Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Diplomová práce

Automatická extrakce klíčových slov pomocí metod trénovaných bez učitele

Plzeň, 2017 Bc. Karel Zíbar

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 15. května 2017

Bc. Karel Zíbar

Poděkování

V první řadě bych chtěl poděkovat panu Ing. Tomáši Brychcínovi, Ph.D., zadavateli mé diplomové práce, za odborné vedení, vstřícný přístup, bezproblémovou a rychlou komunikaci. Další poděkování patří výpočetnímu centru MetaCentrum za možnost používání jejich clusterů k natrénování metod na větším množství dat, bez čehož by nebylo možné tuto diplomovou práci vypracovat. Dále děkuji panu Ing. Lukáši Witzovi za korekturu a další opravy. V neposlední řadě děkuji své rodině za podporu během celého studia a Markétě Šafandové za všeobecnou podporu při jeho dokončování.

Abstract

This thesis deals with different approaches to keyword extraction from text documents. Three well-known methods have been implemented – TF-IDF, LDA and GloVe (keyword extraction by representing words as GloVe's vectors). Their algorithms have been slightly improved so that the methods can use more features. Finally, a new method (denoted as ZKEM), combining all three approaches mentioned above, has been created and properly evaluated in the same way as the others. All methods have been tested and compared with the algorithms participated at international programming competition SemEval 2010. The best method (improved TF-IDF) has achieved 25.95% accuracy. This result would be enough to take second place at SemEval competition.

Abstrakt

Tato práce se zabývá různými přístupy k extrahování klíčových slov z textových dokumentů. Celkem byly implementovány tři dobře známé metody – TF-IDF, LDA a GloVe (extrakce pomocí reprezentace slov vektory GloVe). Jejich algoritmy byly lehce upraveny tak, aby metody mohly využívat více příznaků než před tím. Byla také navržena vlastní metoda (označena jako ZKEM) kombinující všechny výše zmíněné přístupy a otestována stejným způsobem. Všechny tyto metody byly testovány a srovnány s metodami, účastnících se mezinárodní programovací soutěže SemEval 2010. Nejlepší metoda (vylepšená metoda TF-IDF) dosáhla úspěšnosti 25,95 %. Tento výsledek by stačil na druhé místo v soutěži SemEval.

Obsah

Seznam obrázků

Seznam tabulek

1 Úvod

Extrakce klíčových slov, případně celých klíčových frází, je důležitou technikou pro získávání informací z dokumentů (tzv. *information extraction*), jejich rozdělování do skupin na základě podobnosti (tzv. *clustering*), sumarizaci a tak podobně. Klíčová slova nesou totiž důležitou informaci o obsahu dokumentu, z nějž byly získány. S jejich pomocí může uživatel pátrat po hledané informaci s větší efektivitou, protože v případě, že budou ke každému dokumentu přiřazena (tzn. dokumenty budou indexovány), pomůže to v rozhodování, zda daný dokument použít, nebo ne [12]. V praxi si můžeme představit například internetový vyhledávač Google. Ten na základě klíčových slov (a samozřejmě dalších vlastností webových stránek a dokumentů – například meta značek) seřadí výsledky vyhledávání tak, aby co nejlépe odpovídaly hledané frázi. Pokud máme velmi rozsáhlá data (jako je v tomto případě prakticky obsah celého internetu), je ruční přiřazování klíčových slov k jednotlivým stránkám a dokumentů nemožné. Je proto nasnadě tuto úlohu automatizovat.

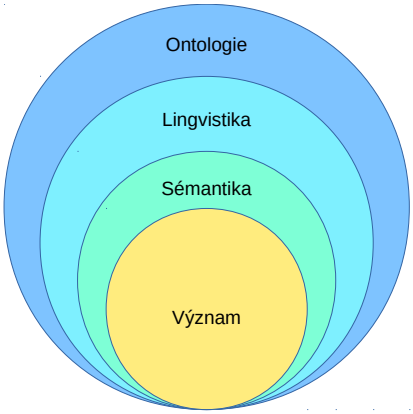
„Klíčovost“ slova nemá jednoznačnou definici a člověk, pokud provádí úlohu extrakce klíčových slov, se řídí především vlastní intuicí. To představuje značný problém při automatizování této úlohy – jak nahradit lidskou intuici za něco, co zvládne počítač? Z tohoto důvodu se jedná o úlohu umělé inteligence a strojového učení (*machine learning*) konkrétně pak z informatické disciplíny porozumění přirozenému jazyku (NLP – *natural language processing*).

V této práci jsou popsány některé z dobře známých metod pro extrakci klíčových slov z rodiny metod s učením bez učitele (*unsupervised methods*). Výhoda těchto metod spočívá v tom, že při fázi trénování nejsou zapotřebí ručně anotovaná data, jejichž vytvoření je v praxi často nákladné a v některých případech i nemožné (více v kapitole 3.1). Popsané metody byly implementovány a navíc byla navržena metoda ZKEM, která kombinuje jejich postupy. Algoritmy metod byly následně vylepšeny tím, že při extrakci klíčových frází je zohledňována například i pozice fráze v článku nebo vzájemný výskyt slov tvořících danou frázi. Všechny metody byly pak srovnávány jak pro extrakci pouze jednoslovných frází tak i víceslovných. Jako referenční hodnoty pro porovnávání dosažených výsledků byly použity výsledky mezinárodní programovací soutěže SemEval 2010, kde byla jedním z úkolů právě extrakce klíčových slov (viz [11]).

2 Sémantika textu

Co je obsahem textu? Respektive jaké jsou elementy, které musejí být ve správném pořadí rozpoznány, aby byl pochopen hlavní význam textu? V každém psaném textu (v rozšířeném smyslu i v mluvené formě jazyka) se vždy vyskytuje několik vět, které sdělují podstatu a hlavní myšlenku, kterou chtěl autor říci – kostru dokumentu. Problém je tuto kostru lokalizovat. Jedná se o klíčový krok, který je nezbytný k jakémukoliv dalšímu porozumění danému textu [26].

Můžeme říci, že text je tvořen různými slovy, ze kterých různí aktéři formují různé větní celky v kombinaci s jinými aktéry. Můžeme také tvrdit, že tato slova jsou pro význam textu různě důležitá. Malé procento z těchto slov, ta nejvýznamnější slova (klíčová slova), pak představují samotný základ textu. Pokud by byla odstraněna, „konstrukce textu“ by se zřítila a jeho význam by se ztratil [26]. Na obrázku 2.1 můžeme vidět jednotlivé abstraktní vrstvy, přes které musíme projít, abychom pochopili význam textu.



Obrázek 2.1: Cesta k významu textu

Ontologie je nejvíce abstraktní vrstvou významu textu (a všech dalších projevů myšlení dalo by se říci). Jedná se o filozofickou disciplínu zabývající se jsoucnem, bytím jako takovým a těmi nejobecnějšími otázkami [19]. Lingvistika, neboli jazykověda, jak už samotný název napovídá, je vědním oborem zkoumajícím přirozený jazyk. Sémantika je pak nauka o významu výrazů z různých strukturních úrovní jazyka – morfémů, slov, slovních spojení a vět.

Sémantická analýza

Sémantická analýza se zabývá porozuměním textu. Jedná se o obor zpracování přirozeného jazyka, při kterém je zjišťován obecný význam slov v textu. Uplatnění metod z oblasti sémantické analýzy můžeme nalézt například při sumarizaci nebo získávání informací z textu (IR – *information retrieval*) [8], klasifikaci dokumentů (*document classification*) [21] nebo, jako v případě této práce, při extrakci klíčových slov z textu. Jednotlivé přístupy metod sémantické analýzy jsou popsány v kapitole 3.

2.1 Distribuční hypotéza

Distribuční hypotéza tvrdí, že můžeme do určité míry odhadnout význam slova v textu na základě ostatních slov, v jejichž okolí se vyskytuje. Přímým důsledkem je pak to, že slova, vyskytující se často v podobném kontextu, můžeme brát jako sémanticky podobná [5].

„If we consider words or morphemes A and B to be more different in meaning than A and C, then we will often find that the distributions of A and B are more different than the distributions of A and C.“

Harris (1954)

„You shall know a word by the company it keeps.“

Firth (1957)

2.2 Bag-of-Words hypotéza

Bag-of-Words hypotéza (někdy také zkráceně BOW hypotéza) je zjednodušující model, na kterém je založena spousta metod extrakce klíčových slov z textových dokumentů. Uvažujme dokument jako konkrétní text, ze kterého chceme získat klíčová slova (v rozšířeném smyslu se nemusí jednat pouze o textovou podobu přirozeného jazyka, ale i například zvukovou podobu [12]).

Podle této hypotézy není pravděpodobnost výskytu slova v dokumentu podmíněná slovy v jeho blízkosti a na celý dokument tak lze nahlížet jako na neuspořádanou množinu slov.

To v praxi dovoluje reprezentovat dokument jako vektor příznaků a můžeme spočítat metriky charakterizující daný text. Častým případem je charakterizování dokumentu jako vektoru četnosti unikátních slov korpusu (kolekci všech dokumentů, které máme k dispozici pro trénování metody). Uvažujme následující dva dokumenty:

```
Lukáš se rád dívá v televizi na fotbal.
      Karel se dívá rád na filmy.
```

Pokud seřadíme unikátní slova v obou dokumentech podle místa, kde se poprvé vyskytla, tedy

[Lukáš, se, rád, dívá, v, televizi, na, fotbal, Karel, filmy],

můžeme pak aplikací Bag-of-Words hypotézy každý dokument charakterizovat následovně:

```
[1, 1, 1, 1, 1, 1, 1, 1, 0, 0]
[0, 1, 1, 1, 0, 0, 1, 0, 1, 1]
```

2.3 Morfologie jazyka

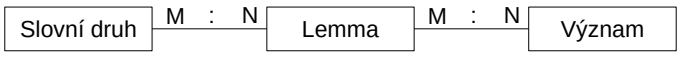
Určování významu (sémantiky) textu dokumentu může být velmi obtížné. Stačí si uvědomit, že se slova vyskytují v různých pádech, tvarech a mnohdy nemusejí být dokonce ani spisovná (v závislosti na typu dokumentu). Je tedy potřeba zajistit, aby se různé gramatické tvary slova rozpoznávaly jako stále stejné slovo a nikoli jako formálně odlišné řetězce znaků. Obecně existují dvě hlavní metody pro nalezení jednotného tvaru slov – lematizace a stemování (*lemmatisation* a *stemming*) [24].

Lematizace: Při lematizování slova se hledá jeho slovníkový tvar. Rozšířeny jsou hlavně dva přístupy jak toho dosáhnout. Lematizer buď musí

obsahovat slovník daného jazyka a také soubor pravidel, podle kterých se slova v tomto jazyce tvoří, nebo si tento slovník s pravidly vytvoří pomocí procesu strojového učení. Porovnání obou těchto přístupů můžeme nalézt ve článku [10], kde jsou srovnávány na úloze získávání informací z textu.

Stemování: Pro hledání stemů („kořenů“) slov se používají heuristické algoritmy nebo, v současné době rozšířenější, algoritmy strojového učení. Nalezený stem se tedy může lišit v závislosti na použité metodě. V článku [6] je popsán algorimtus hledání stemů, který se učí morfologická pravidla z neanotovaného korpusu bez znalosti jazyka nebo dalších informací o textu. Největší výhodou tohoto algoritmu je, že dosahuje stejné úspěšnosti při stemování známých i neznámých slov. Další výhodou je také vysoká úspěšnost i při trénování na malých datech (např 50 000 slov), což je dobré například při použití pro jazyky, pro které není mnoho dostupných dat.

Pokud máme dokument lematizovaný můžeme narazit na problém určení skutečného, tedy správného, významu daného lemmatu. Mezi lemmatem a slovním tvarem existuje obecně relace M:N (viz obrázek 2.2). Existují slova, která jsou homonymní tzn., že mají stejné lemma, ale odlišují se významem (jejich významy nemají nic společného). Příkladem homonimního slova může být slovo „koruna“, které může označovat měnu, korunu stromu nebo královskou korunu. Lehčím příkladem homonymních slov jsou slova polysémní (mnohoznačná). Jedná se opět o slova se stejným lemmatem a různými významy, nicméně tyto významy mezi sebou mají určitou podobnost. Příkladem může být sovo „oko“, což může být zrakový orgán, díra na punčoše nebo pytlácká past. V tomto případě je možná podobnost významů na základě tvaru.



Obrázek 2.2: Vztah mezi slovním druhem, lemmatem a významem

Princip kompozicionality: Pokud budeme uvažovat víceslovná spojení, tedy bigramy, trigramy atp., musíme předpokládat, že význam slovního spojení lze odvodit z jeho částí a pravidel, podle kterých byly zkombinovány (princip kompozicionality). Tuto hypotézu však vyvracejí tzv. idiomy – slovní spojení, jejichž význam nelze odvodit z významu slov, ze kterých jsou tvořeny. Idiomem může být například „ztratil hlavu“, „natáhl bačkory“ nebo v an-

gličtině „pull somebody's leg“. Odhaduje se, že angličtina má přes 25 000 takovýchto výrazů [17].

Stop slova: Pokud se na výskyt slov v textu podíváme ze statistického hlediska, zjistíme, že slov nesoucích nějakou relevantní informaci o dokumentu je poměrně málo. Nejčetnější slova v textu totiž bývají předložky, spojky, zájmena, příslovce nebo členy (vezmeme-li v potaz anglické texty). Takováto slova (*stop-words*) není tedy nutno brát v potaz při sémantické analýze textu. Seznamy takovýchto slov se dají nalézt na internetu například pro angličtinu na stránkách XPO6[1]. Na příklady těchto stop-slov se můžete podívat níže.

```
a, an, and, are, as, at, be, but, by, for, if, in, into, is,
it, no, not, of, on, or, such, that, the, their, then, there,
these, they, this, to, was, will, with
```

[1] `http://xpo6.com/list-of-english-stop-words`

3 Metody extrakce klíčových slov

V této kapitole je nastíněno základní rozdělení metod pro získávání klíčových slov z textových dokumentů. Postup algoritmu většiny metod extrahující klíčová slova z dokumentů by se dal rozdělit do třech následujících kroků [28]:

1. Výběr kandidátů – nejprve je třeba vybrat z textu všechna slova, která mohou být klíčová. To se dá zařídit například odstraněním všech stopslov a různých použitých zkratek, symbolů a jiných označení (například matematických).
2. Určení významnosti slov – v tomto kroku se ke všem vybraným kandidátům určí jejich „míra klíčovosti“. Tu můžeme určit na základě jejich četnosti, pozice v dokumentu, kde se nacházejí (slova v nadpisu budou pravděpodobně více důležitá než ostatní), fontu, kterým jsou napsány nebo podle dalších vlastností slova.
3. Výběr nejlepších – na základě hodnoty, která byla určena v předešlém kroku, je vybrán konečný seznam klíčových slov. K tomuto problému můžeme přistupovat dvěma způsoby – určíme pevný počet nejlepších kandidátů nebo vybereme všechny kandidáty s hodnotou nad předem určenou hranicí (u tohoto přístupu může být problémem určení hranice, kdy kandidáty ještě vybrat a kdy naopak už ne).

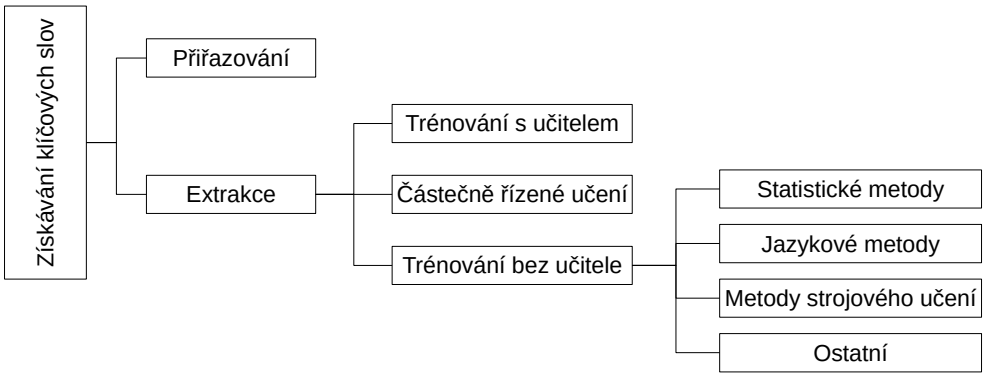
3.1 Rozdělení metod

Metody získávání klíčových slov můžeme rozdělit do dvou kategorií [1] – přiřazování klíčových slov (*keyword assignment*) a extrahování klíčových slov (*keyword extraction*). Rozdíl je v tom, že pokud klíčová slova přiřazujeme, přiřazujeme je z předem definovaných slovníků termů (slov nebo slovních spojení). To ale znamená, že pro daný dokument můžou být určena klíčová slova, která se v něm vůbec nevyskytují. Naopak při extrahování jsou slova z dokumentu analyzována tak, aby se určilo, která z nich jsou nejvíce reprezentativní a vystihují obsah dokumentu nejlépe. Slova získaná tedy tímto způsobem se v dokumentu vždy vyskytují.

Tato práce se zabývá výhradně metodami extrakce klíčových slov. Ty můžeme dále rozdělit na metody učení s učitelem (*supervised methods*), metody

učení bez učitele (*unsupervised methods*) a na metody s částečně řízeným učením (*semi-supervised methods* viz [16]). Metody trénování s učitelem potřebují ke svému správnému natrénování ručně anotovaný dataset – tedy vždy dokument a k němu seznam klíčových frází. To ale v praxi není vždy možné, protože ne všichni autoři ke svým článkům dodávají seznam klíčových slov a ruční anotace je velmi zdlouhavý proces zvážíme-li, že je potřeba každý článek přečíst a zvolit k němu slova, která jej nejvíce vystihují. Ke správné anotaci se také v praxi mnohdy nepoužívá pouze jeden anotátor ale více. To zaručuje větší přesnost. Při velmi velkých datasetech je ale ruční anotace nemožná. Z tohoto důvodu je tato práce zaměřena na metody s trénováním bez učitele.

Metody s učením bez učitele lze dále dělit podle jejich přístupů na statistické (*simple statistic*), jazykové (*linguistic*), metody strojového učení a další, které třeba kombinují několik přístupů najednou [25]. Na celé schéma dělení metod získávání klíčových slov se můžete podívat na obrázku 3.1.



Obrázek 3.1: Rozdělení metod získávání klíčových slov

Statistické metody

Statistické metody nepotřebují žádné předem definované slovníky a jsou tedy doménově i jazykově nezávislé. Statistika může být obecně využita k určení klíčových slov například pomocí metody TF-IDF (viz kapitola 4.2).

Nevýhodou těchto metod je ale to, že v některých odborných dokumentech, jako jsou například lékařské články, se může významově nejdůležitější slovo vyskytnout pouze jednou. Tyto metody jej pak mohou znevýhodnit oproti jiným slovům, která se v dokumentu vyskytují častěji.

Jazykové metody

Jazykové metody používají jazykové vlastnosti slov, vět a dokumentů pro určování klíčových slov. Algoritmy těchto metod v sobě kombinují lexikální analýzu, syntaktickou analýzu, diskurzivní analýzu (zkoumání sociální reality v textu) a tak podobně [25].

Metody strojového učení

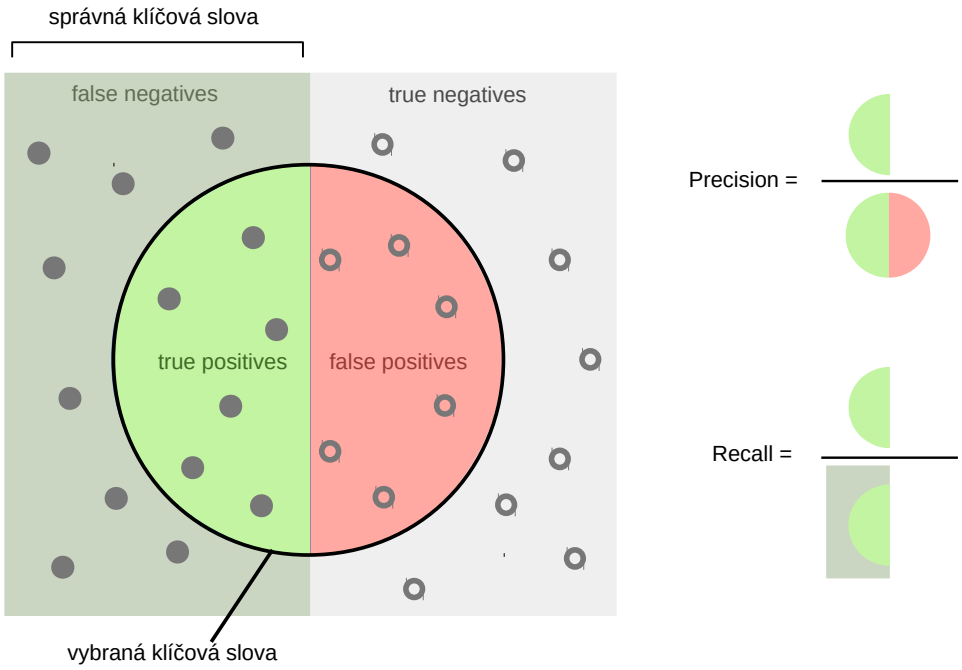
Metody strojového učení využívají trénování s učitelem nebo bez učitele (viz výše). Trénovací model může být vyvozen z algoritmů strojového učení využívajících například Bayessův teorém, SVM (*Support Vector Machines*), C4.5 atd. Protože ale často vyžadují trénovací data jsou tedy tyto metody častěji závislé na doméně dokumentů (jazyku, kategorii dokumentů,...). Model se musí také pokaždé přetrénovat, pokud dojde ke změně této domény.

