VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

METODY DOLOVÁNÍ DAT PRO ANALÝZU TEXTŮ

DATA MINING METHODS FOR TEXT ANALYSIS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Ondřej Kozák

VEDOUCÍ PRÁCE

SUPERVISOR

Mgr. Přemysl Dohnal

BRNO 2022

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Ondřej Kozák ***ID:*** 211155

Ročník: 3 ***Akademický rok:*** 2021/22

NÁZEV TÉMATU:

Metody dolování dat pro analýzu textů

POKYNY PRO VYPRACOVÁNÍ:

Účelem práce je prozkoumat aktuální metodiku a možnosti dolování dat v rámci zpracování textů.

1. Seznamte se s metodami dolování dat pro zpracování textů. Při plnění úkolu využijte zejména Text Analytics Toolbox v programu Matlab, případně jazyk Python, a zhodnoťte jeho vhodnost pro daný účel.
2. Přehledně uveďte metody analýzy textů a prodiskutujte perspektivy jejich využití.
3. Navrhněte příkladné aplikace metod, a to se zaměřením na anglické texty v oblasti elektrotechniky a komunikačních technologií.
4. Realizujte navržené aplikace.
5. Proveďte rozbor a zhodnocení dosažených výsledků.

DOPORUČENÁ LITERATURA:

[1] JO, Taeho. Text Mining. B.m: Springer, Cham, 2019. ISBN 2197-6503.

[2] SEMENOVA, A .V. a V. M. KUREICHIK. Ensemble of classifiers for ontology enrichment. Journal of Physics: Conference Series [online]. 2018, 1015, 032123. ISSN: 1742-6596.

Termín zadání: 7.2.2022 ***Termín odevzdání:*** 23.5.2022

Vedoucí práce: Mgr. Přemysl Dohnal

doc. Ing. Václav Jirsík, CSc.
předseda rady studijního programu

Abstrakt

Tato bakalářská práce se zabývá prozkoumáním aktuální metodiky a možností textového dolování a následné aplikace některých metod. V rámci práce byly popsány metody pro předzpracování, metody pro převedení textu do vektorového prostoru a metody pro analýzu textu a diskutováno jejich možné použití. Na text byly použity jednotlivé metody pro předzpracování a následně bylo demonstrováno převedení do vektorového prostoru jednoduchými metodami jako jsou BOW, Bag of n-grams, TF-IDF nebo metodami se strojovým učením které jsou FastText a GloVe. Na získané vektory byly použity metody LSA, LDA, TextRank, kosinová podobnost, pro získání informací z textu.

Klíčová slova

Textové dolování, TF-IDF, BOW, LSA, LDA, FastText, GloVe, TextRank, kosinová podobnost

Abstrakt

This bachelor thesis explores the current methodology and possibilities of text mining and the subsequent application of some methods. The thesis described methods for preprocessing, methods for converting text to vector space and methods for text analysis and discusses their possible applications. The different preprocessing methods were applied to the text and then the conversion to vector space was demonstrated using simple methods such as BOW, Bag of n-grams, TF-IDF or with machine learning methods which are FastText and GloVe. LSA, LDA, TextRank and cosine similarity methods were applied to the extracted vectors to extract information from the text.

Keywords

Text mining, TF-IDF, BOW, LSA, LDA, FastText, GloVe, TextRank, cosine similarity

KOZÁK, Ondřej. *Metody dolování dat pro analýzu textů*. Brno, 2022. Dostupné také z: https://www.vutbr.cz/studenti/zav-prace/detail/141631. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Přemysl Dohnal.

Prohlášení autora o původnosti díla

Jméno a příjmení studenta:	Ondřej Kozák
VUT ID studenta:	211155
Typ práce:	Bakalářská práce
Akademický rok:	2021/22
Téma závěrečné práce:	Metody dolovaní dat pro analýzu textů

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

.

V Brně dne: ----------

podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Mgr. Přemyslu Dohnalovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne: ----------

podpis autora

Obsah

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

Úvod

Tato práce se zabývá textovým dolováním anglických textů. Textové dolování je obor zabývající se získáváním informací z textů, které jsou i nejsou na první pohled vidět. S rostoucí digitalizací roste množství textů, které pro člověka není možné zpracovat. Tyto texty mohou být komentáře, vědecké články, knihy, emaily a další. K zpracování textů se používají statistické metody, strojové učení a hluboké učení.

Textové dolování je z velké části převádění textu do vektorového prostoru, ze kterého je možné získat informace využitím statistických metod nebo přípravou pro strojové učení. Příkladem textového dolování je vyhledávač, který při zadání vstupních slov generuje další slova, které by uživatel mohl chtít zadat. U komentářů se dá použít textové dolování například k zjištění jejich sentimentu, jestli jsou ohlasy pozitivní nebo negativní. Pomocí textového dolování se dokáže zjistit téma textu a shlukovat podobné texty k sobě, tím se ulehčuje práce při třídění dokumentů a je možné zjistit, jak moc jsou si dokumenty spolu podobné.

1. DOLOVÁNÍ DAT

Tato kapitola se zaměřuje na popis různých druhů dolovaní dat a jejich využití.

1.1 Vztahové dolování dat

Relational data mining je v češtině obecně nazýváno dolování dat [1]. Jedná se o dolování strukturovaných dat. Strukturovaná data jsou taková data, která jsou uspořádaná v tabulkách nebo na webových stránkách. Je u nich přiřazeno, co znamenají např.: hodnota hodnocení je pod označením hodnocení, částka peněz je cena atd. Dobrým příkladem je v obchodě záznam prodejů. Metody analýzy těchto dat je klasifikace, regrese, analýza asociace, shlukovaní a clustering.

1.2 Webové dolování

Webové dolování slouží k získání a objevení informací z webových dokumentů, stránek a služeb. Existují tři typy webového dolování.

Webové dolování obsahu (web content mining) je nejpodobnější textovému dolování zabývá se získáváním užitečných informací z webů. Internetová data jsou různá např.: texty, obrázky, audia, videa atd. Jednou z řešených úloh je shlukovat do skupin dokumenty a stránky podle tématu, jestli spolu souvisí. Dalším úkolem je stručný souhm textu.

Dolování struktury webu (web structure mining) slouží k zjištění struktury stránky a mapování toho jaké má vztahy s jinými stránkami pomocí „link analýzy“, která pomocí hypertextového odkazu shrne všechny odkazy na jiné stránky. Aplikace dolování může být k předvídání tématu stránky podle slov v textu, odkazů na jiné stránky, html tagů, což jsou kategorie při psaní webových stránek např.: nadpis bude mít tag <title>. Tak to fungují vyhledávače a doporučují stránky podle podobných kritérií. Struktura stránky má vliv na to, jak ji přijme vyhledávač. Mapování stránky poukáže na to, jestli na stránce jsou informace jednoduše dostupné a intuitivní pro uživatele. Aby stránka byla doporučena vyhledávačem je důležité, aby informace byly jednoduše dostupné.

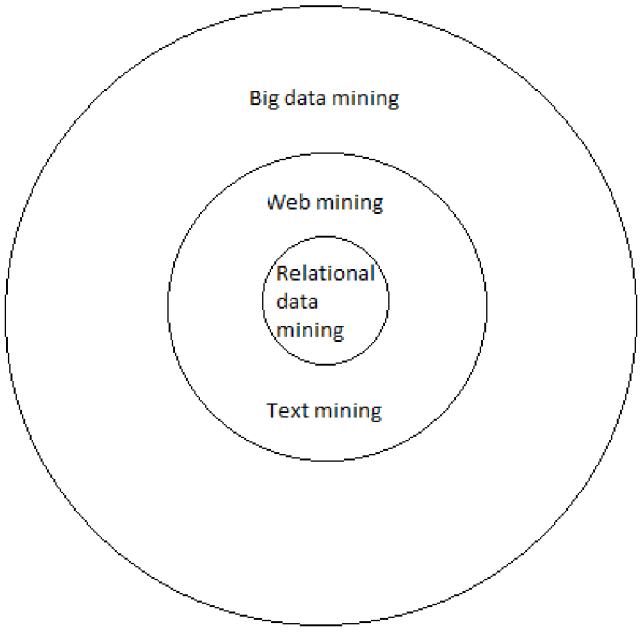
Webové dolování aktivity (web usage mining) se zabývá porozumění chování uživatele na webové stránce. Vstupem tohoto dolování jsou např.: serverové logovací soubory. Na těchto souborech je zaznamenána informace o požadavku uživatele na stránce. Informace obsahují uživatelovu IP adresu, čas, datum, požadovanou stránku, http kód a další. Tyto informace nejsou volně dostupné a má k nim přístup jen správce internetové stránky. Z těchto informací lze pomocí dolovacích metod předvídat chování uživatele nebo upravovat stránku podle chování uživatelů. Webové služby mohou doporučovat, co by uživatel chtěl sledovat, nebo vyhledávat na základě historie používání služby. Podrobnější informace o webovém dolování lze najít na [3].

1.3 Dolování velkých dat

Velká data (big data) je tak velké množství dat, že tradiční aplikace pro zpracování a analýzu dat nejsou schopná tyto data zpracovat [2]. Velká data se získávají z mobilních zařízení, kamer, mikrofonů, senzorů, softwarových logů a mnoha dalších. Vstupní data jsou typu strukturovaného, polostrukturovaného a nestrukturovaného, ale hlavní zaměření je na nestrukturovaná data jako jsou videa, audia texty atd. Výzvou není jen velikost ke zpracování, ale i úložný prostor nebo přesun dat přes síť. Velká data se používají k učení predikujících systémů a k hledání skrytých souvislostí v datech nebo jako trénovací data pro strojové učení a umělou inteligenci. Příklad predikujících systému je využití v bezpečnosti. Po hackerském útoku se mohou získaná data využít k predikovaní dalších útoku. Dále se dají využít v průmyslu, ze záznamu přístrojů se dá predikovat porucha a naplánovat servis před nastáním poruchy. Predikující systémy mají velký vliv pro rozhodování o řízení firmy a propagaci produktů.

1.4 Textové dolování

Textové dolování (text mining) je dolování na nestrukturovaných textech jako jsou články, knížky, vědecké práce, komentáře, emaily atd. Často se objevuje pojem analýza textu, která je součástí textového dolování. Příkladem textového dolování je zjištění frekvence výskytu slov, porozumění textu ve smyslu, že přiřadí ke slovu, jaký to je slovní druh nebo zařadí slovo do známých kategorií. U slova Praha, by bylo přiřazeno, že se jedná o podstatné jméno a že je to město. Další metody jsou předvídaní textu. Příklad je vyhledávač, který doporučuje, co by mohl chtít uživatel napsat. Sentimentální analýza je jedna z metod textového dolování. Tato analýza zjišťuje, jestli je text pozitivní nebo negativní. Velké využití je u komentářů, kde se dá zjistit co si lidé o daném příspěvku myslí. Další z metod je shrnutí obsahu textu. Výsledkem je stručně shrnut text, na který byla analýza použita. Textovým dolováním se tato práce bude zabývat více v příští kapitole.



Obrázek 1.1 Vztah mezi typy dolování

2. PROGRAMOVACÍ JAZYK

Tato kapitola se zabývá výhodami a nevýhodami programu matlab a jeho rozšířením analytic toolbox při zpracování a analýze textu v porovnání s programovacím jazykem python.

2.1 Matlab

Matlab je platforma pro programovaní a numerické výpočty a je vyvíjena společností MathWorks. K dolování dat se používá matlab text analytics toolbox, který obsahuje algoritmy pro předzpracování, analýzu a modelování dat. Matlab analytic toolbox má velkou výhodu, že všechny potřebné věci jsou na jednom místě jako funkce. Jak funkce k předzpracování, tak i samotné analýzy a jejich zobrazení. V matlabu je možné zavolat funkce jednotlivých metod, které jsou na základě strojového i hluboké učení. Jedná se o předem naučené programy, které lze hned využít, ale výsledky nemusí být ideální, protože jsou nacvičeny na obecných trénovacích datech. Pro lepší výsledky se mohou natrénovat na požadovaných datech.

Výhodou je že všechny funkce jsou u sebe a nemusí se vyhledávat a instalovat různé knihovny všechno je v jednom. Nevýhodou je že matlab není moc kompatibilní s ostatními programovacími jazyky a jeho dostupnost také není pro každého.

2.2 Python

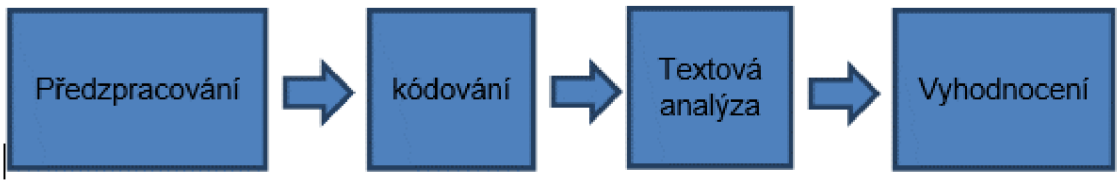
Python je vysokoúrovňový programovací jazyk, který je vyvíjen od roku 1991. Vyšlo už několik verzí, první v roce 1991 a python 2.0 v roce 2000. Poslední velká verze vyšla roku 2008 jako python 3.0, kde došlo ke změně programovací syntaxe a nových funkcí ulehčující programátorův život. Python byl dále vyvíjen, a i nadále jsou vydávány nové verze. Výhodou je jednoduchost a dostupnost. Python je programovací jazyk, není to prostředí jako u matlabu a tak nemá skoro žádné funkce pro dolování dat. K získání funkcí pro dolování dat a analýzu textu si uživatel musí stáhnout a naistalovat různé knihovny jako např.: spacy což je knihovna pro předzpracování textu. Python je volně dostupný (open source). Open source znamená, že je vyvíjen uživateli a díky tomu jsou vytvořeny spousty knihoven.

