá tato práce. Dlouhé dokumenty jsou základním Okapi BM25 penalizovány a tento algoritmus by měl tuto vlastnost odstranit.

Maximální hodnoty 0.155 dosahuje při použití 8 klíčových slov, což se nápadně podobá výsledkům Okapi BM25. Při pohledu na výsledky v grafu 7.6 je vidět, že dokonce i průběhy metrik jsou zcela shodné. Nejde však o záměnu či chybu, výsledná klíčová slova jsou shodná i s pořadím a začínají se lišit až při výběru více než 100 klíčových slov.



Obrázek 7.6: Úspěšnost implementace BM25L



Obrázek 7.7: Porovnání Okapi BM25 a BM25L pro 1000 klíčových slov

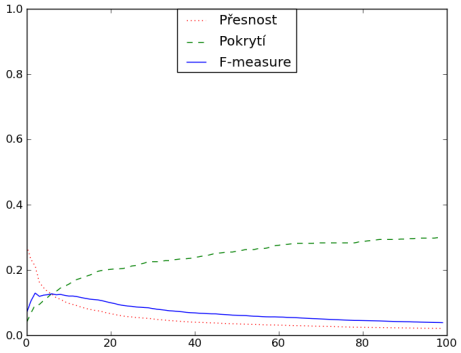
Srovnáním výsledků algoritmů Okapi BM25 a BM25L, které je uvedeno v grafu 7.7 je vidět, že vylepšený algoritmus BM25L dokonce dosahuje při vyšším počtu klíčových slov horších výsleků než jeho originální varianta. Protože jsou výsledná klíčová slova stejná a navíc originální algoritmus dosahuje lepších výsledků, adaptovaný algoritmus BM25L nebude ve výsledném systému použit.

7.3.7 Lexical cohesion

První algoritmus, který zohledňuje víceslovné termíny. Jeho výsledek určuje, jak moc spolu slova v rámci klíčového slova souvisejí.

Maximální hodnoty 0.129 dosahuje tento algoritmus při použití 3 klíčových slov, jak lze vyčíst z výsledného grafu 7.8. Speciální vlastností tohoto algoritmu je, že jako výsledná klíčová slova navrhuje pouze víceslovné výrazy. Maximální přesnost je 0.26, což je v porovnání s celkovým výsledkem velmi pozitivní. Pokrytí je velice špatné, ale to je daň za navrhování pouze víceslovných termínů. V maximální hodnotě dosahuje těsně nad 0.3, což je oproti term frequency hodnota takřka poloviční.

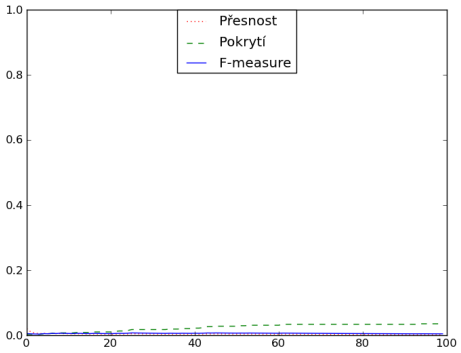
Přestože jsou výsledné hodnoty velice slabé, tento algoritmus má potenciál v tom, že vybírá pouze víceslovné výrazy a je možné, že ve výsledném systému bude mít své místo.



Obrázek 7.8: Úspěšnost implementace lexical cohesion

7.3.8 Weirdness

Myšlenkou tohoto algoritmu je, že frekvence klíčového slova v dokumentu je vyšší než frekvence klíčového slova v obecném korpusu. Protože není k dispozici obecný korpus, bude za něj považován korpus dokumentů, kterým jsou v jenom běhu systému navrhována klíčová slova.



Obrázek 7.9: Úspěšnost implementace weirdness

Maximální hodnoty 0.007 dosahuje tento algoritmus při použití 26 klíčových slov, jak lze vyčíst z grafu 7.9. Jde bezesporu o nejhorší výsledek z celého zkoumání.

Přestože jsou v literatuře výsledky při použití tohoto algoritmu vyhovující, v této práci se zcela neosvědčil. Je to možná proto, že není použit opravdový obecný korpus, ale jen korpus o 100 dokumentech, které jsou k dispozici.