Ostatní metody

Další přístupy k extrahování klíčových slov kombinují výše zmíněné metody a někdy přidávají heuristickou znalost jako je například pozice v textu, délka termu, rozložení slov v něm, html a podobné značky, formátování textu a tak podobně. Na příklad postupů metod z této kategorie se můžeme podívat v kapitole 3.3, kde jsou popsány metody, které dosáhly nejvyšší úspěšnosti v soutěži SemEval 2010 (viz [11]).

3.2 Metrika úspěšnosti metod

Je mnoho způsobů, jak změřit účinnost navržené metody pro extrakci klíčových slov. Nejběžnějším způsobem je určování hodnot *precision* a *recall* a jejich harmonický průměr – F-skóre (někdy též *F-Measure, F-score*) [23]. Uvažujme tedy množinu dokumentů, kde ke každému z nich máme přiřazený seznam klíčových slov. To jsou naše *gold data*. Výstupem testované metody je seznam možných klíčových slov – kandidátů. Každé toto slovo, v závislosti na tom jestli se nalézá mezi gold daty, můžeme podle schématu na obrázku 3.2 zařadit do jedné ze čtyř skupin.



Obrázek 3.2: Precision a Recall

- true positives – slova označená metodou jako klíčová, nalézající se v gold datech (tzv. *hit*)
- false positives – slova označená metodou jako klíčová ale nenalézající se v gold datech (tzv. *miss*)
- true negatives – slova neoznačená metodou jako klíčová a nenalézající se v gold datech
- false negatives – slova neoznačená metodou jako klíčová a nalézající se v gold datech

Pomocí velikostí těchto množin pak můžeme určit hodnoty *precision* (P), *recall* (R) a celkové *F-Skóre* (F) jako:

$$P = \frac{\text{true positives}}{\text{true positives} + \text{false positives}} \tag{3.1}$$

$$R = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}} \tag{3.2}$$

$$F(\beta) = \frac{(1+\beta^2)PR}{\beta^2 P + R}, \tag{3.3}$$

kde β je parametrem funkce F a většinou je volena rovna jedné.

K určování hodnoty F-Skóre můžeme přistupovat dvojím způsobem – určením jako *Makro F-Skóre* nebo *Mikro F-Skóre*. Uvažujme $L = \lambda_j$ pro $j = 1...q$ jako seznam klíčových slov extrahovaných testovanou metodou, binární validační metriku B, která je určována na základě množin slov true positives (tp), true negatives (tn), false positives (fp) a false negatives (fn). Proměnné $tp_\lambda, tn_\lambda, fp_\lambda$ a fn_λ nechť jsou konkrétní množiny pro dokument λ. Pro seznam klíčových slov λ můžeme pak stanovit binární validační metriku B následujícími způsoby [22]:

$$B_{macro} = \frac{1}{q}\sum_{\lambda=1}^{q} B(|tp_\lambda|, |tn_\lambda|, |fp_\lambda|, |fn_\lambda|) \tag{3.4}$$

$$B_{micro} = B\left(\sum_{\lambda=1}^{q}|tp_\lambda|, \sum_{\lambda=1}^{q}|tn_\lambda|, \sum_{\lambda=1}^{q}|fp_\lambda|, \sum_{\lambda=1}^{q}|fn_\lambda|\right), \tag{3.5}$$

kde metrika B může představovat například precision, recall nebo F-Skóre. Macro F-Skóre dává stejnou váhu každé z množin slov pro všechny dokumenty, zatímco Micro F-Skóre dává stejnou váhu množinám pro každý dokument zvlášť. V obou případech F-Skóre ignoruje slova z množiny true negatives (viz rovnice 3.1, 3.2 a 3.3) a její rozsah je tedy dán převážně velikostí množiny true positives.

3.3 Nejlepší známé metody

Níže popsané metody nemají název a jedná se vždy spíše o kombinaci různých postupů a příznaků. Jsou tedy uváděny pod názvy týmů tak, jak byly prezentovány v mezinárodní programátorské soutěži SemEval 2010 (více informací můžete nalézt ve článku [11]). O postupu určování úspěšnosti těchto metod, a vlastně i metod implementovaných v této diplomové práci, se můžete dočíst v kapitole 6, kde je celý postup popsán. Kompletní výsledky níže uváděných metod naleznete v příloze A. Data poskytnutá organizátory soutěže jsou pak popsána v kapitole 5.2.

3.3.1 HUMB

Metoda navržená tímto týmem dosáhla nejvyšší úspěšnosti na všech třech testovacích datasetech. Na extrakci klíčových slov je zde nahlíženo jako na podúlohu extrakce technických výrazů z vědeckých článků a kandidáti jsou vybíráni na základě tří sad příznaků.

První sadou jsou výsledky strukturální analýzy článku. Ta byla prováděna pomocí modulu systému GROBID (GeneRation Of BIbliographic Data), který je zaměřen na automatickou extrakci bibliografických dat (hlavičky, citace atp.) a na rozpoznávání struktur článku (nadpisy, obrázky, tabulky atd.) Druhá sada příznaků zachycuje vlastnosti obsahu založené na „frázovitosti“, „infomativnosti“ a „klíčovosti“ termu. Určení stupně frázovitosti, nebo lépe řečeno soudržnosti víceslovných termů, probíhá určením hodnoty *gdc* (Generalized Dice Coeficient). Jedná se o obdobu hodnoty *pmi* (viz kapitola 4.1). Informativnost slova pak reprezentuje hodnota *tfidf* (viz kapitola 4.2) a klíčovost slova jeho frekvence v globálním korpusu. Třetím a posledním příznakem každého termu je binární indikátor toho, jestli se výraz objevil v databázi GRISP (velký terminologický slovník technických a vědeckých pojmů) nebo Wikipedie.

Na základě všech sad příznaků byl vytvořen model strojového učení. Ze všech typů modelů, se kterými bylo experimentováno, mezi nimi například MLP (Multi-Layer Perceptron) a SVM (Support Vectore Machines), byl vybrán model rozhodovacího stromu, který dosahoval nejlepšího poměru úspěšnosti, stability a náročnosti na natrénování. Po aplikaci modelu na neznámý dokument jsou hodnoty vybraných kandidátů přepočítány s ohledem na pravděpodobnost, že se jedná skutečně o klíčová slova, získanou ze statistik z vý-

zkumného archivu HAL (Hyper Article en Ligne). Vlastní výběr výsledných kandidátů na klíčová slova v neznámém dokumentu pak probíhá následujícím způsobem:

1. Extrakce všech n-gramů (slovo nebo slovní spojení o n slovech) až do hodnoty $n = 5$.
2. Odstranění všech kandidátů začínajících nebo končících na stop-slovo.
3. Odstranění všech kandidátů obsahujících matematický symbol.
4. Normalizování kandidátů převedením na malá písmena a ostemováním stemmerem Porter[1].

Jako trénovací data pro tuto metodu byla použita množina 144 dokumentů poskytnutých organizátory soutěže, která byla rozšířená o korpus z Národní univerzity v Singapuru (dalších 159 článků ze stejných ACM kategorií jako původní trénovací množina). Více informací o této metodě se můžete dočíst ve článku [13].

3.3.2 SZTERGAK

Podle [2] se jedná o systém s učením s učitelem, který funguje na principu Naivního Bayessova klasifikátoru a uvažuje n-gramy až do délky čtyř slov. Prvním krokem tohoto postupu bylo předzpracování trénovacích dat. Podle prvního řádku každého dokumentu (nadpis článku) byl na internetu vyhledán původní dokument v prvních top deseti výsledcích poskytnutých Google API. Dále byly z dokumentů odstraněny všechny řádky nesoucí neužitečné informace. Ty byly identifikovány na základě jejich délky. Následně byly v textu ponechány pouze podstatná a přídavná jména a slovesa, která byla v posledním kroku ostemována pomocí Porter stemmeru. Ke každému takto vzniklému tvaru byla také přidána informace o slovním druhu slova, ze kterého vznikl (zda se jednalo o sloveso nebo podstatné či přídavné jméno).

K extrakci klíčových slov byly použity standardní příznaky jako hodnota *tfidf* (viz kapitola 4.2) a relativní index prvního výskytu termu v dokumentu.

[1] `https://tartarus.org/martin/PorterStemmer`

Další příznaky by se daly rozdělit do čtyř kategorií – příznaky na úrovni fráze, dokumentu, korpusu a externí příznaky. Pro frázi byly určeny příznaky jako délka fráze, seznam slovních druhů každého unigramu ve frázi, tak jak jdou za sebou, a binární příznak toho, jestli původní tvar končil některým suffixem. Na úrovni dokumentu byl sestaven set příznaků zahrnující hodnotu *pmi* (viz kapitola 4.1) a binární ukazatel toho, jestli je daná fráze rozšířením některého akronymu (zkratky složené z prvních písmen termu) v textu. Jako příznak na úrovni korpusu byla určena hodnota *sfisf* (obdoba hodnoty *tfidf* pro sekci dokumentu) a binární indikátor toho, jestli se daná fráze vyskytla mezi autorem přiřazenými klíčovými frázemi v trénovacích datech. Poslední kategorií příznaků jsou příznaky externí. K jejich získání bylo využito Wikipedie a jedná se o ukazatele, jestli pro danou frázi existuje stejnojmenný článek na Wikipedii. Při trénování klasifikátoru byly také brány v potaz i linky na sémanticky podobné články.

3.3.3 WINGNUS

Systém popsaný v [15] pracuje poněkud jiným způsobem než předchozí dva. Jeho hlavní myšlenkou je, že články v prostém textu ztratí velkou část informace o slovech které obsahují – formátování. Nejdůležitější součástí tohoto systému je tedy nalezení původního PDF textu na internetu. K tomu byl speciálně naprogramován crawler, který tyto články vyhledává na Google Scholar podle prvních dvou řádek z daného textového dokumentu. K dokumentům z trénovací množiny (celkem 144 dokumentů) byl tento crawler schopen najít 117 dokumentů, z nichž 116 bylo relevantních. Pro strukturální analýzu PDF dokumentů byl navržen vlastní software – SectLabel, který každé řádce dokumentu přiřadí typ (nadpis, hlavička, text těla atp). Řádky hlavičky jsou dále členěny ještě na abstrakt, intro atp.

Pro využití této analýzy byla formulována hypotéza, že klíčové fráze se nejčastěji objevují v prvních n řádcích každého typu sekce a bylo změřeno, že nejvíce (s hustotou výskytů větší než 0,2) se vyskytuje v nadpisech, hlavičkách, abstraktech, intrech, podobných pracích (related works) a závěrech. Text v těle dokumentu mnoho klíčových frází naopak neobsahuje. Z extrahovaných frází ze stažených PDF dokumentů se jich celkem 97 % skutečně objevovalo v původních textových dokumentech.

Výběr kandidátů z dokumentu probíhal na základě 19 ciferné hodnoty, kde jednotlivé pozice znamenají ((n) značí reálnou číselnou hodnotu a (b) binární):

- 1-3 (n) – hodnota *tfidf*
- 4-5 (n) – první a poslední výskyt termu
- 6 (n) – délka fráze
- 7 (b) – indikuje, zda některá část fráze byla v PDF dokumentu vytištěna tučně nebo kurzívou
- 8 (b) – indikuje, zda se term vyskytl v nadpisu
- 9 (n) – počet, kolikrát se term vyskytl v nadpisech jiných dokumentů (získaných z DBLP databáze)
- 10-14 (b) – ukazatele, zda se term vyskytl v hlavičce, abstraktu, intru, podobných pracích nebo závěru.
- 15-19 (b) – počty, kolikrát se term vyskytl v hlavičce, abstraktu, intru, podobných pracích nebo závěru.

Trénování modelu probíhalo na testovací množině dokumentů poskytnutých organizátory soutěže (144 dokumentů), která byla rozdělena na testovací (104 dokumentů) a validační část (40 dokumentů). Bylo vyzkoušeno různé pořadí výše zmíněných pozic celkového příznaku dokumentu aby se dosáhlo nejlepšího výsledku. Dále byl také proveden experiment s extrahováním kandidátů pomocí regulárního výrazu jen z určitých částí dokumentu, kde bylo zjištěno, že 63 % všech správných klíčových frází lze extrahovat pouze z nadpisu, abstraktu, intra a hlavičky.

3.3.4 SEERLAB

Podobně jako v předchozích případech i metoda popsaná v [20] využívá strukturální analýzy článku, poté vybere možné kandidáty na klíčové fráze, pro které je nakonec určeno skóre, a jsou vybráni nejlepší z nich. Ke strukturální analýze článku je použit regulární výraz, který článek rozdělí do šesti kategorií (více kategorií bylo rozhodnuto, že je irelevantní pro úkol extrakce klíčových slov). Při výběru kandidátů bylo pro zvýšení hodnoty *precision*

rozhodnuto, že sekce dokumentu „Metodology + Experiments“ bude z uvažovaného textu pro extrakci frází vynechána. Ze zbylého textu jsou následně extrahovány bigramy, trigramy a čtyřgramy, které se v dokumentu vyskytly alespoň třikrát a zároveň neobsahují stop-slovo. Některá stop-slova jako například „from“, „of“ nebo „for“ byla ovšem povolena. Aby výsledný systém mohl produkovat i unigramy jako klíčové fráze, bylo do výsledného seznamu kandidátů ještě přidáno třicet nejčetnějších unigramů v dokumentu. Na rozdíl od dalších metod tento postup nevyžaduje žádné přidané informace jako POS tagging a podobně.

Z takto vybraných kandidátů byl vytvořen natrénovaný model Random Forest (RF) klasifikátoru. RF klasifikátor je kolekce rozhodovacích stromů, kde pravděpodobnost, že je daný kandidát skutečně klíčovou frází, je jednoduchá agregace „hlasů“ v každém stromě. Každému kandidátovi je přidělen následující vektor příznaků:

- N – počet slov ve frázi
- ACRO – binární příznak toho, jestli se fráze vyskytla jako akronym v dokumentu
- TF_{doc} – počet kolikrát se fráze vyskytla v dokumentu
- DF – počet kolikrát se fráze vyskytla v trénovací množině dokumentů
- TFIDF – hodnota *tfidf* vypočtená z předchozích dvou
- $\mathrm{TF}_{headers}$ – počet kolikrát se fráze vyskytla v hlavičce dokumentů
- $\mathrm{TF}_{section_i}$ – počet kolikrát se fráze vyskytla v sekci i (nadpis, abstrakt, intro, závěr, ...)

Pro trénování modelu byla využita pouze množina 144 dokumentů poskytnutých organizátory soutěže, která nebyla nijak rozšířena, a bylo zjištěno, že nejlepšími indikátory „klíčovosti“ fráze jsou *tfidf*, *df* a překvapivě i délka fráze. Nejmenší přínos pro úspěšnou extrakci měla hodnota *acro*, protože většina klíčových slov se v textu neobjevuje jako akronym. Podobně jako v předchozí metodě bylo také zjištěno, že klíčová slova se nevyskytují četně v sekci Related Work a těle dokumentu. Naopak velmi četně se vyskytují v intru, abstraktu a závěru.

4 Určování významnosti slova

4.1 Bodová vzájemná informace

Bodová vzájemná informace (*pmi* – Pointwise Mutual information) je metrika, která udává jak se liší pravděpodobnost toho, že se dva náhodné pravděpodobnostní jevy vyskytují souběžně od hodnoty, která by se dala očekávat z pozorování obou jevů. Tedy odlišnost pravděpodobnosti $p(x, y)$ od $p(x)p(y)$. Hodnota *pmi* může být kladná i záporná. Pro dva nezávislé pravděpodobnostní jevy x a y můžeme *pmi* určit následujícím způsobem:

$$\text{pmi}(x, y) = \log \frac{p(x, y)}{p(x)p(y)} = \log \frac{p(x|y)}{p(x)} = \log \frac{p(y|x)}{p(y)}, \tag{4.1}$$

kde v čitateli je pravděpodobnost souběžného výskytu jevů a ve jmenovateli pak součin pravděpodobností výskytu každého z nich [4]. Pro problém extrakce klíčových slov můžeme pravděpodobnosti výskytů nezávislých jevů z rovnice 4.1 zaměnit za frekvence výskytů v korpusu. Uvažujme nyní korpus C jako množinu slov, hodnoty n_{xy} jako počet výskytů bigramu a n_x, n_y jako počet výskytů dílčích unigramů. Celou rovnici pak můžeme upravit do tvaru

$$\text{pmi}(x, y) = \log \left(\frac{\frac{n_{xy}}{|C|}}{\frac{n_x}{|C|} \cdot \frac{n_y}{|C|}} \right). \tag{4.2}$$

Při určování hodnoty *pmi* můžeme narazit na několik problémů. Tato metrika nefunguje příliš přesně v případě, že máme například „řídká“ data. Pokud se některá slova vyskytnou v korpusu pouze jednou ale společně, určená hodnota *pmi* bude vysoká nicméně nebude určovat důležitost daného termu. Stejně tak tomu bude v případě silně závislých slov. Pokud máme slovo, které se objevuje zároveň s jiným, vzorec pro výpočet se zredukuje pouze na tvar $\log(p(y)^{-1})$. To znamená, že čím vzácnější slovo je, tím větší je hodnota *pmi*. Vyšší hodnota *pmi* tedy nemusí znamenat silnou závislost slov, ale jen vzácnější term.

4.2 TF-IDF

Jedná se o jednu z nejjednodušších statistických metod (vic kapitola 3.1), která pracuje s předpokladem, že více četná slova v dokumentu jsou nositeli významnější sémantické informace než méně četná slova.

Tato metrika je složena ze dvou částí – *tf* (Term Frequency), která nabývá větších hodnot, pokud se term vyskytuje v dokumentu častěji a *idf* (Inverse Document Frequency) nabývající větších hodnot pro termy málo četné v ostatních dokumentech korpusu. První část určuje četnost slova v dokumentu a druhá jeho důležitost s ohledem na všechny ostatní dokumenty v korpusu. Určením hodnoty *tfidf* můžeme tedy zjistit, jak moc důležitý je určitý term pro daný dokument [29].

Nevýhodou této metody je, že upřednostňuje více četná slova dokumentu před těmi méně četnými. To může mít ale za následek ignorování slov významově důležitých, která se v textu objevila například jen jednou, jak bylo již uvedeno u přehledu statistických metod (viz kapitola 3.1).

Složku *tf* získáme sečtením všech výskytů slova w_i v dokumentu d_j. To ovšem nestačí, protože by pak docházelo ke zvýhodňování delších dokumentů nad kratšími. Delší dokumenty by obsahovaly dané slovo vícekrát. Proto je potřeba počet výskytů slova normalizovat vydělením délkou dokumentu. To je vidět na vztahu

$$\mathrm{tf}_{i,j} = \frac{n_{i,j}}{|d_j|}, \tag{4.3}$$

kde $n_{i,j}$ je výskyt slova slova w_i v dokumentu d_j.

Složku *idf* můžeme dostat určením počtu dokumentů, které obsahují slovo w_i a dosazením do rovnice

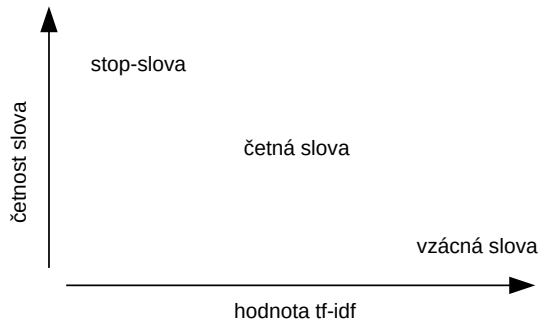
$$\mathrm{idf}_i = \log_{10} \frac{|D|}{|\{j : w_i \in d_j\}|}, \tag{4.4}$$

kde D je množina všech dokumentů korpusu a hodnota ve jmenovateli označuje počet dokumentů, které obsahují slovo w_i.

Celkovou hodnotu *tfidf* dostaneme součinem obou jejích částí (viz rovnice 4.3 a 4.4).

$$\text{tfidf}_{i,j} = \text{tf}_{i,j} \cdot \text{idf}_i = \frac{n_{i,j}}{|d_j|} \cdot \log_{10} \frac{|D|}{|\{j : w_i \in d_j\}|} \tag{4.5}$$

Z předchozích vztahů můžeme usoudit, že pokud je slovo v daném dokumentu více četné (a tudíž složka *tf* vychází větší), jedná-li se o slovo vyskytující se hojně v korpusu dokumentů, je složkou *idf* znevýhodňováno a celková hodnota *tfidf* vychází tedy menší. Podle závislosti výsledné hodnody *tfidf* na četnosti v dokumentu, můžeme všechna slova rozdělit do několika skupin jak je vidět na obrázku 4.1.



