VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Určování podobnosti dokumentů s použitím tradičních výpočetních metod a spolupráce davu

Document Categorization Using Traditional Algorithms and Crowd Sourcing

2018 Bc. Barbora Cigánková

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání diplomové práce

Student: **Bc. Barbora Cigánková**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: Určování podobnosti dokumentů s použitím tradičních výpočetních metod a spolupráce davu
Document Categorization Using Traditional Algorithms and Crowd Sourcing

Jazyk vypracování: čeština

Zásady pro vypracování:

Textové dokumenty si mohou být podobné bez ohledu na formátování, či pořadí odstavců, vět, záměnou slov za jejich synonyma. Použití ontologií a webového inženýrství je dobrým způsobem, jak podobnost dokumentů řešit. Existují algoritmy a výpočetní postupy, jako TF-IDF nebo shlukovací algoritmy, které mají v řadě kategorizačních úloh velmi dobré výsledky. Na jejich použití a na spolupráci davu při hodnocení kvality vyhodnocení podobnosti je založena tato práce.

Cílem je navrhnout a implementovat prototyp řešení, které kategorizuje textové dokumenty na základě jejich podobnosti.
1. Implementujte jeden z algoritmů pro kategorizaci dokumentů.
2. Pomocí spolupráce uživatelů navrhněte hodnocení kvality kategorizace a způsob jejího použití pro další zkvalitnění kategorizace.
3. Realizujte dvě případové studie s použitím volně dostupných dat a nebo dat dodaných vedoucím projektu.
4. Vyhodnoťte úspěšnost určování podobnosti a navrhněte její další zlepšování.

Seznam doporučené odborné literatury:

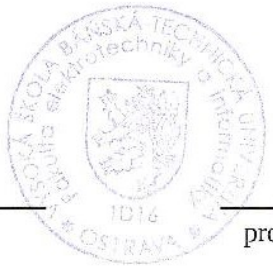
[1] Morbus Iff, Tara Calishain (2003): Spidering Hacks
[2] http://guidetodatamining.com/
[3] Maas, Andrew L., Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. "Learning word vectors for sentiment analysis." Proceedings of the 49th Annual Meeting of ACL: Human Language Technologies-Vol. 1. ACL, 2011. 142-150.
[4] Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig. "Linguistic Regularities in Continuous Space Word Representations." HLT-NAACL. 2013. 746-751.
[5] Miller, George A. "WordNet: a lexical database for English." Communications of the ACM 38.11 (1995): 39-41.
[6] Šajgalík, M., Barla, M., Bieliková, M.: Exploring Multidimensional Continuous Feature Space to Extract Relevant Words. In: Proc. of SLSP 2014, Springer-Verlag, 2014

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **doc. RNDr. Petr Šaloun, Ph.D.**

Datum zadání: 01.09.2017

Datum odevzdání: 30.04.2018



doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry

prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracovala samostatně. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

V Ostravě 30. dubna 2018 .....

Ráda bych poděkovala vedoucímu diplomové práce doc. RNDr. Petru Šalounovi, Ph.D. za odbornou pomoc a konzultaci při vytváření této práce. Zároveň bych chtěla poděkovat všem studentům Ostravské univerzity v Ostravě a neformálním pečujícím, kteří se do této práce zapojili při sběru dat a hodnocení kategorizace.

Abstrakt

Diplomová práce se zabývá kategorizací textových dokumentů a jejím následným zlepšováním pomocí spolupráce davu. Jejím cílem je návrh a vytvoření prototypu klasifikátoru textových dokumentů na základě jejich podobnosti a návrh zhodnocení a následné zlepšování kategorizace s využitím spolupráce davu. Ke kategorizaci dokumentů byl vybrán algoritmus N-gramů, který byl následně implementován v jazyce Java. Dále bylo vytvořeno rozhraní pro spolupráci davu s využitím CMS WordPress. Účelem rozhraní je, kromě sběru dat, také zhodnocení správnosti kategorizace, na základě kterého je následně rozšiřována testovací sada dokumentů klasifikátoru, čímž je úspěšnost kategorizace zvyšována. Obě části práce by měly sloužit jako základ pro chystaný projekt TAČR Éta mezi Ostravskou univerzitou v Ostravě a Vysokou školou báňskou - Technickou univerzitou Ostrava.

Klíčová slova: Kategorizace, textové dokumenty, přirozený jazyk, podobnost dokumentů, N-gramy, spolupráce davu, WordPress, Java, PHP

Abstract

The master thesis deals with categorization of text documents and its improvement through crowdsourcing. Its goal is to design and implement text documents classifier prototype based on documents similarity and to design evaluation and improvements of categorization using crowdsourcing. For categorization the N-grams algorithm has been chosen, which was implemented in Java. Next, interface for crowdsourcing was created using CMS WordPress. In addition to data collection, the purpose of interface is to evaluate categorization accuracy, which leads to extension of classifier's test data set, thus the categorization is more successful. Both parts of the thesis should serve as base for prepared project between University of Ostrava and VŠB - Technical university of Ostrava.

Key Words: Categorization, text documents, natural language, documents similarity, N-grams, crowdsourcing, WordPress, Java, PHP

Obsah

Seznam použitých zkratek a symbolů

API	–	Application Programming Interface
BOW	–	Bag of Words
CMS	–	Systém správy obsahu (Content Management System)
CSS	–	Kaskádové styly (Cascading Style Sheets)
cURL	–	Client Uniform Resource Locator
DAO	–	Data access object
DB	–	Databáze
ECM	–	Správa podnikového obsahu (Enterprise Content Management)
HTML	–	Hypertextový značkovací jazyk (Hypertext Markup Language)
IDF	–	Inverse Document Frequency
IIS	–	Internet Information Services
JSON	–	JavaScript Object Notation
LSA	–	Latentní sémantická analýza
LSI	–	Latentní sémantická indexace
MS SQL	–	Microsoft Structured Query Language
MySQL	–	My Structured Query Language
NLP	–	Zpracování přirozeného jazyka (Natural Language Processing)
PHP	–	Hypertextový preprocesor (Hypertext Preprocessor)
PoC	–	Ověření konceptu (Proof of Concept)
R&D	–	Výzkum a vývoj (Research and Development)
REST	–	Representational state transfer
RSS	–	Rich Site Summary
SVD	–	Singulární rozklad matice (Singular Value Decomposition)
SVM	–	Support Vector Machine
TF-IDF	–	Term Frequency-Inverse Document Frequency
URL	–	Uniform Resource Locator
WCM	–	Správa webového obsahu (Web Content Management)
WP	–	WordPress

Seznam obrázků

Seznam tabulek

Seznam výpisů zdrojového kódu

Úvod

Oblast využití počítačů se stále zvětšuje. V dnešní době je čím dál častější zapojení práce s počítačem do výuky již těch nejmenších. Z těchto důvodů je více než žádoucí komunikaci člověka s počítačem stále ulehčovat. Tento úkol je prvním cílem oboru nazývaného zpracování přirozeného jazyka (zkráceně NLP), přičemž se přirozeným jazykem myslí jazyk, kterým se dorozumívají lidé. NLP je využíván v mnoha oblastech ať už při extrakci informací, korektuře textu či rozpoznávání řeči. Svou roli však také hraje při kategorizaci textu, kde společně s dalšími procesy předzpracovává vstupní text do podoby přijatelné klasifikátorem.

Kategorizace dokumentů umožňuje automatické třídění textů do předem určených kategorií, což eliminuje nutnost manuálního roztřiďování. Využití této úlohy je také velmi široké např. automatické zařazování vědeckých článků, filtrování spamů, třídění dokumentů, ale také odhalení plagiátorství. V posledním zmiňovaném případě se využívá kategorizace za pomocí určování podobnosti dokumentů, která využívá různých metrik pro určení míry podobnosti. Přístupů ke kategorizaci je však spousta od těch běžnějších jako je metoda k-nejbližších sousedů či rozhodovací strom až po ty komplexnější jako jsou například neuronové sítě nebo algoritmus podpůrných vektorů.