7.3.9 Modifikace weirdness

Protože je myšlenka weirdness jednoduchá a zajímavá, byl proveden pokus s obměnou této myšlenky. V čitateli celkového weirdness zlomku je frekvence klíčového slova v dokumentu (jde vlastně o term frequency), ve jmenovateli je frekvence klíčového slova v obecném korpusu (corpus frequency). Poměrem těchto frekvencí získáme výsledné skóre. Ovšem pokus na reálných datech ukázal velmi slabé výsledky. Jak se ale ukázalo dříve, samotné term

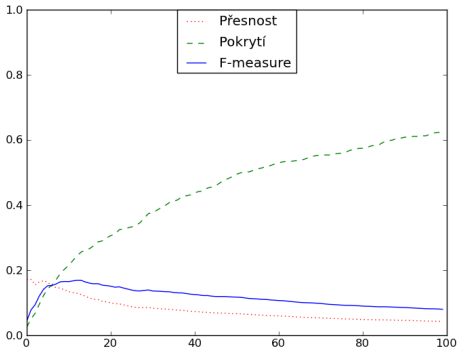
frequency dává výsledky slušné. Ke zhoršení výsledků term frequency došlo porovnáním s corpus frequency pomocí dělení. Byl tedy proveden pokus, kdy se term frequency a corpus frequency porovnávaly pomocí odečtení, jak je uvedeno v rovnici 7.1.

$$\text{Modifikace weirdness}(t) = fs(t) - fg(t), \tag{7.1}$$

kde t je ohodnocovaný termín, *fs(t)* je frekvence termínu v dokumentu a *fg(t)* je frekvence termínu v obecném korpusu.

Důsledky odečtení:

- Pokud je frekvence výskytu kandidátního slova v dokumentu nižší než v celém korpusu, dostává záporné skóre
- Pokud je frekvence výskytu kandidátního slova v dokumentu stejná jako v celém korpusu, dostává nulové skóre
- Pokud je frekvence výskytu kandidátního slova v dokumentu vyšší než v celém korpusu, dostává skóre vyšší než 0.



Obrázek 7.10: Úspěšnost implementace modifikace weirdness

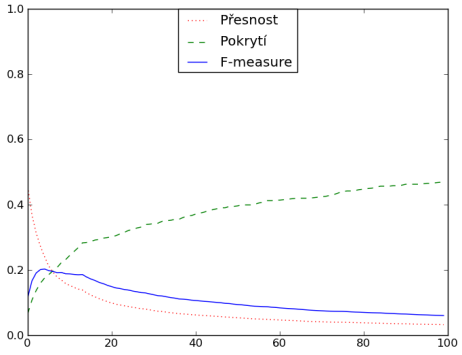
V grafu 7.10 jsou výsledky pro tento algoritmus. Term frequency dosahuje v maximu hodnoty 0.158; modifikovaná verze weirdness dosahuje v maximu hodnoty 0.169, obě při použití 13 klíčových slov. Také pokrytí při použití 100 klíčových slov je lehce vyšší oproti term frequency.

Pokud by ve výsledném systému měl figurovat základní algoritmus term frequency, bylo by lepší jej nahradit tímto algoritmem upravené verze weirdness.

7.3.10 C-value

Algoritmus navržený pro víceslovné termíny. Zcela ignoruje jednoslovné termíny a navíc penalizuje ty, které se vyskytnou jako součást delšího termínu.

Jak lze vyčíst z grafu 7.11, maximální hodnoty 0.203 dosahuje při použití 5 klíčových slov. Maximální přesnosti dosahuje v hodnotě 0.45, to je nejvyšší dosažený výsledek. Zároveň jako klíčová slova navrhuje dvojslovné, ale často i delší termíny. Při použití 100 klíčových slov dosahuje přesnosti 0.47, vzhledem k tomu, že nevybírá jednoslovná klíčová slova, je to vynikající výsledek.



Obrázek 7.11: Úspěšnost implementace C-vallue

Tento algoritmus nebyl kvůli své časové náročnosti v originální implementaci použit, což je chyba. Dodává velké množství kvalitních víceslovných výrazů a ve výsledném systému musí mít své místo.

7.4 Kombinace algoritmů

Pro kombinaci algoritmů je využit postup z kapitoly 2.3.12. Pro každý algoritmus jsou pro každý dokument spočítány výsledky všem kandidátním slovům. Kandidátní slova jsou seřazena podle skóre algoritmu, jako výstupní skóre je brána převrácená hodnota pořadí ve výsledku.

7.4.1 Použité algoritmy

Do výsledného systému nejsou zařazeny algoritmy, jejichž myšlenku využívá úspěšnější algoritmus nebo jsou správně navržená klíčová slova obsažena v rámci jiného algoritmu.