Obrázek 4.1: Rozdělení slov podle hodnoty tfidf

TF-IDF víceslovných termů

Určování hodnoty *tfidf* funguje stejně pro unigramy jako i bigramy, trigramy a další víceslovné termy. Rovnice 4.5 se nijak nemění. V práci je použit tento postup a zároveň byl vyzkoušen i postup, kdy se celková hodnota *tfidf* víceslovného termu $\boldsymbol{w}$ o n slovech určuje jako geometrický průměr hodnot *tfidf* jednotlivých unigramů v něm viz vzorec 4.6.

$$\text{tfidf}_{\boldsymbol{w}} = \left(\prod_{i=1}^{n} \text{tfidf}_{w_i} \right)^{\frac{1}{n}} \tag{4.6}$$

Tento postup je dále označován jako metoda TF-IDF-avg.

4.3 Latentní Dirichletova alokace

Latentní Dirichletova alokace (Latent Dirichlet Allocation – LDA) je generativní pravděpodobnostní model, který má široké pole použitelnosti a to zejména ve zpracování přirozeného jazyka. Využití však nalézá i v disciplínách jako je získávání vizuálních informací (*content-based image retrieval*) nebo bioinformatice [3]. V této práci je algoritmus LDA použit pro extrakci klíčových slov z textových dokumentů.

4.3.1 Popis modelu

Model LDA je založen na Bag-of-Words hypotéze (viz kapitola 2.2) a díky tomu dovoluje na dokument nahlížet jako na pravděpodobnostní rozdělení předem daného počtu témat. Podobně pak na každé téma lze nahlížet jako na pravděpodobnostní rozdělení slov.

Zaveďme nyní následující značení:

- K – počet různých témat
- M – počet dokumentů
- V – počet slov ve slovníku
- N – počet slov ve všech dokumentech
- w_i – slovo na pozici i v korpusu
- z_i – téma na pozici i v korpusu
- d_i – index dokumentu na pozici i v korpusu

LDA předpokládá následující generativní proces pro každý dokument d v korpusu D (grafickou podobu tohoto procesu pak můžeme vidět na obrázku 4.2):

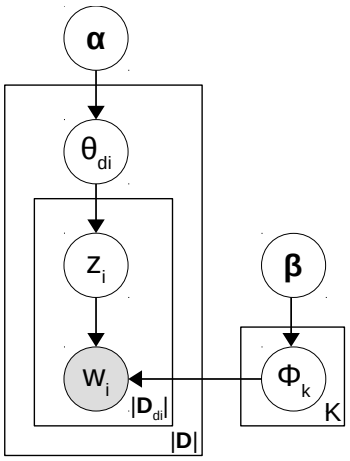
1. Pro každý dokument $d_m \in D$ vybere dirichletovo rozdělení $\theta_m \sim Dir(\boldsymbol{\alpha})$ přes témata z_i, kde $i \in < 1, K >$, $m \in < 1, M >$ a $\boldsymbol{\alpha}$ je vektor hyperparametrů Dirichletova rozdělení.

2. Pro každé téma vybere dirichletovo rozdělení $\varphi_k \sim Dir(\boldsymbol{\beta})$ přes slova w_i, kde $i \in < 1, V >$ a $\boldsymbol{\beta}$ je vektor hyperparametrů Dirichletova rozdělení.

3. Pro každou pozici i v korpusu:

   (a) Vyber téma z_i z pravděpodobnostního rozdělení θ_{d_i}

   (b) Vyber slovo w_i z pravděpodobnostního rozdělení φ_{z_i}



Obrázek 4.2: Grafická reprezentace LDA

Na obrázku 4.2 si můžeme povšimnout tří úrovní reprezentace LDA. Vektory parametrů $\boldsymbol{\alpha}$ a $\boldsymbol{\beta}$ se nacházejí na „korpusové“ úrovni, což znamená, že jsou nastaveny pouze jednou a platí pro celý korpus. Proměnné θ_{d_i} a φ_k jsou na „dokumentové“ úrovni a jsou tedy vzorkovány vždy pro daný dokument. Konečně proměnné z_i a w_i jsou na „slovní“ úrovni a jsou vzorkovány pro každé slovo znovu.

Na tento generativní proces můžeme nahlížet tak, že každý dokument lze vygenerovat z témat, která jsou v něm rozložena podle Dirichletova pravděpodobnostního rozdělení s vektorem parametrů $\boldsymbol{\alpha}$ a analogicky každé téma lze vygenerovat ze slov rozložených v něm podle Dirichletova pravděpodobnostního rozdělení s vektorem parametrů $\boldsymbol{\beta}$. Pravděpodobnost slova w_i v dokumentu můžeme pak určit pomocí následujícího vztahu:

$$p(w_i) = \sum_{j=1}^{K} p(w_i|z_i = j)p(z_i = j), \tag{4.7}$$

kde $p(w|z)$ indikuje, jak významné je slovo v daném tématu, zatímco $p(z)$ je významnost tématu v celém dokumentu (tato hodnota bude rozdílná pro všechna témata) [9]. Například v časopisu, který publikuje pouze články o matematice a neurovědě bychom mohli vyjádřit pravděpodobnostní rozdělení nad slovy ve dvou tématech – jedním spojeným s matematikou a druhým s neurovědou. Obsah těchto témat by byl dán výskytem slov s pravděpodobností $p(w_i|z_i)$. Téma matematika by mělo vysokou pravděpodobnost slov jako „teorie“, „funkce“, „prostor“, zatímco v tématu neurověda by byla nejpravděpodobnější slova jako „synapse“, „neurony“ nebo „hypothalamus“. To zda daný dokument obsahuje jedno z těchto témat, by pak bylo pravděpodobnostním rozdělením nad všemi tématy s pravděpodobností $p(z)$, která také uvádí, jak jsou mezi sebou témata v dokumentu promíchaná. Fakt, že jedno slovo v dokumentu může pocházet z několika různých témat odlišuje tento model od standardního Bayesovského klasifikátoru. Ten předpokládá, že každé jedno slovo v dokumentu patří právě do jednoho tématu. Dokumentu pak klasifikátor vždy přiřadí třídu, ze které bylo v dokumentu obsaženo nejvíce slov (nebo nejvýznamnějších slov atp.).

Tento generativní proces je ale založen na několika zjednodušujících předpokladech. Prvním z nich je Bag-of-Words hypotéza (viz kapitola 2.2), jak bylo zmíněno již výše. Dále se jedná například o dimenzi prostoru témat. Ta je od počátku známá a neměnná. Pro každý dokument z korpusu tedy od počátku předpokládáme, že obsahuje právě K témat, což nemusí být vždy nejlepší. Pokud nastavíme $K = 200$ a na vstupu bude nějaký krátký, například jednoodstavcový dokument, těžko v něm budeme moci rozdělit slova mezi všech dvě stě témat. Na důkladnější popis tohoto procesu se můžete podívat ve článku [7].

4.3.2 Použití pro extrakci klíčových slov

Pro určení „klíčovosti“ termu v dokumentu tedy dále předpokládejme, že máme dokument a generativní proces, který byl schopen daný dokument rozdělit do předem definovaného počtu K témat. Pro každé téma pak byla určena pravděpodobnost výskytu v dokumentu a stejně tak pro každé slovo máme k dispozici pravděpodobnosti výskytu v každém tématu.

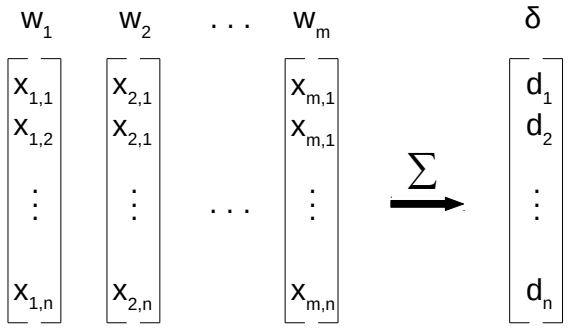
Pro unigramy je jako výsledné skóre použita samotná pravděpodobnost jejich výskytu ve všech tématech, jak je zapsáno v rovnici 4.7. V případě víceslovných termů je pak jako „míra klíčovosti“ použit geometrický průměr

pravděpodobností výskytu dílčích unigramů. Pro víceslovný n-gram $\boldsymbol{w}$ o délce n slov určíme skóre tedy jako

$$lda_{\boldsymbol{w}} = \left(\prod_{i=1}^{n} p(w_i) \right)^{\frac{1}{n}}.$$

4.4 Extrakce pomocí vektorové reprezentace

Při tomto postupu je každé slovo v dokumentu reprezentováno n-dimenzionálním vektorem a celý dokument součtem vektorů slov, které se v něm vyskytují. Pro dokument d o m slovech je tento postup znázorněn na obrázku 4.3.



Obrázek 4.3: Vektorová reprezentace dokumentu

Vektory slov nám utvářejí matici dokumentu, kterou označme W. K vytvoření vektoru dokumentu δ se v praxi častěji než normální součet (viz obrázek 4.3 výše) využívá vážený součet. Každý vektor je tedy vynásoben vahou slova, jež reprezentuje. V případě extrakce klíčových slov může být vahou například hodnota *tfidf*, čímž dojde k potlačení méně důležitých slov (spojek, částic atp.) a výsledný vektor dokumentu bude tedy z větší části tvořen vektory významnějších, v případě *tfidf* četnějších, slov. Matematicky tento proces můžeme zapsat jako

$$\delta = \sum_{i=1}^{m} W_i \cdot \lambda_i,$$

kde W_i představuje vektor a λ_i váhu slova i a m je počet unikátních slov v dokumentu d (tedy počet sloupců matice W).

Obecně existuje více různých metod, které převádějí slova na vektory tak, aby tyto vektory reprezentovaly určité vlastnosti daných slov. O metodě, jejíž výsledné vektory byly v práci použity, si můžete přečíst v následující kapitole 4.4.1.

4.4.1 GloVe – použitá reprezentace slov

GloVe (Global Vectors) je projektem Stanfordské univerzity a jedná se o algoritmus s učením bez učitele, který je trénován na agregovaných statistikách vzájemného výskytu slov získaných z korpusu. Výsledkem algoritmu je vektorová reprezentace slova představující lineární strukturu ve vektorovém prostoru – tedy jeho význam [18]. Všechny vytvořené vektory získané z několika různých korpusů spolu se zdrojovými kódy a demoaplikací jsou ke stažení na webové stránce projektu[1].

Postup vytváření vektorů ke slovům je založen na sestavení matice vzájemného výskytu (co-occurrence matrix) X, kde X_{ij} je vždy pravděpodobnost výskytu slova i ve stejném kontextu se slovem j. K sestavení této matice postačí jeden průchod přes celý korpus, nicméně pro velká data si i jeden průchod může vyžádat poměrně hodně času. Složitost modelu závisí na počtu nenulových prvků matice X a protože je toto číslo vždy menší než celkový počet různých slov v korpusu C, nemůže být složitost větší než $O(|C|^2)$. Nicméně typické velikosti slovníků použitých pro sestavování vektorů GloVe se pohybují okolo stovek tisíc slov a $|C|^2$ tedy může být v řádu stovek miliard, což je větší než většina korpusů. Model GloVe má dva hlavní případy užití – nalezení nejbližších sousedů a lineární substruktury [18].

Určením euklidovské vzdálenosti (nebo kosinové podobnosti viz kapitola 4.4.2) mezi dvěma vektory, poskytuje efektivně získanou informaci o jazykové nebo sémantické podobnosti slov, které vektory reprezentují. Čím větší shoda mezi vektory, tím jsou si slova podobnější. V některých případech mohou nejbližšími sousedy být taková slova, která se často nenacházejí v normálním lidském slovníku. Například pro slovo „žába“ můžeme dostat nejpodobnější slova jako „rosnička“, „ropucha“, „kuňka“ ale také vzácnější slova jako „leptodactylidae“ či „eleutherodactylus“. Příklad využití hledání nejbližších sousedů slova můžeme nalézt například u webového vyhledávače

[1] `http://nlp.stanford.edu/projects/glove/`

Google a jemu podobných nástrojů. Ty tak například nemusejí brát v úvahu jen uživatelem zadané slovo, ale i k slova sémanticky podobná a rozšířit tak výsledek hledání.

Metrika podobnosti použitá v případě hledání nejbližších sousedů slova poskytuje jednoduchou skalární hodnotu, která určuje příbuznost dvou slov. Toto zjednodušení ale může být v některých případech problematické protože slova, která takto porovnáváme téměř vždy vykazují více komplikovaných vztahů, než může být zachyceno jedním číslem. Například slovo „muž“ může být považováno za podobné slovu „žena“, protože obě popisují člověka. Na druhou stranu ale mohou být považována za opačná. Abychom byly schopni v takovém případě zachytit více rozdílů mezi oběma porovnávanými slovy, vyplatí se určit vektor rozdílu obou vektorů – nalezneme tak lineární substrukturu. Model GloVe je navržen tak, aby tento vektor rozdílu obsahoval co možná nejvíce informací. Základním konceptem, který rozlišuje slovo „muž“ od slova „žena“ je pohlaví. To však může být určeno i pomocí jiných ekvivalentních párů slov – „král“ a „královna“ nebo „bratr“ a „sestra“. Chceme-li uvést toto pozorování matematicky, můžeme říci, že vektory rozdílů muž-žena, král-královna a bratr-sestra budou zhruba shodné. Máme-li tedy slovo x a dvojici slov y-z, můžeme tímto postupem najít slovo w, které bude ve stejném vztahu (synonymum, antonymum atp.) ke slovu x, jako je slovo z ke slovu y.

4.4.2 Použití pro extrakci klíčových slov

Uvažujme, že máme slovník vektorů získaný nějakou metodou (viz kapitola 4.4.1). Následně spočteme vektor dokumentu δ tak, že provedeme vážený součet vektorů slov, vyskytující se v něm. Pakliže se některé slovo v dokumentu vyskytuje vícekrát, je možné se zachovat dvěma způsoby – každé slovo započítáme tolikrát, kolikrát se v dokumentu vyskytlo, nebo pouze jednou. V případě této práce je použit druhý postup protože jako váha je zde použita hodnota *tfidf*, která v sobě nese jakousi informaci o frekvenci termu.

To, jak moc důležitá jednotlivá slova v dokumentu jsou, můžeme pak určit například pomocí tzv. kosinové podobnosti. Jednoduše spočteme kosinus úhlu mezi vektorem dokumentu a vektorem slova. Kosinová podobnost je definována jako:

$$\text{sim}(A, B) = \frac{A \cdot B}{||A|| \cdot ||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}, \tag{4.8}$$

kde A a B jsou porovnávané vektory s dimenzí n.

Použití pro víceslovné termy

Pokud bychom chtěli určit vektor například nějakého bigramu nebo trigramu, můžeme tak učinit analogicky, jako jsme určili vektor dokumentu δ – tedy provedeme vážený součet všech dílčích vektorů reprezentující unigramy v n-gramu. Tento vektor pak dosadíme společně s vektorem dokumentu δ do metody *sim* (viz vzorec 4.8). Tím získáme důležitost daného víceslovného termu. V této práci je navíc tato hodnota ještě vynásobena hodnotou metriky *pmi* pro dané slovo. Výsledek pak určuje konečnou „klíčovost“ termu.

4.5 Zíbarova metoda extrakce klíčových slov

Zíbarova metoda extrakce klíčových frází (dále jen ZKEM – Zibar's Keyword Extraction Method) je metoda navržená pro účely této práce kombinující postupy metod TF-IDF, LDA a GloVe (viz předchozí kapitoly). Jako všechny tyto metody také u ZKEM se jedná o algoritmus s učením bez učitele. Hlavní myšlenkou této navržené metody je to, že každý příznak určený pro dané slovo (s_{tfidf}, s_{lda}, s_{glove}) v sobě nese jinou informaci o textu. Metoda TF-IDF upřednostňuje četnější slova v textu a tudíž příznak s_{tfidf} nese informaci o frekvenci slova v textu a v korpusu. Metoda LDA určuje pravděpodobnost slova v každém z automaticky vybraných témat v dokumentech. Dostáváme tak příznak s_{lda} založený na pravděpodobnostech výskytu slova v korpusu. Výsledek metody GloVe pak vypovídá o tom, jak moc vektor slova ovlivňuje výsledný vektor dokumentu. Příznak s_{glove} tedy nese informaci o podobnosti slova s celým dokumentem.

Cílem navržení této metody bylo zjistit, jestli při zkombinování metod fungujících na základě odlišných příznaků lze dosáhnout lepší úspěšnosti než při použití dílčích metod.

4.5.1 Výběr kandidátů

Všechny implementované metody byly trénovány na ručně upravených korpusech (viz kapitola 5.1). Oba korpusy byly již lematizovány a tokenizovány. Nebylo tedy potřeba při výběru klíčových frází žádné jiné manipulace s tvarem slov. Relevantní slova pro extrakci byla rozdělena mezerou a každý článek (případně věta) korpusu byl na jedné řádce (viz kapitola 5.1).

Jako kandidáti na klíčové fráze byly uvažovány unigramy, bigramy a trigramy, protože tvoří největší procentuální zastoupení v odpovědích v gold datech (viz tabulka 6.1). Z textu zpracovávaného dokumentu byly tedy vybrány n-gramy až do délky 3, které nezačínají ani nekončí na stop-slovo, nebo, v případě unigramů, nejsou stop-slovem.

4.5.2 Ohodnocení kandidátů

Jak bylo řečeno, metoda ZKEM kombinuje tři metody – TF-IDF, LDA a GloVe. Skóre spočtené pro term každou z těchto metod je pak použito jako jeden příznak pro výsledné skóre. Tyto příznaky jsou pak zkombinovány do jedné hodnoty podle vzorce

$$s = s_{\text{tfidf}}^{x} \cdot s_{\text{lda}}^{y} \cdot s_{\text{glove}}^{z},$$

kde x, y a z jsou jejich váhy. Hodnoty vah byly nastaveny na $x = 1$, $y = 0$ a $z = 0$ podle výsledků experimentu popsaného v kapitole 7.3.

Dalšími použitými příznaky pro určení konečného skóre slova jsou váhy kapitol (nebo ohodnocených částí článku), ve kterých se daný term vyskytl. V této práci byl článek vždy rozdělen na tři části – nadpis (s vahou 5), abstrakt (s vahou 3) a „to ostatní“ (s vahou 1) (více informací viz kapitola 7.2). Skóre slova je pak vynásobeno součinem vah kapitol, ve kterých se vyskytlo. Jako poslední příznak je použita hodnota *pmi* (viz kapitola 4.1), která, pro víceslovné termy, udává informaci o tom, „jak moc“ k sobě všechny unigramy termu patří. Zabrání se tak možnosti, aby byl jako klíčová fráze vybrán term, který byl sice ohodnocen vysokým skórem, ale sémanticky nedává smysl.

Výsledná ohodnocující funkce metody ZKEM má tedy pro term t následující tvar:

$$f(t) = \text{pmi}_t \cdot \prod_{c \in C_t} w_c \cdot s^x_{\text{tfidf, t}} \cdot s^y_{\text{lda, t}} \cdot s^z_{\text{glove, t}},$$

, kde C_t je množina indexů kapitol, ve kterých se term t vyskytl, c je index kapitoly, w_c udává váhu dané kapitoly, x, y, z jsou koeficienty příznaků a $s_{xxx,t}$ udává skóre určené příslušnou metodou pro term t.

Aplikujeme-li hodnoty pro x, y a z získané z experimentu (viz kapitola 7.3), lze tvar zjednodušit na:

$$f(t) = \text{pmi}_t \cdot \prod_{c \in C_t} w_c \cdot s^x_{\text{tfidf, t}}. \tag{4.9}$$

5 Data

5.1 Vytvoření trénovacích korpusů

V práci byly použity dva korpusy pro trénovaní – originální trénovací korpus poskytnutý organizátory soutěže SemEval 2010 a korpus Wikipedie, který byl vybrán proto, aby se dalo zjistit, jaká bude úspěšnost metod trénovaných na obecných datech. Korpus Wikipedie obsahuje přes osm a půl milionu článků z různých témat, což by mělo zajistit metodám větší obecnost a měly by fungovat srovnatelně na článcích z různých oblastí. Wikipedia na svých stránkách poskytuje volně stažitelný dump[1] všech článků pro všechny jazyky. Tato práce je zaměřena na extrakci klíčových slov z anglicky psaných textů, a proto byl stažen dump anglické Wikipedie. Jedná se o surová XML data opatřená navíc meta značkami obsahujícími informace o autorovi článku, datu vydání, datech úprav, podobných článcích, referencích článku atp. Tyto informace nejsou pro úkol extrakce klíčových slov příliš podstatná a formátování textu pomocí html značek také v implementovaných metodách není bráno v potaz (narozdíl od některých metod zmíněných v kapitole 3.3). Proto bylo potřeba z tohoto korpusu tento šum odstranit.

K vybrání textu článků ze staženého dumpu byl použit Wikipedia Extractor[2], který celý korpus rozdělí do souborů po zvolené velikosti. V každém souboru je pak vždy několik článků oddělených od sebe značkou `<doc>`. Tyto soubory byly následně spojeny do jednoho, a to tak, že každý článek wikipedie byl vložen na jeden řádek.

Jak bylo řečeno v kapitole 2.3 je před vlastním zpracováváním textu (sčítání frekvencí, analýza typů slov atp.) nutno tento korpus ještě lematizovat. K tomu byla v této práci využita knihovna Stanford CoreNLP [14] vytvořená na Stanfordské univerzitě kolektivem lidí, zabývajícím se problémy zpracování přirozeného jazyka. Tato knihovna poskytuje sadu nástrojů pro hloubkovou analýzu textu – určování základního tvaru slov (lemmat), slovního druhu (part-of-speech), druhu pojmenované entity (zda-li je slovo názvem firmy, jménem osoby atp.), normalizování datumů, časů nebo číselných kvantifikátorů, označování gramatických částí vět, frází nebo slovních závislostí, označování slov, která odkazují na stejné entity, označování sentimentu slov, a spoustu dalšího viz [27].