I když jsou dnešní algoritmy pro kategorizaci poměrně úspěšné, jako u všech výstupů počítačových algoritmů je i zde vhodná evaluace správnosti výsledků. Vzhledem k tomu, že člověk, i přes poměrně vysokou úspěšnost umělých klasifikátorů, je stále v této oblasti přesnější, je možné využít jako referenční systém právě člověka. Jelikož kolektivní inteligence má větší sílu než schopnosti jednotlivce, nabízí se v tomto ohledu aplikace crowdsourcingu. Crowdsourcing neboli spolupráce davu je metodika při níž je určitá kreativní či inovativní činnost outsourcována na dav. Tato praktika zažívá v posledních letech značný nárůst určitě i díky výhodám, které firmě přináší např. nižší finanční náklady, uvolnění lidských zdrojů pro jiné projekty apod.

V rámci této práce bude vyvíjena aplikace s funkcí kategorizace textových dokumentů v přirozeném jazyce. Výsledky tohoto klasifikátoru budou nejprve ověřeny na dvou datových sadách, následně pak za využití metodiky crowdsourcing. Komunikace s davem bude probíhat pomocí webové aplikace běžící v rámci redakčního systému (zkráceně CMS) WordPress. Tento CMS byl vybrán nejen z důvodu jeho oblíbenosti (viz kapitola 4.1 WordPress). Výsledky práce budou sloužit jako PoC pro připravovaný projekt v rámci programu Éta Technologické agentury ČR.

Textová část této práce je strukturována do šesti částí. První z nich se zabývá kategorizací textových dokumentů. Tato část popisuje teoretické základy klasifikace a stručný popis některých využívaných algoritmů. Druhá kapitola obsahuje téma zpracování dokumentů v přirozeném jazyce, jeho využití a jednotlivé fáze. Blíže se pak zabývá algoritmy pro lemmatizaci a stemmizaci a jejich aplikaci v českém jazyce. Následující kapitola, Spolupráce davu, uvádí definice pojmu, výhody a využití metodiky, motivaci davu k zapojení se do těchto projektů, ale také srovnání

s podobnými koncepty. Ve čtvrté kapitole je popsáno téma redakční systémy se zaměřením na tři v současnosti nejpoužívanější, přičemž WordPress je popsán i z hlediska šablon a pluginů, a to kvůli jeho následnému využití v praktické části práce, kterou se zabývají následující kapitoly. Kapitola Realizace prototypu popisuje návrh, architekturu a implementaci klasifikátoru textových dokumentů a rozhraní pro crowdsourcing. Poslední kapitola práce zahrnuje zhodnocení úspěšnosti klasifikace implementovaného klasifikátoru na dvou datových sadách a následně na datech získaných pomocí spolupráce davu.

1 Kategorizace textových dokumentů

Kategorizace dokumentů je téma spadající především do informačních věd. Úkolem klasifikátorů dokumentů obecně je zařadit dokument do jedné či více kategorií na základě jeho obsahu. V oblasti dolování informací, konkrétně dolování textu (ang. text mining), se však také jedná o proces automatického učení kategorizačních schémat využívaných pro přímou kategorizaci nových, nezařazených dokumentů [1].

Některé přístupy ke kategorizaci dokumentů bývají označovány jako algoritmy pro kategorizaci na základě podobnosti dokumentů. Součástí těchto algoritmů je stanovení metriky, např. kosinové podobnosti, pro výpočet podobnosti dvou dokumentů. Tato metrika je pak používána jak během učení klasifikátoru, tak během samotné kategorizace. Před samotnou kategorizací bývá třeba provést dva kroky. Prvním je úprava textu do podoby, ve které může být automaticky zpracován. Tato fáze zahrnuje např. odstranění tagů, odstranění stop slov a další předzpracování textu (viz kapitola 2 Zpracování dokumentů v přirozeném jazyce). Druhým krokem je převod dokumentu do tvaru čitelného samotným klasifikátorem. Mnohé přístupy využívají extrakci vlastností daného textu, pro které je následně vypočítávána jejich váha. Tyto vlastnosti jsou poté reprezentovány jako vektory charakterizující přítomnost slov či syntaktických celků [1].

Mnoho klasifikátorů využívá k reprezentaci textu přístup označovaný jako bag-of-words model (zkráceně BOW) [1]. Jedná se o zjednodušenou reprezentaci textu, využívanou především při zpracovávání přirozeného jazyka a při získávání informací, při které je text dokumentu převeden na sadu jednotlivých slov, a to bez ohledu na gramatiku a pořadí slov, avšak se zachování jejich možné duplicity. Při kategorizaci se pro slova v dané vytvořené sadě počítá frekvence výskytu, které mohou být následně použity jako vstupy pro trénovaní klasifikátoru.

Po potřebné transformaci textu do podoby akceptovatelné klasifikátorem je dalším krokem učení klasifikátoru, které probíhá na základě trénovací sady, která obsahuje dokumenty, s již označenou cílovou třídou, a následně samotná kategorizace dokumentů. Dnešní klasifikátory využívají buď statistických metod nebo metod strojového učení. Kategorizace dokumentů může být rozdělena do dvou kategorií [1]:

- kategorizace s dohledem (ang. supervised document classification),
- kategorizace bez dohledu (ang. unsupervised document classification).

Dohled u prvního typu kategorizace může být zastoupen lidskou zpětnou vazbou, kdežto druhý typ kategorizace probíhá bez přístupu k vnějším informacím.

Mezi hlavní algoritmy, které jsou využívány pro kategorizaci dokumentů patří např. rozhodovací stromy, n-gramy, neuronové sítě, Bayesův klasifikátor ad. [1].

1.1 Naivní Bayesův klasifikátor

Naivní Bayesův klasifikátor patří mezi pravděpodobnostní klasifikátory založené na Bayesově teorému. Ten je formulován následovně [2]:

Nechť je dán úplný systém vzájemně neslučitelných jevů $Y_1, Y_2, ..., Y_n$ a libovolný jev X, který může nastat jen současně s některým z jevů Y_i. Pak pravděpodobnost, že nastane jev Y_i, za předpokladu, že nastal jev X je

$$P(Y_i|X) = \frac{P(X|Y_i) \cdot P(Y_i)}{P(X)}, \tag{1.1}$$

kde

$$P(X) = \sum_{k=1}^{n} P(Y_k) \cdot P(X|Y_k). \tag{1.2}$$

Tato věta udává, jak podmíněná pravděpodobnost nějakého jevu souvisí s opačnou podmíněnou pravděpodobností.

Předpokladem naivního Bayesovského klasifikátoru je myšlenka, že (ne)přítomnost jedné vlastnosti dané třídy není nijak závislá na (ne)přítomnosti vlastnosti jiné [1]. Jinak řečeno předpokládá, že jednotlivé atributy jsou na sobě nezávislé, a tedy že efekt, který má hodnota atributu na danou třídu, není ovlivnitelný hodnotami ostatních atributů. Tento předpoklad nám umožňuje upravit podmíněnou pravděpodobnost $P(X|Y)$ do podoby

$$P(X|Y_i) = \prod_{k=1}^{n} P(X_k|Y_i), \tag{1.3}$$

kde X_k označuje k-tý atribut prvku X a Y_i označuje i-tou třídu.

Při klasifikaci, pak použijeme upravenou Bayesovu větu

$$P(Y_i|X) = \frac{\prod_{k=1}^{n} P(X_k|Y_i) \cdot P(Y_i)}{P(X)}, \tag{1.4}$$

kterou vypočteme pro každou třídu, do které může být prvek zařazen. Výsledná třída pak bude ta s nejvyšší podmíněnou pravděpodobností.

Výhodou naivního Bayesovského klasifikátoru je potřeba malého množství trénovacích dat k určení parametrů, např. průměru a směrodatné odchylky hodnot, potřebných ke klasifikaci [1].

1.2 TF-IDF

Pro ohodnocení jednotlivých vlastností textu (ang. term-weighting) se často využívá například metodika Term Frequency-Inverse Document Frequency (zkráceně TF-IDF). Jedná se o statistické měřítko určující důležitost slov v daném dokumentu [3].