- *Term frequency*, základní algoritmus je obsažený v *TF-IDF*, proto není použit.
- *TF-IPF*, obměna *TF-IDF*, zcela se neosvědčil a oproti *TF-IDF* nepřináší žádné použitelné výsledky.
- *BM25L*, vylepšení algoritmu *Okapi BM25*, které se v praxi ukázalo jako mírně horší než originál.
- *Lexical cohesion*, algortmus se zajímavou myšlenkou, ovšem výsledky překrývá algoritmus *C-value*.
- *Weirdness*, algoritmus, který se zcela neosvědči.
- *Modifikace weirdness*, protože nepřináší oproti *term frequency* výrazně lepší výsledky a oproti *TF-IDF* žádné lepší výsledky, není použit.

K dalšímu zpracování jsou zařazeny algoritmy *TF-IDF*, *RIDF*, *Okapi BM25* a *C-value*.

7.4.2 Odhad vah

Při odhadování vah jednotlivým algoritmům jsou spolu porovnávány vždy 2 algoritmy, jednomu roste váha od 0 k 1, druhému naopak klesá od 1 k 0. Optimální váha je v místě, kde dává kombinace těchto 2 algoritmů maximální výsledek. Tímto způsobem byly zjištěny optimální kombinace pro všechny 4 algoritmy.

V dalším kroku byly tyto výsledky ještě jednou kombinovány, a to tak, aby se ve výsledky objevily všechny 4 algoritmy. Průběh byl úplně stejný jako v předchozím koku, jen se místo výsledku jednoho algoritmu bral agregovaný výsledek.

Tímto způsobem byly nalezeny váhy jednotlivým algoritmům, které dávají nejvyšší souhrnný výsledek, a to *F-measure*. Tyto váhy jsou uvedeny v tabulce 7.1.

Algoritmus	**Váha**
TF-IDF	0.546
RIDF	0.073
Okapi BM25	0.007
C-value	0.364

Tabulka 7.1: Výsledné váhy při použití 4 algoritmů

Protože mají algoritmy *Okapi BM25* a *RIDF* velice malou váhu, je proveden ještě pokus tyto 2 algoritmy úplně vynechat a jsou použity pouze *TF-IDF* a *C-value*. Přiřazení váhy je provedeno stejně jako dříve, jednomu algoritmu roste váha od 0 k 1, druhému klesá od 1 k 0. Výsledné váhování pro tyto 2 algoritmy jsou uvedeny v tabulce 7.2.

Algoritmus	**Váha**
TF-IDF	0.6
C-value	0.4

Tabulka 7.2: Výsledné váhy při použití 2 algoritmů

Je zajímavé, že váhy vyšly přesně na *0.6* a *0.4*, proto bylo váhování ověřeno spuštěním ještě pro 100 hodnot rovnoměrně rozložených mezi 0.39 a 0.41 (a k nim odpovídající 0.59 a 0.61) a nalezené nastavení vah bylo potvrzeno.

7.5 Výsledná konfigurace algoritmů

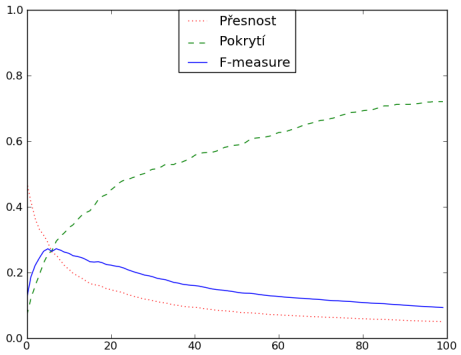
Jak je vidět v tabulce 7.3, jsou rozdíly ve výsledcích při použití 2 a 4 algoritmů opravdu minimální. Maximální F-measure je dokonce o trošku vyšší při použití pouze 2 algoritmů, avšak toto maximum je až při použití 8 klíčových slov. Při použití 6 klíčových slov jsou výsledky F-measure shodné.

Princip Occamovy břitvy říká, že jednodušší je lepší než složitější. Vzhledem k minimálním rozdílům ve výsledcích a také vzhledem k tomuto principu jsou do výsledného systému použity pouze 2 algoritmy, a to *TF-IDF* s váhou 0.6 a *C-value* s váhou 0.4.