Při lematizaci korpusu Wikipedie byl také vytvořen slovník obsahující dvě stě tisíc nejčetnějších podstatných a přídavných jmen v celém korpusu, která navíc obsahují jen písmena anglické abecedy (i v anglické Wikipedii se vyskytují např. slova s přehlasovanými souhláskami atp.), pomlčku, středník a číslici. Samozřejmě slova začínající pomlčkou, středníkem nebo číslicí nebyla brána v úvahu. Číslice byly ponechány z toho důvodu, že je některá klíčová slova v gold datech obsahují (například „3d“, „p2p“ či „iPhone 5s“). Slovník byl poté manuálně rozšířen o některá stop-slova, která můžeme nalézt v gold-datech. Jednalo převážně o předložky a členy jako například „the“, „for“, „of“. Dokumenty, které byly po odstranění irelevantních slov kratší než 150 znaků, byly následně ještě odstraněny.

Podobně jako korpus Wikipedie byl předzpracován i korpus SemEvalu. Ten se skládal celkem ze 144 článků (viz kapitola 5.2). Namísto toho, aby byl každý článek na jednom řádku, byl každý z nich rozdělen po větách a ty byly uloženy každá na jeden řádek. To z důvodu poskytnutí většího kontextu metodám. K filtraci slov, která nejsou relevantní, byl použit slovník vytvořený na korpusu Wikipedie a v korpusu SemEvalu byla ponechána tedy jen slova vyskytující se v tomto slovníku.

Průběh změn obou korpusů je zachycen v tabulkách 5.2 a 5.1, kde jsou zaznamenány počty řádků, slov a velikost celého korpusu v jednotlivých krocích jejich předzpracování.

korpus	počet řádků	počet slov	velikost
originální text	154 842	1 159 341	6,7 MB
profiltrovaný text	36 296	513 797	3,1 MB

Tabulka 5.1: Statistiky korpusu SemEval

korpus	počet řádků	počet slov	velikost
původní XML	923 584 949	6 010 421 252	55 GB
pouze články	8 644 167	2 310 422 472	12 GB
profiltrovaný text	4 442 638	1 303 082 835	7,7 GB

Tabulka 5.2: Statistiky korpusu Wikipedie

[1] `https://dumps.wikimedia.org`
[2] `http://medialab.di.unipi.it/wiki/Wikipedia_Extractor`

5.2 Data ze soutěže SemEval 2010

Data poskytnutá pořadateli soutěže SemEval byla sesbírána z ACM elektronické knihovny (převážně z konferenčních a work-shop materiálů). Jedná se o anglické články rozsahem mezi šesti a osmi stránkami textu zahrnujícím tabulky i obrázky. Pro zajištění různorodosti korpusu byly vybrány články z různých vědních oborů patřících do čtyř 1998 ACM kategorií – C.24 (Distributed Systems), H3.3 (Information Search and Retrieval), I2.11 (Distributed Artificial Intelligence – Musliagent Systems) a J4 (Social and Behavioral Sciences – Economics). Z vybraných článků byly posléze sestaveny tři datasety (trial, training a test) se stejným procentuálním rozložením těchto témat (viz tabulka 5.3). Trial dataset je pouze podmnožinou training datasetu a jedná se o ukázku formátu dat (pro účely soutěže). Protože původní formát článků byl PDF, byly převedeny nástrojem `pdftotext` do prostého textu pro lepší následné zpracovávání. Slova, původně rozdělena pomlčkou v původním PDF souboru do dvou řádků, byla při tomto procesu složena opět do jednoho celku [11].

Dataset	Celkem	Kategorie			
		C	H	I	J
Trial	40	10	10	10	10
Training	144	34	39	35	36
Test	100	25	25	25	25

Tabulka 5.3: Počty dokumentů v jednotlivých datasetech

Všechny vybrané články obsahují klíčová slova přiřazená autorem (součást originálního PDF dokumentu) a navíc k nim byla dodána klíčová slova přiřazená čtenářem. Každý z padesáti čtenářů dostal pět článků, ze kterých měl vybral klíčové fráze, které se přímo vyskytují v textu dokumentu (včetně nadpisů a popisků u tabulek či obrázků). Bohužel ne všechny vybrané fráze, se skutečně v textu vyskytovaly. V průměru se jednalo přibližně o 15 % ze všech vybraných frází, které se v textu dokumentů nevyskytovaly. V případě klíčových slov poskytnutých autorem bylo ovšem takovýchto frází více (celkem 19 %). Z tohoto důvodu je maximální možná úspěšnost metod testovaných na těchto datasetech 85 % pro klíčové fráze přiřazené čtenářem a 81 % pro klíčové fráze přiřazené autorem. Obě množiny klíčových slov byly následně ještě zkombinovány do třetího souboru odpovědí (viz tabulka 5.4).

Některé fráze se ve článcích mohou vyskytovat ve více podobách. Proto byla gold data klíčových frází doplněna ještě o další možné tvary fráze. Akceptovatelné tvary jsou „*A of B* → *B A*“ (např. policity of school, school policity) a „*A's B* → *A B*“ (např. school's policity, school policity). V klíčových frázích poskytnutých autorem jsou takovéto fráze 3, ve čtenářem přiřazených jich je 44 a v kombinovaných tedy celkem 47. Celkové počty frází přiřazených čtenářem, autorem a kombinací těchto dvou můžete vidět v tabulce 5.4.

Dataset	Autor	Čtenář	Kombinované
Trial	149	526	621
Training	559	1824	2223
Test	387	1217	1482

Tabulka 5.4: Počty klíčových frází

Aby se předešlo neúplné validaci z důvodu, že metoda poskytla některou z frází v jiném tvaru (jiném pádu nebo číslu), než ve kterém se vyskytuje v množině odpovědí, byla pořadateli tato gold data poskytnuta v ostemované i lematizované verzi. Při vlastní validaci je pak možné si vybrat buď lematizovaná nebo ostemovaná gold data. V této práci byly používány ostemované verze kandidátů na klíčová slova.

6 Testování

Podle pokynů pro validaci navržených systémů ze soutěže SemEval 2010 (viz článek [11]) bylo ze všech kandidátů poskytnutých některou z implementovaných metod vybráno 15 nejlepších podle skóre, které k nim bylo přiřazeno. Pro správné vyhodnocení úspěšnosti implementovaných metod byl použit validační skript poskytnutý organizátory soutěže, který byl psaný v programovacím jazyce Perl. Ten jako vstupní parametr požaduje textový soubor s lematizovanými nebo ostemovanými klíčovými frázemi, které vybral testovaný systém. Formát souboru musí být následovný:

```
FILENAME␣:␣KEYWORD_LIST
```

K ostemování vybraných kandidátů byla použita javovská implementace Porter Stemmeru[1], který byl (ve verzi psané v Perlu) použit i při soutěži. Při použití jiného stemmeru by bylo možné, že by docházelo ke zbytečným chybám vlivem jiného tvaru slov. Na příklad vygenerovaného souboru s klíčovými frázemi se můžete podívat níže:

```
C-1 : uddi,dht,registri,uddi registri,servic,grid,uddi kei,...
C-3 : processor,applic,node,resourc,adapt,grid,cluster,...
C-4 : packet,loss,probabl,frame,conceal,burst,scheme,...
C-6 : content,polici,storag,manag,spectrum,store,system,...
J-30 : social-choic function,optim,mechan,social-choic,...
J-31 : strategi,nash equilibrium,optim,player,game,...
```

Každá testovaná metoda tedy musela jako svůj výstup vygenerovat takovýto soubor, na kterém byl posléze puštěn validační skript, jehož výstup byl nepatrně pozměněn tak, aby vytiskl zároveň hodnoty pro micro a macro F-Skóre (viz kapitola 3.2) a také jméno testované metody. Bylo tak učiněno z důvodu přehlednosti testování metod při dávkovém spouštění. Obě metriky byly určeny vždy pro top 5, 10 a 15 kandidátů načtených z výše zmíněného souboru a pro všechny tři soubory s odpověďmi (klíčové fráze přiřazené autorem, čtenářem a kombinací předešlích dvou). Více informací o procesu určování úspěšnosti metod je sepsáno v kapitole 6.1.

[1] `http://tartarus.org/~martin/PorterStemmer`

6.1 Proces evaluace

Práce byla testována na testovací množině poskytnuté organizátory soutěže SemEval 2010 (viz tabulka 5.3). Tradičně se úspěšnost metod automatické extrakce klíčových slov stanovuje pomocí počtu top-N kandidátů, kteří se přesně shodují s gold daty. V některých případech se do celkového počtu správně extrahovaných klíčových frází mohou započítávat i fráze, které se s gold daty nezcela shodují. Může se jednat například pouze o část správné klíčové fáze nebo například celou frázi s následujícím či předchozím slovem (tzv. *near-miss*) atp. Případně se někdy také za správné považují sémanticky podobné fráze. K tomu je ale zapotřebí natrénování validátoru na velkém korpusu či sestavení slovníku sémanticky podobých frází [11].

Pro účely soutěže SemEval 2010, a tedy i pro tuto práci, byl použit tradiční postup – tedy určením počtu frází poskytnutých metodou, které se přesně shodují s klíčovými frázemi v gold datech a spočtením micro-average *precision* (P), *recall* (R) a *F-Skóre* s parametrem $\beta = 1$ (F). Vyhodnocování metod probíhá na testovacím datasetu viz kapitola 5.2, který obsahuje celkem 100 dokumentů ze všech vybraných kategorií (viz tabulka 5.3). Hodnoty P, R a F byly určeny pro nejlepších 5, 10 a 15 kandidátů, vybraných metodou, a to pro každou ze tří množin odpovědí (klíčové fráze přiřazené autorem, čtenářem a kombinací dvou předešlých).

Pro účely této práce byly jako klíčové fráze dokumentu uvažovány n-gramy až do délky $n = 3$. Celkové rozdělení počtů jednotlivých n-gramů v testovacích datech zobrazuje tabulka 6.1 dále. Z té je patrné, že největší procento zastoupení mají v gold datech bigramy a po té trigramy následované unigramy. Pro n-gramy delší než tři slova bylo rozhodnuto, že je nemá smysl z dokumentů extrahovat. Pokud by se extrahovaly docházelo by k poklesu hodnoty *precision*, a tím i ke snížení celkové úspěšnosti metody.

n	Autor	Čtenář	Kombinované
1	79	185	235
2	146	448	536
3	46	198	228
4	6	59	65
5	2	22	24
6	0	2	2
7	0	1	1

Tabulka 6.1: Četnosti n-gramů v testovacím datasetu

6.2 Parametry testovaných metod

Každá z testovaných metod – TF-IDF, LDA a GloVe byla spouštěna s různými parametry, aby bylo dosaženo co nejlepších výsledků. Navíc byly také trénovány na obou předzpracovaných korpusech (viz kapitola 5.1) pro možnost porovnání úspěšnosti při natrénování na obecných a specifických datech. Pro TF-IDF byly uvažovány dva přístupy. Prvním byl ten, že hodnota *tfidf* víceslovného n-gramu je určována stejně jako pro unigramy. Druhým přístupem (označeným jako metoda TF-IDF-avg) bylo určování hodnoty *tfidf* víceslovného termu jako geometrického průměru hodnot *tfidf* jednotlivých unigramů, ze kterých je term složen (viz vzorec 4.6).

Měnícím se parametrem pro metodu LDA byl pouze počet témat. V této práci bylo vyzkoušeno 50, 100 a 200 témat. Ostatní parametry metody jako například parametry Dirichletova rozdělení α a β zůstaly neměnné.

Metodu GloVe není, na rozdíl od předchozích metod TF-IDF a LDA, potřeba trénovat na připraveném korpusu. Vektory slov byly již vytvořeny a jsou volně ke stažení na oficiálních stránkách projektu[1]. Jediným měnícím se parametrem metody byla tedy pouze použitá sada vektorů. Všechny použité vektory můžete vidět v tabulce 6.2 dále.

[1] `https://nlp.stanford.edu/projects/glove/`

Počet slov	Dimenze	Zdroj
840B	300	Common Crawl
42B	300	Common Crawl
27B	25	Twitter
27B	50	Twitter
27B	100	Twitter
27B	200	Twitter
6B	50	Wikipedia
6B	100	Wikipedia
6B	200	Wikipedia
6B	300	Wikipedia

Tabulka 6.2: Použité vektory GloVe

Dále pro přehlednost nechť je zavedeno značení pro jednotlivé verze metod TF-IDF, TF-IDF-avg, LDA a GloVe jako `tfidf`, `tfidf-avg`, `lda(z)`, kde z je počet témat, a `glove(X-Yd)`, kde X je počet slov korpusu, ze kterého byly použité vektory sestavovány, a Y je dimenze sestavených vektorů.

7 Experimenty

Po implementování metod TF-IDF, LDA, GloVe a ZKEM bylo možné jejich úspěšnost o něco zvýšit. Byly tedy navrženy tři experimenty (viz následující kapitoly), které měly najít optimální nastavení pro některou fázi extrakce klíčových slov tak, aby bylo dosaženo maximální možné úspěšnosti. Jako referenční metoda pro testování posloužila ve většině případů metoda TF-IDF protože dosahovala v základní implementaci nejlepších výsledků. Pro každý experiment pak byla určena hodnota úspěšnosti pro kombinovaná klíčová slova a top 15 kandidátů (více viz kapitola 6). Tato hodnota pak byla porovnávána při spuštění metody s různými parametry.

7.1 Profiltrování korpusu Wikipedie

Jako první experiment pro zvýšení úspěšnosti implementovaných metod bylo navrženo filtrování korpusu Wikipedie. Důvodem byla malá, takřka žádná, úspěšnost metody LDA při natrénovaní na tomto korpusu. To bylo přičítáno velké rozmanitosti témat, která Wikipedie obsahuje. Testovací množina narozdíl od Wikipedie obsahuje, stejně jako trénovací množina ze soutěže SemEval 2010, články pouze ze 4 kategorií, které jsou navíc všechny technického rázu.

Při natrénování metody LDA například pro 200 témat na korpusu Wikipedie, dojde k tomu, že metoda vybere z celého korpusu daný počet témat, vypočítá pravděpodobnosti výskytu daného tématu a pravděpodobnosti výskytu jednotlivých slov v něm. Ve výsledku tedy máme spočítané pravděpodobnosti pro 200 témat, ale dokumenty z testovací množiny jsou tématicky podobné například jen se dvěma z nich. Při vlastní extrakci pak metoda LDA nedokáže rozlišit mezi slovy z těchto dvou témat.

Z korpusu Wikipedie tedy bylo potřeba odstranit články, které nejsou tématicky podobné s články z testovací množiny. Jako tématicky podobné články byly označeny ty, které obsahovaly alespoň 10, 20 nebo 30 procent slov, která jsou uvedena v gold datech jako správná klíčová slova k testovací množině dokumentů (gold slov). Z původního předzpracovaného korpusu Wikipedie byly tedy vytvořeny tři další korpusy s články, které splňovaly podmínku minimálního obsahu těchto gold slov viz tabulka 7.1. Pro možnost

porovnání je přidán původní korpus Wikipedie obsahující všechny články a označený jako /.

Na každém nově vytvořeném korpusu byla natrénována metoda LDA pro 50, 100 a 200 témat. Pro porovnání byla na všech korpusech vyzkoušena také metoda TF-IDF. Všechny naměřené výsledky jsou zaznamenány v tabulce 7.2. Jako označení korpusu je zde zvolen minimální obsah gold slov v každém článku.

stupeň filtrování	počet dokumentů	počet slov	velikost
/	4 442 638	1 303 082 835	7,7 GB
10 %	191 919	414 774 489	2,4 GB
20 %	18 734	100 678 124	596 MB
30 %	1 163	12 119 703	72 MB

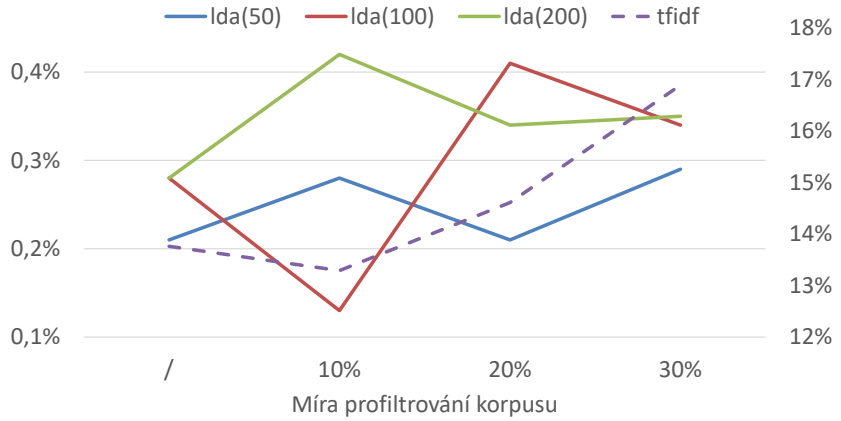
Tabulka 7.1: Profiltrované korpusy Wikipedie

metoda	stupeň filtrování			
	/	10 %	20 %	30 %
tfidf	13,76	13,29	14,61	16,89
lda(50)	0,21	0,28	0,21	0,29
lda(100)	0,28	0,13	0,41	0,34
lda(200)	0,28	0,42	0,34	0,35

Tabulka 7.2: Úspěšnost metod na profiltrovaných korpusech Wikipedie [%]

Hodnoty z tabulky 7.2 jsou zobrazeny také v grafu na obrázku 7.1. Na ose X jsou všechny vytvořené korpusy, na hlavní ose Y jsou hodnoty úspěšnosti metod LDA a vedlejší osa Y pak zobrazuje hodnoty metody TF-IDF.

Jak je vidět z tabulky a grafu, úspěšnost pro metodu LDA se nijak výrazně nezlepšila. To je způsobeno tím, že metoda LDA vyzdvihuje nejpravděpodobnější slova ve všech tématech (viz kapitola 4.3). Ta jsou ovšem mnohem více obecná, zatímco slova označená jako klíčová, jsou spíše konkrétnější. Jako příklad se můžeme podívat na klíčová slova, která metoda vybrala z prvního dokumentu testovací množiny a porovnat je se klíčovými slovy, která měla být správně určena (slova jsou uvedena v ostemovaném tvaru):



Obrázek 7.1: Úspěšnosti zvolených metod ve vyfiltrovaném korpusu

Slova vybraná metodou:
`system, problem of servic, internet, internet and run, network of multipl, network, set of search, network of uddi, architectur to support, user, system on top, servic, match for system, servic to uddi, includ`

Správná klíčová slova:
`grid servic discoveri, uddi, distribut web-servic discoveri architectur, dht base uddi registri hierarchi, deploy issu, bamboo dht code, case-insensit search, queri, longest avail prefix, qo-base servic discoveri, autonom control, uddi registri, scalabl issu, soft state, dht, web servic, grid comput, md, discoveri`

Při srovnání obou množin vidíme, že například pro slovo „system“ je zde několik částečně shodných slov, ale jinak byla vybrána slova, která se sice v článku vyskytla, dá se říci, že se jedná o častá slova v technických dokumentech, avšak jako klíčová slova nebyla označena ani čtenáři článku ani jeho autorem. Z toho se dá tedy usoudit, že to, jak je slovo pravděpodobné v dokumentu, nemá na jeho „klíčovost“ vliv a je lepší klíčová slova z článku extrahovat například na základě jejich četnosti, jako je tomu u metody TF-IDF.

7.2 Vážení kapitol

Jak bylo zmíněno v kapitole 3.3, některé z metod, které byly vytvořeny pro účely soutěže SemEval 2010, používaly při načítání dokumentů z testovací množiny speciálně vytvořený parser (nebo v jednodušším případě regulární výraz), který načítaný článek rozdělil do automaticky nalezených kapitol. (abstrakt, intro, závěr,...). Z těchto kapitol byly některé vyloučeny a vlastní algoritmus metody byl pak testován na zbylých. Případně byl použit příznak složený z binárních ukazatelů a pokud se slovo v dané kapitole vyskytlo, byl patřičně změněn příslušný ukazatel.

Pro účely této práce byl navržen jednoduchý parser, který testovaný dokument rozdělí na tři části – nadpis, abstrakt a "to ostatní". Zároveň se některé kapitoly z článku vyloučily jako například poděkování nebo reference. Také meta informace o článku, které jsou uvedeny vždy před abstraktem byly odstraněny. Jednalo se převážně o informace o autorech článku, kde a kdy byl článek vydán, o datech úprav atp. Všechny tyto odstraněné části článku nebyly brány v úvahu, protože nebudou obsahovat žádná klíčová slova a zanášely by tak značnou chybu do celkového určování skóre jednotlivých kandidátů na klíčová slova.