3. TEXTOVÉ DOLOVÁNÍ

Tato kapitola se podrobněji zaměřuje na dolování textu. Jakou problematikou se zabývá a jaké různé metody analýzy a zobrazení výsledků textového dolování jsou používány.

3.1 Proces textového dolování

Jak bylo zmíněno v minulé kapitole 1.4 textové dolování se zaměřuje na nestrukturovaný text. Takový typ dat není možné zpracovat pomocí běžných počítačových programů a kvůli tomu je nutné nejdřív text upravit. Tomuto procesu se říká předzpracování. Na předzpracovaná data se dají použít některé metody analýzy textu, ze kterých se dostanou užitečné informace a zároveň zakódují text do numerických vektorů, protože metody strojového učení pracují s numerickými vstupy. Na numerické data se využijí metody analýzy textu a poté už zbývá jen vyhodnocení.



Obrázek 3.1 Proces textového dolování

3.2 Předzpracování textu

Předzpracováním je upraven vstupní text, který byl získán z dokumentů. Tyto dokumenty mohou být různého typu jako např.: pdf, docx a další. Úpravy se mohou lišit podle jazyka, kterým je dokument napsaný a co v textu je analyzováno. Základními operacemi pro anglické texty jsou:

- Tokenizace
- Odstranění stažených tvarů
- Odstranění diakritických znamének
- Převedení na velká nebo malá písmena
- Stemování
- Lematizace
- Korekce pravopisu
- Odstranění slov bez většího významu (stop words)

3.2.1 Tokenizace

Tokenizace je proces rozdělení textu do menších částí nazývaných tokeny. Nejčastějším typem je slovní tokenizace, kde jeden token odpovídá jednomu slovu textu. Jsou i tokenizace na věty, sekvence slov atd. Při rozdělení textu se rozděluje na mezerách mezi slovy nebo interpunkčními znaménky. Nejjednodušším způsobem, jak provést tokenizaci v pythonu je převést řetězcový text do listu a rozdělovat text podle mezer a interpunkčních znamének. Tento způsob je jednoduchý, ale zbytečně náročný pro počítač. Knihovna NLTK nabízí tokenizátory různých typů.

3.2.2 Odstranění stažených tvarů

V angličtině dochází ke zkracování tvarů slov např.: he‘s je zkrácenina he is, I’m je zkrácenina I am. Těmito tvary dochází k přidávání interpunkčních znamének a při použití tokenizace podle interpunkčních znamének by docházelo k vzniku tokenů, které by nemusely dávat smysl. K zabránění vzniku těchto tokenů se musí slovo rozšířit zpátky do základního tvaru.

3.2.3 Odstranění diakritických znamének

V anglickém jazyce se nevyskytují diakritická znaménka, ale mohlo by se stát, že člověk píšící text nemusí být rodilý mluvčí a přepsal se. Odstraněním diakritických znamének by se opravily některé chyby.

3.2.4 Převedení na velká nebo malá písmena

Převedení na malá nebo velká písmena se dělá, protože by se v textu mohl vyskytnou šum různě rozházených velkých písmen jako např.: „HeLlO“. V různých metodách analýzy textu by mohlo dojít, že by slovo bylo několikrát napsané jinak, a to by ovlivnilo přesnost metody. U četnosti výskytu slova bychom dostali, že slovo se tam vyskytuje pětkrát, ale to by nebyla pravda, protože je ještě několikrát jinak napsané. U velkých písmen na začátku věty by mohlo dojít, že slovo s velkým prvním písmenem bude považováno za důležitější a dále by to ovlivňovalo přesnost metod. Ne vždycky je požadováno převést písmena na jednotlivý tvar, ale nejčastěji se převádí na malá písmena.

3.2.5 Korekce pravopisu

Text opravujeme ze stejného důvodu jako u převodu na velká nebo malá písmena. Snažíme se normalizovat text, aby se neztratily důležité informace z textu způsobem různých tokenů, které by jinak vznikly. Jsou různé korekce textu opravuji se chyby typu překlepů např.: „wordd“. Dále se opravují pravopisné chyby.

3.2.6 Stemování

Stemování je proces získání základního tvaru slova. Tento proces probíhá odstraňováním přípon a předpon. V anglickém jazyce se využívá pouze odstraňovaní přípon slova jako

jsou ed, ing, s. Stemováním mohou vznikat i slova, která nedávají smysl, protože algoritmus stemování odebírá jen koncovky a některá slova mají jiná pravidla. Pro nepravidelná slovesa se stemování nehodí.

3.2.7 Lematizace

Lematizace je velice podobná stemování. Účelem je taktéž získat základ slova, ale rozdílem je, že použitím lematizace nemůže vzniknout slovo, které nedává smysl. Proces lematizace porovnává slova podle různých slovníků a tím se považuje za mnohem pomalejší způsob získání kořene slova.

3.2.8 Odstranění slov bez většího významu (stop words)

V jakémkoli jazyce se vyskytují slova bez většího významu a z těchto slov nejsou získány žádné důležité informace. Tyto slova jsou v textu zátěží a zbytečně by zpomalovaly programy k analýze textu. V anglickém jazyce se jedná o slova jako např.: a, the, is, are a další.

3.3 Analýza textu

Analýzu textu využíváme ke zjištění informací z textu. Základní informace jdou získat algoritmy, které nám převedou text do numerické formy. Z těchto algoritmů lze získat např.: četnost výskytu slova nebo sousloví, přiřadit váhu slova, která určí důležitost slova v textu a další. Kombinací několika algoritmů lze získat komplexnější informace jako podobnost textu, téma textu, stručné shrnutí textu a další. Jedná se o tyto algoritmy:

- Bag of Words Model
- Bag of N-Grams Model
- TF-IDF Model
- Word2Vec Model
- The continuous Bag of Words Model
- The Skip-Gram Model
- The fastText Model
- The GloVe Model

Výše zmíněné algoritmy se také využívají k přípravě textu na další analýzu pomocí strojového nebo hlubokého učení pro metody jako jsou:

- LSA Model
- LDA Model
- TextRank
- BM25
- VADER
- Klasifikace

3.3.1 Bag of Words Metoda

Tento model převádí text do vektorového prostoru. Každému slovu je přirazena souřadnice (am, an) am souřadnice říká v kolikátém tokenu se vyskytuje slovo. Každému novému slovu je přiřazena souřadnice an a pokud se slovo opakuje má tuto souřadnici stejnou jako nalezená první instance tohoto slova. U an souřadnice se slova třídí podle abecedy. Tento model se dá zobrazit jako souřadnice s odpovídající hodnotou nebo jako matice, kde je zobrazena jen hodnota v souřadnicích. Z tohoto modelu lze zjistit počet všech slov nebo četnost výskytu jednoho slova.

3.3.2 Bag of N-Grams Metoda

Bag of N-Grams model funguje na stejném principu jako Bag of Words model. Rozdílem je, že umožnuje hledat množství frází a slovních spojení. U tohoto modelu je možné zvolit, jak velkou frázi je nutné vyhledat. Těmto shlukům slov se říká gramy. Při hledání například dvojce by se zaznamenaly všechny dvojce jdoucí po sobě a setřídily podle abecedy.

3.3.3 TF-IDF Metoda

TF-IDF stojí za term frequency-inverse document frequency, takže frekvence výskytu slova × inverzní frekvence dokumentu. Tento algoritmus se využívá na více dokumentů najednou k zjištění, jak jsou určitá slova v dokumentu důležitá. K určení důležitosti se ke slovu přidává váha, která se zvedá s každým výskytem slova. Kdyby se při předzpracováním neodstranila slova bez většího významu tak by tyto slova měly největší váhu, aby se tomuto zabránilo tak je důležité vědět, jak často se tyto slova vyskytují ve všech dokumentech. Tedy když se slovo vyskytuje ve všech dokumentech často tak se snižuje jeho váha. [4]
TF je jak často se slovo vyskytuje v dokumentu podělený počtem slov v dokumentu aby se normalizovali rozdíly v délce dokumentů.[5]

$$TF(t) = \frac{Jak\ často\ se\ slovo\ vyskytuje\ v\ dokumentu}{Počet\ slov\ v\ dokumentu} \tag{3.1}$$

IDF říká, jak důležité slovo je v dokumentu. Počítá se podle rovnice 3.2. kde je logaritmus z počtu dokumentů děleno počtem dokumentů s obsahem slova. Často se tato rovnice píše s plus 1 aby se zabránilo dělení nulou a logaritmu nuly.

$$IDF(t) = log\left(\frac{Počet\ dokumentů}{Počet\ dokumentů\ s\ obsahem\ slova}\right) \tag{3.2}$$

TF-IDF se vypočítá.

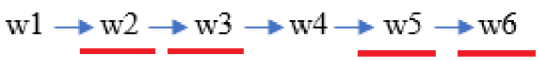
$$TFIDF = TF \cdot IDF \tag{3.3}$$

3.3.4 Word2Vec Model

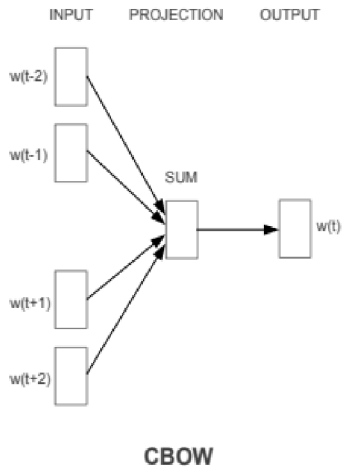
Word2Vec model byl vydán společností Google v roce 2013 a využívá metody hlubokého učení specificky neuronové sítě. Zaměřuje se na vkládání slov (word embeding). Word embeding je proces kde se slova převádějí na vektory. Základní převod slova na vektor je, když se vezme slovo ze slovníku, které se převádí na vektor. Vektor tohoto slova bude mít všude nuly a jedničku na indexu místa kde se slovo vyskytuje. Příklad je, slovník o 5000 slovech a vybrané slovo je na 2000 pozici. Vektor, který vznikne má 5000 dimenzí a v každé dimenzi je nula až na pozici 2000, kde je jednička. Z tohoto převodu slova na vektor se nezískají skoro žádné užitečné informace a pro zpracování pomocí strojového nebo hlubokého učení je to příliš náročné, protože vzniklé vektory jsou příliš velké. Také převody na vektor se dělají například s Bag of Words modelem. Pomocí word embeding je možné tyto vektory zmenšit a zjistit některé vlastnosti slov podle, kterých poté budou převedeny na vektory. Převod probíhá tak, že se určí, kolik vlastností se chce zjistit a podle toho bude vektor velký. Slova s podobným významem budou mít vektory blízko sebe. Word2Vec má dvě metody kterýma se word embeding provádí. The continues Bag of Words a Skip-Gram model.

3.3.5 Continuous Bag of Words Model (CBOW)

Je metoda Word2Vec modelu, která používá neuronové sítě k předvídání slova podle kontextu textu, na který byla naučena. V této metodě se zvolí na kolik slov se bude pozorovat kolem hledaného slova v by vstupem byly čtyři slova která jsou červeně podtržená a výsledkem by bylo slovo w4. Více o CBOW v praci [6].



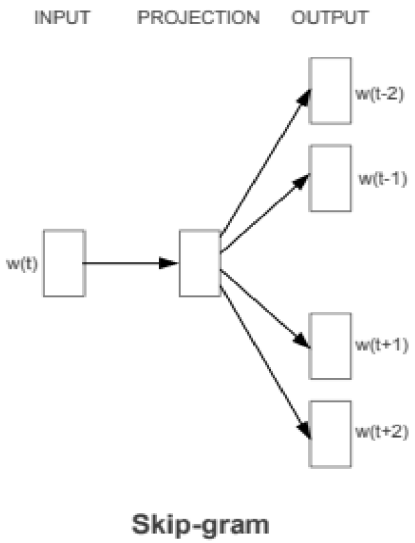
Obrázek 3.2 Výběr kontextových slov jako vstupy pro CBOW



Obrázek 3.2 CBOW architektura [6]

3.3.6 Skip-Gram model

Je též metodou Word2Vec a je přesným opakem CBOW. Ve Skip-gram modelu je vstup slovo, od kterého hledáme slova kontextová. Určuje se tam jak velké okolí slova, které se bude prozkoumávat a hledají se pravděpodobnosti slov kontextových. Tato metoda je náročnější z důvodu, že se musí provádět mnohem více predikcí, než u CBOW metody kde se predikuje jedno slovo. Použitím tohoto modelu se nevyužívá náročné násobení



Obrázek 3.3 Skip-gram architektura [6]

matic a dokáže se na jednom počítači naučit více jak 100 miliard slov za den, jak uvádí ve své práci o Skip-Gram modelu Tomas Mikolov [7].

3.3.7 FastText model

FastText model byl představen Facebookem, dnes Metou, v roce 2016. Je to rozšíření myšlenky Word2Vec konkrétně Skip-gram modelu. Jedná se o open source knihovnu pro textovou reprezentaci a klasifikaci. Obsahuje předučené modely s anglickými vektory slov naučených z wikipedie a webcrawlu nebo vícejazyčné vektory slov předučené na 157 různých jazycích. Tento model využívá skip-gram model. Místo toho, aby se využíval přímo na slova, tak se slova rozdělují na shluky písmen, kterým se říká n-gramy. Ve své práci [17] Tomas Mikolov uvádí příklad slova where a n = 3 slovo se rozdělí na <wh, whe, her, ere, re> a <where>. Celá slova se dávají do závorek (<>), aby se dala odlišit od slov rozdělených. Rozdělováním slov by se měla zaručit lepší vektorová reprezentace slov, které se tak často nevyskytují. Slova, která se nevyskytovala v knihovně při učení by mohla být reprezentována, složením rozdělených částí slov.