V grafu 7.12 jsou zobrazeny sledované metriky pro finální nastavení systému. Stejného zaokrouhleného maxima metriky F-measure, a to 0.272, dosahuje systém při použití 6 nebo 8 klíčových slov. Nejvyšší přesnosti 47.47 % dosahuje systém při použití jednoho klíčového

Metrika	2 algoritmy	4 algoritmy
Přesnost při 1 slově	0.47474	0.47474
Pokrytí při 100 slovech	0.72013	0.72164
Maximum F-measure	0.27192	0.27179
F-measure při 6 slovech	0.27179	0.27179

Tabulka 7.3: Sledované metriky pro různý počet použitých algoritmů



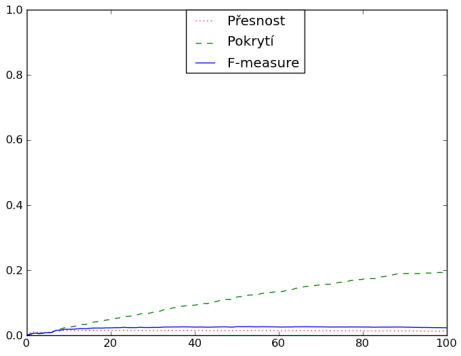
Obrázek 7.12: Vyhodnocení metrik pro výsledný systém

slova, při použití 100 klíčových slov dosahuje pokrytí 72.01 % a při použití 1000 klíčových slov (není zaznamenáno v grafu) dosahuje systém pokrytí 96.47 %.

Hodnota pokrytí při použití 1000 klíčových slov, 96.47 %, znamená, že byla nalezena téměř všechna referenční klíčová slova, jen se neumístila na čelních pozicích.

7.6 Porovnání s originálním systémem

Původní systém Tomáše Strachoty byl vyvíjen na českých a anglických normách a anglických článcích, zato systém vyvíjený v rámci této práce se soustřeďuje na závěrečné práce v češtině. Přesto by mělo jít tyto systémy porovnat.



Obrázek 7.13: Vyhodnocení metrik pro původní systém

Při vyhodnocování původního systému se vyskytl problém, výsledky původní implementace jsou velice slabé, vyhodnocení metrik pro 100 dokumentů je uvedeno v grafu 7.13.

Problém je ve způsobu počítání výsledného skóre klíčových slov. V původní implementaci je skóre počítáno pro celý korpus. To v praxi znamená, že úspěšně nalezené klíčové slovo v jednom dokumentu dostane vysoké ohodnocení a dostane se správně na první místo mezi navrženými klíčovými slovy. V jiném dokumentu, kde se stejné kandidátní slovo nalézá zcela náhodně a mělo by dostat nízké ohodnocení, dostane stejné vysoké ohodnocení a také se umístí na prvním místě.

Většina navržených klíčových slov je navržena špatně, protože získala dobré ohodnocení z jiných dokumentů. Tomu také odpovídají metriky z grafu 7.13, nejvyšší hodnoty F-measure 0.024 dosahuje systém při použití 51 klíčových slov.

Porovnání originálního systému a systému vytvořeného v rámci této práce se nepovedlo. Při vyhodnocování pomocí metod popsaných v této práci původní systém selhává.

Kapitola 8

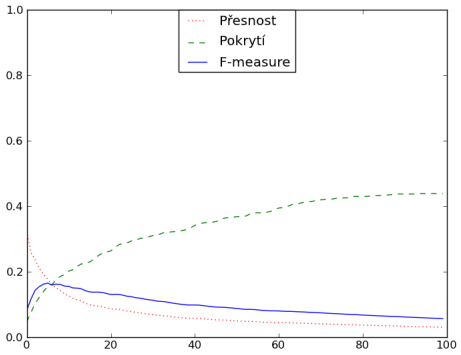
Převod z lemma

Doposud celý systém i jeho vyhodnocení pracovalo pouze s lemmaty klíčových slov. Především u víceslovných klíčových slov je ale vhodné lemmata nepoužívat a raději výsledné klíčové slovo uvést ve skloněném tvaru.

Převod probíhá podle originální implementace tak, jak je popsán v kapitole 3.5.

8.1 Výsledky bez převodu

Nejprve je otestováno, jestli je vůbec převod z lemma vůbec potřebný, k tomu je stále využíván evaluátor z kapitoly 4.