Hlavní myšlenkou rozdělení článku do těchto částí bylo to, že pokud se slovo vyskytlo v abstraktu, bude patrně důležitější než slova obsažená jinde. Autoři samovolně vybírají do abstraktu taková slova, aby článek dobře vystihovala. Podobně je tomu i s nadpisem. Můžeme také předpokládat, že slovo v nadpisu je patrně o něco více důležitější než slovo v abstraktu. Dalším předpokladem může být, že pokud se slovo vyskytuje v nadpisu, abstraktu i těle, bude patrně jedno z nejdůležitějších a je tedy pravděpodobné, že bude i klíčovým slovem. Otázkou tedy zůstává, o kolik jsou slova v abstraktu a nadpise významnější než slova obsažená v těle článku. To bylo potřeba zjistit experimentálně.

Pro tento experiment byla vybrána metoda TF-IDF pro svoji nejvyšší úspěšnost při natrénování na korpusu SemEvalu. Při experimentu byla pak porovnávána procentuální úspěšnost pro nejlepších 15 kandidátů a kombinovaná gold data (více v kapitole 6). Určování celkového skóre bylo tedy lehce pozměněno tak, že skóre spočtené metodou TF-IDF bylo vynásobeno součinem vah částí článku, ve kterých se dané slovo vyskytlo. Nejprve byl místo součinu vah použit jejich součet ale protože bylo dosahováno menší úspěšnosti, nebyl tento postup použit.

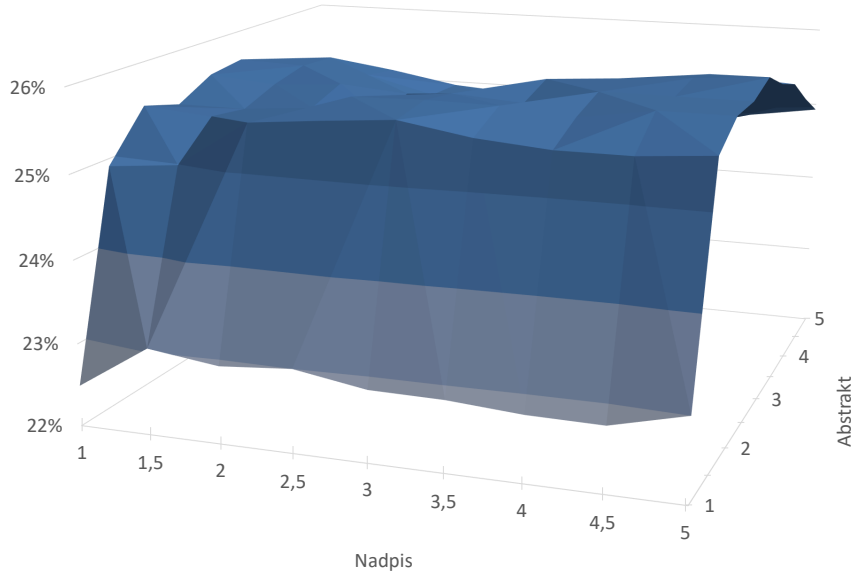
Metoda TF-IDF byla hrubou silou pouštěna pro různé kombinace vah nadpisu a abstraktu z intervalu $< 1; 5 >$ (s krokem 0,5) aby se zjistilo, pro jakou kombinaci je dosaženo nejvyšší úspěšnosti. Úspěšnost metody pro každou kombinaci vah je zaznamenána v tabulce 7.3.

Pro lepší zvýraznění toho, jak vybrané váhy ovlivňují výslednou úspěšnost se můžete podívat na graf na obrázku 7.2, kde na přední ose jsou naneseny volené váhy pro nadpis, na pravé ose jsou váhy pro abstrakt a celková úspěšnost pro danou kombinaci vah určuje výška grafu.

A \ N	1	1,5	2	2,5	3	3,5	4	4,5	5
1	22,49	23,04	22,93	23,00	22,86	22,85	22,79	22,78	23,00
1,5	24,91	24,99	25,55	25,63	25,70	25,56	25,49	25,49	25,56
2	25,47	25,40	25,47	25,75	25,77	25,76	25,69	25,82	25,82
2,5	25,33	25,33	25,40	25,34	25,69	25,63	25,82	25,82	25,82
3	25,56	25,49	25,55	25,48	25,42	25,77	25,83	25,89	**25,95**
3,5	25,62	25,59	25,52	25,31	25,31	25,62	25,66	25,79	25,72
4	25,35	25,56	25,44	25,29	25,23	25,23	25,48	25,58	25,57
4,5	24,81	24,94	24,96	24,88	24,81	24,81	24,81	24,93	25,23
5	24,88	24,86	24,95	24,88	24,88	24,81	24,74	24,80	24,93

(N: nadpis, A: abstrakt)

Tabulka 7.3: Procentuální úspěšnost metody TF-IDF v závislosti na vahách pro nadpis a abstrakt



Obrázek 7.2: Závislost úspěšnosti na zvolených vahách pro nadpis a abstrakt

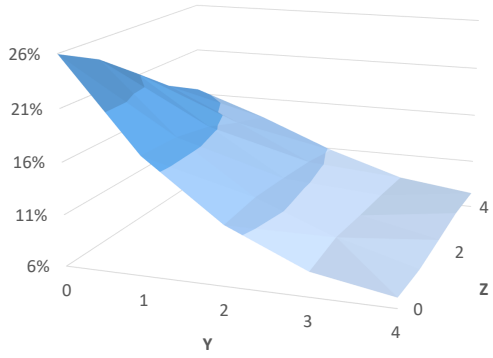
Z tabulky a grafu je patrné, že nastavením ideální váhy pro slova v nadpisu a abstraktu můžeme úspěšnost extrakce klíčových slov zvýšit přibližně o 3,5 %. Ideální nastavení vah se pak pohybuje okolo hodnot 2 - 3 pro abstrakt a 3,5 - 5 pro nadpis. Nejlepších výsledků bylo při testování dosahováno nastavením váhy 3 pro abstrakt a 5 pro nadpis. Nejhorší výsledky jsou, jak se dalo čekat, v případě, že se uvažují slova v abstraktu za stejně důležitá jako ostatní – tedy váha je 1. Pokud jsou slova v abstraktu zdůrazněna jakoukoliv vahou větší než 1, je dosahováno lepší úspěšnosti. Optimální vektor vah je tedy $[5, 3, 1]$ pro nadpis, abstrakt a „to ostatní“.

7.3 Nalezení vah příznaků metody ZKEM

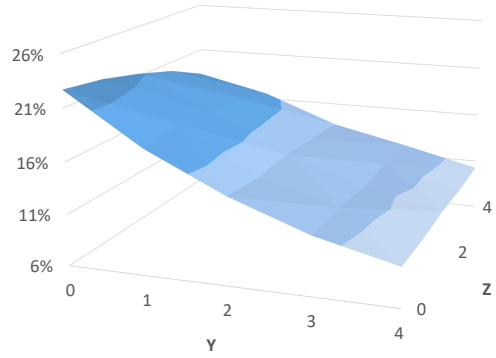
Jako další experiment bylo navrženo hledání optimálního nastavení koeficientů příznaků metody ZKEM (viz kapitola 4.5). Pro koeficient x byly voleny hodnoty z množiny $\{1; 2; 3; 4\}$ a pro koeficienty y a z z množiny $\{0; 1; 2; 3; 4\}$. U koeficientu x nebyla nulová hodnota uvažována z toho důvodu, že příznak s_{tfidf}, který je tímto koeficientem ovlivňován, má na celkové skóre největší příznivý vliv a tudíž se nesmí zanedbat. Celkem bylo tedy vyzkoušeno 100 kombinací všech tří koeficientů.

Na grafech na obrázku 7.3 můžeme vidět závislost úspěšnosti metody ZKEM na volených hodnotách pro koeficienty y a z (koeficient x je vždy pevně zvolen). Můžeme si povšimnout že, nejvyšších hodnot dosahují všechny grafy v bodě $[0; 0]$, čili nenulové hodnoty nabývá jen pevně zvolený koeficient x. Čím vyšších hodnot koeficienty y a z nabývají, tím je úspěšnost metody menší. Jedná se tedy o nepřímou úměru. S rostoucími hodnotami x se průběh grafu nijak výrazně nemění. Jediná změna nastává ve zmenšování sklonu plochy grafu. Optimální nastavení vah příznaků s_{tfidf}, s_{lda} a s_{glove} je tedy podle tohoto experimentu $x = 1$, $y = 0$ a $z = 0$.

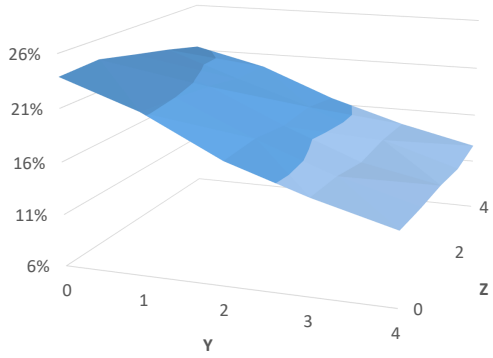
Z tohoto výsledku ovšem vyplývá, že optimálním nastavením koeficientů příznaků u metody ZKEM, metoda zanedbá příznaky pro možnou klíčovou frázi určené metodami LDA a GloVe a její postup je tedy vlastně degradován na postup metody TF-IDF, jejíž příznak je, jako jediný, brán v potaz. Tím tedy nemůže metoda ZKEM dosáhnout lepší úspěšnosti než metoda TF-IDF s navrženými úpravami.



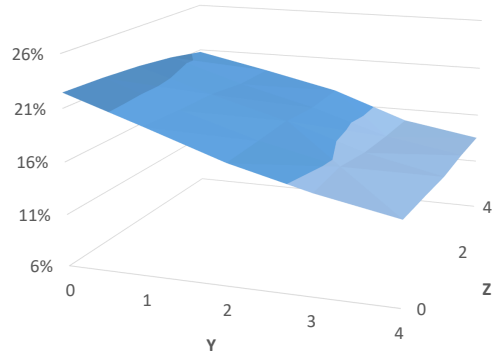
(a) Pro $x = 1$



(b) Pro $x = 2$



(c) Pro $x = 3$



(d) Pro $x = 4$

Obrázek 7.3: Závislost úspěšnosti metody ZKEM na volených koeficientech příznaků

8 Výsledky

Výsledky všech implementovaných metod byly srovnávány s výsledky mezinárodní programátorské soutěže SemEval[1], která je zaměřena na algoritmy z oblasti zpracování přirozeného jazyka. V roce 2010 byla jedním z úkolů právě extrakce klíčových slov (viz [11]), které se účastnilo celkem devatenáct týmů. Nejlepší navržená metoda dosáhla v průměru 27,5 % úspěšnosti. Na výsledky ostatních metod se můžete podívat v příloze A. Některé z nejlepších metod byly také popsány v kapitole 3.3. V této práci byla pro možnost porovnání implementovaných metod použita stejná testovací data, metrika i stejný postup vyhodnocování výsledků jako při soutěži.

V této kapitole jsou uvedeny výsledky všech samostatně naměřených metod – TF-IDF, LDA, GloVe a ZKEM (viz kapitola 4). Všechny metody byly otestovány podle postupu uvedeném v kapitole 6.1. Při testování metod TF-IDF, LDA a GloVe byl použit stejný parser dokumentů jako pro metodu ZKEM se stejným nastavením vah jednotlivých částí článků, jak bylo popsáno v kapitole 7.2. Bylo tak učiněno z důvodu dosahování lepších výsledků. Navíc je také u těchto metod použita metrika *pmi* stejným způsobem jako u metody ZKEM.

Všechny implementované metody byly natrénovány na obou předzpracovaných korpusech – SemEvalu a Wikipedie (viz kapitola 5.1). Testování pak probíhalo ve dvou fázích – 1) extrahovány byly z dokumentů pouze unigramy a 2) extrahovány byly uni-, bi- i trigramy. Důvodem tohoto dvoufázového testování bylo to, aby bylo možné posoudit, o kolik procent se zvýší úspěšnost dané metody, pokud jsou extrahovány i víceslovné termy.

Ve výsledkových tabulkách jsou zaneseny výsledky pro všechny soubory s odpověďmi. Pro každý dataset je pak určena úspěšnost pro top 5, 10 a 15 klíčových slov určených příslušnou metodou. To je vždy reprezentováno hodnotou *micro F-Skóre* (F). Pro úplnost jsou také uvedeny hodnoty *precision* (P) a *recall* (R).

Pro lepší orientaci v tabulkách zaveďme následující označení kombinací korpusu, na kterém byla metoda trénována, maximální délky n-gramu, který byl extrahován, a souboru s odpověďmi jako:

[1] `http://semeval2.fbk.eu/semeval2.php`

`[S/W] - [1/3] (- [A/Č/K])`,

kde první část tedy označuje korpus, na kterém byly metody trénovány (`S` pro SemEval a `W` pro Wikipedii). Druhá část je maximální hodnota n pro n-gramy (`1` znamená že byly extrahovány pouze unigramy a `3`, že se extrahovaly uni-, bi- a trigramy). Poslední část nechť je nepovinná a značí gold data, se kterými byli kandidáti metody porovnáváni (A – autor, Č – čtenář a K – kombinace obou dvou).

8.1 Výsledky metod

8.1.1 Výsledky metody TF-IDF

Jak bylo řečeno v kapitole 6.2, metoda TF-IDF byla testována ve dvou verzích – klasické verzi tfidf a verzi označené jako tfidf-avg (viz kapitola 4.2). Algoritmus metody TF-IDF-avg je pro unigramy totožný s algoritmem původní metody TF-IDF. Výsledky obou metod naměřené při extrakci unigramů budou tedy shodné. Rozdílná úspěšnost metod se dá očekávat až pro víceslovné termy. V následujících tabulkách jsou zachyceny výsledky pro obě verze metody.

#	D	Top 5 kandidátů			Top 10 kandidátů			Top 15 kandidátů		
		P	R	F	P	R	F	P	R	F
S - 1	A	9,80	11,78	10,41	7,20	16,76	9,82	5,20	18,43	7,92
	Č	11,60	4,97	6,92	9,40	8,00	8,58	7,40	9,37	8,21
	K	17,60	6,12	9,04	14,00	9,85	11,48	10,87	11,40	11,04
S - 3	A	18,20	24,55	20,35	13,00	34,47	18,44	10,40	40,77	16,24
	Č	25,60	10,97	15,26	23,10	19,84	21,17	20,40	26,12	22,72
	K	33,00	11,54	16,98	29,30	20,36	23,80	25,67	26,75	**25,95**
W - 1	A	8,60	11,04	9,47	6,60	16,11	9,17	5,00	18,24	7,70
	Č	11,20	4,85	6,72	8,40	7,15	7,66	7,87	10,02	8,74
	K	16,20	5,76	8,44	12,70	8,91	10,39	11,20	11,83	11,42
W - 3	A	9,80	12,82	10,88	7,70	19,79	10,83	6,07	23,37	9,45
	Č	14,80	6,34	8,82	12,70	10,79	11,58	11,27	14,36	12,53
	K	19,60	6,94	10,19	16,90	11,88	13,84	14,80	15,56	15,05

Tabulka 8.1: Výsledky metody TF-IDF [%]

#	D	Top 5 kandidátů			Top 10 kandidátů			Top 15 kandidátů		
		P	R	F	P	R	F	P	R	F
S - 3	A	9,80	11,78	10,41	7,20	16,76	9,82	5,20	18,43	7,92
	Č	11,60	4,97	6,92	9,40	8,00	8,58	7,40	9,37	8,21
	K	17,60	6,12	9,04	14,00	9,85	11,48	10,87	11,40	11,04
W - 3	A	8,60	11,15	9,51	7,50	19,22	10,54	5,80	22,52	9,04
	Č	11,40	4,94	6,84	9,20	7,87	8,41	8,67	11,12	9,66
	K	6,00	5,73	8,38	13,50	9,46	11,03	12,00	12,80	**12,28**

Tabulka 8.2: Výsledky metody TF-IDF-avg [%]

Z tabulek 8.1 a 8.2 je vidět, že nejlepších výsledků je dosahováno při použití klasické metody TF-IDF, která je natrénována na korpusu SemEvalu a extrahují se n-gramy až do délky tří slov, které se porovnávají s kombinovanými klíčovými slovy (tedy pro kombinaci S-3-K). Za zmínku stojí také vysoká úspěšnost pro nejlepších pět kandidátů a kombinaci S-3-A. Jedná se totiž o největší dosaženou úspěšnost při srovnávání s klíčovými slovy, která vybral sám autor článku, a tudíž se jedná o ta „nejcennější“.

Zajímavým úkazem jsou pak rozdíly v úspěšnosti metody TF-IDF v její upravené verzi TF-IDF-avg. V případě extrahování n-gramů do délky 3, dosahuje metoda při natrénování na korpusu SemEvalu pouze poloviční úspěšnosti, zatímco při natrénování na korpusu Wikipedie jen lehce zaostává za standardní metodou TF-IDF (přibližně o 2 %). To je nejspíše způsobeno tím, že korpus Wikipedie obsahuje daleko více dokumentů, a proto jsou hodnoty určené pro unigramy přesnější než na malých datech SemEvalu.

8.1.2 Výsledky metody LDA

Metoda LDA byla testována pro 50, 100 a 200 témat viz tabulky 8.3, 8.4 a 8.5. Pro testování byly použity verze metod natrénovaných na desetiprocentním korpusu Wikipedie, protože bylo experimentálně zjištěno, že dosahují největší úspěšnosti. Korpus SemEvalu byl pro trénování použit v původním stavu, jaký je popsán v kapitole 5.1.

Jak je vidět z tabulek 8.3 až 8.5, metoda LDA není asi nejlepší volbou pro extrahování klíčových slov z dokumentů. Důvod, jak bylo již vysvětleno v kapitole 7.1, je ten, že metoda vyzdvihuje sice nejvíce pravděpodobná slova v dokumentech, nicméně tato slova jsou spíše obecná a nejsou ve většině případů označena za klíčová ani autorem ani čtenářem článku. Z tabulek mů-

#	D	Top 5 kandidátů			Top 10 kandidátů			Top 15 kandidátů		
		P	R	F	P	R	F	P	R	F
S - 1	A	2,60	3,90	3,09	1,70	4,98	2,52	1,53	6,28	2,44
	Č	4,40	2,04	2,77	3,10	2,83	2,95	3,00	3,96	3,39
	K	6,20	2,39	3,42	4,20	3,18	3,59	3,93	4,31	4,08
S - 3	A	5,60	7,73	6,40	3,70	10,24	5,35	3,60	15,21	5,74
	Č	6,60	3,05	4,15	5,60	5,01	5,24	5,73	7,42	6,40
	K	9,80	3,63	5,26	7,50	5,41	6,21	7,27	7,70	**7,39**
W - 1	A	0,20	0,33	0,25	0,10	0,33	0,15	0,07	0,33	0,11
	Č	0,80	0,34	0,48	0,80	0,70	0,74	1,07	1,34	1,18
	K	0,80	0,29	0,43	0,80	0,57	0,66	1,07	1,11	1,08
W - 3	A	0,20	0,33	0,25	0,10	0,33	0,15	0,07	0,33	0,11
	Č	0,80	0,34	0,48	0,80	0,70	0,74	1,07	1,35	1,18
	K	0,80	0,29	0,43	0,80	0,57	0,66	1,07	1,12	1,09

Tabulka 8.3: Výsledky metody LDA pro 50 témat [%]

#	D	Top 5 kandidátů			Top 10 kandidátů			Top 15 kandidátů		
		P	R	F	P	R	F	P	R	F
S - 1	A	2,20	3,23	2,59	1,80	5,11	2,63	1,47	6,06	2,34
	Č	5,00	2,28	3,11	3,60	3,22	3,38	2,87	3,73	3,22
	K	6,40	2,45	3,52	4,70	3,49	3,98	3,80	4,14	3,93
S - 3	A	4,80	6,48	5,44	4,60	12,23	6,58	3,47	14,23	5,50
	Č	7,40	3,42	4,65	7,00	6,05	6,44	6,07	7,83	6,77
	K	9,80	3,65	5,28	9,00	6,35	7,37	7,40	7,87	**7,54**
W - 1	A	0,20	0,33	0,25	0,10	0,33	0,15	0,20	0,81	0,31
	Č	0,60	0,27	0,37	1,20	1,06	1,11	1,27	1,62	1,41
	K	0,60	0,20	0,30	1,20	0,88	1,01	1,40	1,50	1,44
W - 3	A	0,20	0,33	0,25	0,10	0,33	0,15	0,20	0,81	0,31
	Č	0,60	0,27	0,37	1,30	1,13	1,20	1,33	1,69	1,48
	K	0,60	0,20	0,30	1,30	0,93	1,08	1,47	1,55	1,50

Tabulka 8.4: Výsledky metody LDA pro 100 témat [%]

žeme ovšem vysledovat rostoucí tendenci úspěšnosti se zvyšujícím se počtem témat. Maximální rozdíl je však pouze 1,5 %. Použití metody LDA tedy není vhodné pro extrahování klíčových slov. Tato metoda nicméně může být velmi dobře uplatnitelná v jiných úlohách zpracování přirozeného jazyka například pro kategorizaci dokumentů či podobné případy.