První část názvu metodiky, tedy TF – Term Frequency, označuje počet výskytů daného slova v dokumentu zkrácený o celkový počet slov v dokumentu. Tato normalizace se provádění z důvodu minimalizace „zvýhodnění“ dlouhých dokumentů, které mohou mít, vzhledem ke své délce, frekvenci výskytu určitého výrazu výrazně vyšší než dokumenty kratší. Druhá část, reprezentující důležitost výrazů, tedy IDF (ang. Inverse Document Frequency), je charakterizována jako logaritmus celkového počtu dokumentů zkrácený o počet dokumentů, ve kterém se daný výraz nachází. Zkráceně lze výpočet obou složek znázornit takto [3]:

- TF(t) = (počet výskytů výrazu t v dokumentu) / (celkový počet výrazů v dokumentu),
- IDF(t) = log (celkový počet dokumentů / počet dokumentů obsahujících výraz t).

Vysokou váhu budou mít tedy dokumenty s vysokou frekvencí výskytu daného výrazu v dokumentu a nízkým celkovým počtem výskytu výrazu v sadě ostatních dokumentů. Díky tomu se TF-IDF může využít k detekci tzv. stop slov. Stop slova jsou nejčastěji se vyskytující se slova v textu, například předložky a spojky, které však nehrají žádnou rolí při samotné analýze textu. Mezi tato slova můžeme zařadit slova jako a, nebo, pro, za apod.

Jednou z hlavních nevýhod TF-IDF je ignorování klíčových sémantických spojení mezi slovy a slovními významy a srovnávání dokumentů pouze na základě frekvence výskytů slov.

Různé varianty TF-IDF jsou často využívány ve vyhledávačích jako hlavní nástroj pro skórování a ohodnocení relevance dokumentu, který je odpovědí na uživatelem vyhledávaný výraz [1].

1.3 Latentní sémantická analýza

Latentní sémantická analýza (zkráceně LSA), také známá jako latentní sémantická indexace (zkráceně LSI), je technika využívaná při zpracování přirozeného jazyka. Jedná se o analýzu vztahů mezi sadou dokumentů a výrazy v nich obsaženými. Na rozdíl od klasického zpracování přirozeného jazyka či programů využívajících prvků umělé inteligence, nevyužívá LSI žádný člověkem vytvořený slovník, znalostní bázi, gramatiku či syntaktický parser. Vstupem LSI je pouhý text rozdělený do smysluplných částí jako jsou věty nebo odstavce [4].

LSI využívá matematický postup zvaný jako singulární rozklad matic (ang. Singular Value Decomposition, zkráceně SVD). Jedná se o metodu numerické lineární algebry, při které dojde k rozložení čtvercové matice na tři další, přičemž platí, že výsledkem maticového násobení vzniknuvších matic je matice původní.

SVD rozklad reálné $m \times n$ matice A se značí [5]

$$A = USV^T, \tag{1.5}$$

kde S je nezáporná diagonální matice $m \times n$ obsahující na své diagonále tzv. singulární čísla. Matice U, resp. V^T, označují $m \times m$, resp. $n \times n$, ortogonální matice, jejichž sloupce jsou označovány jako pravé, resp. levé singulární, vektory.

LSA je postavena na principu, který říká, že slova, která jsou obsažená v jednom kontextu mívají stejný či podobný význam [1]. LSA tedy předpokládá, že slova, která jsou si významově podobná, se budou nacházet v podobných částech textu.

1.3.1 Postup

Postup LSI je popsán v [4] a [5]. První krok představuje převod textu do podoby matice. V matici představují řádky unikátní slova textu a sloupce jeho jednotlivé části. Každá buňka matice pak obsahuje počet výskytů daného slova v dané části textu.

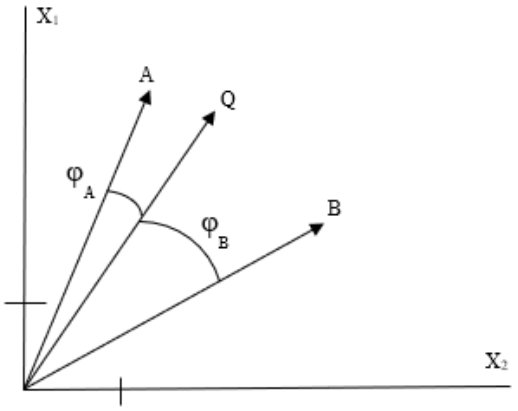
Druhým krokem je úprava hodnot matice obsahující dvě části. První částí je zlogaritmování hodnot všech buněk matice, tedy frekvencí jednotlivých slov. Druhou částí je vydělení hodnoty každé buňky vypočtenou entropií daného slova (řádku). Ta se vypočte jako součet výrazu $p \cdot log(p)$ pro všechny buňky daného řádku. Účinkem této transformace je určení váhy každého výskytu slova přímo, odhadem jeho důležitosti v dané části textu, a nepřímo, pomocí míry, do jaké víme, že výskyty slova podávají informaci o tom, v jaké části textu se slovo objevilo.

Hlavní část LSA se skládá ze dvou kroků. Prvním je aplikace SVD rozkladu matice, při kterém dojde k odstranění šumu a přebytečných informací z dat. Jelikož může výpočet vlastních čísel blízkých nule a jejich vlastních vektorů zhoršovat kvalitu vyhledávání, je v reálných výpočtech uvažováno pouze k největších singulárních čísel a jim příslušejících singulárních vektorů [5]. Vznikne tedy k-tá aproximace (A_k) matice A, která je sestavena vybráním pouze k-prvních singulárních čísel matice S, zatímco zbývající jsou zanedbány. Při aproximacích maticemi A_k nízkých hodností se do sloupcových vektorů promítají hodnoty z jiných sloupců dle vazeb mezi nimi [5]. Právě tato vlastnost umožňuje zachytit skryté vazby mezi dokumenty a termy.

Druhým krokem je výpočet koeficientu podobnosti mezi transformovanými daty. Ten je možný provést mnoha způsoby. Jedním z nejpoužívanějších metod je výpočet tzv. kosinovy podobnosti, která je vyjádřena jako [5]

$$cos_j^{\varphi} = \frac{(q, D_j)}{\sqrt{(q,q)} \cdot \sqrt{(D_j, D_j)}}, \tag{1.6}$$

kde q, resp. D_j, označují transformovaný dokument, a $1 \leq j \leq n$. Grafické znázornění kosinovy podobnosti ve dvoudimenzionálním prostoru znázorňuje obrázek 1 kosinová podobnost ve 2D prostoru. Symboly A, B, Q reprezentují dokumenty. Symboly φ_A a φ_B označují úhly mezi A, Q, resp. mezi B, Q. Dokument A je více podobný dokumentu Q než dokumentu B, neboť $\varphi_A < \varphi_B$. Malý úhel mezi dokumenty tedy znázorňuje jejich podobnost.



Obrázek 1: Kosinová podobnost ve 2D prostoru [5]

1.4 Algoritmus podpůrných vektorů

Algoritmus podpůrných vektorů (ang. Support Vector Machine, zkráceně SVM) je metoda strojového učení s učitelem využívaná při binární kategorizaci a regresní analýze. Je založena na konceptu rozhodovacích rovin[1], které definují hranice rozhodnutí [6].

SVM je metoda, která ke kategorizaci využívá nadroviny[2] (ang. hyperplanes) v multidimenzionálním prostoru, které oddělují objekty jednotlivých tříd.

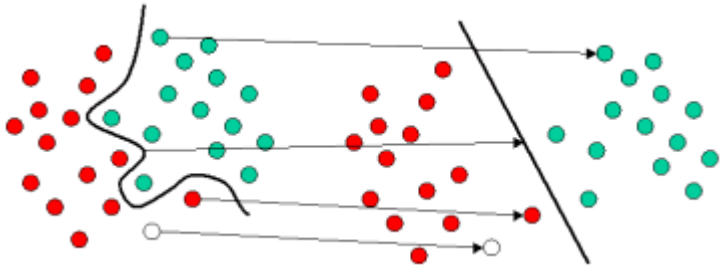
Hlavní myšlenkou SVM je pomocí transformace objektů, prováděné sadou matematických funkcí, známých jako jádrové funkce (ang. kernels funtions), umožnit lineární rozdělení objektů různých tříd [6]. Základem je tedy najít optimální nadrovinu, což je nadrovina s co nejširším maximálním odstupem (ang. maximal margin). Jinak řečeno se jedná o nadrovinu, u níž platí, že vzdálenost mezi nebližšími body od roviny je co největší. Pro popis nadroviny přitom stačí pouze body ležící na okraji maximálního odstupu. Tyto body se nazývají podpůrné vektory (ang. support vector) [7]. Ostatní body nejsou pro nadrovinu zapotřebí. Metoda SVM je tedy schopna najít ty trénovací příklady, které jsou pro nalezení nadroviny podstatné. Velikost trénovací množiny potřebné pro naučení klasifikátoru je tedy mnohem menší než trénovací množina původní, což je výraznou výhodou této metody [7]. Graficky je úloha transformace objektů znázorněná na obrázku obrázek 2.