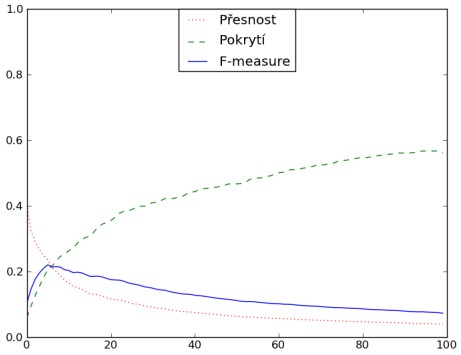


Obrázek 8.1: Vyhodnocení metrik při porovnávání přesného tvaru slov

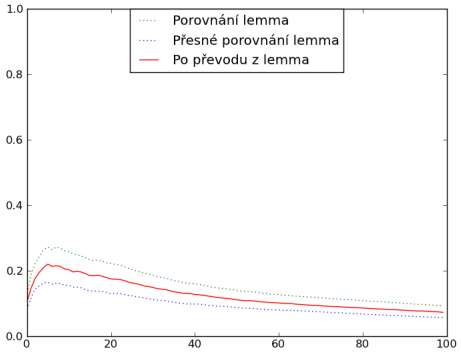
V grafu 8.1 jsou vypočteny sledované metriky při přesném porovnávání tvaru navržených klíčových slov oproti referenčním. Nejvyšší hodnoty 0.165 dosahuje systém při 6 klíčových slovech. Jde o propad úspěšnosti o 40 % oproti výsledky dosaženém porovnávání lemmat. Tímto se jeví, že je převod z lemma potřebný.

8.2 Provedený převod

V grafu 8.2 jsou vypočteny sledované metriky při přesném porovnávání již převedených slov. Nejvyšší hodnoty 0.221 dosahuje systém při použití 6 klíčových slov. Jde jen o propad o 18.75 % oproti porovnávání lemma, což je významný pokrok.



Obrázek 8.2: Vyhodnocení metrik při porovnávání přesného tvaru slov po převodu z lemma



Obrázek 8.3: Porovnání Výsledných F-measure při porovnávání lemmat, při přesném porovnávání a při přesném porovnávání po převodu z lemma

Nakonec jsou všechny výsledky porovnány. V grafu 8.3 jsou porovnány průběhy F-measure při porovnávání lemmat (zeleně), porovnávání lemmat s referenčním tvarem slov (modře) a porovnávání převedených klíčových slov z lemmat s referenčním tvarem klíčových slov.

8.3 Druhy chyb

Při převodu výsledných klíčových slov z lemmat došlo k významnému vylepšení výsledného skóre, avšak skóre stále nedosahuje úspěšnosti základního porovnávání lemmat. Při zkoumání příčiny bylo zjištěno, že výsledné tvary klíčových slov jsou jazykově správné. Důvod zhoršení výsledného skóre je jednoduchý, některé vytvořené tvary neodpovídají přesně referenčním tvarům.

Většina neshod je dána špatně určeným číslem, například klíčové slovo nalezené systémem je „*porucha chování*“, ale referenční termín je „*poruchy chování*“. Pro klíčové slovo je vybráno nejčastěji se vyskytující číslo v dokumentu, ale referenční klíčové slovu tomuto nemusí odpovídat. Tomuto se nelze vyhnout, protože referenční klíčová slova jsou vytvořena čistě autory dokumentů.

Další, méně častou chybou, je chyba v pádu. Systém například určil klíčové slovo jako

„*Hra na kytaru*“, ale odpovídající referenční termín je „*Hra na kytaře*“. Této chybě se, stejně jako předchozí, nedá vyhnout, opět záleží na autorech dokumentů.

8.4 Zhodnocení převodu

Přestože je při striktním vyhodnocování na základě přesné shody systém horší než při porovnávání lemmat, je převod z lemmat velice dobrý. Při ručním procházení navržených klíčových slov a jejich tvarů nebyl nalezen žádný, který by byl jazykově špatný. Naopak bylo potěšující vidět, jak jsou správně převedena i některá komplikovaná slovní klíčová slova, jako například *kyselá důlní voda*, *metoda konečných prvků*, *genderové stereotypy* nebo *staří lidé* (pocházející z lemma *starý člověk*).

Protože převod probíhá opravdu dobře, nebyla část originálního systému provádějící tento převod nijak upravena, je použita čistě originální implementace.

Kapitola 9

Závěr

Cílem této práce bylo vytvořit systém pro automatický návrh klíčových slov. Podkladem pro moji práci byl framework vytvořený Tomášem Strachotou. Abych mohl na jeho práci úspěšně navázat, bylo nutné nejprve opravit některé méně závažné chyby, hlavně byl ale přepsán způsob počítání výsledného skóre. Po této změně je skóre klíčového slova počítáno pro každý dokument zvlášť. Může se to zdát jako záležitost zcela samozřejmá, nicméně v původní implementaci bylo skóre klíčového slova počítáno pro celý korpus a výsledky tím byly hodně poznamenány.