#	D	Top 5 kandidátů			Top 10 kandidátů			Top 15 kandidátů		
		P	R	F	P	R	F	P	R	F
S - 1	A	2,80	3,92	3,20	2,10	5,81	3,03	1,60	6,38	2,52
	Č	5,00	2,26	3,09	3,80	3,35	3,53	3,20	4,14	3,58
	K	7,00	2,61	3,77	5,20	3,78	4,34	4,27	4,57	4,37
S - 3	A	6,20	8,63	7,09	4,90	13,24	7,01	4,07	16,65	6,43
	Č	9,00	4,07	5,57	8,10	6,95	7,42	6,73	8,58	7,47
	K	12,20	4,46	6,47	10,20	7,19	8,34	8,53	8,96	**8,64**
W - 1	A	0,20	0,33	0,25	0,10	0,33	0,15	0,20	0,78	0,32
	Č	1,20	0,50	0,70	1,30	1,11	1,19	1,33	1,70	1,48
	K	1,20	0,43	0,63	1,30	0,95	1,09	1,47	1,54	1,49
W - 3	A	0,20	0,33	0,25	0,10	0,33	0,15	0,20	0,78	0,32
	Č	1,20	0,50	0,70	1,60	1,40	1,48	1,47	1,91	1,65
	K	1,20	0,43	0,63	1,60	1,16	1,34	1,60	1,68	1,63

Tabulka 8.5: Výsledky metody LDA pro 200 témat [%]

8.1.3 Výsledky metody GloVe

Metoda GloVe byla testována pro všechny dostupné sady vektorů poskytnutých na stránkách projektu (viz kapitola 6.2). V této kapitole jsou uvedeny pro přehlednost pouze nejlepší výsledky, kterých bylo dosaženo použitím sady vektorů 840B-300d. Výsledky pro ostatní použité sady vektorů naleznete v příloze B.

#	D	Top 5 kandidátů			Top 10 kandidátů			Top 15 kandidátů		
		P	R	F	P	R	F	P	R	F
S - 1	A	5,20	6,96	5,85	3,70	9,59	5,24	3,07	11,62	4,78
	Č	6,40	2,76	3,83	5,00	4,35	4,62	3,93	5,10	4,41
	K	9,00	3,28	4,78	7,00	5,06	5,83	5,73	6,18	5,90
S - 3	A	10,20	14,04	11,59	7,00	18,93	10,02	6,13	24,70	9,66
	Č	11,80	5,12	7,08	10,00	8,52	9,12	9,40	11,88	10,41
	K	16,00	5,85	8,50	12,70	9,08	10,49	12,00	12,66	**12,20**
W - 1	A	6,00	7,49	6,53	4,30	10,68	6,01	3,40	12,75	5,28
	Č	8,00	3,52	4,85	6,70	5,82	6,18	5,07	6,60	5,69
	K	11,00	4,01	5,83	9,20	6,58	7,61	7,00	7,51	7,19
W - 3	A	6,00	7,75	6,76	4,30	11,11	6,20	3,40	13,18	5,41
	Č	8,00	3,32	4,69	6,70	5,56	6,08	5,07	6,31	5,62
	K	11,00	3,75	5,59	9,20	6,28	7,46	7,00	7,16	7,08

Tabulka 8.6: Výsledky metody GloVe pro sadu vektorů 840B-300d [%]

Tato metoda má, ve srovnání s metodou LDA, o něco větší úspěšnost (pro korpus SemEvalu přibližně o 4 % a pro korpus Wikipedie o 8 %). Stále se však nemůže vyrovnat metodě TF-IDF. Z tabulky 8.6 a tabulek B.1 až B.9 můžeme vypozorovat, že čím vyšší je dimenze vektoru a čím více slov bylo k sestavení vektoru použito, tím lépe metoda funguje. Proto je nejlepšího výsledku dosaženo použitím sady vektorů s dimenzí 300, které byly vytvářeny z korpusu obsahujícího 840B slov.

8.1.4 Výsledky metody ZKEM

Provedeným experimentem (viz kapitola 7.3) bylo zjištěno, že nastavením optimálních koeficientů jednotlivých příznaků metody, je metoda ZKEM degradována na metodu TF-IDF. Nejvyšší možná úspěšnost, které může metoda dosáhnout, je tedy rovna úspěšnosti metody TF-IDF, která je zanesena v tabulce 8.1.

8.2 Srovnání úspěšnosti metod

Porovnání všech implementovaných metod bylo provedeno ze dvou hledisek 1) jaké n-gramy jsou metodou upřednostňovány a 2) celková dosažená úspěšnost pro nejlepších 5, 10 a 15 kandidátů.Všechny metody byly také porovnávány ve všech verzích, aby bylo zjištěno, která z nich je „ta nejlepší“.

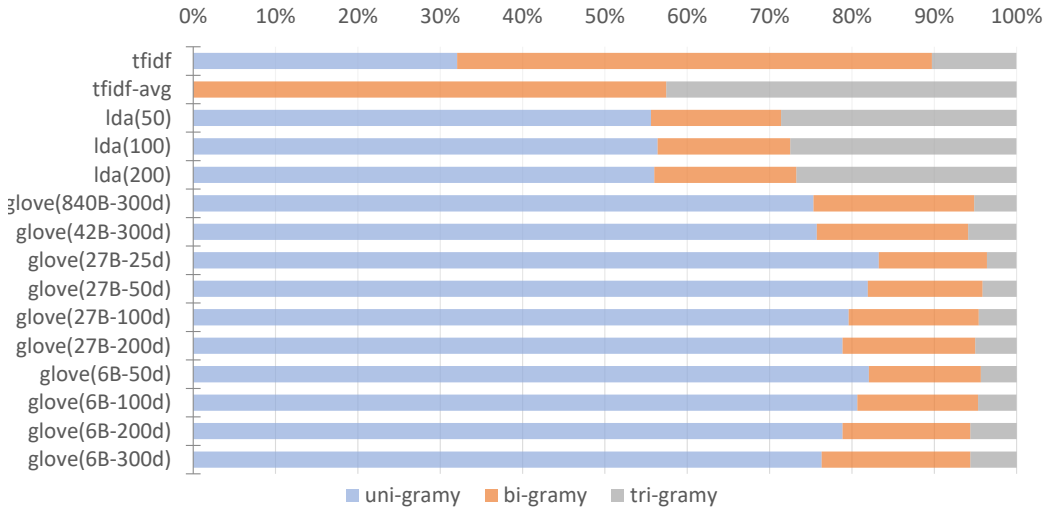
8.2.1 Zastoupení n-gramů ve výsledcích

Na obrázcích 8.1 a 8.2 můžeme vidět procentuální zastoupení unigramů, bigramů a trigramů v kandidátech vybraných každou metodou. Pro každou metodu bylo tedy určeno zastoupení v extrahovaných kandidátech pro všechny dokumenty z testovací množiny – celkem tedy v 1 500 klíčových slovech (15 pro každý ze 100 dokumentů). Důvodem tohoto porovnání bylo zjistit, jestli nějaká metoda nedává přednost například trigramům, jejichž zastoupení v gold datech je (viz tabulka 6.1) menší než u bigramů a unigramů.

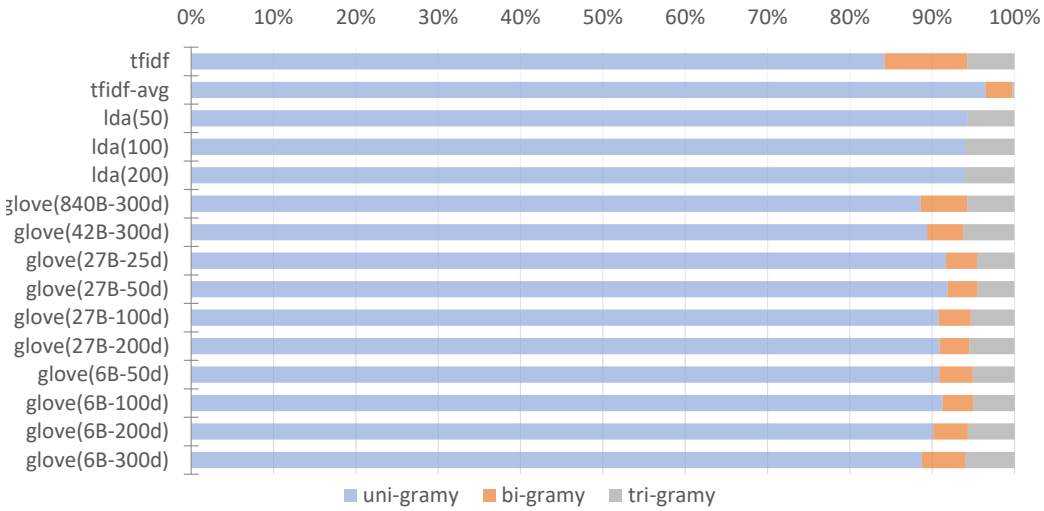
V případě metod trénovaných na korpusu SemEvalu (obrázek 8.1) je vidět, že nejlepší poměr víceslovných a jednoslovných kandidátů má metoda TF-IDF, kde je ve výsledcích obsaženo nejvíce bigramů a poté unigramů

a trigramů. To je právě optimální poměr. Přibližně stejné množství bigramů extrahuje i její upravená verze – metoda TF-IDF-avg. Nicméně tím, že jsou zde průměrovány hodnoty unigramů ve víceslovných termech dochází k tomu, že výsledné skóre víceslovných termů převáží skóre jednoslovných a metoda tedy prakticky zanedbává extrakci unigramů. Tím nejspíše dochází k tomu, že jsou výsledné hodnoty zkresleny a tím pádem dosahuje tato metoda menší úspěšnosti.

Všechny verze metody LDA a GloVe pak extrahují vždy přibližně stejné množství unigramů i bigramů, Přičemž metody LDA při natrénování na korpusu Wikipedie bigramy zanedbávají. Zajímavým úkazem je také to, že pokud jsou metody natrénovány na korpusu Wikipedie, upřednostňují jednoslovné termy nad těmi víceslovnými. To má patrně také vliv na celkově menší úspěšnost metod, protože je menší šance, že se vybraný unigram skutečně nachází v gold datech. Na obrázku 8.2 si můžeme všimnout, že ze všech verzí metody GloVe je nejvíce bigramů obsaženo ve výsledcích verze `840B-300d`, která pak dosahuje nejlepších výsledků.



Obrázek 8.1: Zastoupení n-gramů pro kombinaci S-3



Obrázek 8.2: Zastoupení n-gramů pro kombinaci W-3

8.2.2 Dosažená úspěšnost metod

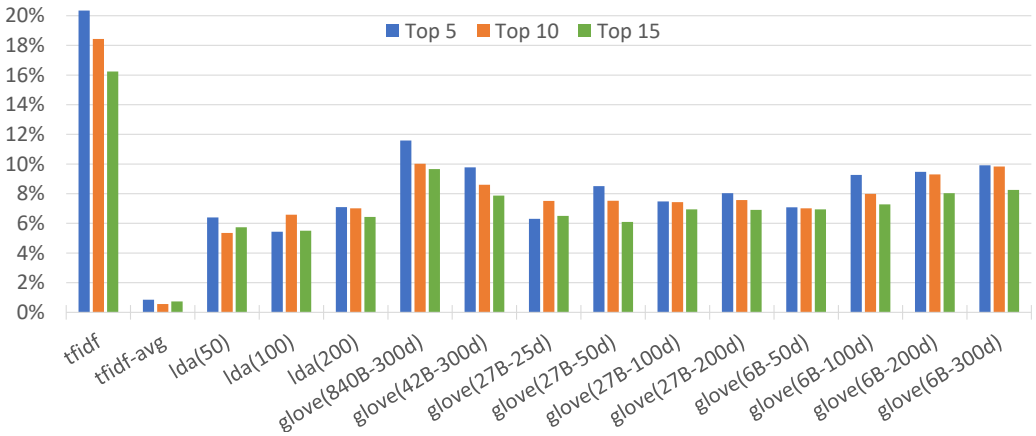
Na obrázcích 8.3 až 8.8 můžete vidět grafy, na kterých jsou zaneseny úspěšnosti pro všechny metody a jejich verze při extrakci n-gramů do délky 3. Úspěšnost metod při extrahování pouze jednoslovných klíčových frází můžete vidět v příloze C na obrázcích C.1 až C.6. Pro každou metodu je vyznačena úspěšnost pro top 5, 10 a 15 kandidátů na klíčová slova vybraných metodou.

Z grafů si můžeme povšimnout, že nejhorší průměrné výsledky mají jednoznačně metody LDA, které dosahují maximálně lehce přes 8 %, a to pro kombinaci S-3-K. Při natrénování metody na korpusu Wikipedie se dá říct, že metody vůbec nefungují. Jejich výsledky jsou v tomto případě pod 2 % úspěšnosti. Pro oba korpusy si ale můžeme všimnout, že čím více témat metoda rozpoznává, tím lépe funguje. Tedy nejlepších výsledků dosahuje verze metody lda(200). To, proč dosahují metody LDA tak malé úspěšnosti, je zapříčiněno tím, že vyzdvihují hodně obecná slova. Ta však nejsou označována za klíčová ani autorem ani čtenářem (více viz 7.1).

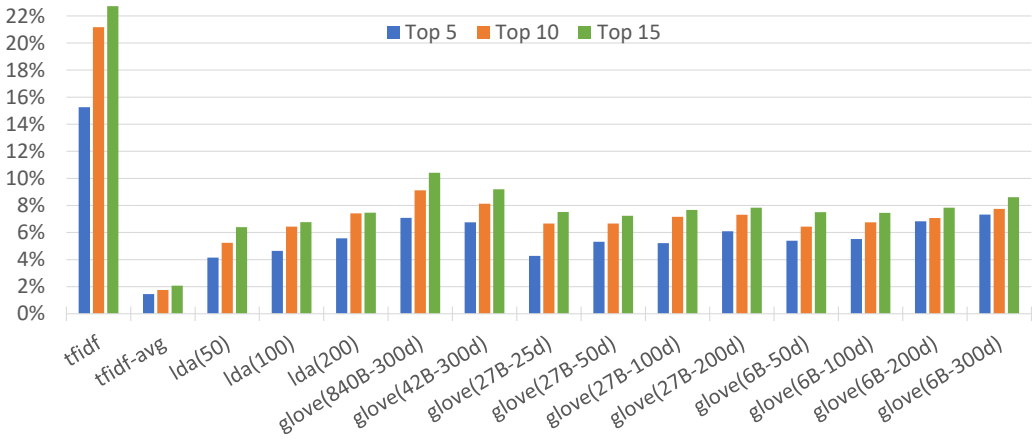
Vektory používané metodou GloVe k vlastní extrakci se nijak nevytvářejí a proto by všechny verze metody měly mít stejnou úspěšnost jak pro korpus SemEvalu tak Wikipedie. V algoritmu metody je však začleněno používání hodnot *tfidf* a *pmi*, které už podléhají natrénování na daných korpusech a ovlivňují tak celkovou úspěšnost. Metody tak dosahují pouze průměrné,

někdy mírně nadprůměrné úspěšnosti (4-12 %). Dá se také říci, že fungují nejlépe pro případ, kdy jsou kandidáti metody testováni oproti klíčovým slovům přiřazeným autorem. Nejlepších výsledků pak dosahuje metoda glove(840B-300d), která používá vektory s největší dimenzí a vytvořené z největšího korpusu.

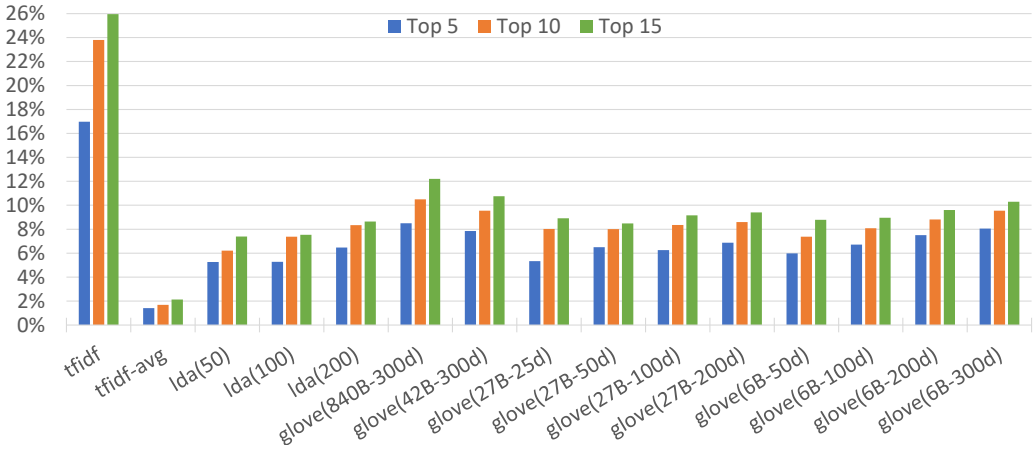
Největší úspěšnosti dosáhla metoda TF-IDF. Ta drží prvenství při natrénování na korpusu SemEvalu i Wikipedie i co se týká extrahování jednoslovných i víceslovných termů. Nejvýše se úspěšnost vyšplhala k 25,95 %. Jedná se sice o nejjednodušší přístup ale ukázalo se, že zatím i o nejlepší. Verze TF-IDF-avg této metody při extrakci trigramů funguje lépe, pokud je natrénována na korpusu Wikipedie. Při natrénování na korpusu SemEvalu dojde k tomu, že jsou vlivem průměrování dílčích hodnot *tfidf* prakticky zanedbány unigramy a metoda tak poskytuje pouze bi- a trigramy. To má za následek velmi malou úspěšnost metody (dokonce menší než u metod LDA). Při natrénování na korpusu Wikipedie však tato metoda jen o něco zaostává za klasickou metodou TF-IDF (přibližně o 4 %).



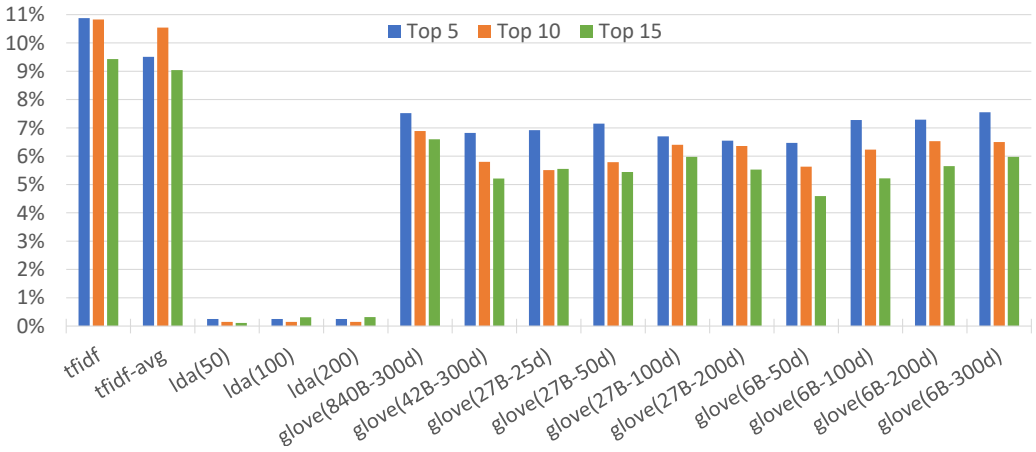
Obrázek 8.3: Úspěšnost metod pro kombinaci S-3-A



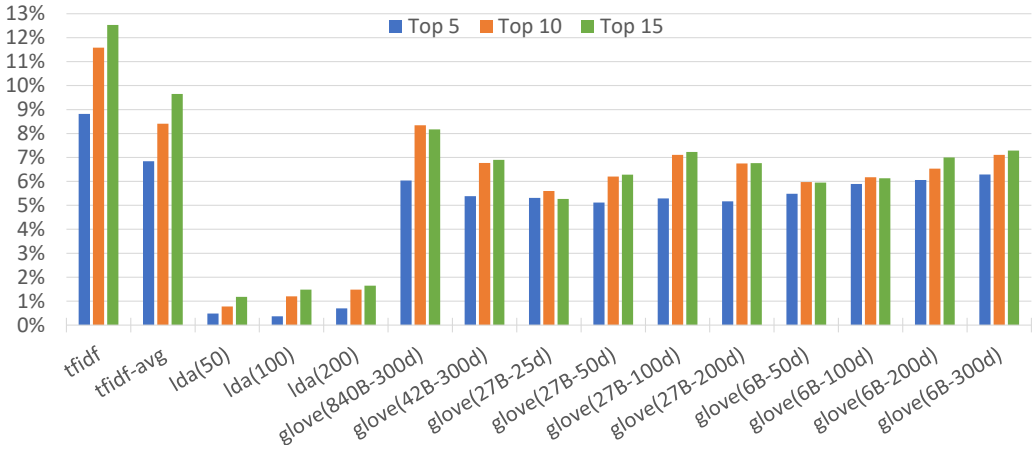
Obrázek 8.4: Úspěšnost metod pro kombinaci S-3-Č



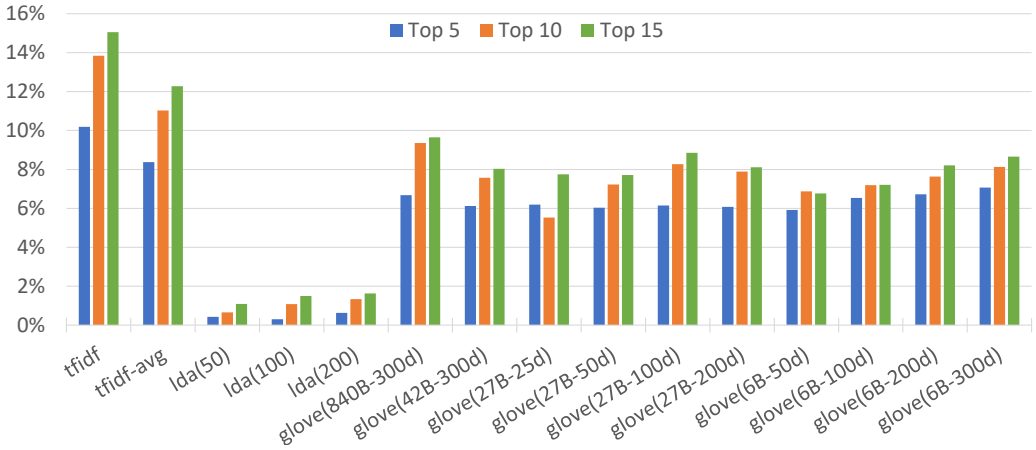
Obrázek 8.5: Úspěšnost metod pro kombinaci S-3-K



Obrázek 8.6: Úspěšnost metod pro kombinaci W-3-A



Obrázek 8.7: Úspěšnost metod pro kombinaci W-3-Č



Obrázek 8.8: Úspěšnost metod pro kombinaci W-3-K

9 Závěr

V této práci byly popsány a implementovány tři známé metody pro extrakci klíčových frází z textových dokumentů – TF-IDF, LDA a GloVe. Všechny tyto metody spadají do rodiny metod s trénováním bez učitele tzn., že nebylo potřeba vytvářet ručně anotovaná data pro jejich trénování. Navržena a implementována byla také vlastní metoda (metoda ZKEM), která kombinuje postupy všech výše zmíněných metod. Účelem bylo zjistit, zda je možné při zkombinování různých přístupů dosáhnout vyšší úspěšnosti než při použití pouze některého z nich.