3.3.8 GloVe model

Je model, který jako Word2Vec model vytváří husté matice s malým počtem dimenzí oproti modelu bag of words a TF-IDF. Rozdíl je, že model World2Vec se zabývá lokální statistikou [13]. Lokální statistika je na datech konkrétního dokumentu a ve své práci Jeffrey Pennington [14] mluví o tom, že modely jako Word2Vec ztrácí informace z hlediska všech zkoumaných dokumentů naopak LSA je model, který se zabývá převážně globální statistikou a nemá podrobné informace z konkrétního dokumentu. Model GloVe se snaží zahrnout obě statistiky, aby dosáhl podrobnějšího zmapování slov. Model Glove nevyužívá neuronové sítě k získání vektorů, ale jsou přímo vytvořeny na slovech tak, aby vektor dvou slov byl roven logaritmu počtu kolikrát se slova objeví vedle sebe [15].

3.3.9 LSA model

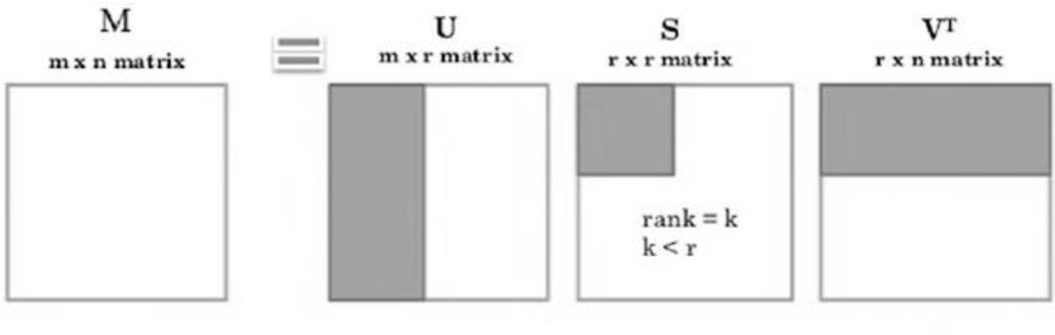
LSA stojí za latent semantic analasys (latentní sémantická analýza) metoda, která zkoumá vztahy mezi slovy a určuje témata textu. Proces LSA začíná získáním textů pro analýzu. Odstraní se slova bez většího významu jinak by mohlo dojít k tomu, že nejdůležitější slovo bude např.: the. Z předzpracovaného textu se vytvoří matice slov, která je jednoduchou reprezentací slov v dokumentu ve vektorech. Na vytvoření této matice se může použít metoda bag of words. Na této matici se provede rozklad na singulární hodnoty a zredukuje se množství dimenzí, které vznikly metodou bag of words. Rozklad na singulární hodnoty je postup, který lze použít na každou matici složenou z reálných dat, kde se daná matice rozloží na součin tří matic se speciálními vlastnostmi, které lze vidět v Obrázek 3.4 více o rozkladu na singulární hodnoty na webu [19]. Dále si uživatel může zvolit kolik témat chce v textu najít, ale nic mu nezaručuje, jestli budou všechny dávat smysl. Vyhodnocení muže být vyjádření v eukleidovském prostoru kde při dvou tématech slova související s prvním tématem zobrazovala na ose x a druhé na ose y. LSA lze využít na zjištění tématu a shlukování slov k daným tématům ale může se využít i

k souhrnu textu kde se vypočítá váha pro každou větu a nejdůležitější věty se zahrnou do souhrnu. Výpočet váhy pro věty se provede rovnicí 3.4.

$$Váhy = \sqrt{\sum_{i=0}^{k} S_i V_i^T} \tag{3.4}$$

$$S_i = 0 \; iff \; S_i < \frac{1}{2} S \tag{3.5}$$

V rovnici 3.4 je Si definovaný pomocí rovnice 3.5 kde Si je matice singulárních hodnot a V^T je matice s vektory vět.



Obrázek 3.4 Rozklad na singulární hodnoty [20]

3.3.10 LDA model

LDA model neboli latent dirichlet allocation (latentní dirichletovo rozdělení) je další tematický model. Tematické modelování je bezdozorovou klasifikací textu, který je rozdělován na skupiny slov, které vyjadřují nebo souvisí s nějakým určitým tématem. Témata nemusí být známá do konce provedení této metody. Metoda tematického modelování se používá k automatické organizaci dokumentů, porozumění textu a k souhrnu velkého množství textu. LDA se použije stejně jako LSA model na předzpracovaný text ve formě vektorů, který byl vytvořen přes bag of words. Ke každému slovu se přidělí náhodně jedno z předvoleného počtu témat. Poté se projde každé slovo a vypočítá u něj pravděpodobnost, jestli se vyskytuje v daném tématu. V prvním kroku se snaží zjistit kolik slov patří do daného tématu z konkrétního dokumentu. Slovo, které se zkoumá se nezapočítává. Pokud hodně slov z dokumentu patří do tématu je větší pravděpodobnost, že konkrétní zkoumané slovo do něj taky patří. Další krokem je

zkoumání, jak toto slovo moc patří do tématu z hlediska všech vstupních dokumentů. Tento proces se provádí na každé slovo a několikrát na celý dokument, aby se zpřesnilo rozřazení.[8]

3.3.11 TextRank

Je grafická metoda pro shrnutí obsahu textu. Funguje na základě algoritmu od Googlu PageRank, která hodnotí, jaká je pravděpodobnost, že se uživatel dostaví na určitou stránku z hlediska odkazu na stránkách a podle velikosti této pravděpodobnosti hodnotí stránky. Při použití textRanku se určí, kolik vět je požadováno a poté je text rozdělen na věty, to se zajistí metodou na předzpracovaní – tokenizací. Dalším krokem je word embeding tedy převedení textu na vektory. Převod na vektory je zajištěn metodami GloVe, Word2Vec nebo i bag of words a TF-IDF. Preferují se modely GloVe a Word2Vec z důvodu že modely bag of words a TF-IDF ignorují posloupnost slov a při použití na delší texty vzniká velké množství vektorů. Poté se musí vytvořit matice, která bude zkoumat podobnosti vět. Podobnost vět se vypočítá přes kosinovou podobnost nebo algoritmem BM25, který bude probírán v další podkapitole. Kosinová podobnost je úhlová vzdálenost vektorů. Lze si to představit, jako vykreslení vektorů do eukleidovského prostoru a změřením úhlu mezi nimi, tak cosinus tohoto úhlu určí jejich podobnost. Hodnota může nabývat v rozmezí hodnot [-1;1]. Hodnota blížící se k 1 znamená, že slova jsou si podobná, pokud se blíží 0 tak spolu nemají nic společného, ale když se blíží -1 tak to jsou opaky. Místo stránek jsou v TextRanku věty a místo odkazů jsou si vypočítány podobnosti vět, ze který se věty ohodnotí. Více o TextRank v práci [9].

3.3.12 BM25

BM25 stojí za best matching (nejlépe se shodující) a je to algoritmus, který je používán ve vyhledavačích. Vyhodnocuje stránky na základě kladených otázek. Jestli jsou nějaké podobnosti mezi kladenými otázkami a obsahem stránek. Když se podá dotaz, který má více jak jedno slovo, tak se dokumenty ohodnotí podle každého slova a pak se tyto hodnoty sečtou, tím se dosáhne konečného hodnocení obsahu stránek na větší dotazy. Dále se používá pro zjištění podobnosti dokumentů a textů. Je známý jako okapi BM25, protože první využití tohoto modelu bylo v City University v Londýně, kde v letech 1980-90 byl sestaven systém k získávání informací jménem Okapi, který byl poprvé využit na reálných datech. Tento model je založený na několika hodnotících funkcích. Jednou z hlavních metod je rozšířená metoda TF-IDF. Hodnocení textu se získá přes rovnici 3.4.

$$BM25(d_j, q_{1:N}) = \sum_{i=1}^{N} IDF(q_i) \cdot \frac{TF(q_i, d_j) \cdot (K+1)}{TF(q_i, d_j) + k \cdot (1 - b + b \cdot \frac{|d_j|}{L}}} \quad (3.4)$$

Rovnice 3.4 dává hodnocení BM25 pro dotazy obsahující slova q 1:N v dokumentu dj. TF(qi, dj) je počet, kolikrát se slovo qi objevilo v dokumentu dj. |dj| je délka dokumentu dj ve slovech. L je průměrná délka všech dokumentů které se hodnotí. K a b jsou volné parametry. K se většinou volí 2 a b 0.75. IDF je inverzní frekvence dokumentu. DF(qi) je počet dokumentů s výskytem slova qi. N je počet slov v dotazu.

$$IDF(q_i) = \ln \frac{N - DF(q_i) + 0.5}{DF(q_i) + 0.5} \quad (3.5)$$

U Rovnice 3.5 je rozdíl v N. Kde N je celkový počet dokumentů. Největší vlivy na hodnocení dokumentů má frekvence výskytu slova v text, inverzní frekvence dokumentu, kterou se vyloučí častá slova, která se vyskytují ve všech dokumentech. Délka dokumentu má taky velký vliv, protože když se párkrát hledané slovo objeví ve velkém dokumentu tak neznamená, že se tím zabývá oproti tomu, když bude velký výskyt hledaného slova v kratším textu. Více o BM25 v této práci [10].

3.3.13 VADER

VADER neboli valance aware dictionary and sentiment reasoner (valenční slovník a argumentace sentimentu). Je sentimentální slovník specializovaný na sociální služby [11]. Tento slovník byl vytvořen na základě prozkoumání vlastností existujících a ověřených sentimentálních slovníku, které v té době byly UWC, ANEW, GI. Doplněny o výrazy běžně se používající k vyjádření emocí na sociálních sítích jako jsou emotikony, slang a zkráceniny. Bylo ohodnoceno přes 9000 možných slov pomocí metody the wisdom of the crowd [12]. Z toho bylo vybráno kolem 7500 slov, které byly pečlivě ohodnoceny. Hodnocení je rozloženo na stupnici od -4 do 4 kde -4 je nejvíce negativní a 4 je positivní a 0 je neutrální. VADER vyhodnotí, jak moc je text negativní, positivní a neutrální. Hodnoty normalizuje na stupnici od -1 do 1 kde 1 je pozitivní a -1 negativní.

3.3.14 Klasifikace

Textová klasifikace je metoda rozřazování textu do předem definovaných kategorií. Klasifikace textu se může dělat manuálně kde text klasifikuje člověk, ale tato metoda je příliš zdlouhavá. Místo toho je automatická klasifikace kde se text klasifikuje pomocí systému zakládající se na pravidlech, systému založeném na strojovém učení nebo využití obou systémů dohromady. Systémy zakládající se na pravidlech musí mít slovník se slovy k dané kategorii. Aplikace tohoto systému může být roztřídění slov do kategorii nebo text jako celek, kde převažující slova kategorie definuje klasifikaci. Systémy se strojovým

učením mají dvě metody jedna je učení bez dohledu kde se model zajímá o vyskytující se vzory v textu a skrytých vlastností. Zakládá se na metodách shrnutí textu a tematických modelů. Učení s dohledem jsou algoritmy, které jsou učeny na datech určených pro učení. Z toho, co se naučily na trénovacích datech dokážou využít i na data co nikdy neviděly. Hlavní druhy učení s dozorem je klasifikace a regrese [16]. Nejčastější algoritmy ke klasifikaci jsou Naive Bayes, SVM (support vector machines), random forest a další.

4. NÁVRH METOD PRO ANGLICKÉ TEXTY

Tato kapitola se zabývá návrhy pro použití metod analýzy textu z kapitoly 3 a jak by se daly aplikovat na anglické texty, které se zaměřují na oblast elektrotechniky a komunikačních technologií.

4.1 Návrh použití Bag of words, Bag of N-grams, TF-IDF

Předzpracuje se text a postupně se na něm vyzkouší tyto metody. Nejprve budou vyzkoušeny na samotný dokument a poté na sbírku dokumentů. Výsledkem bude vektorová matice, která bude vyhodnocena a budou zjištěny nejčastěji používaná slova v textu u metody TF-IDF bude zjištěno jaká jsou nejdůležitější slova a jejich váhy. Zobrazením výsledku bude tabulka nejčastějších slov z dokumentu.

4.2 Podobnost textu pomocí kosinové podobnosti

Text bude předzpracován a pomocí metody TF-IDF se převede do vektorového prostoru. Na vektory textu se využije metoda kosinové podobnosti, pomocí které se získají podobnosti mezi jednotlivými větami. Výsledky podobností budou zobrazeny v teplotní mapě.

4.3 Shrnutí textu metodami LSA a TextRank

Text se předzpracuje a převede do vektorového prostoru. Zvolí se počet vět pro finální souhrn a provedou se metody LSA a TextRank.

4.4 Použití tematických metod pro získání témata textu

Jako vstup bude několik dokumentů, které se načtou jako jeden text a předzpracují se. Převedou se do vektorového prostoru a využijí se metody LSA a LDA pro získání nejdůležitějších slov v tématu. K tématům se poté přiřadí jednotlivé dokumenty.