Pro zkonstruování optimální nadroviny používá SVM iterativní trénovací algoritmus, který slouží k minimalizaci chybové funkce, podle které lze SVM rozdělit do čtyř skupin [6]:

- SVM pro klasifikaci typu 1 (také znám jako C-SVM klasifikace),
- SVM pro kategorizaci typu 2 (také znám jako nu-SVM klasifikace),
- SVM pro regresní analýzu typ 1 (také znám jako epsilon-SVM regrese),
- SVM pro regresní analýzu typ 2 (také znám jako nu-SVM regrese).

[1] Rozhodovací rovina odděluje objekty zařazené do různých tříd.

[2] Nadrovinou v afinním bodovém prostoru An rozumíme jeho podprostor dimenze n – 1 [27].



Obrázek 2: Původní objekty (levá strana) transformovány pomocí matematických funkcí, zvaných jádra, a rozděleny optimální nadrovinou (pravá strana) [6]

SVM umožňuje práci jak se spojitými, tak s kategoriálními proměnnými. Pro kategoriální proměnné je však vytvořena fiktivní proměnná s hodnotami 0 nebo 1. Kategoriální proměnná s hodnotami A, B, C je tedy pomocí fiktivní proměnné reprezentována jako A: {1 0 0}, B: {0 1 0}, C: {0 0 1} [6].

1.5 N-gramy

N-gram je definován jako sled N po sobě jdoucích položek z dané posloupnosti například slov či písmen. Sled dvou po sobě jdoucích položek bývá označován jako bigram, sled tří položek jako trigram. Od čtyř výše se používá označení N-gram, kde N je nahrazeno počtem za sebou jdoucích elementů.

N-gramy se nejčastěji využívají pro reprezentaci textu, kde se jedná právě o sled slov. Druhým možný využitím je kategorizace dokumentů založená na jejich podobnosti. Při kategorizaci dokumentů se naopak častěji využívá sledu písmen, přičemž začátek a konec slova bývá označen speciálním znakem, jako například podtržítkem [8]. Příkladem může být slovo TEXT, ze kterého vzniknou N-gramy [8]:

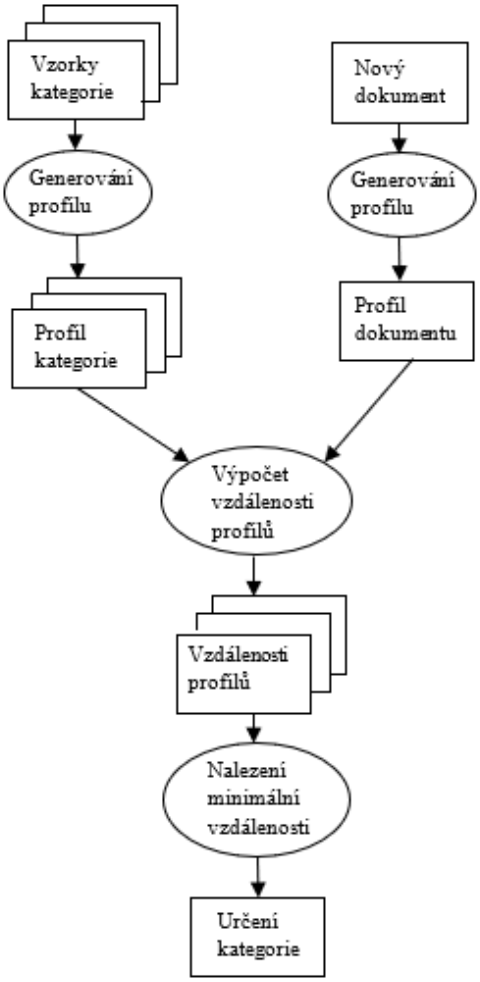
- bigram: _T, TE, EX, XT, T_
- trigram: _TE, TEX, EXT, XT_, T___
- 4gram: _TEX, TEXT, EXT_, XT___, T____

Obecně platí, že každá skupina N-gramů (bigramů, trigramů ad.) pro řetězec o délce k, bude obsahovat $k+1$ N-gramů. Značná výhoda kategorizace dokumentů pomocí N-gramů je její nezávislost na jazyku dokumentů, jelikož zde nemusí neprobíhat žádné zpracování textu typu stemmizace či lemmatizace. Další výhodou je určitá tolerance k pravopisným chybám a překlepům.

Nevýhodou je však velké množství generovaných N-gramů. Tato nevýhoda však může být redukována např. odstraněním stop slov či výše zmiňovanou stemmizací či lemmatizací či jinou redukcí délky textu, čímž však přicházíme o první výhodu algoritmu, tedy nezávislosti na jazyku dokumentu.

1.5.1 Postup

Obecný postup kategorizace dokumentů pomocí N-gramů je vyjádřen obrázkem obrázek 3. Prvním krokem této metody kategorizace je určení profilů obsahujících frekvence výskytů N-gramů pro každý dokument z trénovací sady [8]. Z těchto profilů jsou následně vytvořeny profily charakteristické pro každou kategorii zvlášť. Při kategorizaci nového dokumentu je nejprve stanoven profil frekvencí N-gramů daného dokumentu. Poté je, pomocí výpočtu vzdálenosti, tento profil srovnán s profily všech kategorií. Dokument je následně zařazen do kategorie s jejíž profilem má nejmenší vzdálenost.



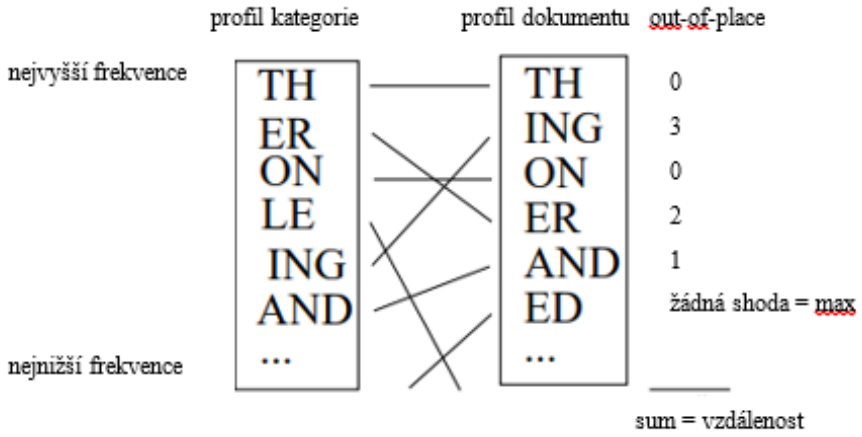
Obrázek 3: Postup kategorizace dokumentů pomocí N-gramů [8]

Aplikace jednotlivých fází celého postupu se může případ od případu více či méně lišit. V této práci budou popsány jednotlivé kroky tak, jak jsou uvedeny v [8].

Prvním krokem je vytvoření profilu dokumentu. Ten obsahuje několik jednoduchých částí. První je tokenizace textu, kde jednotlivé tokeny představují jednotlivá slova, přičemž číslice a interpunkce jsou z tokenizace vynechány. Následně jsou označeny začátky a konce tokenů

předem daným speciálním znakem. Poté jsou z tokenů postupně vytvářeny N-gramy o více či jednom N, jejichž frekvence jsou ukládány např. do hashovací tabulky. Po výpočtu všech frekvencí pro daný dokument je vytvořen sestupný seznam vytvořených N-gramů dle jejich počtu výskytu. Tento seznam tvoří profil daného dokumentu. Pro redukci počtu N-gramů, a tudíž délky profilů, využil Cavnar a spol. [8] teorii Zipfova zákona. Ten říká, že pokud četnost výskytů slov v textu seřadíme od nejvyšší po nejmenší mají tyto četnosti tvar pravidelné klesající křivky. V článku [8] také podotýkají, tato křivka nám naznačuje, že kvalita kategorizace na základě N-gramů nebude příliš citlivá na délce použitých profilů. Autoři zároveň dodávají, že během provedených experimentů měly nejlepší úspěšnost profily s délkou 400 začínající kolem 300. pozice. Odebrání prvních přibližně 300 slov odůvodňují tím, že na těchto pozicích se nacházejí slova, která jsou v daném jazyce nejvíce používaná [8].