V algoritmu každé metody bylo navíc začleněno použití metriky *pmi* (viz kapitola 4.1), díky čemuž se, do určité míry, podařilo zamezit extrakci nesmyslných víceslovných termů. Pro zvýšení základní úspěšnosti metody byly navrženy tři experimenty, z nichž nejlépe dopadl experiment, při kterém se hledalo optimální nastavení vah slov vyskytujících se v nadpisu, abstraktu a těle dokumentu. Díky správnému nastavení těchto vah byla úspěšnost zvýšena v nejlepším případě až o 4 %. Při experimentech bylo také zjištěno, že algoritmus metody ZKEM dosahuje nejlepších výsledků, pokud jsou ignorovány příznaky termu určené metodami LDA a GloVe. Z toho důvodu může metoda ZKEM dosáhnout maximálně stejné úspěšnosti jako metoda TF-IDF.

Jako trénovací data byly použity celkem dva korpusy – korpus anglické Wikipedie, který bylo potřeba nejprve předzpracovat, a trénovací kolekce dokumentů poskytnutých organizátory mezinárodní programátorské soutěže SemEval 2010.

Testování metod probíhalo stejným způsobem jako při soutěži SemEval 2010. Byl použit stejný validační script i stejná testovací data. Díky tomuto způsobu testování je možné porovnávat implementované metody v této práci s metodami vytvořenými při soutěži. Každá z metod byla testována pro případ extrahování jednoslovných i víceslovných klíčových frází, a to jak při natrénování na korpusu SemEvalu, tak Wikipedie. V případě extrahování víceslovných klíčových frází byly uvažovány fráze až do délky tří slov.

Metody dosahovaly při natrénování na korpusu Wikipedie celkově horší úspěšnosti než při natrénování na korpusu SemEvalu. To je způsobeno nejspíše tím, že Wikipedie obsahuje články z velkého množství témat, zatímco testovací množina dokumentů je tvořena články pouze ze čtyř ACM kategorií a to navíc technického rázu. Trénovací korpus SemEvalu je tvořen články

ze stejných kategorií a proto při trénování metod na tomto korpusu bylo dosahováno až dvojnásobné úspěšnosti. Nutno však říci, že ačkoliv metody při natrénování na korpusu Wikipedie fungují hůře, jejich pole působnosti je větší a úspěšnost by měla být přibližně stejná při extrakci klíčových slov z jakéhokoliv obecného anglického textu.

Při testování se jako nejlepší metoda ukázala metoda TF-IDF, která s navrženými úpravami dokázala ze všech testovacích článků správně extrahovat 25,95 % klíčových slov (všechny výsledky jsou zaneseny v tabulce 8.1). Při porovnání s výsledky ostatních metod, které byly implementovány při soutěži SemEval 2010 (viz tabulka A.1) je vidět, že vyšší úspěšnosti dosáhla pouze metoda HUMB s 27,5 %. Při soutěži by se tedy upravená metoda TF-IDF umístila na druhém místě. Umístění na druhém místě považuji za velmi dobrý výsledek této práce.

Literatura

[1] Slobodan Beliga, Ana Meštrović, and Sanda Martinčić-Ipšić. An overview of graph-based keyword extraction methods and approaches. *Journal of information and organizational sciences*, 39(1):1–20, 2015.

[2] Gábor Berend and Richárd Farkas. Sztergak: Feature engineering for keyphrase extraction. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 186–189. Association for Computational Linguistics, 2010.

[3] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[4] Gerlof Bouma. Normalized (pointwise) mutual information in collocation extraction. In *Proceedings of the Biennial GSCL Conference*, volume 156, 2009.

[5] Tomá Brychcín. *Distributional Semantics in Language Modeling*. Disertační práce, Západočeská univerzita v Plzni, 2015.

[6] Tomáš Brychcín and Miloslav Konopík. Hps: High precision stemmer. *Information Processing & Management*, 51(1):68–91, 2015.

[7] William M Darling. A theoretical and practical implementation tutorial on topic modeling and gibbs sampling. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 642–647, 2011.

[8] Ofer Egozi, Shaul Markovitch, and Evgeniy Gabrilovich. Concept-based information retrieval using explicit semantic analysis. *ACM Transactions on Information Systems (TOIS)*, 29(2):8, 2011.

[9] Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences*, 101(suppl 1):5228–5235, 2004.

[10] Jakub Kanis and Lucie Skorkovská. Comparison of different lemmatization approaches through the means of information retrieval performance. In *International Conference on Text, Speech and Dialogue*, pages 93–100. Springer, 2010.

[11] Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 21–26. Association for Computational Linguistics, 2010.

[12] Feifan Liu, Deana Pennell, Fei Liu, and Yang Liu. Unsupervised approaches for automatic keyword extraction using meeting transcripts. In *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics*, pages 620–628. Association for Computational Linguistics, 2009.

[13] Patrice Lopez and Laurent Romary. Humb: Automatic key term extraction from scientific articles in grobid. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, pages 248–251, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[14] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60, 2014.

[15] Thuy Dung Nguyen and Minh-Thang Luong. Wingnus: Keyphrase extraction utilizing document logical structure. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 166–169. Association for Computational Linguistics, 2010.

[16] Jiří Pavlík. Částečně řízené učení algoritmů strojového učení (semi-supervised learning). Diplomová práce, Mendelova univerzita v Brně, Provozně ekonomická fakulta, Brno, 2016.

[17] Francis Jeffry Pelletier. The principle of semantic compositionality. *Topoi*, 13(1):11–24, 1994.

[18] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

[19] Barry Smith. Objects and their environments: from aristotle to ecological ontology. *The life and motion of socio-economic units: GISDATA*, 8, 2001.

[20] Pucktada Treeratpituk, Pradeep Teregowda, Jian Huang, and C Lee Giles. Seerlab: A system for extracting key phrases from scholarly documents. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 182–185. Association for Computational Linguistics, 2010.

[21] Peter D Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424. Association for Computational Linguistics, 2002.

[22] Vincent Van Asch. Macro-and micro-averaged evaluation measures. *Tech. Rep.*, 2013.

[23] Cornelis Joost Van Rijsbergen. Information retrieval. 1979.

[24] Lukáš Witz. Extrakce sociálních sítí ze zpravodajských textů. Bakalářská práce, Západočeská univerzita v Plzni, 2014.

[25] Chengzhi Zhang. Automatic keyword extraction from documents using conditional random fields. *Journal of Computational Information Systems*, 4(3):1169–1180, 2008.

[26] About text analysis and semantics. [online]. [cit. 2017-04-22]. Dostupné z: `https://www.semantic-knowledge.com/text-analysis.htm`, 2010.

[27] Stanford. Software. the stanford natural language processing group. [online]. [cit. 2017-01-30]. Dostupné z: `https://nlp.stanford.edu/software`, 2013.

[28] Getting started with keyword extraction. text mining online. [online]. [cit. 2016-11-06]. Dostupné z: `http://textminingonline.com/getting-started-with-keyword-extraction`, 2014.

[29] Tf/idf with google n-grams and pos tags. language processing. [online]. [cit. 2016-11-05]. Dostupné z: `http://trimc-nlp.blogspot.cz/2013/04/tfidf-with-google-n-grams-and-pos-tags.html`, 2013.

A Výsledky soutěže SemEval 2010

V následujících třech tabulkách jsou zaznamenány výsledky soutěže SemEval 2010 z úlohy číslo 5 Automatic Keyphrase Extraction from Scientific Articles. Pro každý systém byla určena hodnota *micro-average precision* (P), *recall* (R) a *F-Skóre* (F).

Systém	Top 5 kandidátů			Top 10 kandidátů			Top 15 kandidátů		
	P	R	F	P	R	F	P	R	F
HUMB	39,0	13,3	19,8	32,0	21,8	26,0	27,2	27,8	27,5
WINGNUS	40,2	13,7	20,5	30,5	20,8	24,7	24,9	25,5	25,2
KP-Miner	36,0	12,3	18,3	28,6	19,5	23,2	24,9	25,5	25,2
SZTERGAK	34,2	11,7	17,4	28,5	19,4	23,1	24,8	25,4	25,1
ICL	34,4	11,7	17,5	29,2	19,9	23,7	24,6	25,2	24,9
SEERLAB	39,0	13,3	19,8	29,7	20,3	24,1	24,1	24,6	24,3
KX FBK	34,2	11,7	17,4	27,0	18,4	21,9	23,6	24,2	23,9
DERIUNLP	27,4	9,4	13,9	23,0	15,7	18,7	22,0	22,5	22,3
Maui	35,0	11,9	17,8	25,2	17,2	20,4	20,3	20,8	20,6
DFKI	29,2	10,0	14,9	23,3	15,9	18,9	20,3	20,7	20,5
BUAP	13,6	4,6	6,9	17,6	12,0	14,3	19,0	19,4	19,2
SJTULTLAB	30,2	10,3	15,4	22,7	15,5	18,4	18,4	18,8	18,6
UNICE	27,4	9,4	13,9	22,4	15,3	18,2	18,3	18,8	18,5
UNPMC	18,0	6,1	9,2	19,0	13,0	15,4	18,1	18,6	18,3
JU CSE	28,4	9,7	14,5	21,5	14,7	17,4	17,8	18,2	18,0
LIKEY	29,2	10,0	14,9	21,1	14,4	17,1	16,3	16,7	16,5
UvT	24,8	8,5	12,6	18,6	12,7	15,1	14,6	14,9	14,8
POLYU	5,6	5,3	7,9	14,6	10,0	11,8	13,9	14,2	14,0
UKP	9,4	3,2	4,8	5,9	4,0	4,8	5,3	5,4	5,3

Tabulka A.1: Úspěšnost systémů testovaných na kombinovaných seznamech klíčových slov [%]

Systém	Top 5 kandidátů			Top 10 kandidátů			Top 15 kandidátů		
	P	R	F	P	R	F	P	R	F
HUMB	30,4	12,6	17,8	24,8	20,6	22,5	21,2	26,4	23,5
KX FBK	29,2	12,1	17,1	23,2	19,3	21,1	20,3	25,3	22,6
SZTERGAK	28,2	11,7	16,6	23,2	19,3	21,1	19,9	24,8	22,1
WINGNUS	30,6	12,7	18,0	23,6	19,6	21,4	19,8	24,7	22,0
ICL	27,2	11,3	16,0	22,4	18,6	20,3	19,5	24,3	21,6
SEERLAB	31,0	12,9	18,2	24,1	20,0	21,9	19,3	24,1	21,5
KP-Miner	28,2	11,7	16,5	22,0	18,3	20,0	19,3	24,1	21,5
DERIUNLP	22,2	9,2	13,0	18,9	15,7	17,2	17,5	21,8	19,5
DFKI	24,4	10,1	14,3	19,8	16,5	18,0	17,4	21,7	19,3
UNICE	25,0	10,4	14,7	20,1	16,7	18,2	16,0	19,9	17,8
SJTULTLAB	26,6	11,1	15,6	19,4	16,1	17,6	15,6	19,4	17,3
BUAP	10,4	4,3	6,1	13,9	11,5	12,6	14,9	18,6	16,6
Maui	25,0	10,4	14,7	18,1	15,0	16,4	14,9	18,5	16,1
UNPMC	13,8	5,7	8,1	15,1	12,5	13,7	14,5	18,0	16,1
JU CSE	23,4	9,7	13,7	18,1	15,0	16,4	14,4	17,9	16,0
LIKEY	24,6	10,2	14,4	17,9	14,9	16,2	13,8	17,2	15,3
POLYU	13,6	5,7	8,0	12,6	10,5	11,4	12,0	14,9	13,3
UvT	20,4	8,5	12,0	15,6	13,0	14,2	11,9	14,9	13,2

Tabulka A.2: Úspěšnost systémů testovaných na seznamech klíčových slov přiřazených čtenářem [%]

Systém	Top 5 kandidátů			Top 10 kandidátů			Top 15 kandidátů		
	P	R	F	P	R	F	P	R	F
HUMB	21,2	27,4	23,9	15,4	39,8	22,2	12,1	47,0	19,3
KP-Miner	19,0	24,6	21,4	13,4	34,6	19,3	10,7	41,6	17,1
ICL	17,0	22,0	19,2	13,5	34,9	19,5	10,5	40,6	16,6
Maui	20,4	26,4	23,0	13,7	35,4	19,8	10,2	39,5	16,2
SEERLAB	18,8	24,3	21,2	13,1	33,9	18,9	10,1	39,0	16,0
SZTERGAK	14,6	18,9	16,5	12,2	31,5	17,6	9,9	38,5	15,8
WINGNUS	18,6	24,0	21,0	12,6	32,6	18,2	9,3	36,2	14,8
DERIUNLP	12,6	16,3	14,2	9,7	25,1	14,0	9,3	35,9	14,7
KX FBK	13,6	17,6	15,3	10,0	25,8	14,4	8,5	32,8	13,5
BUAP	5,6	7,2	6,3	8,1	20,9	11,7	8,3	32,0	13,2
JU CSE	12,0	15,5	13,5	8,5	22,0	12,3	7,5	29,0	11,9
UNPMC	7,0	9,0	7,9	7,7	19,9	11,1	7,1	27,4	11,2
DFKI	12,8	16,5	14,4	8,5	22,0	12,3	6,6	25,6	10,5
SJTULTLAB	9,6	12,4	10,8	7,8	20,2	11,3	6,2	24,0	9,9
Likey	11,6	15,0	13,1	7,9	20,4	11,4	5,9	22,7	9,3
UvT	11,4	14,7	12,9	7,6	19,6	11,0	5,8	22,5	9,2
UNICE	8,8	11,4	9,9	6,4	16,5	9,2	5,5	21,5	8,8

Tabulka A.3: Úspěšnost systémů testovaných na seznamech klíčových slov přiřazených autorem [%]

B Další výsledky metody GloVe

#	D	Top 5 kandidátů			Top 10 kandidátů			Top 15 kandidátů		
		P	R	F	P	R	F	P	R	F
S - 1	A	3,40	4,38	3,74	2,40	6,33	3,43	2,00	7,59	3,12
	Č	4,60	1,94	2,71	3,50	2,97	3,19	2,80	3,56	3,11
	K	6,60	2,32	3,42	4,80	3,41	3,96	4,00	4,20	4,07
S - 3	A	6,40	8,37	7,08	4,90	13,09	7,01	4,40	17,79	6,95
	Č	9,00	3,90	5,40	7,00	6,04	6,43	6,73	8,60	7,50
	K	11,60	4,06	5,98	9,00	6,33	7,37	8,60	9,12	8,79
W - 1	A	4,80	6,07	5,25	2,90	7,08	4,04	2,33	8,64	3,62
	Č	6,80	2,95	4,09	4,60	4,02	4,26	3,67	4,84	4,14
	K	9,20	3,30	4,83	6,20	4,48	5,17	5,00	5,45	5,17
W - 3	A	5,80	7,58	6,47	4,00	10,11	5,63	2,93	11,31	4,59
	Č	9,20	3,94	5,48	6,50	5,60	5,97	5,33	6,86	5,95
	K	11,20	4,05	5,92	8,30	5,93	6,87	6,60	7,07	6,77

Tabulka B.1: Výsledky metody GloVe s použitím sady vektorů 6B-50d [%]

#	D	Top 5 kandidátů			Top 10 kandidátů			Top 15 kandidátů		
		P	R	F	P	R	F	P	R	F
S - 1	A	4,20	5,26	4,55	2,40	6,18	3,39	1,87	7,26	2,93
	Č	4,40	1,86	2,60	3,40	2,86	3,08	2,73	3,44	3,02
	K	7,00	2,42	3,58	4,90	3,38	3,98	3,87	4,02	3,91
S - 3	A	8,40	10,95	9,26	5,60	14,89	7,99	4,60	18,85	7,28
	Č	9,20	3,98	5,52	7,40	6,31	6,76	6,73	8,47	7,45
	K	13,00	4,56	6,71	9,90	6,90	8,08	8,87	9,18	8,96
W - 1	A	5,60	7,02	6,10	3,40	8,66	4,79	2,53	9,53	3,94
	Č	7,20	3,18	4,39	4,60	4,05	4,27	3,80	5,00	4,29
	K	10,20	3,69	5,38	6,50	4,69	5,40	5,20	5,66	5,38
W - 3	A	6,60	8,45	7,28	4,40	11,36	6,23	3,33	12,89	5,22
	Č	9,80	4,25	5,89	6,70	5,80	6,17	5,47	7,09	6,13
	K	12,40	4,48	6,54	8,70	6,21	7,19	7,00	7,55	7,21

Tabulka B.2: Výsledky metody GloVe s použitím sady vektorů 6B-100d [%]

#	D	Top 5 kandidátů			Top 10 kandidátů			Top 15 kandidátů		
		P	R	F	P	R	F	P	R	F
S - 1	A	4,40	5,71	4,88	3,00	7,68	4,24	2,20	8,38	3,43
	Č	5,20	2,17	3,04	3,90	3,25	3,51	3,13	4,01	3,49
	K	7,60	2,64	3,90	5,70	3,93	4,62	4,47	4,72	4,55
S - 3	A	8,40	11,38	9,48	6,50	17,53	9,30	5,07	21,15	8,04
	Č	11,40	4,92	6,83	7,80	6,57	7,07	7,07	8,95	7,83
	K	14,40	5,12	7,51	10,80	7,55	8,82	9,47	9,95	9,61
W - 1	A	5,60	7,13	6,14	3,60	9,11	5,07	2,87	11,09	4,48
	Č	7,60	3,39	4,65	5,20	4,55	4,81	4,40	5,78	4,96
	K	10,60	3,89	5,64	7,20	5,21	5,99	6,00	6,54	6,21
W - 3	A	6,60	8,48	7,29	4,60	11,94	6,53	3,60	14,13	5,65
	Č	10,00	4,39	6,06	7,10	6,13	6,53	6,27	8,06	7,00
	K	12,60	4,62	6,72	9,20	6,60	7,63	8,00	8,57	8,21

Tabulka B.3: Výsledky metody GloVe s použitím sady vektorů 6B-200d [%]

#	D	Top 5 kandidátů			Top 10 kandidátů			Top 15 kandidátů		
		P	R	F	P	R	F	P	R	F
S - 1	A	4,60	5,99	5,11	3,00	8,08	4,31	2,33	9,28	3,69
	Č	5,80	2,46	3,43	3,90	3,31	3,55	3,20	4,15	3,58
	K	8,20	2,92	4,27	5,70	4,05	4,70	4,53	4,90	4,67
S - 3	A	8,80	11,88	9,91	6,80	18,96	9,83	5,20	21,79	8,26
	Č	12,20	5,28	7,32	8,50	7,22	7,74	7,80	9,80	8,61
	K	15,40	5,50	8,05	11,60	8,22	9,54	10,13	10,64	10,29
W - 1	A	5,80	7,16	6,27	3,90	9,84	5,48	3,07	11,90	4,80
	Č	8,00	3,63	4,96	5,70	4,97	5,26	4,60	6,04	5,18
	K	11,20	4,13	5,99	7,90	5,67	6,55	6,47	6,97	6,66
W - 3	A	7,00	8,61	7,55	4,60	11,84	6,50	3,80	15,19	5,98
	Č	10,20	4,59	6,29	7,70	6,70	7,11	6,53	8,40	7,29
	K	13,20	4,87	7,07	9,80	7,04	8,13	8,47	9,00	8,66

Tabulka B.4: Výsledky metody GloVe s použitím sady vektorů 6B-300d [%]