4.5 Reprezentace slov v textu pomocí modelu fastText a GloVe

Text se předzpracuje a na předzpracovaném textu bude naučen model fastText, z kterého budou získány vektory slov, které se následně vykreslí. Pro Glove se sežene výstup z předem naučeného modelu, ve kterém se budou hledat slova v textu. Při nálezu se ke slovu přiřadí vektor daného slova.

5. PRAKTICKÁ ČÁST

Tato část se zabývá aplikací metod a předzpracováním v programovacím prostředí pycharm. Jaké knihovny jsou použity a výsledky použití těchto metod. Metody jsou zkoušeny na poskytnutých anglických dokumentech, které se zabývají oblasti elektrotechniky a komunikačních technologií.

5.1 Předzpracování

Tokenizace je jeden z nejdůležitějších procesů předzpracování. Jednoduchou tokenizaci umožnuje knihovna NLTK [24]. NLTK je open source knihovna zabývající se zpracováním jazyka, která nabízí přes 50 jazykových korpusů a lexikálních zdrojů a knihovny na předzpracování, které jsou tokenizace, stemování a další. Při tokenizaci je možné si vybrat mezi slovní a větnou tokenizací. Použitím větné tokenizace je získán list, kde každá instance tohoto listu bude jedna věta z textu. Tyto tokeny jsou pročištěny tím, že jsou odstraněna diakritická znaménka pomocí modulu unicode, který umožnuje přístup k databázi znaků a definuje jejich vlastnosti. Má funkci k normalizaci textu. Dále jsou odstraněny stažené tvary pomocí knihovny contractions, která obsahuje slovník s nejčastějšími staženými tvary. Projdeme každý slovo textu a pokud se nalezne stažený tvar tak ho je opraven. Všechna písmena jsou převedena na malá a jsou odstraněny diakritická znaménka a speciální znaky pomocí knihovny regular expression také nazývané re, která ulehčuje práci se řetězcovými proměnnými. Jako poslední věc jsou odstraněna slova bez většího významu. Po těchto úpravách vzniknou prázdné tokeny, protože v sobě měly číslo nebo jen diakritické znaménko tak je nutné odstranit tyto prázdné tokeny. Dále je potřeba ještě převést slova do základního tvaru pomocí lematizace. Lematizaci je provedena pomocí knihovny spacy [23], která se zabývá zpracováním přirozeného jazyka. Zpracovává text pomocí modulů, které se nazývají pipelines. Příklad pipline je parser, který ke slovům přiřazuje jejich slovní druh. Uživatel knihovny spacy má možnost si vytvořit vlastní pipelines. Při použití se rozdělí tokeny vět na tokeny slov, protože některé moduly jako lemanizace potřebuje pracovat s jednotlivými slovy. Text je předzpracovaný a lze použit některé metody analýzy textu.

5.2 Zjištění četností a důležitosti slov v textu

Pro tyto aplikace se využily metody převedení textu do vektorů pro jednoduší prací. K nejjednoduššímu převedení textu na vektory slouží metody bag of words, bag of n-grams a TF-IDF. Po převedení textu na vektory se lehce použijí funkce jako sumace nebo zjištění maximální hodnoty, která se v matici vyskytuje, z které získáme požadované informace.

5.2.1 Použití bag of words pro získání frekvence výskytu slov

Bag of words bylo použito pomocí knihovny scikit-learn [18]. Tato knihovna je zdarma dostupná a soustředí se na strojové učení pro programovací jazyk python. K převodu slov na vektory byla využita funkce CountVectorizer, u které je možné zvolit škálu hodnot při výskytu slova. Defaultně se jedná o bag of words model, ale je možnost ho upravit na bag of n-grams, tím že se zvolí velikost n-gramu. Výstupem funkce je velmi velká matice, kde řádek reprezentuje větu a sloupec slovo. K získání frekvence výskytu slov byl sečten každý sloupec zvlášť funkcí sum z knihovny numpy [25]. Pomocí knihovny pandas [26] se získají slova, která jsou v matici. Knihovna pandas slouží pro analýzu dat, zobrazení do přehlednější formy a umožnuje importovat různé formáty dat. Pomocí funkce DataFrame jsou slova zobrazeny do tabulky s maticí frekvence výskytu slova nebo vektorem součtu všech sloupců. K setřídění slov od nejčastějšího po nejméně časté byl vytvořen vnořený list, do kterého byly vloženy všechny dvojce slov a jejich četností. Při sestavě toho listu byly všechny prvky převedeny na řetězce a díky tomu správně nefunguje funkce pro list v pythonu sort z důvodu, že třídil prvky podle hodnot řetězce, a ne podle čísla v něm. Tím docházelo k tomu, že slovo, které se vyskytlo padesátkrát bylo pod slovem, které se vyskytlo jen pětkrát. Řešením je převod číselné hodnoty v listu na integer a poté nechat setřídit. K přehlednějšímu výsledku byla použita knihovna tabulate, která se zabývá přehledným zobrazením výsledků v terminálu používaného prostředí.

Words	Ocurence
magnetic	36
field	32
sample	21
susceptibility	19
material	14
reaction	10
specimen	8
model	8
value	7
space	7

Obrázek 5.1 Výsledek použití metody BOW

5.2.2 Použití bag of n-grams

Bag of n-grams bylo použito podobně jako bag of words jediná změna je v možnosti zvolení, jak dlouhé slovní spojení bude hledáno. Jako u metody BOW vznikne matice, kde řádky jsou věty a sloupce jsou slova. Pokud se slovo nachází ve větě přičte se v tom řádku jednička což je nejvyšší předem nadefinovaná váha. Pro zjištění nejčastějších slovních spojení se sečtou sloupce pro každé slovo zvlášť. Pomocí funkce Dateframe je možno získat slova, která se v textu vyskytují a přidat k nim jejich výskyt. Výsledky byly zobrazeny pomocí knihovny tabulate která byla využita stejně jak v u bag of words až na pár pythonovských uprav s listy.

Words	Ocurence
magnetic susceptibility	13
magnetic field	11
reaction field	10
magnetic flux	6
static magnetic	4
flux density	4
field vicinity	4
spherical cylindrical	3
paramagnetic diamagnetic	3
homogeneous field	3

Obrázek 5.2 Výsledek použití Bag of n-Grams

5.2.3 TF-IDF metoda

Metoda TF-IDF byla použita knihovnou sklearn neboli scikit learn. Je to knihovna pro strojové učení v pythonu. Tato knihovna má funkce pro získání dat z obrázků a textů. Jednou z funkcí je převedení textu do matice pomocí metody TF-IDF. Tuto metodu je možné využít bez knihoven naprogramováním rovnic 3.2 a 3.3. Využitím funkce TfidfVectorizer je získána matice z textu, která vypadá stejně jak u předchozích metod, ale místo přidání jedničky při výskytu slova se vypočítává váha přes uvedené rovnice. Z této matice lze zjistit slova s největší důležitostí. Je možné zjistit nejdůležitější slova v ohledu na celý text nebo jen slova s největší váhou ve větě. Sečtením každého sloupce v matici jsou získány slova s největší důležitostí v textu, které jsou zobrazeny v obrázku Obrázek 5.3. V obrázku Obrázek 5.4 jsou vidět slova s největší váhou v matici. Při

předzpracování textu vznikl problém, že vznikly věty, které měly jen jedno slovo a z toho důvodu tato slova měly nejvyšší váhu. K vyřešení tohoto problému se muselo upravit předzpracování textu a zabránit vzniku takových vět.

Words	Sum of weights for all occurencies in text
magnetic	4.69575
field	4.51633
sample	3.79137
susceptibility	3.33033
material	2.74214
reaction	2.16156
space	1.75265
specimen	1.74237
value	1.69771
calculate	1.67738

Obrázek 5.3 Slova s největší váhou po součtu vah všech výskytů slov

Words	highest weights
release	0.742385
source	0.669974
vector	0.645753
surround	0.63067
magnitude	0.61174
show	0.582308
table	0.569095
plot	0.568343
bz_emnc	0.568343
locate	0.553961

Obrázek 5.4 Slova s největší váhou

5.2.4 Vyhodnocení výsledků

Pomocí metody bag of words byly získány nejvíce používaná slova s jejich výskytem, které jsou zobrazeny v obrázku Obrázek 5.1. Tato metoda je vhodná pro zjištění počtu slov, které se v textu nachází a zároveň se získají jedinečné vektory pro slova nebo věty k dalšímu využití. Nevýhodou je velikost vektorů, která je ale problém u všech výše uvedených metod. Pomocí bag of n-grams byla zjištěna nejčastější dvoj slovní spojení,

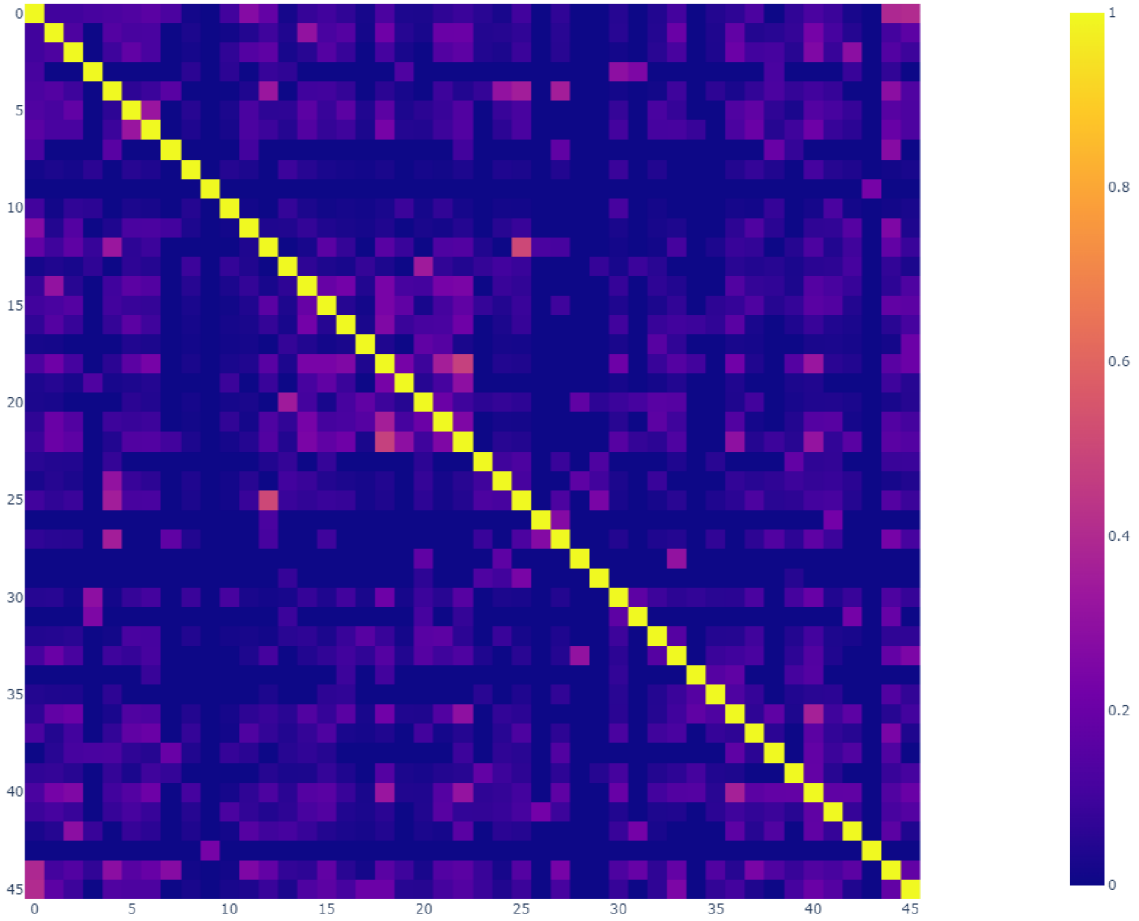
která se v textu vyskytují a jsou vyobrazeny v obrázku Obrázek 5.2. Pomocí metody TF-IDF byla zjištěna slova s nejvyšší váhou což jsou slova, která lze předpokládat za nejvýznamnější slova v textu obrázek Obrázek 5.3. V obrázku Obrázek 5.4 jsou sečteny všechny váhy pro výskyt slova. Při porovnání slov z obrázku Obrázek 5.3 a Obrázek 5.4 je vidět že některá slova s nižším počtem výskytu mají celkově větší váhu než slova s větším výskytem. A potvrzuje se, že slova s větším výskytem mají ve větách menší váhu, jak je vidět v obrázku Obrázek 5.4 kde slovo magnetic, které je nejčastější slovo v textu se ani v tabulce nevyskytuje. Tyto metody jsou vhodné k použití pro zjištění zastoupení slov v textu, četnosti jednotlivých slov a převedení textu do vektorového prostoru.

5.3 Podobnost textu a shlukování podobných vět

Využívá se zde metoda TF-IDF a kosinová vzdálenost pro zjištění podobnosti vět v dokumentu. Díky tomu můžeme přidávat jednotlivé chyby jako nové věty do textu a zjišťovat jejich výskyt.

5.3.1 Kosinová podobnost

Kosinovou podobnost byla použita knihovnou scikit learn, která již byla použita v kapitole 5.2. Tato knihovna obsahuje funkci cosine_similarity, která vypočítává úhel mezi jednotlivými vektory. Podobnost se určuje podle kosinové hodnoty daného uhlu. Pro přidělení vektorů k vypočítání podobnosti bere jako argument matici vektorů. Text převeden do vektorů byl vytvořen v předešlých metodách. Matice z metody TF-IDF je použita pro vypočítání podobnosti mezi každou větou a výsledkem vznikne matice podobností, kde na diagonále jsou jedničky. Jedničky se tam vyskytují, z důvodu porovnávání věty, kde se věta porovnává sama se sebou. Výsledkem jsou stejné vektory, z čehož vyplívá že mají mezi sebou nulový úhel a kosinus nulového úhlu je jedna, této hodnotě se říká kosinová vzdálenost. Tato matice je příliš velká a nepřehledná pro zobrazení z toho důvodu byla vykreslena jako teplotní mapa, kterou lze vidět na obrázku Obrázek 5.5 Teplotní mapa matice podobností. Pro porovnávání dokumentů je nejjednodušší sloučit dokumenty do jednoho a provést transformaci do vektorů. Poté provést výpočet do matice podobností, tím vzniknou vektory vět stejně dlouhé a nemusejí se sjednocovat vektorovou délkou.