Pro výzkum se podařilo obstarat kvalitní trénovací sadu dokumentů - část databáze závěrečných prací z Masarykovy univerzity. Ke každé závěrečné práci jsou autorem uvedena klíčová slova a oproti těmto referenčním klíčovým slovům jsou porovnávána klíčová slova navržená systémem.

Systém je zaměřený především na závěrečné práce. Vzory, kterými jsou vybírána kandidátní slova jsou odvozeny od klíčových slov, které autoři uvedli ke svým závěrečným pracím. Zároveň je ke každé závěrečné práci autorem přiřazeno malé množství klíčových slov, a proto i systém navrhuje malé množství, konkrétně 6 klíčových slov.

V rámci této práce byl vytvořen také nástroj pro vyhodnocení úspěšnosti návrhu klíčových slov. Tento nástroj porovnává automaticky navržená klíčová slova s referenčně zadanými klíčovými slovy, vyhodnocuje metriky přesnost, pokrytí a F-measure a vytváří grafy těchto metrik. Tento nástroj výrazně usnadnil vyhodnocování systému a velká část grafů použita v této práci je vygenerována právě tímto nástrojem.

Výsledky systému při použití různých hodnotících algoritmů byly podrobeny detailnímu zkoumání, aby se do výsledné konfigurace dostaly opravdu jenom ty algoritmy, které dávají dobré výsledky.

V celé práci je vyhodnocování prováděno striktně strojově a výsledkem je, že průměrně čtvrtina navržených klíčových slov je navržena správně (shodují se s referenčními klíčovými slovy). Tento výsledek se může zdát slabý, ale je třeba si uvědomit, že jde o porovnávání s klíčovými slovy, které si vybral autor práce. Pokud se ale podíváme na klíčová slova navržená systémem, uvidíme, že často se vyskytující *false positive* je vhodné klíčové slovo, jen si jej autor nevybral. Příklady klíčových slov navržených systémem jsou uvedeny v příloze B.

Jako možné pokračování v této práci vidím prozkoumání jiných typů dokumentů, než závěrečných prací. Očekávám, že se zvolené metody ohodnocování kandidátních slov budou chovat jinak pro krátké dokumenty. Jistě by bylo také zajímavé vidět výsledky navrženého systému pro dokumenty, které mají přiřazeny velké množství klíčových slov.

Při kombinaci více algoritmů jsou výsledky lepší, než při použití kteréhokoliv algoritmu samostatně. Je pravděpodobné, že existuje způsob kombinace, který přinese další

zlepšení. Zkusil bych zkombinovat samotné algoritmy a ne až jejich výsledky. K této myšlence mě dovedl úspěch *TF-IDF*, který kombinuje algoritmy *term frequency* a *inverse document frequency* pouhým násobením a dosahuje nejlepších výsledků.

Literatura

[1] Klíčová slova a jejich využití v Knihovně Jiřího Mahena v Brně. *Inflow: information journal*, ročník 1, č. 10, 2008 [cit. 2013-04-02], ISSN 1802-9736.
URL http://www.inflow.cz/klicova-slova-jejich-vyuziti-v-knihovne-jiriho-mahena-v-brne

[2] Ahmad, K.; Gillam, L.; Tostevin, L.; aj.: University of surrey participation in TREC8: Weirdness indexing for logical document extrapolation and retrieval (wilder). In *The Eighth Text REtrieval Conference (TREC-8)*, 1999.

[3] DATAMIN: Evaluation methods in text categorization. 2011-09-05 [cit. 2013-04-09].
URL http://datamin.ubbcluj.ro/wiki/index.php?title=Evaluation_methods_in_text_categorization&oldid=2019

[4] Fajmon, B.; Růžičková, I.: *Matematika 3*. Fakulta elektrotechniky a komunikačních technologií VUT v Brně.

[5] 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, č. 2, 2000: s. 115–130.

[6] Hajič, J.: Czech "Free"Morphology. 2001 [cit. 2013-04-11].
URL http://ufal.mff.cuni.cz/pdt/Morphology_and_Tagging/Morphology/

[7] Hajič, J.; Hajičová, E.; Hlaváčová, J.; aj.: *Průvodce PDT 2.0*. 2006.