Druhý krok, výpočet vzdáleností profilu kategorizovaného dokumentu od profilů jednotlivých kategorií, probíhá, dle [8], pomocí jednoduché statistické metody označované jako „out-of-place“. Tato metoda určuje rozdíl mezi místem N-gramu v jednom profilu oproti místu N-gramu v druhém profilu. Pro každý N-gram v profilu dokumentu je tedy nalezena pozice daného N-gramu v profilu kategorie a spočítán počet míst o které se liší. Pokud se však N-gram v profilu kategorie nevyskytuje, hodnota „out-of-place“ je stanovena na maximum. Následný součet všech vypočítaných hodnot stanovuje vzdálenost mezi danými profily, a tedy danými dokumenty.



Obrázek 4: Příklad metody "out-of-place"pro výpočet vzdálenosti profilů dvou dokumentů [8]

Dokument je následně zařazen do kategorie, jejíž profil je nejméně vzdálen, tedy do té kategorie, jejíž dokumenty jsou nejpodobnější.

2 Zpracování dokumentů v přirozeném jazyce

V dnešní době je zpracování přirozeného jazyka (ang. Natural Language Processing, zkráceně NLP) častým tématem, a to především díky snaze o zjednodušení komunikace člověka s počítačem. Primárním cílem tohoto oboru je umožnění takovéto komunikace v přirozeném jazyce, tedy v jazyce, kterým se lidé dorozumívají mezi sebou. Hlavním rozdílem mezi jazykem přirozeným a umělým je fakt, že přirozený jazyk se vyvíjí v průběhu času a mívá více či méně odchylek, výjimek a nejednoznačností [9]. Oproti tomu, umělý jazyk má jasně stanovená, dopředu vymyšlená, pravidla, která se mění jen velmi zřídka. Právě díky výše zmíněným výjimkám, představuje zpracování přirozeného jazyka, tedy jeho převedení do formy zpracovatelné počítačem, určitou výzvu.

Možné využití NLP je široké. Mezi obory zaměření a aplikace využívající NLP patří např. rozpoznávání řeči, strojový překlad, extrakce informací ať už z textu či řeči, jazykové modelování pro rozpoznávání a syntézu řeči a mnohé další [9].

Software založený na zpracování přirozeného jazyka vyžaduje konzistentní znalostní bázi, jako je například podrobný slovník, jazyková a gramatická pravidla, ontologie a synonyma atd. [10]. Proces NLP obsahuje několik fází, během nichž jsou využívány různé metodiky např. k rozluštění nejednoznačností přirozeného jazyka pomocí taggování částí řeči, k extrakci vztahů, stejně jako porozumění a rozpoznávání přirozeného jazyka [10]. Mezi tyto fáze patří různé typy počítačové analýzy přirozeného jazyka [9]:

- morfologická analýza,
- syntaktická analýza,
- sémantická analýza.

Morfologická analýza se zabývá slovem, jakožto nejmenší smysluplnou jednotkou. Pomocí elektronického slovníku přiřazuje slovu jeho základní tvar, slovní druh a další morfologické kategorie. Oproti tomu syntaktická analýza se zabývá větnými celky a formálním popisem jejich struktury[3]. Poslední zmíněná, sémantická, analýza má za úkol zachytit význam výrazu či většího celku textu. Morfologická analýza je z těchto tří uvedených analýz nejvíce prozkoumaná a nejlépe algoritmizovatelná [9]. Na druhou stranu, sémantická analýza je proveditelná nejhůře a to především, díky homonymii slov.

Téměř před každou prací s textem v přirozeném jazyce je za potřebí provést několik kroků v rámci předzpracování samotného textu. Kromě jeho převedení na malá písmena a odebrání speciálních znaků, se jedná o odstranění stop slov či o tokenizaci.

Jak již bylo zmíněno výše, stop slova jsou slova, která sama o sobě nenesou žádný význam (např. předložky, spojky apod.). Při odstranění stop slov nejde o nic jiného než o nalezení takovýchto slov v celém textu, podle předem stanového seznamu, a jejich následné odstranění z dále

[3]Syntaktická analýza češtiny skýtá určité problémy, a to zejména z důvodu volného slovosledu jazyka.

zpracovávaného textu. Takto upravený text je následně podroben tokenizací, při které dochází k rozdělení vstupního textu na slova a čísla. Jednotlivé celky jsou pak souhrnně označovány jako tokeny.

V rámci zpracování textu jsou také často využívány stemmovací a lemmatizační algoritmy. Stemming i lemmatizace jsou metody pro převod různých forem slova (skloňované, časované atd.) na slovo se společnou základnou či na kořen daného slova. Liší se však používanými přístupy. Tyto procesy jsou velice užitečné například při počítání frekvence slov či frází v textu, jelikož díky nim nemusíme brát zřetel na možné tvary daného slova vznikající časováním, skloňováním či přidáním přípon nebo předpon.

2.1 Stemmery a lemmatizéry

Stemmery a lemmatizéry mají, v dnešní době, velké uplatnění v oblasti webových prohlížečů, například pro full textové vyhledávání, ale také právě v oblasti zpracování přirozeného jazyka. Jak již bylo řečeno, cílem obou algoritmů je převod slov na jejich základní tvar či kořen.

Stemmery pracují se samostatnými slovy nezávisle na kontextu a díky tomu nemohou rozlišovat mezi různými významy slov. Jejich základním principem je pouhé „useknutí“ předpon a přípon slova, čímž vznikne tzv. stem. Existuje několik možností, jak stemming provést. Tím nejjednodušším způsobem je Brute force algoritmus. Ten pracuje na základě převodní tabulky mezi jednotlivými tvary a jejich základním tvarem [11]. Tento způsob je sice rychlý a poměrně jednoduchý, ale takováto databáze bývá, v závislosti na jazyku, poměrně rozsáhlá. Navíc tímto způsobem není možné stemmovat nová slova, která v databázi zanesena nejsou. Druhým způsobem je tzv. affix-stripping[4] algoritmus, který nevyužívá převodní tabulku, ale seznam pravidel pro odstranění předpon a přípon, která mohou vypadat například takto: pokud slovo končí na -ovy, -ovy odstraň [11]. Některé verze affix-stripping algoritmu využívají, pro zlepšení výsledků, porovnávání vytvořeného stemu s databází všech morfologických základů slov. Pokud vytvořený stem v databázi nalezen není, je použit alternativní přístup např. ve formě aplikace jiného pravidla. Jiný přístup např. přiřazuje daným pravidlům prioritu, což řeší situaci, kdy je možné využít více než jedno pravidlo. Affix-stripping algoritmus je také využívám ve formě sufix-stripping, tedy odstraňování pouze přípon. Jedním z nejznámějších algoritmů tohoto typu je Porterův stemmer pro anglický jazyk [11]. Nevýhody affix-stripping algoritmu jsou ve výjimkách a nepravidelných slovesech či ve faktu, že ne všechny slovní druhy (či celé jazyky) mají jasně definované předpony a přípony. Časté je také spojení více přístupů najednou. Jednou z možností je například vytvoření převodní tabulky pouze pro výjimky a nepravidelná slovesa a v případě, že analyzované slovo v této tabulce není, použije se následně affix-stripping algoritmus.

Oproti tomu, lemmatizace využívá morfologickou analýzu slov. Lemmatizace pracuje s detailní databází pravidel, která jednoznačně definují dvojice jednotlivých forem slov a jejich zá-

[4] Afix je hromadné označení pro předponu i příponu.

kladních tvarů (tzv. lemmat) a to za předpokladu, že daná forma slova splňuje určitá morfologická kritéria např. slovní druh, čas nebo číslo [12].