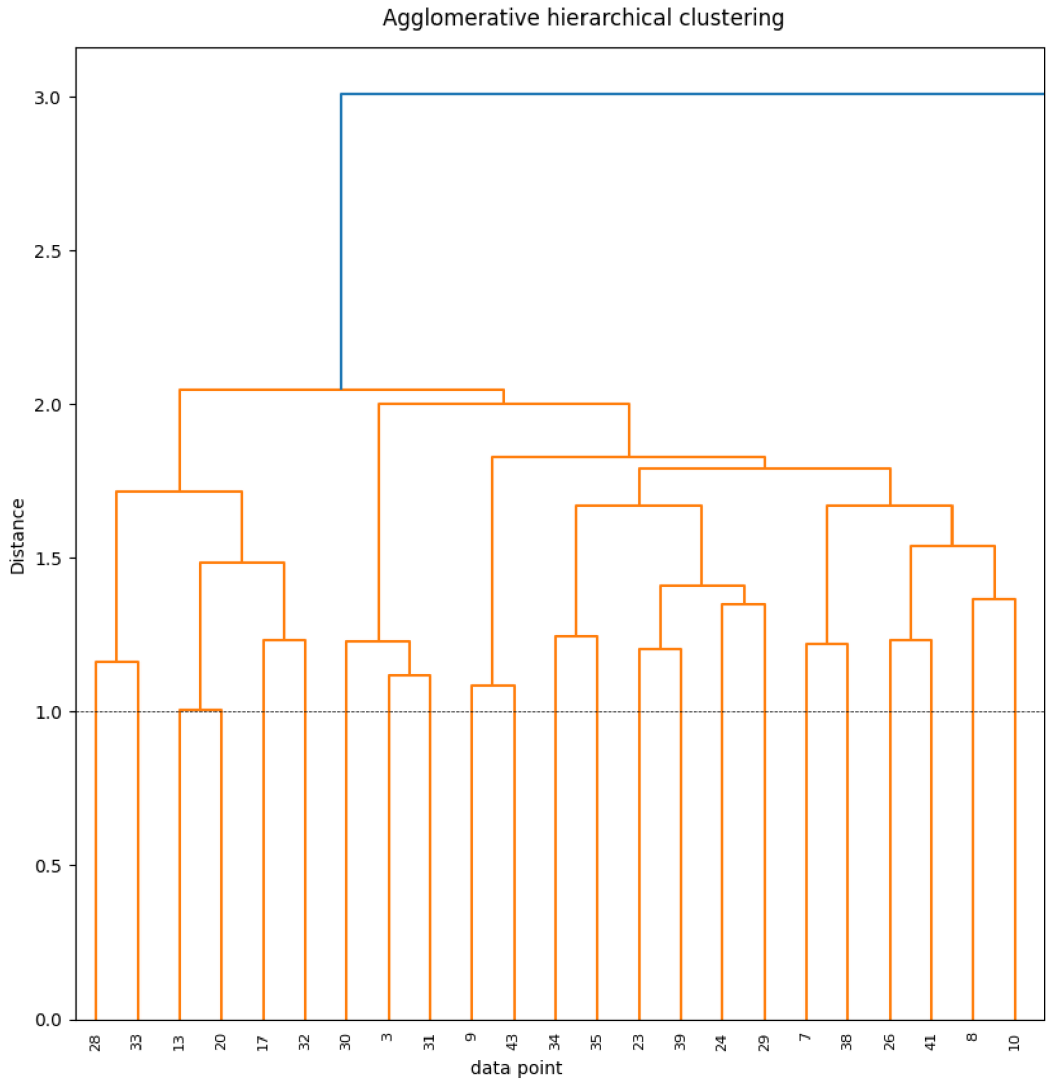


Obrázek 5.5 Teplotní mapa matice podobností

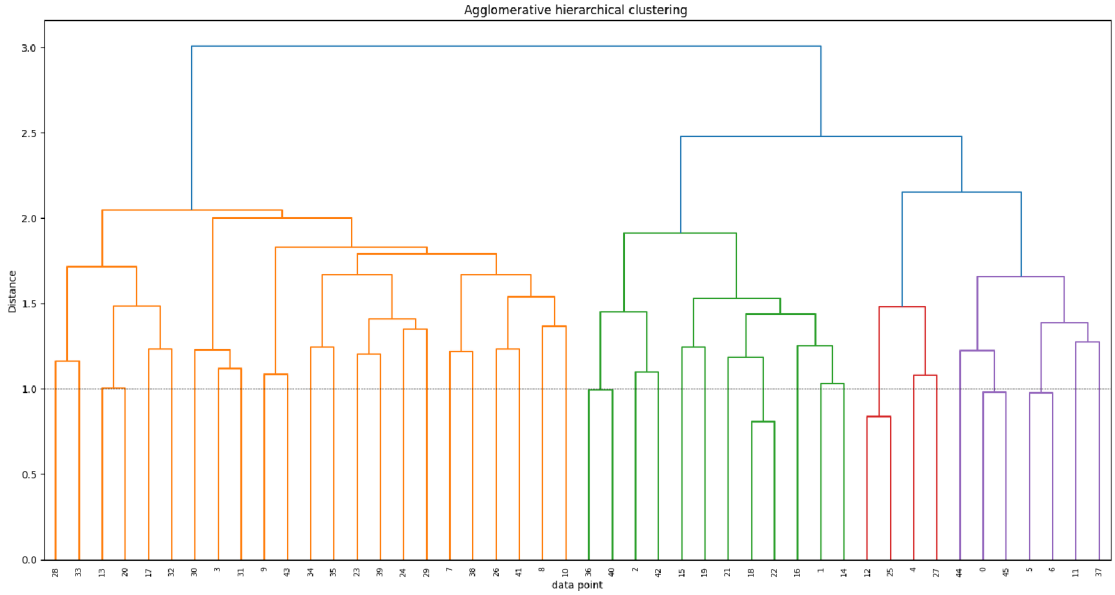
5.3.2 Aglomerativní hierarchické shlukování

Je metoda, která vytváří hierarchii podle podobností, jak na sobě věty závisí. Shlukuje věty k sobě podle podobností. U Aglomerativního hierarchického shlukování se jedná o postup kde se z mnoha shluku postupně stává jeden, tomuto postupu se říká bottom-up neboli od spodu nahoru. Existuje i opačná metoda která se nazývá dělící hierarchické shlukování. Hierarchické shlukování bylo použito pomocí knihovny scipy [27], která je využívaná pro vědecké a technické výpočty. Tato knihovna obsahuje funkci linkage, která provádí aglomerativní hierarchické shlukování. Vstupem této funkce je požadována matice vzdáleností, jež byla získána z kosinové podobnosti jako další parametr se určuje metoda. Metoda je algoritmus, pomocí kterého se přiřazuje vektor ke shluku. Používané metody jsou single, complete, centroid, ward linkage a další. Centroid bere průměrnou hodnotu shluku, která se vyskytuje ve středu shluku, od kterého se měří vzdálenost. Single bere nejbližší krajní hodnotu shluku a complete je pravý opak bere nejvzdálenější hodnotu shluku. Ward linkage hledá vzdálenost mezi nejbližšími prvky shluku a pak od této hodnoty přiřazuje nejbližší větu. Tato metoda byla využita v tomto příkladě. Výsledky se

často zobrazují v teplotních mapách nebo v dendogramu. Část dendogramu je zobrazena v obrázku Obrázek 5.6. Graf zobrazuje pouze část dat z důvodu přehlednosti. V grafu jsou vidět, jaké věty jsou si podobné a jak se shlukují k sobě. Na svislé ose je vzdálenost, jak se věty od sebe liší a na vodorovné ose jsou věty.



Obrázek 5.6 Část dendogramu z hierarchického shlukování



Obrázek 5.7 Celý dendogram hierarchického shlukování

5.3.3 Vyhodnocení výsledků

Pomocí kosinové podobnosti byla získána matice se vzdálenostmi jednotlivých vět od sebe. Podle teplotní mapy je vidět že nejvíce podobné věty jsou 26. s 13. a 23. s 19. Tyto věty po předzpracování zní:

> individual chapter characterize modeling reaction field vicinity variously shape sample spherical cylindrical cuboidal different material aluminium copper
>
> center space hold spherical cylindrical cuboidal sample copper aluminium material exhibit different magnetic susceptibility

Jedná se o 13. a 26. větu v upraveném textu a výsledek kosinové podobnosti je 0.5. Věty obsahují osm stejných slov, takže jsou velmi podobné. Tato metoda se dá využít k zjištění podobností textu pro zjištění plagiátorství, pokud jsou k dispozici předchozí práce pro daná témata, dále se dá využít k vyhledávání konkrétních výrazů v textu například chyb.

Z Aglomerativní hierarchické shlukování vyšly výsledky, kde se shlukují věty podobné, ale nejsou to věty, co vyšly nejpodobněji pomocí kosinové podobnosti, z důvodu použití algoritmu ward, kde se počítá průměr shluku po sloučení a poté se porovnává s původním shlukem. Tento proces se dělá pro každou variaci a vybírá se ta která nejméně změní průměr shluku. Hierarchické shlukování se dá využít například pro sjednocení témat vědeckých prací kde by sjednotilo práce zabývající se podobnou problematikou.

5.4 Souhrn textu

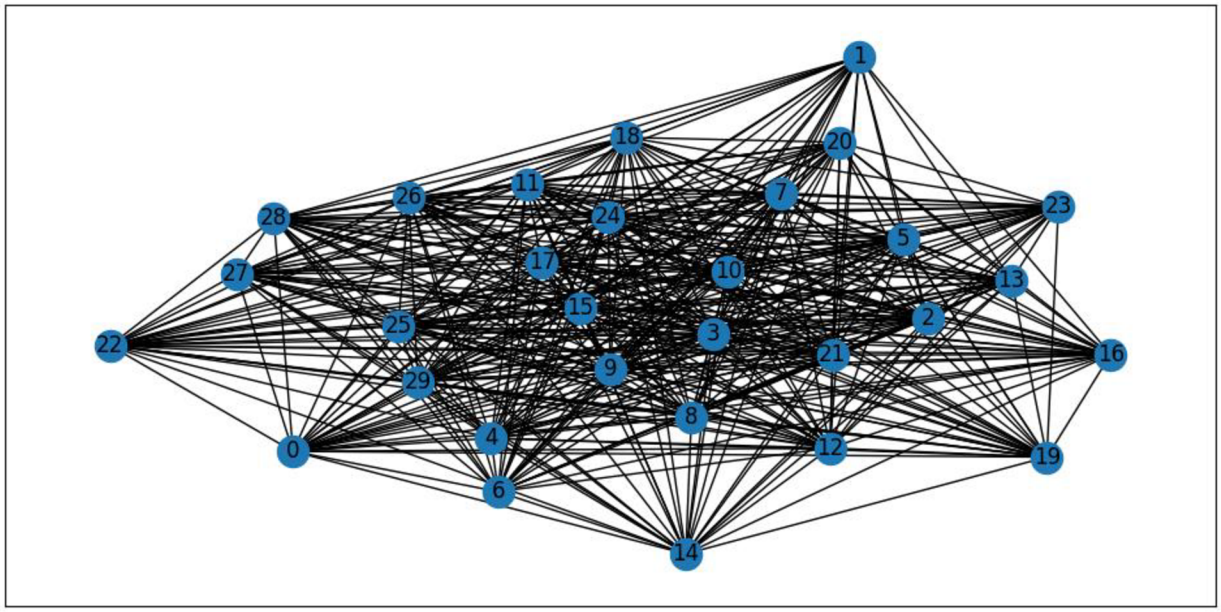
Souhrn textu je složitá problematika nejen tím, jak text shrnout ale také jak dlouhý má být nebo kolik témat se má zvolit pro tematické metody které se využívají. V této kapitole byly využity metody LSA a TextRank pomocí, kterých byly určeny nejdůležitější věty pro souhrn.

5.4.1 LSA

K Shrnutí textu pomocí metody LSA se jako první provede předzpracování textu. Předzpracování textu je důležitou částí z důvodu různé úpravy textu vycházejí různé výsledky. V tomto případě bylo vyzkoušeno shrnutí pro předzpracovaný text, kde nebyly odstraněny slova bez většího významu a čísla. V druhém případě byly odstraněny. Dále byl text převeden na vektory pomocí metody TF-IDF. V tuto chvíli je text připraven pro použití metody LSA. V knihovně scipy je funkce svd, která umožňuje vykonat rozklad na singulární hodnoty. Využitím této funkce jsou získány potřebné hodnoty a pomocí rovnic 3.4 a 3.5 se vypočítají váhy pro věty v textu. Vybere se jaký počet vět je požadován v souhrnu, podle toho se vybere tolik vět s nejvyšší váhou. Souhrny textu jsou uvedeny v příloze z důvodu použití této metody na rozsáhlý text.