[8] Hajič, J.; Hanová, H.; Hladká, B.; aj.: Manual for Morphological Annotation. Technická Zpráva TR-2005-27, UK MFF CKL, 2005.

[9] Heffner, C. L.: Research Methods. 2004, [cit. 2013-05-12].
URL http://allpsych.com/researchmethods/errors.html

[10] Kowalczyk, M.: konwert - interface for various character encoding conversions. 1998 [cit. 2013-04-10].
URL http://manpages.ubuntu.com/manpages/lucid/man1/konwert.1.html

[11] Linhart, J.; kolektiv: *Slovník cizích slov*. Dialog, 2004, ISBN 80-85843-61-7.

[12] Lv, Y.; Zhai, C.: When documents are very long, BM25 fails! In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '11, New York, NY, USA: ACM, 2011, ISBN 978-1-4503-0757-4, s. 1103–1104, doi:10.1145/2009916.2010070.

[13] Manning, C. D.; Raghavan, P.; Schütze, H.: *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008, ISBN 0521865719, 9780521865715.

[14] Manning, C. D.; Schütze, H.: *Foundations of statistical natural language processing*. Cambridge, MA, USA: MIT Press, 1999, ISBN 0-262-13360-1.

[15] MUNI: Absolventi a závěrečné práce. 2013 [cit. 2013-04-14].
URL http://is.muni.cz/thesis/

[16] Navigli, R.: Word Sense Disambiguation: A Survey. *ACM Computing Surveys*, ročník 41, č. 2, 2009.

[17] Park, Y.; Byrd, R. J.; Boguraev, B. K.: Automatic glossary extraction: beyond terminology identification. In *Proceedings of the 19th international conference on Computational linguistics - Volume 1*, COLING '02, Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, s. 1–7, doi:10.3115/1072228.1072370.

[18] Pérez-Iglesias, J.; Pérez-Agüera, J. R.; Fresno, V.; aj.: Integrating the Probabilistic Models BM25/BM25F into Lucene. *CoRR*, ročník abs/0911.5046, 2009.

[19] Robertson, S.: Understanding inverse document frequency: On theoretical arguments for IDF. *Journal of Documentation*, ročník 60, 2004: str. 2004.

[20] Sasaki, Y.: The truth of the F-measure. *Teach Tutor mater*, 2007: s. 1–5.

[21] Sklenák, V.: *Data, informace, znalosti a Internet*. C.H. Beck pro praxi, C.H. Beck, 2001, ISBN 9788071794097.

[22] Strachota, T.: *Automatické navrhování klíčových slov*. diplomová práce, FIT VUT v Brně, Brno, 2010.

[23] Zhang, Z.; Iria, J.; Brewster, C.; aj.: A Comparative Evaluation of Term Recognition Algorithms. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco: European Language Resources Association (ELRA), may 2008, ISBN 2-9517408-4-0.

Příloha A

Obsah CD

- Technická zpráva ve formátu PDF a zdrojové texty pro LaTeX
- Zdrojové texty systému navrhující klíčová slova a spustitelný zkompilovaný bytekód
- Zdrojové kódy evaluátoru
- Demonstrační nastavení systému pro jeho snadné spuštění a několik připravených označkovaných dokumentů

Příloha B

Klíčová slova navržená některým dokumentům

Pro každý ukázkový dokument jsou uvedena klíčová slova přiřazená autorem dokumentu, takzvaná referenční klíčová slova a klíčová slova přiřazená systémem. Tučně jsou vyznačena klíčová slova, která jsou ve shodě s referenčními klíčovými slovy.

Každá podkapitola je pojmenovaná podle dokumentu, který je v ní zkoumán.

B.1 Lidová kultura Kyjovska a Ždánicka

Klíčová slova: Kyjovsko, Ždánsko, lidová kultura, historie, nářečí, hudební folklor, taneční folklor, dětský folklor, výroční zvyky a obyčeje

Systémem navržená klíčová slova: **Kyjovsko**, lidové písně, písně, **lidová kultura**, Kyjov, lidové tance, tanec, cimbálová muzika, Dubňany, František Synek, slovácká, Synek František, lidová, slovácký rok, slovácko, moravské slovensko, muzika, ústav lidové kultury, svatobořice, kulturní středisko, ...

B.2 Motivace studentů ke studiu oboru Informační studia a knihovnictví na MU podle genderu

Klíčová slova: Informační studia a knihovnictví, gender, genderová propast, genderové stereotypy, knihovny, knihovnictví, motivace ke studiu

Systémem navržená klíčová slova: genderový, **genderové stereotypy**, **gender**, bakalářský program,intracelurární, magisterský program, oborů studium, studium oborů, žena, informační studia, **knihovnictví**, obor isk, muži, isk obor, isk, divide gender, stereotypy, masarykova univerzita, gender divide, tek, kabinet informačních studií, ...