2.1.1 Stemmery a lemmatizéry pro český jazyk

Český jazyk je jeden z jazyků obtížnějších pro stemming i lemmatizaci. Čeština využívá poměrně velké množství předpon a má komplikovanější skloňování než většina jazyků. Díky tomu neexistuje mnoho kvalitních frameworků či softwarových knihoven věnujících se tomuto tématu.

Jedno možné řešení nabízí knihovna Lucene. Tento vyhledávací engine od Apache nabízí analyzátor pro český jazyk, který obsahuje jednak sadu českých stop slov, jednak český stemmer, ale také nabízí možnost tokenizace textu, jejíž průběh můžeme ovlivnit přidáním filtrů například pro transformaci textu na malá písmena či pro odstranění již zmíněných stop slov. Nevýhodou této knihovny je absence českého lemmatizátoru. Tento nedostatek můžeme kompenzovat například českým morfologickým analyzátorem, vyvinutým v rámci brněnské Masarykovy univerzity, s názvem Majka. Lemmatizátor Majka je vylepšenou verzí svého předchůdce s názvem Ajka. Majka je kompletně založená na konečných automatech, proto je také rychlejší a flexibilnější než jeho předchůdce. V základní podobě systém k zadanému slovnímu tvaru přiřadí [13]:

- základní tvar a gramatickou značku,
- všechna slova patřící ke stejnému lemmatu,
- všechna možná slova s diakritikou.

3 Spolupráce davu

Koncept spolupráce davu bývá definován jako obchodní praktika, při které dochází k outsourcování dané aktivity davem [14].

Tento výraz je častěji znám pod anglickým pojmem crowdsourcing. Tento výraz vznikl složeninou dvou slov – crowd, tedy dav, a sourcing, tedy dolování či získávání. Díky této složenině je výraz také někdy překládán jako čerpání z davu či využití davu.

Právě anglická verze tohoto výrazu, tedy crowdsourcing, byla prvně použita Jeffem Howem a Markem Robinsonem, v časopise Wired Magazine v roce 2005. Ti jej využili pro popis činnosti označované jako outsourcování davem. V roce 2006 pak Howe uveřejnil, jako první, definici pojmu, která zněla [15]:

„Crowdsourcing represents the act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call.“

Podle Howa představuje tedy crowdsourcing převedení určité činnosti z firemních zaměstnanců na blíže nedefinovanou skupinu lidí. Oproti čistému outsourcingu se však nejedná o jinou firmu či organizaci, ale o velkou skupinu lidí (dav) oslovenou všeobecnou výzvou. Tato výzva je nejčastěji šířena přes Internet. Rozdíl mezi outsourcingem a crowdsourcingem názorně zobrazuje obrázek 5.

Nejdůležitější částí definice je zmínka právě o všeobecné výzvě. Úkol tedy není outsourcován na předem vybranou skupinu expertů. Do práce může být zapojen kdokoliv. Jediná selekce, která bývá prováděna je selekce posteriorní, tedy výběr dosažených výsledků. Mnohdy, však k výběru nedochází vůbec a výsledky jsou pouze sdružovány.

Crowdsourcing je založen na myšlence kolektivní inteligence. Tento koncept vystihuje věta „Všichni společně jsem chytřejší než každý zvlášť.“ [16]. Kolektivní inteligence je také známá jako moudrost davu (ang. wisdom of crowd). Z tohoto hlediska lze crowdsourcing také definovat jako nástroj na shromáždění kolektivní inteligence pro určitý úkol [16].

Jelikož se během několika let od článku Jeffa Howa objevilo mnoho různých definic, provedl Enrique Estellés-Arolas a Fernando González Ladrón-de-Guevara, výzkumní pracovníci Technické univerzity ve Valencii, analýzu 40 definic dostupných v literatuře, aby vytvořili jednu ucelenou globální definici [14]:

„Crowdsourcing is a type of participative online activity in which an individual, an institution, a nonprofit organization, or company proposes to a group of individuals of varying knowledge, heterogeneity, and number, via a flexible open call, the voluntary undertaking of a task. The undertaking of the task; of variable complexity and modularity, and; in which the crowd should participate, bringing their work, money, knowledge and/or experience, always entails mutual benefit. The user will receive the satisfaction of a given type of need, be it economic, social

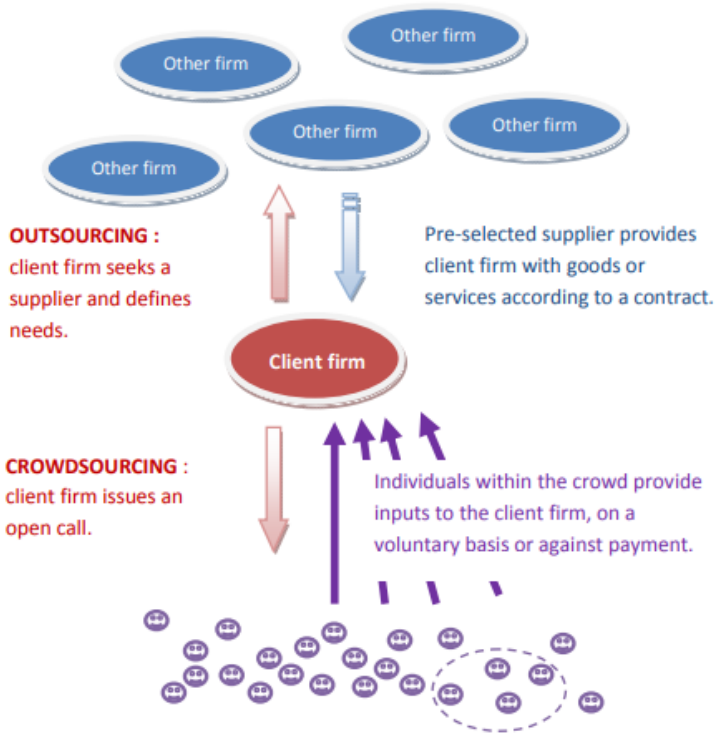
recognition, self-esteem, or the development of individual skills, while the crowdsourcer will obtain and use to their advantage that which the user has brought to the venture, whose form will depend on the type of activity undertaken. "

Výše zmíněná definice obsahuje odpovědi na 8 otázek týkajících se tří prvků crowdsourcingu. Tyto otázky určili Estellés-Arolas a Fernando Ladrón-de-Guevara na základě analyzovaných definic [14]:

- O davu:
1. Kdo ho tvoří.
2. Co dělají.
3. Co dostanou na oplátku.

- O iniciátorovi
1. Kdo to je.
2. Co dostanou za práci davu.

- O procesu:
1. Typ procesu.
2. Typ použité výzvy.
3. Použité médium.



Obrázek 5: Outsourcing vs. Crowdsourcing [15]

3.1 Spolupráce davu, otevřené a uživatelské inovace a open-source

Kromě outsourcingu bývá spolupráce davu spojována také s koncepty otevřené inovace (ang. Open Innovation), uživatelské inovace (ang. User Innovation) a open-source [15] [16]. Mezi těmito čtyřmi pojmy je poměrně tenká hranice, avšak pár rozdílů lze charakterizovat.

První zmiňovaný koncept, koncept otevřené inovace, nabádá organizace k porušení tradičních postupů týkajících se vývoje inovací. V tradičním pojetí jsou firmy zvyklé být uzavřené před vnějším světem ve smyslu nevyužívání či úplného odmítání externích znalostí a zdrojů

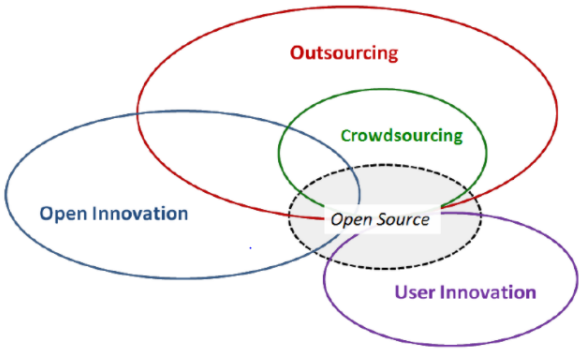
[16]. Popisovaný koncept naopak doporučuje výzkum a vývoj (ang. Research and Development, zkráceně R&D) outsourcovat z jiných organizací [16].