5.4.2 TextRank

Pro TextRank se použije předzpracovaný text jako pro LSA metodu, aby bylo možné porovnat výsledky mezi sebou. Vezme se i výstup z TF-IDF na který se provede kosinová podobnost, aby byla zjištěna podobnost mezi větami. Pro TextRank se využije knihovna networkx [21], která slouží k vytváření, manipulaci a studiu struktury, dynamiky a funkcí komplexních sítí. Pomocí této knihovny se převede matice podobností na třídu graf, kde funkce from_numpy_array zajistí převod. Graf se vykreslí pomocí funkce draw_networkx a je možné ho vidět na obrázku Obrázek 5.8, kde byl vyzkoušen na recenzi, z důvodu přehlednosti grafu pro demonstraci. V grafu jsou věty vyobrazeny jako uzly a každý uzel má číslo, které reprezentuje o kolikátou větu se v textu jedná. Z každého uzlu jsou spoje, které reprezentují vztah mezi větami, pokud kosinová podobnost není nula tak je tam spoj. Při použití na dlouhý dokument se výstup funkce draw_networkx stává nepřehledným shlukem dat. Funkcí pagerank z knihovny networkx se získají skoré mezi větami. Vyberou se věty s největším skoré a seřadí se podle toho, jak se věty vyskytují v textu.

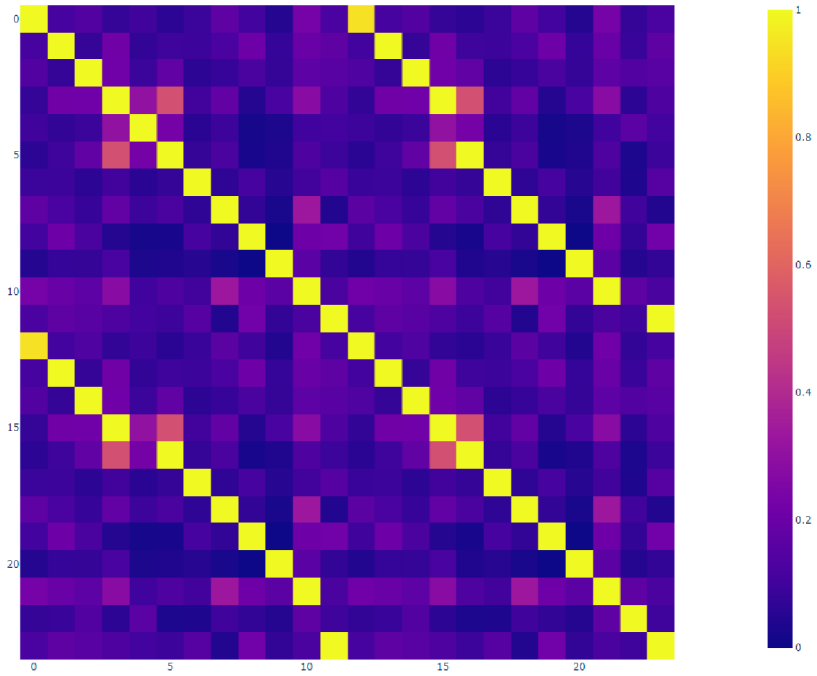


Obrázek 5.8 Graf podobností

5.4.3 Vyhodnocení souhrnu textu

Metoda LSA je jednoduchá pro použití, ale problém nastává při zvolení počtu témat pro matici singulárních hodnot, v tomto případě byl využit dokument Sample_text_1.docx který je přiložen v příloze, na kterém bylo vyzkoušeno různé množství témat. Rozdíl ve výsledcích se objevil v případě, kdy počet témat byl méně jak tři, kde se výsledky liší ve dvou větách oproti shrnutí s větším počtem témat. Pomocí TextRank je souhrn totožný jak při využití metody LSA s počtem témat vyšším jak tři. Vyhodnocení mezi TextRank a LSA metodou je vidět na obrázku Obrázek 5.9, kde je teplotní mapa kosinové podobnosti pro porovnání výsledků shrnutí. V teplotní mapě je prvních dvanáct vět metody LSA pro méně témat jak tři a zbytek vět metody TextRank. Z teplotní mapy je vidět, že pátá a jedenáctá věta se liší. Na diagonále se nachází sloučené shrnutí obou metod. Vysoké podobnosti pod a nad diagonálou jsou zrcadlené výsledky podobnosti metody TextRank.

Tyto metody nejsou vhodné pro shrnutí vědeckých textů z důvodu, že tyto metody zkoumají jen jak spolu slova souvisí a jak se vyskytují v textu a tím se ztrácí spousta informací. Hodnotí jen důležitost vět, tím že se vyberou nejlépe hodnocené věty nejde shrnout vědeckou práci. Tyto metody jsou výtečné pro použití na jednoduché texty, které se zabývají jednou věcí. Příklad takového textu jsou komentáře, recenze produktu a filmu. U shrnutí recenze filmu, když se ztratí velká část vět, tak to čtenářovi tolik nevadí, protože u recenze každá věta má účel ohodnotit daný produkt a čtenářovi nebude chybět, jestli polovinu kritiky neslyšel.



Obrázek 5.9 Teplotní mapa podobnosti shrnutí textu

5.5 Metody pro určení tématu textu

V této kapitole je popsáno využití metod pro získání tématu textu. Výsledkem jsou slova a dokumenty, která jsou přiřazená k jednotlivým tématům. Využity metody jsou LSA a LDA. Témata jsou přiřazována k šesti dokumentům, dokumenty se zabývají magnetickým polem, implementací a návrhem věcí do průmyslu 4.0 a využití umělé inteligence. Všechny dokumenty jsou poskytnuty v příloze. Pro jednodušší využití všechny dokumenty byly převedeny a uloženy v textových souborech pro jednotnost formátu dokumentů.

5.5.1 LSA

Tato metoda byla využita v souhrnu textu v kapitole 5.4.1 kde se provedla pomocí funkce svd z knihovny scipy. V této aplikaci byla využita knihovna sklearn, která obsahuje třídu TruncatedSVD. Jako první se načtou dokumenty a provede se předzpracování. Předzpracování obsahuje odstranění slov bez většího významu, převod na zaklad slova lematizací, převedení na malá písmena, tokenizace na slova a odstraněné čísel. Dalším krokem je převézt text na numerickou formu pomocí metody bag of n-grams. Poté se získají jednotlivá slova matice, metodou get_feature_names_out. Z knihovny skleam se použije třida TruncatedSVD. Tato třída má několik parametrů, které se nemusí zadat z důvodu předem nastavených defaultních parametrů. V tomto případě byl změněn

parametr pro počet témat z defaultní hodnoty na šest, protože jsou hledána témata ze šesti textů. Do konstruktoru třídy byla zadána matice z metody bag of n-grams. Dalším krokem je získání hodnot z této třídy. Hodnoty se získají metodou componets. Z této metody jsou získány váhy jednotlivých slov pro dané téma. K získání jednotlivých slov jsou zapotřebí jejich indexy, které jsou získány funkcí z knihovny numpy, která se jmenuje argsort. Tato funkce vrací indexy pole po setřídění. Po setřídění je k dispozici dvacet indexů slov s nejvyšší váhou, ke každému tématu. Ke každému indexu je nalezena váha a reálné slovo které se skrývá za indexem. Získaná slova se roztřídí podle znaménka vah, tedy na záporná nebo kladná. Orientace znaménka poukazuje na podtéma v dokumentu, slova záporná i kladná jsou důležitá, ale mohou poukazovat na jiná témata které se v textu nacházejí spolu. V Tabulka 5.1 jsou uvedeny slova k tématům.

Tabulka 5.1 Nejdůležitější slova témat z metody LSA

Topic1	Topic2	Topic3	Topic4	Topic5
power	power	detection	sensor	field
sensor	Loss	image	smartjacket	structure
system	mechanical	object	operator	magnetic
loss	measurement	detect	case	magnetic field
mechanical	measure	algorithm	data	wa
crossref	electrical	neural	layer	coil
communication	energy	network	wireless	rf
energy	uncertainty	learning	high	distribution
measurement	production	classi	wifi	result
asset	communication	set	standard	element
production	Asset	neural network	consumption	mr
component	product	optical	product	material
measure	Ua	rate	production	resonant
product	sensor	production	machine	sample
case	Opc	product	pp	used
data	manufacturing	asset	management	frequency
technology	opc ua	power	simulation	flux
method	industry	ua	system	shown
device	component	communication	service	grid
electrical	model	opc	time	value

5.5.2 LDA

Prvním krokem bylo provedeno stejně jak u LSA předzpracování textu a zároveň bylo využito matice z metody bag of n-grams. LDA model byl využit pomocí knihovny sklearn ve které se jmenuje LatentDirichletAlocation. Použité parametry funkce n_components, max_iter, learning_method, batch_size, n_jobs. Do parametr n_components je vložen

žádaný počet témat, který je v tomto případě šest. Do parametru max_iter byla zadaná hodnota sto, která určuje kolikrát se projdou trénovací data. Do learning_method se zadává metoda pro aktualizaci hodnot a vybírá se mezi metodou online a batch, kde online se používá pro větší množství vstupních dat. Batch_size se používá jen s online metodou a zadává kolik dokumentů se v každé iteraci využije. Pomocí argumentu n_jobs se určuje kolik jader procesoru se využije pro trénování dat. Z knihovny numpy funkcí argsort jsou získány indexy dvaceti slov s nejvyšší váhou. Pomocí indexu se vypíšou slova k tématům. Knihovnou pandas byl vytvořen datový typ dataframe, kterým vznikla tabulka kde řádky odpovídají danému dokumentu a sloupce tématům, tato tabulka je zobrazena v obrázku Obrázek 5.10. V tabulce je vidět ke kterému tématu dokument patří a je vidět že poslední tři témata nejsou dominantní v žádném dokumentu. Těchto témat je možné se zbavit tím že je zadáno méně témat, ale dosáhne se toho, že důležitá slova z ostatních témat se začnou přelévat do témat co zůstala a při přiřazení tématu k textu by mohlo nastat, že všechny dokumenty budou souviset převážně s jedním tématem.

Tabulka 5.2 Nejdůležitější slova témat z metody LDA

Topic1	Topic2	Topic3	Topic4	Topic5
field	detection	Power	sensor	Sensor
magnetic	image	Sensor	system	Systém
structure	object	Systém	case	Power
magnetic field	algorithm	Loss	component	Data
wa	network	communication	detection	Crossref
coil	detect	asset	model	Asset
sample	sensor	production	asset	Component
material	system	component	technology	Case
rf	set	crossref	opc	Production
susceptibility	neural	mechanical	data	Network
result	method	energy	crossref	Technology
distribution	classi	product	communication	communication
element	learning	data	power	Loss
mr	neural network	case	i4	Noc
value	pp	ua	product	Smartjacket
flux	rate	opc	device	Energy
frequency	individual	measurement	network	Layer
used	optical	industry	datum	De
shown	proceeding	technology	de	Device
magnetic flux	base	manufacturing	production	Datum

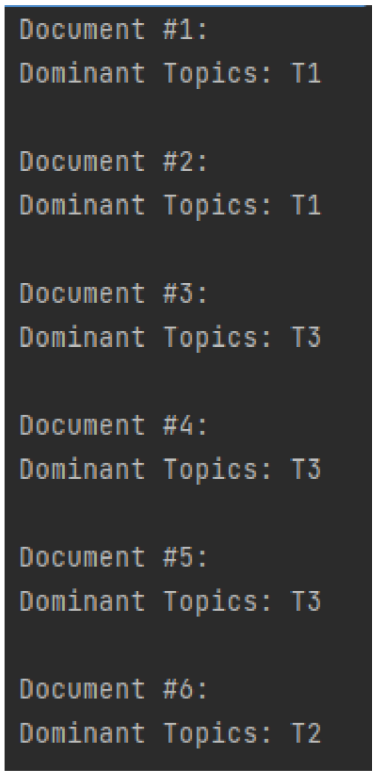
5.5.3 Vyhodnocení metod pro určení tématu

Metodou LSA byla získána slova pro zvolený počet témat. Mezi těmito slovy se někdy nachází slova, která jsou použita pro popis udělané práce např. used, result, set atd. Tyto slova jsou vybrána z důvodu vysokého výskytu v textu a vstup do funkce je text převeden na matici vektorů metody bag of n-grams, kde je modelu předána matice s frekvencí výskytu slov. Využitím metody TF-IDF by bylo možné zjistit některá důležitější slova. V Obrázek 5.10 jsou k dokumentům zobrazeny témata, která se v nich vyskytují. První téma je dominantní téma dokumentu. Výsledky odpovídají tématům v dokumentech, kde v prvním a druhém dokumentu mluví o magnetických polích a cívkách. Dokumenty dva až čtyři se zabývají průmyslem 4.0 a dokument šest se zabývá umělou inteligencí.

```
Document #1:
Dominant Topics: ['T5', 'T1', 'T6']

Document #2:
Dominant Topics: ['T6', 'T5', 'T1']

Document #3:
Dominant Topics: ['T1', 'T4', 'T2']

Document #4:
Dominant Topics: ['T1', 'T2', 'T4']

Document #5:
Dominant Topics: ['T1', 'T2', 'T3']

Document #6:
Dominant Topics: ['T3', 'T1', 'T2']
```

Obrázek 5.10 Přiřazená témata k jednotlivým dokumentům

Metodou LDA bylo dosaženo podobných výsledků jen s rozdílem zobrazení jednoho tématu pro dokument. Jelikož matice mimo maxima má stejné hodnoty kde se nedají vyhodnotit další témata. Výsledky LDA metody jsou vidět v Obrázek 5.11. Tyto metody je možné využít k setřídění dokumentů podle stejných témat. S využitím klasifikačních modelů by se výsledky mohly roztřídit nebo zjistit možné spolupráce. Nebo na již setříděné dokumenty použít některou z metod zjištění podobností a zjistit, jestli už někdo podobnou práci nedělal. Dalším využitím může být, využití nejdůležitějších slov tématu pro klíčová slova dokumentu.