#	D	Top 5 kandidátů			Top 10 kandidátů			Top 15 kandidátů		
		P	R	F	P	R	F	P	R	F
S - 1	A	4,80	6,01	5,20	3,10	7,68	4,32	2,47	9,29	3,83
	Č	5,20	2,23	3,10	3,80	3,24	3,47	3,40	4,37	3,80
	K	8,00	2,81	4,13	5,80	4,05	4,73	4,93	5,21	5,03
S - 3	A	8,80	11,52	9,76	6,10	15,66	8,60	5,00	20,23	7,87
	Č	11,20	4,90	6,76	8,90	7,61	8,13	8,33	10,48	9,20
	K	15,00	5,38	7,85	11,70	8,18	9,54	10,67	11,07	10,76
W - 1	A	5,20	6,56	5,70	3,40	8,36	4,75	2,53	9,39	3,93
	Č	7,60	3,33	4,60	5,30	4,63	4,90	4,27	5,60	4,81
	K	10,40	3,78	5,51	7,40	5,29	6,12	5,73	6,21	5,92
W - 3	A	6,20	7,89	6,82	4,10	10,53	5,80	3,33	12,76	5,21
	Č	9,00	3,88	5,38	7,40	6,33	6,77	6,20	7,93	6,90
	K	11,60	4,19	6,12	9,20	6,53	7,57	7,87	8,35	8,03

Tabulka B.5: Výsledky metody GloVe s použitím sady vektorů 42B-300d [%]

#	D	Top 5 kandidátů			Top 10 kandidátů			Top 15 kandidátů		
		P	R	F	P	R	F	P	R	F
S - 1	A	3,40	4,10	3,63	2,20	5,19	3,03	2,00	7,31	3,09
	Č	3,80	1,57	2,21	3,90	3,24	3,51	3,13	3,96	3,47
	K	6,40	2,19	3,24	5,60	3,82	4,51	4,67	4,86	4,72
S - 3	A	5,80	7,21	6,31	5,30	13,87	7,51	4,13	16,57	6,50
	Č	7,20	3,07	4,27	7,30	6,21	6,66	6,80	8,57	7,52
	K	10,40	3,61	5,33	9,90	6,82	8,02	8,87	9,12	8,92
W - 1	A	5,00	6,56	5,59	3,50	8,81	4,91	2,67	10,26	4,16
	Č	6,40	2,70	3,77	4,70	4,01	4,28	4,13	5,27	4,59
	K	9,20	3,29	4,82	6,90	4,90	5,68	5,87	6,25	6,00
W - 3	A	6,20	8,14	6,92	3,90	10,00	5,51	3,53	13,96	5,55
	Č	9,00	3,80	5,31	6,20	5,21	5,60	5,67	7,16	6,27
	K	11,80	4,22	6,19	8,10	5,69	6,63	7,60	8,04	7,75

Tabulka B.6: Výsledky metody GloVe s použitím sady vektorů 27B-25d [%]

#	D	Top 5 kandidátů			Top 10 kandidátů			Top 15 kandidátů		
		P	R	F	P	R	F	P	R	F
S - 1	A	3,40	4,02	3,60	2,20	5,28	3,04	1,93	7,23	3,00
	Č	4,20	1,73	2,43	3,30	2,72	2,96	3,07	3,93	3,41
	K	6,60	2,25	3,34	4,90	3,32	3,93	4,40	4,67	4,49
S - 3	A	7,60	10,13	8,51	5,30	14,00	7,53	3,87	15,70	6,10
	Č	9,00	3,81	5,32	7,40	6,16	6,67	6,53	8,28	7,24
	K	12,60	4,41	6,50	9,90	6,81	8,01	8,40	8,73	8,48
W - 1	A	5,40	6,83	5,90	3,00	7,50	4,20	2,73	10,30	4,25
	Č	6,20	2,65	3,68	4,90	4,21	4,48	4,07	5,22	4,53
	K	9,20	3,31	4,83	6,70	4,78	5,53	5,87	6,29	6,02
W - 3	A	6,40	8,39	7,15	4,10	10,54	5,79	3,47	13,52	5,44
	Č	8,60	3,69	5,12	6,80	5,79	6,20	5,67	7,20	6,28
	K	11,40	4,14	6,03	8,80	6,22	7,23	7,53	8,04	7,71

Tabulka B.7: Výsledky metody GloVe s použitím sady vektorů 27B-50d [%]

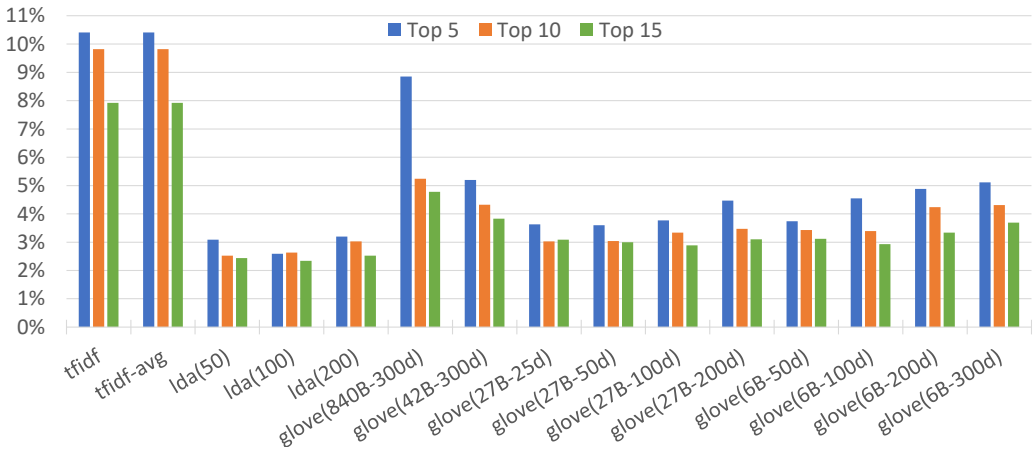
#	D	Top 5 kandidátů			Top 10 kandidátů			Top 15 kandidátů		
		P	R	F	P	R	F	P	R	F
S - 1	A	3,60	4,16	3,77	2,40	5,86	3,34	1,87	6,89	2,89
	Č	4,60	1,91	2,68	3,60	3,07	3,28	3,07	3,91	3,40
	K	7,20	2,44	3,63	5,20	3,64	4,24	4,33	4,57	4,41
S - 3	A	6,80	8,69	7,48	5,20	14,02	7,43	4,40	17,88	6,94
	Č	8,80	3,74	5,21	7,90	6,66	7,16	6,93	8,76	7,67
	K	12,20	4,24	6,26	10,30	7,14	8,36	9,07	9,44	9,16
W - 1	A	5,00	6,49	5,53	3,70	9,50	5,23	3,07	11,89	4,81
	Č	6,60	2,84	3,93	5,60	4,85	5,15	4,60	5,97	5,15
	K	9,60	3,47	5,06	8,00	5,76	6,64	6,67	7,21	6,87
W - 3	A	6,00	7,89	6,70	4,50	11,79	6,40	3,80	15,05	5,98
	Č	8,80	3,83	5,29	7,80	6,65	7,11	6,47	8,37	7,23
	K	11,60	4,22	6,15	10,00	7,16	8,27	8,60	9,26	8,85

Tabulka B.8: Výsledky metody GloVe s použitím sady vektorů 27B-100d [%]

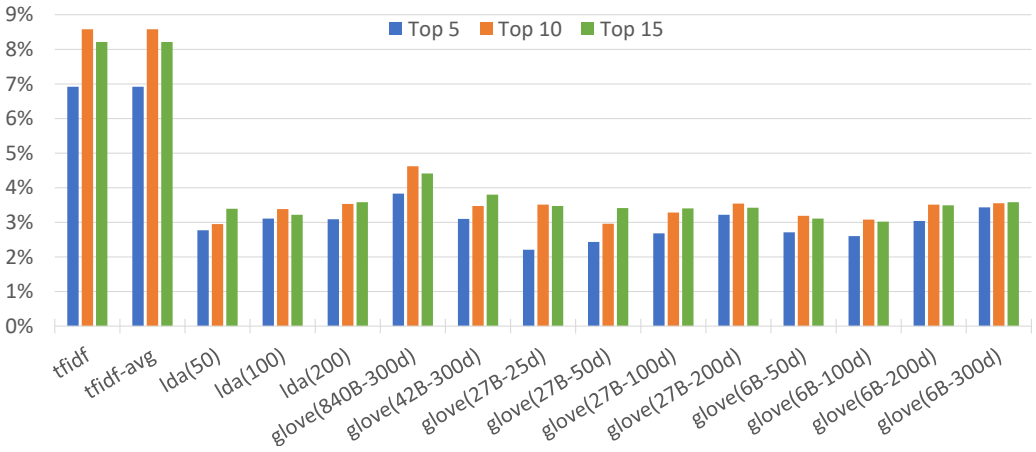
#	D	Top 5 kandidátů			Top 10 kandidátů			Top 15 kandidátů		
		P	R	F	P	R	F	P	R	F
S - 1	A	4,20	5,03	4,47	2,50	6,06	3,47	2,00	7,43	3,10
	Č	5,40	2,32	3,22	3,90	3,31	3,54	3,07	3,94	3,42
	K	8,20	2,87	4,23	5,60	3,92	4,57	4,47	4,72	4,55
S - 3	A	7,20	9,60	8,04	5,30	14,39	7,57	4,40	17,47	6,91
	Č	10,20	4,39	6,09	8,10	6,80	7,31	7,13	8,90	7,84
	K	13,20	4,68	6,87	10,60	7,36	8,60	9,33	9,65	9,40
W - 1	A	4,80	6,08	5,24	3,60	9,04	5,05	2,87	10,94	4,48
	Č	6,80	2,95	4,08	5,10	4,42	4,69	4,33	5,66	4,87
	K	9,80	3,52	5,14	7,40	5,27	6,10	6,20	6,69	6,38
W - 3	A	6,00	7,58	6,55	4,50	11,54	6,36	3,53	13,64	5,53
	Č	8,60	3,73	5,17	7,40	6,32	6,75	6,07	7,80	6,76
	K	11,60	4,16	6,08	9,60	6,80	7,89	7,93	8,45	8,11

Tabulka B.9: Výsledky metody GloVe s použitím sady vektorů 27B-200d [%]

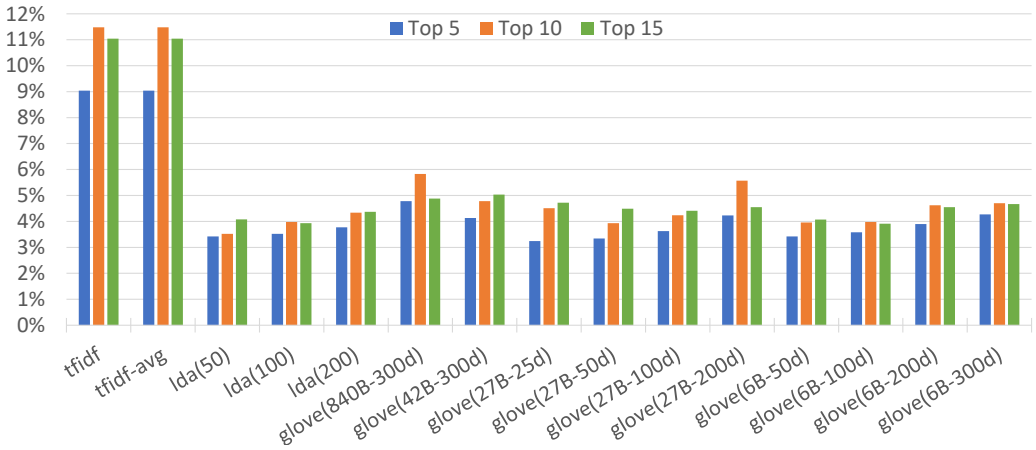
C Srovnání metod při extrahování unigramů



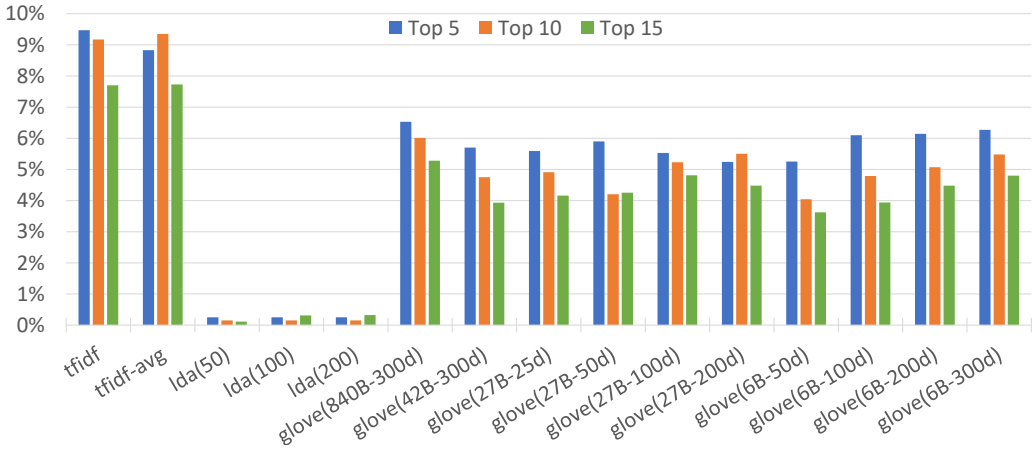
Obrázek C.1: Úspěšnost metod pro kombinaci S-1-A



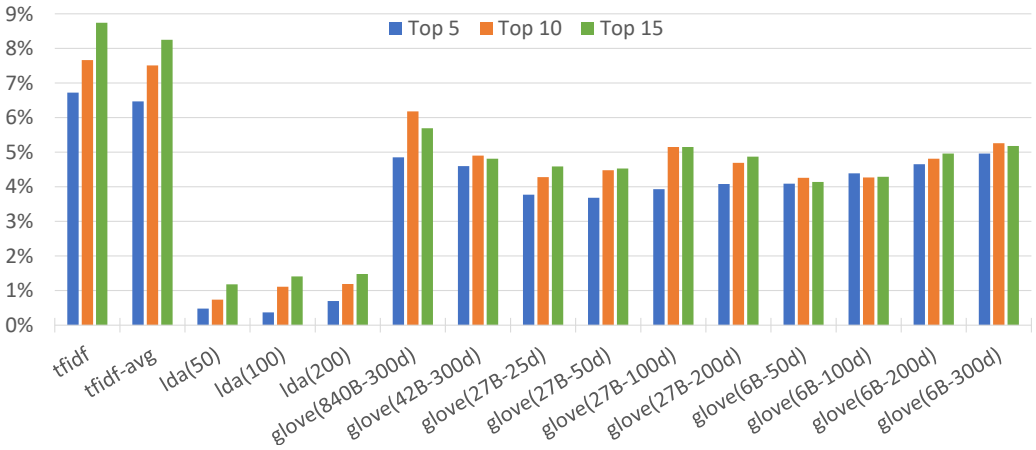
Obrázek C.2: Úspěšnost metod pro kombinaci S-1-Č



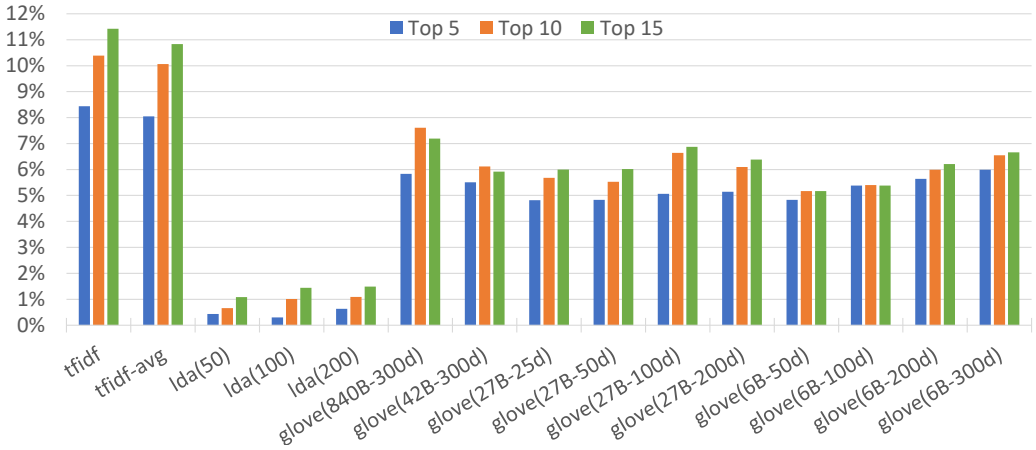
Obrázek C.3: Úspěšnost metod pro kombinaci S-1-K



Obrázek C.4: Úspěšnost metod pro kombinaci W-1-A



Obrázek C.5: Úspěšnost metod pro kombinaci W-1-Č



Obrázek C.6: Úspěšnost metod pro kombinaci W-1-K

D Uživatelská dokumentace

D.1 Sestavení aplikace

Aplikace vytvořená pro účely této práce byla psána v jazyce Java SE (verze 1.8.0_101) a jako nástroj pro správu závislostí a automatizaci buildů byl použit Maven. Pro vlastní sestavení stačí v konzoli zadat následující příkaz.

```
$ mvn package
```

Aplikace se automaticky sestaví a ve složce `target` by pak měl být spustitelný soubor `keywordExtraction.jar`.

Během fáze testování metod je aplikací volán script `verification.pl` a je proto nutné mít v počítači nainstalovanou podporu pro programovací jazyk Perl (při tvorbě práce byla použita verze v5.18.2).

Doporučený operační systém, na kterém aplikace poběží, je některý z Unix-based systémů (Linux nebo případně OSX). To z toho důvodu, že aplikace byla trénována a testována na clusterech výpočetního centra MetaCentrum, které běží pod Linuxem.

D.2 Parametry spuštění

- **-h** – ukáže help text.
- **--help** – ukáže help text.
- **-r** – nastavení cesty ke kořenové složce aplikace. Musí být nastavena cesta do složky se strukturou jaká je popsána v kapitole D.3.
- **--corpus** – nastavení jména souboru, na kterém budou metody trénovány.
- **-e** – za tento parametr lze napsat názvy metod, které mají být natrénovány, otestovány a serializovány do souboru. Přípustná jména metod jsou `tfidf`, `tfidf-avg`, `lda` (spustí metodu LDA pro všechny témata – 50, 100 a 200), `lda(50)`, `lda(100)`, `lda(200)`, `vector-tfidf`

(spustí metodu GloVe pro všechny vektory uložené v adresáři vektorů viz níže) a `vector-tfidf-best` (spustí metodu GloVe ve verzi glove(840B-300d)).

- **-E** – parametr se stejnou funkčností jako `-e` s tím rozdílem, že vybrané metody jsou deserializovány z výstupní složky aplikace. Serializované metody jsou hledány v adresáři `out` (viz kapitola D.3).
- **--vocabulary** (nepovinný) – nastavení jména souboru serializovaného slovníku. Pokud není nastaven, jsou brána v úvahu všechna slova v souborech.
- **--stats** (nepovinný) – nastavení jména souboru, ve kterém jsou uloženy napočítané statistiky pro trénovací korpus. Pokud je nastaven tento parametr, nebudou se statistiky znovu napočítávat a načtou se ze souboru.
- **-c** (nepovinný) – počet klíčových frází, které budou extrahovány z testovacích dokumentů. Pokud není nastaven počet tímto parametrem, extrahuje se výchozí počet klíčových frází – 15.

Následujícím příkazem spustíme aplikaci ve složce `keywordextraction` pro metody TF-IDF, TF-IDF-avg, LDA a GloVe ve všech svých verzích. Budou natrénovány na korpusu `train/corpus.txt`, extrahovat se budou jen fráze složené ze slov uložených ve slovníku `train/vocabulary.bin` a načtou se statistiky ze souboru `train/statistics.txt`. Každá z metod z každého testovacího souboru extrahuje 15 klíčových frází.

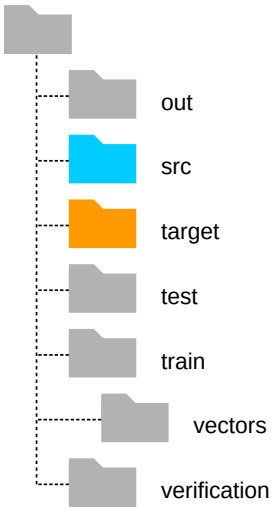
```
$ java -jar ./target/keywordExtraction.jar \
  -r /Users/Zibby/Workspace/keywordextraction\
  --corpus corpus.txt \
  --vocabulary vocabulary.bin \
  -stats statistics.txt \
  -c 15 \
  -e tfidf tfidf-avg lda vector-tfidf
```

D.3 Adresářová struktura aplikace

Na obrázku D.1 je vyobrazena struktura adresářů v kořenové složce projektu. Do adresáře `out` jsou ukládány výsledky testování metod, serializovány natrénované metody a všechny další výstupy programu. Složka `src` obsahuje

zdrojové kódy programu, které jsou pomocí nástroje Maven přeloženy a sestaveny do složky `target`. Testovací soubory se soutěže SemEval 2010 jsou umístěny ve složce `test` a správná klíčová slova od autora, čtenáře i kombinace obou dvou jsou pak ve složce `verification` společně se validačním scriptem.

Všechny soubory potřebné k natrénování metod musejí být umístěny ve složce `train`. To se týká trénovacího korpusu, uloženého slovníku, seznamu stop-slov a sad vektorů pro metodu GloVe. Vektory musejí být uloženy v adresáři `train/vectors`.



Obrázek D.1: Adresářová struktura aplikace