I když je základem crowdsourcingu i otevřených inovací stejná myšlenka – znalosti jsou distribuovány a otevření R&D procesů vede ke konkurenční výhodě – lze mezi nimi najít i jasně definované rozdíly [15]. Prvním rozdílem, který je částečně patrný již z názvů konceptů je v účastnících procesu. V konceptu otevřené inovace se jedná o vztah mezi organizací a organizací, kdežto v crowdsourcingu jde o vztah firmy a davu. Druhý rozdíl je společný i pro další pojem, uživatelské inovace. Jak již z názvů plyne, tyto koncepty jsou, oproti crowdsourcingu, zaměřené pouze na proces inovací.

Uživatelská inovace se, stejně jako otevřená inovace, odklání od tradičního pojetí procesů R&D. Při tradičních postupech pochází inovace pouze od firmy (ang. manufacturer centred innovation), kdežto v pojetí uživatelských inovací pocházejí inovace právě od uživatelů (ang. user centred innovation) [15]. Jelikož se oba tyto koncepty týkají práce lidí mimo organizaci, jejich společným problémem je motivace lidí. Dalším společným prvkem je využití internetu a dalších informačních a komunikačních technologií. Na druhou stranu, jedním z hlavních rozdílů mezi spoluprací davu a uživatelskými inovacemi je způsob řízení projektu – řízení organizací oproti řízení uživateli. Druhým rozdílem jsou pracovníci. U uživatelských inovací jsou projekty vyvíjeny koncovými uživateli daného projektu. U crowdsourcingu se do procesu vývoje může zapojit kdokoliv. Rozdíl v zaměření na inovace, v případě uživatelských inovací, je již zmíněn výše.

Posledním pojmem je Open-Source. Tento koncept je nejčastěji spojován s vývojem softwaru ve smyslu volně přístupných zdrojových kódů otevřených ke změnám a sdílení. Howe [15] označil crowdsourcing jako rozšíření open-source principů do dalších průmyslových odvětví, avšak spolupráce davu není otevřená ve stejném smyslu jako open-source. I když je vývoj otevřený pro všechny, výsledky crowdsourcingu bývají často organizací patentovány a jejich volné sdílení či úprava nejsou umožněny [15].

Schenk společně s Guittardem [15] znázornili vztah mezi těmito čtyřmi koncepty následujícím schématem.



Obrázek 6: Vztah spolupráce davu, otevřených a uživatelský inovací, open-source a outsourcingu [15]

Aitamurto, Leiponen a Tee [16] toto schéma následně upravili. V jejich verzi tvoří crowdsourcing podmnožinu uživatelských inovací, které dále tvoří podmnožinu inovací otevřených. Open-source pak, stejně jako ve znázorněném schématu, kombinuje zbylé tři koncepty.

3.2 Výhody

Výhody čerpání z davu pro firmu jsou zřejmé – uvolnění svých zaměstnanců pro jinou práci, a především nižší náklady. Následující schéma zobrazuje výhody crowdsourcingu oproti ostatním možnostem dodání služeb. Vrchol pyramidy přitom představuje vlastní kapacity firmy, střed je zastoupen outsourcingem (viz výše) a nejspodnější vrstva představuje crowdsourcing [17]. Z tohoto schématu vyplývá, že čím výše v pyramidě se pohybujeme tím vyšší jsou pro nás náklady, tím vyšší je časová náročnost a tím menší různorodost řešení máme k dispozici [17]. Jinými slovy řečeno, čím více využíváme vlastní zdroje tím více zaplatíme, tím déle to bude trvat a tím menší variabilitu řešení budeme mít.

Obrázek 7: Porovnání crowdsourcingu, outsourcingu a insourcingu [17]

3.3 Motivace davu

Značný problém crowdsourcingu může představovat motivace potenciálních pracovníků k zapojení se do práce na projektu. V některých případech stačí pouze možnost zdokonalit si své schopnosti, přispět společnosti k vývoji něčeho důležitého nebo zvýšit si reputaci v rámci dané komunity. Mnohdy však bývá splnění práce, zadané v rámci crowdsourcingu, odměněno finančně, přičemž může být odměna předána všem úspěšným řešitelům nebo pouze nejlepšímu z nich.

Vnitřní motivace člověka k vykonání dané činnosti bývá podstatně účinnější než motivace vnější. Tento fakt je potvrzen i dále uvedenou studií.

Nikolas Kauffman a spol. [18] vytvořili, na základě existujících teorií (včetně teorie klasické motivace), model popisující motivaci pracovníků k účasti v projektech založených na spolupráci davu. Ze sebedeterminační teorie motivace Deciho a Ryana převzali dva pojmy – vnitřní (ang.

intrinsic) a vnější (ang. extrinsic) motivace [18]. V této teorii je vnitřní motivace chápána jako ta, která pochází z uspokojení, které přináší činnost sama. Oproti tomu, motivace vnitřní je brána pouze jako nástroj k dosažení výsledků. Oba typy autoři dále dělí na podkategorie. U každé z nich také uvádí faktory, které je ovlivňují např. čím více se pracovník s úkolem ztotožní tím větší bude jeho vnitřní motivace vycházející z potěšení. Autoři také uvádějí, že dělení na vnitřní a vnější motivaci je spíše teoretické a za důležitější považují dělení na nižší úrovni.

Dělení motivace společně s jednotlivými faktory je znázorněno v tabulce 1, která byla vytvořena na základě schématu uvedeném v práci Kaufmanna a spol. [18].

Tabulka 1: Kombinovaný model motivace

	Kategorie motivace	*Faktory*
Vnitřní motivace	Motivace založená na potěšení	Rozmanitost využitých dovedností
		Hmatatelný výsledek úkolu
		Autonomie v úkolu
		Přímá zpětná vazba
		Zábava
	Motivace založená na komunitě	Identifikace s komunitou
		Sociální kontakt
Vnější motivace	Motivace založená na okamžité odměně	Výplata
	Motivace založená na pozdější odměně	Signalizace pro okolí
		Vylepšení dovedností užitečných do budoucna
	Sociální motivace	Shoda s hodnotami společnosti
		Povinnosti a normy určené třetí stranou
		Nepřímá zpětná vazba

3.4 Využití

Možnosti využití crowdsourcingu jsou široké. Na jedné straně může být využit k rutinním úlohám jako je sběr dat, či překlad krátkých textů. Na straně druhé může dopomoci k řešení komplexních úloh v rámci inovativních projektů [15].

Spolupráce davu bývá v praxi často využívána ve spojitosti s marketingem. V posledních letech jeho využití vzrůstá. Zde je několik příkladů z praxe [19]:

- **Waze** – Aplikace týkající se dopravní situace, využívající informace od uživatelů pro registraci dopravních uzavírek, zácp, měřených úseků apod.

- **Lego** – Firma využila dav pro vytvoření nových produktů. Uživatel může navrhnout nový produkt a zároveň hlasovat pro návrhy druhých, přičemž produkt s nejvíce hlasy bude vyroben a jeho autor bude odměněn 1 % čistého výnosu.
- **Samsung** – využívá dav k hledání inovativních řešení pro stávající elektronické produkty a technologie.
- **Lays** – Stejně jako McDonalds i firma Lays využila crowdsourcing pro vytvoření nové příchuti. Díky vítězné příchuti se zvedly tržby firmy o 8 %.
- **Greenpeace** – Tato organizace pobídla své stoupence k vytvoření sarkastických hlášek, které následně použila k negativní reklamě ropné společnosti Shell.

4 Redakční systémy

Text následujících kapitol, týkajících se tématu redakční systémy, bude částečně převzat z mé bakalářské práce [20].

Redakční systém (ang. Content Management System, zkráceně CMS) je aplikace nebo sada souvisejících programů pro vytvoření a správu digitálního obsahu. CMS se používají ve dvou oblastech – pro správu podnikového obsahu (ang. Enterprise Content Management, zkráceně ECM) a správu webového obsahu (ang. Web Content Management, zkráceně WCM) [21]. Systém ECM integruje funkce správy dokumentů, správy digitálních prostředků a uchovávání záznamů. Poskytuje uživatelům přístup založený na rolích k digitálnímu obsahu organizace. Na druhou stranu WCM usnadňuje spolupráci při tvorbě a správě webových stránek. Software ECM často obsahuje funkci publikování WCM, ale webové stránky ECM obvykle zůstávají za firewallem organizace [21].