Obrázek 5.11 Výsledky metody LDA

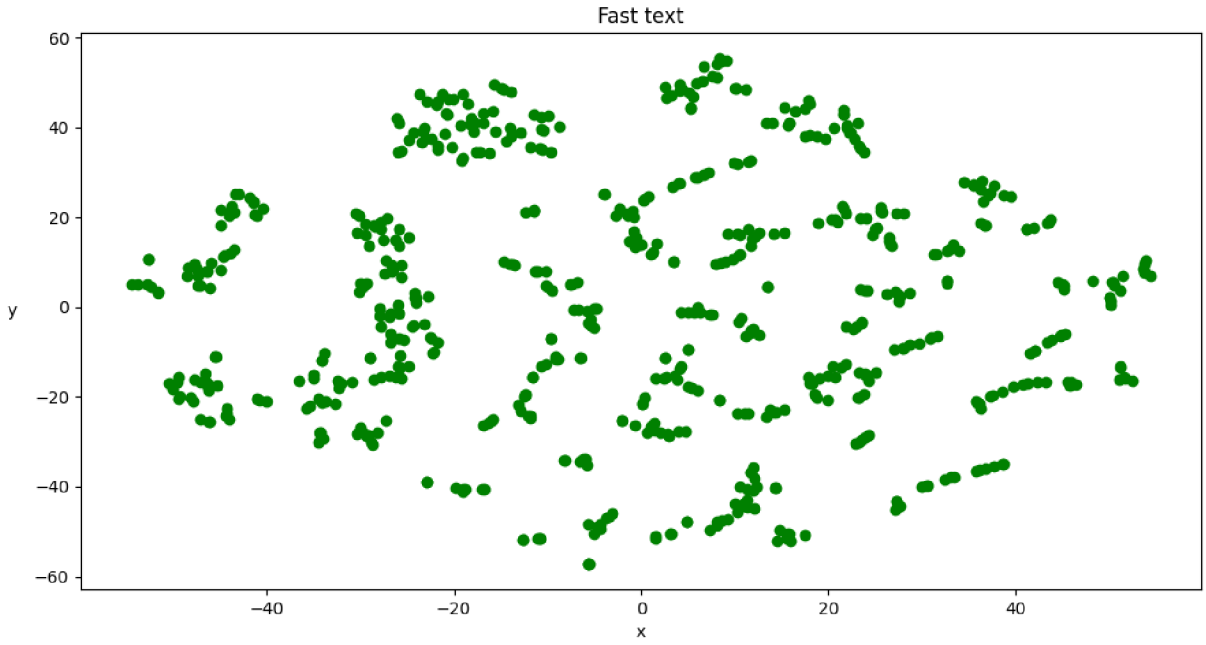
5.6 Použití metod pro reprezentaci a klasifikaci konkrétního textu

Tato kapitola se zabývá reprezentací slov v textu a převedení na vektory pomocí metod skip-gram a CBOW, které se využívají v Fasttext a GloVe metodách.

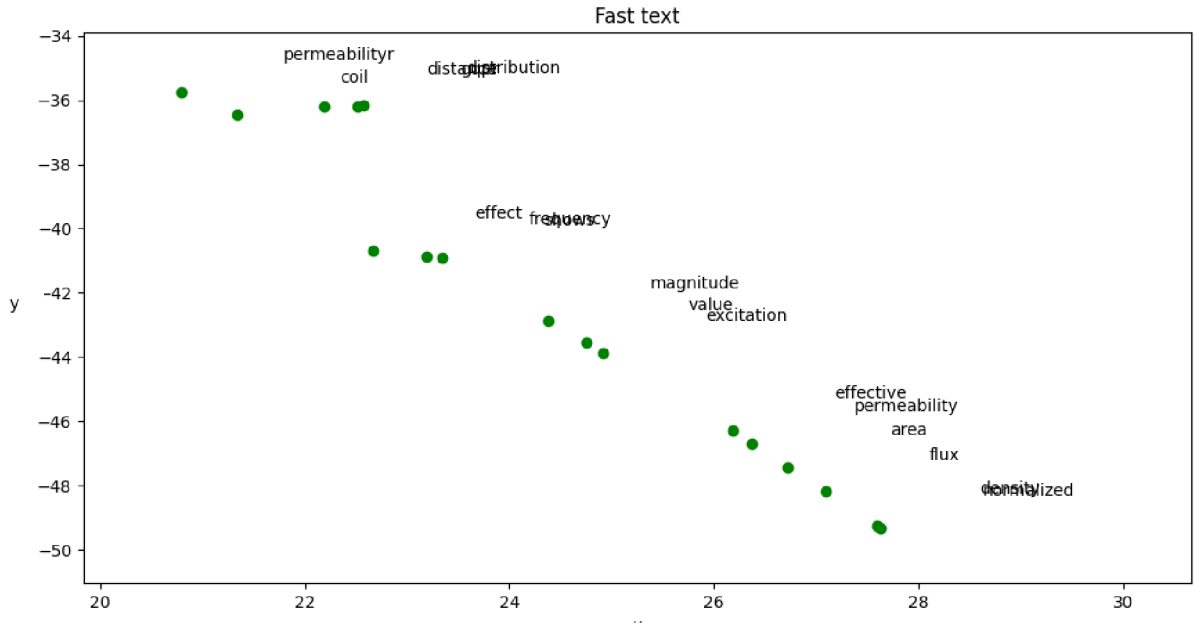
5.6.1 FastText

Pro využití fastText metody byla použita knihovna gensim [22]. Prvním krokem je načtení textu a tokenizace na věty. Tokenizovaný text se předzpracuje a výstup se uloží do textového dokumentu. Textový dokument je poté použit jako vstupní data pro třídu fastText, která metodou build_vocab z textového souboru vytvoří vlastní textový korpus. Vytvoří se instance třídy fastText, kde jsou zadány parametry vector_size, window, min_count, sg, sample a epochs. Vector_size omezuje velikost vektorů a byl nastaven na hodnotu sto, window parametrem se nastaví maximální vzdálenost mezi slovy ve větě a byl nastaven na hodnotu dvacet pět, parametrem min_count se ignorují slova výskytem menším nebo rovným, než je nastavena jeho hodnota. Hodnota min_count byla nastavena na pět, parametrem sg se vybírá mezi metodou skip-gram nebo CBOW nastavena byla

hodnot jedna, která odpovídá metodě skip-gram, a parametr epochs byl nastaven na padesát, který nastavuje kolik iterací se provede. Na instanci třídy fastText se použije metoda train, které se do parametru zadá textový dokument jako vstupní text kolik iterací se má provést, počet dokumentů a všechny slova v dokumentu. Výsledky modelu se zobrazí pomocí knihovny sklearn a třídou TSNE. Tato funkce se zabývá vizualizací dat s velkým počtem dimenzí.



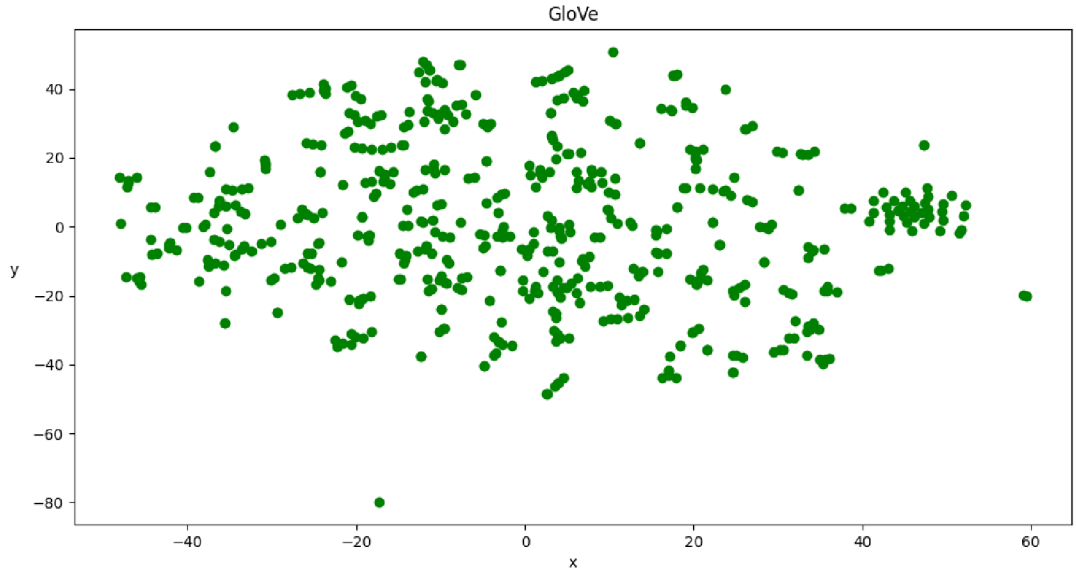
Obrázek 5.12 Vizualizace fastText reprezentace textu



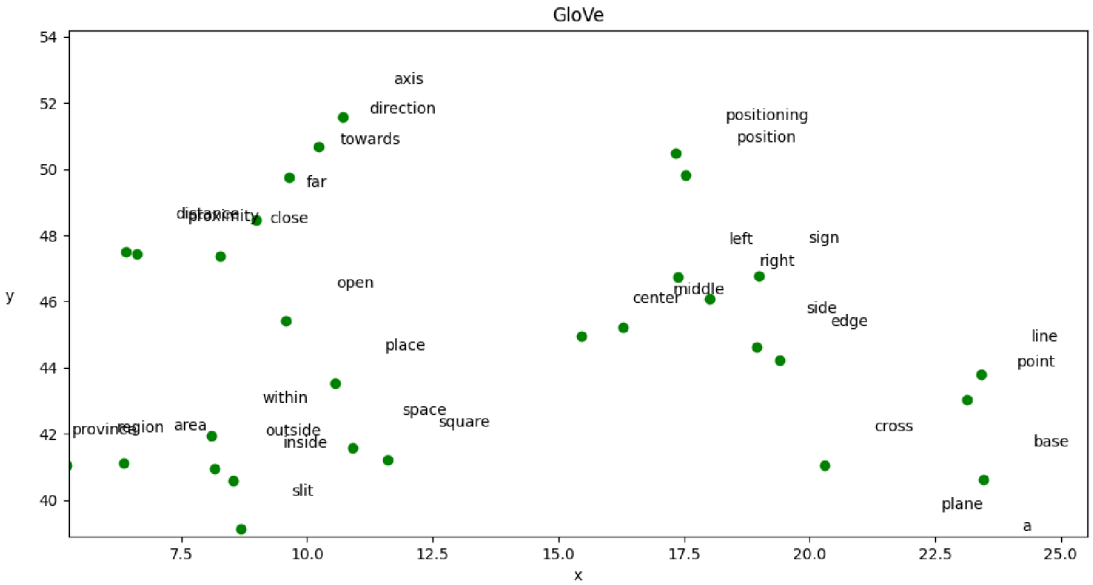
Obrázek 5.13 Zobrazení shluku z fastText reprezentace textu

5.6.2 GloVe

Metoda GloVe je využita pomocí knihovny spacy, kde je využit již naučený korpus en_core_web_lg, který obsahuje vektory 1,1 milionů slov, což je polovina slov, které nabízí projekt GloVe. Prvním krokem je předzpracovat vstupní text provedením lematizace kterou následuje tokenizace na věty a úprava odstraněním slov bez většího významu, odstranění čísel, převedení na malá písmena atd. Načte se korpus en_core_web_lg funkcí load. Získají se jednotlivá slova textu a porovnají se s textem v korpusu, pokud se v něm nachází je získán jeho vektor.



Obrázek 5.14 Zobrazení vektorů GloVe modelu



Obrázek 5.15 Zobrazení shluk GloVe modelu

5.6.3 Vyhodnocení modelů fastText a GloVe

Modely fastText a Glove byla získána reprezentace slov v textu a při porovnání výstupu metody fastText v obrázku a GloVe v Obrázek 5.14 je vidět rozdíl že fastText model je více roztříděn do shluků. Rozdíl je v tom, že fastText model byl naučen na konkrétní text kdežto u model GloVe byly získány vektory slov z předem naučeného modelu. Z bližšího pohledu v Obrázek 5.13 a Obrázek 5.15 je vidět že metodou fastText jsou získány shluky slov, která se spolu v textu vyskytují, zato model GloVe shlukuje slova, která se používají v podobném kontextu jako např: left, right, side, midle, center. Všechno jsou slova zabývajícím se pozicemi. Tyto modely se využívají pro důkladnější převedení textu do vektorového prostoru pro využití v další metodách. Mezi metodami jako je BOW nebo TF-IDF se tyto metody liší ve velikosti vektorů. Na velké texty metoda BOW má vektory dlouhé podle počtu slov v textu a metoda GloVe vytváří zhuštěné matice s malým počtem dimenzí.

6. ZÁVĚR

V první časti bakalářské práce byly popsány typy dolování dat a příklady jejich využití. V druhé časti byly uvedeny programovací prostředí, programovací jazyky, kterými se má textová analýza provádět a byly zhodnoceny jejich výhody a nevýhody.

V třetí kapitole bylo popsáno textové předzpracování, proč se dělá a různé metody, kterými se zabývá. Dále se zabývá metodami textové analýzy, kde jsou popsány metody od kódování textu do vektorového prostoru až po metody strojového učení, které využívají tento zakódovaný text.