B.3 Erozní ohrožení půd vybraného území Jižní Moravy

Klíčová slova: dálkový průzkum Země, půdní eroze, erozní ohrožení půd, letecký snímek, vizuální interpretace

Systémem navržená klíčová slova: eroze, **letecké snímky**, erozní, vodní eroze, půdní, geologický ústav praha, snímky, svahů sklony, sklony svahů, svah, větrná eroze, půda, výřezy snímků, výřezy leteckých snímků, erozní procesy, ...

B.4 Přírodní medicína jako alternativní způsob léčby obyvatel, aneb návrat ke kořínkům

Klíčová slova: přírodní, alternativní a vědecká medicína, fytoterapie, samoléčba

Systémem navržená klíčová slova: přírodní medicína, medicína, alternativní medicína, metody medicíny, léčivé rostliny, metody přírodní medicíny, rostliny, vědecká medicína, léčivé, volně prodejné léky, léky, metody alternativní medicíny, byliny, prodejné léky, synteticky vyráběné léky, ...

Příloha C

Morfologická analýza ukázkové věty

Slovní forma	Lemma	Morfologický tag
Některé	některý	PZFP1----------
kontury	kontura	NNFP1-----A----
problému	problém	NNIS2-----A----
se	se_^(zvr._zájmeno/částice)	P7-X4----------
však	však	J^----------
po	po-1	RR--6----------
oživení	oživení_^(*3it)	NNNS6-----A----
Havlovým	Havlův_;S_^(*3el)	AUIS7M---------
projevem	projev	NNIS7-----A----
zdají	zdát	VB-P---3P-AA---
být	být	Vf--------A----
jasnější	jasný	AAFP1----2A----
.	.	Z:----------

Příloha D

Struktura morfologického tagu

Pozice	**Popis**
1	Slovní druh
2	Slovní poddruh
3	Rod
4	Číslo
5	Pád
6	Rod vlastníka
7	Číslo vlastníka
8	Osoba
9	Čas
10	Stupeň
11	Negace
12	Rod
13	Rezervováno
14	Rezervováno
15	Varianta, styl

Příloha E

Atributy analytické roviny

Název atributu	**Popis**
lemma	lemma
tag	morfologické kategorie, tag
form	tvar slova po příp. úpravách
afun	analytická funkce, neboli typ vztahu k nadřízenému (řídícímu) uzlu
lemid	bližší identifikace lemmatu (zejména pro víceslovná lemata)
mstag	morfologicko-syntaktický tag
origf	původní tvar slova
origap	formátovací informace předcházející původní tvar slova
gap1	formátovací informace předcházející tvar slova, část 1
gap2	formátovací informace předcházející tvar slova, část 2
gap3	formátovací informace předcházející tvar slova, část 3
ord	pořadové číslo slovního tvaru (form) ve větě

Příloha F

Ovládání evaluátoru

Spuštění probíhá z příkazové řádky:

`python evaluator.py [OPTION] [FILE]`
kde jako `FILE` je očekáván výstup z automatického navrhování klíčových slov.
Na standardní výstup je vypsána hodnota maxima F-measure a počet klíčových slov, při které bylo maxima dosaženo.

Možné přepínače jsou:

`--lemma`	Výsledky budou srovnávány s lemmaty referenčních termínů
`[-g|--graph=]GRAPH`	Do souboru `GRAPH` bude uložen výsledný graf, pokud není zadán, je graf vykreslen do nového okna.
`[-w|--graphwidth=]W`	Parametr `W` určuje, kolik bude v grafu zobrazeno výsledků (maximum osy x). Pokud není zadán je výchozí hodnota 100.

Příloha G

Klíčová slova navržená této práci

V této příloze jsou uvedena klíčová slova, která navrhl systém pro tuto práci.

klíčová slova
algoritmy
kandidátní slova
frequency
referenční klíčová slova
kandidátní
term frequency
referenční slova
metriky
lemma
inverse frequency
pokrytí
navržené klíčové slovo
referenční
navržené slovo
strom
počet slov
tvar slova
inverse
počet klíčových slov
inverse term
závěrečné práce
term inverse
frequency frequency
implementace
korpus
slovo
ohodnocovaný termín