V čtvrté kapitole byly navrženy aplikace některých metod textové analýzy. Jedním z návrhu je reprezentace textu ve vektorovém prostoru, shrnutí a získání témat z textu.

Pátá kapitola byla věnována praktické části, kde jsou zpracovány aplikace navržených metod. Pro zpracování se využilo prostředí pycharm a programovací jazyk python. Jako první byly popsaný metody předzpracování. Úspěšně se provedlo předzpracování pro odstranění slov bez většího významu, odstranění zkrácených tvarů a odstranění speciálních znaků. U lematizace se vyskytl problém, že knihovna, která byla použita neznala základy některých slov. Dále byly demonstrovány metody BOW, bag of n-grams, TF-IDF pro převedení textu do vektorového prostoru a vyčtení z těchto hodnot četnost výskytů a důležitost slov. Pomocí vektorové reprezentace byly využity metody pro zjištění podobností přes kosinovou podobnost a shlukování vět pomocí hierarchického shlukování. Další demonstrací je shrnutí textu metodami LSA a TextRank kde bylo zjištěno že tyto metody jsou nevhodné pro shrnutí textu akademických prací, ale excelují pro jednoduché texty jako recenze a komentáře.

Předposlední částí bylo získání témat z textu metodami LSA a LDA, které vrátily nejdůležitější slova a roztřídily je do témat k, kterým byly přiřazeny jednotlivé dokumenty. Jako poslední byla provedena reprezentace slov v textu pomocí modelů fastText a GloVe kde bylo zobrazeno rozprostření slov ve vektorovém prostoru a zobrazena jejich klasifikace, jak slova podobná mají vektory blízko u sebe.

Zadání bylo, dle mého názoru, splněno. Možné změny by mohli být výběr jiných metod pro analýzu textu. A pro zlepšení by mohlo být využití pravé databáze slov pro metodu GloVe a lepší předzpracování.

LITERATURA

[1] Džeroski S. (2001) Data Mining in a Nutshell. In: Džeroski S., Lavrač N. (eds) Relational Data Mining. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-04599-2_1

[2] SOWMYA R AND SUNEETHA K R, "DATA MINING WITH BIG DATA," *2017 11TH INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS AND CONTROL (ISCO)*, 2017, PP. 246-250, DOI: 10.1109/ISCO.2017.7855990.

[3] SRIVASTAVA, JAIDEEP & COOLEY, ROBERT & DESHPANDE, MUKUND & TAN, PANG-NING. (2000). WEB USAGE MINING: DISCOVERY AND APPLICATIONS OF USAGE PATTERNS FROM WEB DATA.. SIGKDD EXPLORATIONS. 1. 12-23.

[4] MAGIC OF TF-IDF [ONLINE]. ANALYTICS VIDHYA, 2020 [CIT. 2021-12-22]. DOSTUPNÉ Z: HTTPS://MEDIUM.COM/ANALYTICS-VIDHYA/MAGIC-OF-TF-IDF-202649D39C2F

[5] TFIDF [ONLINE]. [CIT. 2021-12-22]. DOSTUPNÉ Z: HTTP://WWW.TFIDF.COM/

[6] MIKOLOV, TOMAS. EFFICIENT ESTIMATION OF WORD REPRESENTATIONS IN VECTOR SPACE: ARXIV:1301.3781V3 [CS.CL] [ONLINE]. CORNELL UNIVERSITY, 2013 [CIT. 2021-12-25]. DOSTUPNÉ Z: HTTPS://ARXIV.ORG/PDF/1301.3781.PDF. COMPUTATION AND LANGUAGE. CORNELL UNIVERSITY.

[7] MIKOLOV, TOMAS. DISTRIBUTED REPRESENTATIONS OF WORDS AND PHRASES AND THEIR COMPOSITIONALITY: ARXIV:1310.4546V1 [CS.CL] [ONLINE]. CORNELL UNIVERSITY, 2013 [CIT. 2021-12-25]. DOSTUPNÉ Z: HTTPS://ARXIV.ORG/PDF/1301.3781.PDF. COMPUTATION AND LANGUAGE. CORNELL UNIVERSITY.

[8] KULSHRESTHA, RIA. A BEGINNER'S GUIDE TO LATENT DIRICHLET ALLOCATION(LDA). MEDIUM.COM [ONLINE]. SAN FRANCISCO: KULSHRESTHA, 2019, 2019 [CIT. 2021-12-27]. DOSTUPNÉ Z: HTTPS://TOWARDSDATASCIENCE.COM/LATENT-DIRICHLET-ALLOCATION-LDA-9D1CD064FFA2

[9] MIHALCEA, R. & TARAU, P. (2004), TEXTRANK: BRINGING ORDER INTO TEXTS, IN 'PROCEEDINGS OF EMNLP-04 AND THE 2004 CONFERENCE ON EMPIRICAL METHODS IN NATURAL LANGUAGE PROCESSING'.

[10] AMATI G. (2009) BM25. IN: LIU L., ÖZSU M.T. (EDS) ENCYCLOPEDIA OF DATABASE SYSTEMS. SPRINGER, BOSTON, MA. HTTPS://DOI.ORG/10.1007/978-0-387-39940-9_921

[11] HUTTO, C. AND GILBERT, E. (2014) "VADER: A PARSIMONIOUS RULE-BASED MODEL FOR SENTIMENT ANALYSIS OF SOCIAL MEDIA TEXT", PROCEEDINGS OF THE INTERNATIONAL AAAI CONFERENCE ON WEB AND SOCIAL MEDIA, 8(1), PP. 216-225. AVAILABLE AT: HTTPS://OJS.AAAI.ORG/INDEX.PHP/ICWSM/ARTICLE/VIEW/14550 (ACCESSED: 28DECEMBER2021).

[12] TURNER, BRANDON & STEYVERS, MARK. (2011). A WISDOM OF THE CROWD APPROACH TO FORECASTING.
[13] SRINIVASAN S. (2017) LOCAL AND GLOBAL SPATIAL STATISTICS. IN: SHEKHAR S., XIONG H., ZHOU X. (EDS) ENCYCLOPEDIA OF GIS. SPRINGER, CHAM. HTTPS://DOI.ORG/10.1007/978-3-319-17885-1_700
[14] JEFFREY PENNINGTON, RICHARD SOCHER, AND CHRISTOPHER D. MANNING. 2014. GLOVE: GLOBAL VECTORS FOR WORD REPRESENTATION
[15] REDDY, SANJANA. GLOVE AND FASTTEXT — TWO POPULAR WORD VECTOR MODELS IN NLP. SAP [ONLINE]. WALLDORF: REDDY, 2019 [CIT. 2021-12-28]. DOSTUPNÉ Z: HTTPS://BLOGS.SAP.COM/2019/07/03/GLOVE-AND-FASTTEXT-TWO-POPULAR-WORD-VECTOR-MODELS-IN-NLP/
[16] STRECHT, PEDRO & CRUZ, LUÍS & SOARES, CARLOS & MOREIRA, JOÃO & ABREU, RUI. (2015). A COMPARATIVE STUDY OF CLASSIFICATION AND REGRESSION ALGORITHMS FOR MODELLING STUDENTS' ACADEMIC PERFORMANCE.
[17] Bojanowski, Piotr & Grave, Edouard & Joulin, Armand & Mikolov, Tomas. (2016). Enriching Word Vectors with Subword Information. Transactions of the Association for Computational Linguistics. 5. 10.1162/tacl_a_00051.
[18] Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.
[19] HOLČÍK, JIŘÍ, KOMENDA, MARTIN (EDS.) A KOL. MATEMATICKÁ BIOLOGIE: E-LEARNINGOVÁ UČEBNICE [ONLINE]. 1. VYDÁNÍ. BRNO: MASARYKOVA UNIVERZITA, 2015. ISBN 978-80-210-8095-9.
[20] TEXT ANALYTICS WITH PYTHON: A PRACTITIONER'S GUIDE TO NATURAL LANGUAGE PROCESSING. SECOND EDITION. APRESS, 2019. ISBN 978-1-4842-4353-4.
[21] ARIC A. HAGBERG, DANIEL A. SCHULT AND PIETER J. SWART, "EXPLORING NETWORK STRUCTURE, DYNAMICS, AND FUNCTION USING NETWORKX", IN PROCEEDINGS OF THE 7TH PYTHON IN SCIENCE CONFERENCE (SCIPY2008), GÄEL VAROQUAUX, TRAVIS VAUGHT, AND JARROD MILLMAN (EDS), (PASADENA, CA USA), PP. 11–15, AUG 2008
[22] REHUREK, R. & SOJKA, P., 2011. GENSIM–PYTHON FRAMEWORK FOR VECTOR SPACE MODELLING. NLP CENTRE, FACULTY OF INFORMATICS, MASARYK UNIVERSITY, BRNO, CZECH REPUBLIC, 3(2).
[23] HONNIBAL, M. & MONTANI, I., 2017. SPACY 2: NATURAL LANGUAGE UNDERSTANDING WITH BLOOM EMBEDDINGS, CONVOLUTIONAL NEURAL NETWORKS AND INCREMENTAL PARSING.
[24] BIRD, S., KLEIN, E. & LOPER, E., 2009. NATURAL LANGUAGE PROCESSING WITH PYTHON: ANALYZING TEXT WITH THE NATURAL LANGUAGE TOOLKIT, " O'REILLY MEDIA, INC."
[25] HARRIS, C.R. ET AL., 2020. ARRAY PROGRAMMING WITH NUMPY. NATURE, 585, PP.357–362.

[26] McKinney, W. & others, 2010. Data structures for statistical computing in python. In Proceedings of the 9th Python in Science Conference. pp. 51–56.

[27] Virtanen, P. et al., 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, 17, pp.261–272.

Příloha A - Shrnutí textu

A.1 Shrnutí textu pomocí metody LSA

the paper describe the procedure to verify the validity of the general relationship for calculate the susceptibility from the reaction field for sample of various shape and material

the quantity be define as the ratio between the magnetization m of a material in a magnetic field and the field intensity h all material can be classify into three group by the magnetic susceptibility value

the magnitude of such a deformation depend on the difference between the magnetic susceptibility of the sample s and its vicinity v the volume and shape of the sample and the magnitude of the basic field b

the difference between a change in the magnetic field in the vicinity of the specimen and the value of the static magnetic field b be refer to as the reaction field b

where bv be the reaction field in the vicinity of the specimen vs be the volume of the specimen and b be the static magnetic field

for an irrotational field the above equation assume the form the material relation be then outline by equation the model reaction field b of the sample be show in the cross section plane fig

to calculate the magnetic susceptibility we employ the course of the reaction field in the x axis

the magnetic susceptibility of the model material be then equal to where the sign in front of the fraction depend on the material use

the value bmax and bmin represent the value of b in the space of the sample with bmax denote the field close to the inner side of the boundary of the sample and bmin express the field close to the corresponding outer side

the magnetic susceptibility of the simulate sample be calculate by use the value of the reaction field b from fig

in the table below we summarize the result of the magnetic susceptibility of all the sample

base on model the magnetic field in the vicinity of sample of non ferromagnetic material we consider the formula applicable to all the paramagnetic and diamagnetic material sample

A.2 Shrnutí textu pomocí metody TextRank

the paper describe the procedure to verify the validity of the general relationship for calculate the susceptibility from the reaction field for sample of various shape and material

the quantity be define as the ratio between the magnetization m of a material in a magnetic field and the field intensity h all material can be classify into three group by the magnetic susceptibility value

the magnitude of such a deformation depend on the difference between the magnetic susceptibility of the sample s and its vicinity v the volume and shape of the sample and the magnitude of the basic field b

the difference between a change in the magnetic field in the vicinity of the specimen and the value of the static magnetic field b be refer to as the reaction field b

we have from which it be evident that the magnetic flux density outside the specimen be change result in a shape that can be consider the superposition of the homogeneous field b and the reaction field b

where bv be the reaction field in the vicinity of the specimen vs be the volume of the specimen and b be the static magnetic field

for an irrotational field the above equation assume the form the material relation be then outline by equation the model reaction field b of the sample be show in the cross section plane fig

to calculate the magnetic susceptibility we employ the course of the reaction field in the x axis

the magnetic susceptibility of the model material be then equal to where the sign in front of the fraction depend on the material use

the value bmax and bmin represent the value of b in the space of the sample with bmax denote the field close to the inner side of the boundary of the sample and bmin express the field close to the corresponding outer side

the magnetic susceptibility of the simulate sample be calculate by use the value of the reaction field b from fig

base on model the magnetic field in the vicinity of sample of non ferromagnetic material we consider the formula applicable to all the paramagnetic and diamagnetic material sample

Příloha B - Obsah přiloženého CD

Příloha obsahuje programy, kterými byly dosaženy výsledky, dokumenty, na které bylo textové dolování použito a textový dokument, který obsahuje verze použitých knihoven. Příkazem pip install requirements.txt dojde k nainstalování knihoven.

Obsah přiloženého CD

- Zdrojové kódy
  - preprocesing.py
  - kosinova_podobnost.py
  - FastText.py
  - GloVe.py
  - Text_summary.py
  - Topic_modeling_scikit_learn.py
  - corpus_processing.py
  - zobrazeni.py
- Použité dokumenty
  - Sample_text_1.docx
  - Sample text 2.docx
  - Sample_text_3.docx
  - sensors-19-01592.pdf
  - sensors-21-02004-v2.pdf
  - sensors-21-02388-v2.pdf
  - sensors-21-04244-v2.pdf
  - soubor nipstxt
  - batman_review.docx
  - remotesensing-13-01878-v2.pdf
- requirements.txt