V tomto textu se zaměříme na využití redakčních systémů v kontextu webových stránek. Zkratka CMS bude tedy využívána ve smyslu WCM systémů.

Tyto systémy umožňují i laickému uživateli poměrně snadnou administraci jeho webových stránek i bez znalosti programovacích a skriptovacích jazyků jako například HTML, PHP, JavaScriptu apod. Využití možností CMS výrazným způsobem snižuje náklady na celý provoz internetových stránek, jelikož není nutné si kupovat služby třetích stran na jejich aktualizaci.

Ve svých začátcích byly redakční systémy používány především při tvorbě a administraci osobních stránek a blogů. V dnešní době, při možnosti rozšíření CMS různými doplňky, se redakční systémy používají také pro elektronické obchody, diskuzní fóra či sociální sítě.

Jedním z hlavních důvodů využití redakčního systému je častá změna či aktualizace obsahu webových stránek a požadavek na uživatelsky příznivou a snadnou administraci. Z pohledu vývojářů představují CMS velkou výhodu v připravených šablonách a doplňcích, které umožňují snadnější tvorbu samotných stránek. Na nynějším trhu je k dispozici velké množství redakčních systémů, jak komerčních, jejichž cena je se odvíjí od funkčnosti systému, tak freeware, tedy CMS, které jsou dostupné zdarma. K dispozici jsou také tak zvané open source CMS. U těchto systémů má uživatel možnost, při dodržení určitých podmínek, prohlížet a zasahovat do jejich zdrojového kódu.

V následujícím textu budou v krátkosti popsány jedny z nejoblíbenější redakčních systémů, mezi které patří zejména WordPress, Joomla!, Drupal. Tyto CMS se řadí mezi open source.

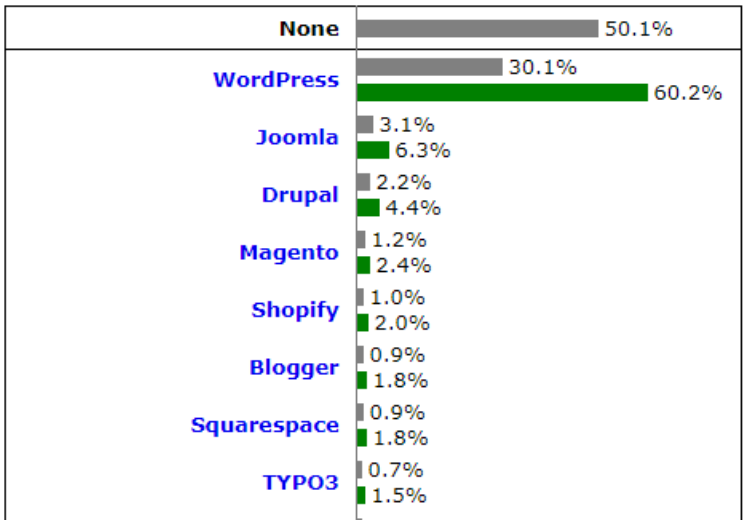
4.1 WordPress

WordPress (zkráceně WP) je v dnešní době nejvyužívanější redakční systém dostupný zdarma, což potvrzuje také graf na obrázku 8. Ten říká, že 30 % všech webových stránek a zároveň 60,2 % webových stránek založených na CMS je vytvořeno a administrováno pomocí WP [22].

WP je vhodný pro vytvoření blogů či firemních stránek, ale také e-shopů. Využívá skriptovacího jazyka PHP a databáze MySQL a MariaDB.

Byl vytvořen v roce 2003. Jeho kořeny však sahají až do roku 2001. V roce 2005 vznikla také hostingová služba WordPress.com. Zde se uživatel může registrovat, čímž získá doménu 3. řádu, a vytvořit si svůj vlastní blog, přičemž není nutná jakákoliv instalace či nastavování. Druhou možností je nasazení WP na vlastní server, který však musí mít podporu Apache a výše zmíněných PHP a MySQL. Výhodou této volby je vývoj webových stránek na lokálním počítači, avšak za potřeby stažení, instalace a konfigurace systému, která však není složitá.

WP je jedním z uživatelsky nejpříznivějších a nejlépe hodnocených open source CMS na trhu s velkým množstvím dostupných pluginů.



Obrázek 8: Použití různých CMS (údaje k datu 11. 3. 2018) [22]

Příklady webových stránek založených na CMS WordPress:

- TheWaltDisneyCompany.com,
- KatyPerry.com,
- Chicago.Suntimes.com.

Z důvodu velkého procenta využití WordPressu na trhu byl tento CMS použit také v praktické části této práce (viz kapitola 5). Následující dvě kapitoly budou tedy obsahovat podrobnější informace o šablonách a doplňcích WordPressu a o možnostech jejich vlastního vytvoření.

4.1.1 Šablony pro WordPress

Šablona WordPressu se skládá ze sady PHP skriptů, které dohromady vytváří design a funkcionalitu daných webových stránek. Tyto šablony umožňují uživateli měnit vzhled stránky jednak přes administrátorské rozhraní WP, ale také úpravou samotných skriptů. Komunita WordPressu vytvořila nespočet šablon, které jsou jak komerční (za poplatek), tak volně stažitelné. Již po

instalaci WordPressu či vytvoření domény přes hostingovou službu, máte k dispozici tři šablony, avšak pár kliknutími je možné stáhnout, nainstalovat a ihned i využít šablony další.

Každá šablona se skládá ze tří hlavních typů souborů [23]. Prvním jsou kaskádové styly obsažené v souboru style.css. Pomocí tohoto souboru je určena vizuální podoba stránek. Druhým typem je PHP skript s funkcemi – functions.php. Ten, pokud je v šabloně obsažen, je automaticky načten při inicializaci WP. Posledním typem jsou soubory šablony, které řídí způsob načtení a zobrazení dat z databáze WP.

WordPress však uživatele neomezuje pouze na šablony již vytvořené, ale umožňuje vytvořit si šablonu vlastní. K tomu je však již samozřejmě nutná znalost PHP, HTML a CSS (popř. JavaScriptu). Zároveň je vytvoření vlastní šablony možné provést dvěma způsoby – vytvoření zcela nové šablony, vytvoření tzv. potomka šablony (ang. child theme) - přičemž druhá zmiňovaná varianta je jednodušší.

Potomek šablony dědí funkcionalitu i design své nadřazené šablony [24]. Obojí však může změnit či rozšířit. Využití potomka šablony je doporučeno při úpravě šablon. Výhodou tohoto přístupu je především perzistence změn při aktualizaci nadřazené šablony.

Obsah takto vytvořené šablony je, při nejmenším, soubor s kaskádovými styly a PHP skript s funkcemi šablony (viz výše) [24]. Pro správné zobrazení šablony na administrační stránce WP musí soubor style.css na svém začátku obsahovat hlavičku s názvem šablony, popisem, jménem autora, verzí šablony apod. Soubor s funkcemi musí naopak přidat kaskádové styly nadřazené šablony do vytvořeného potomka, a to pomocí pár řádků PHP kódu [24]:

```php
<?php
    function my_theme_enqueue_styles() {
       $parent_style = 'parent-style';

        wp_enqueue_style( $parent_style, get_template_directory_uri() . '/style
            .css' );
        wp_enqueue_style( 'child-style',
             get_stylesheet_directory_uri() . '/style.css',
      array( $parent_style ),
      wp_get_theme()->get('Version')
        );
     }

     add_action( 'wp_enqueue_scripts', 'my_theme_enqueue_styles' );
?>
```

Výpis 1: Přidání kaskádových stylů nadřazené šablony do vytvořeného potomka (PHP)

Dále by měl obsahovat definici funkcí použitých ve více šablonových souborech nebo nastavit nabídku možností umožňující uživatelům šablony nastavit barvu, styly a další aspekty šablony.

Jako u většiny práce s WP, i tato činnost může být zjednodušena pomocí vytvořeného volně stažitelného doplňku. Doplněk s názvem Child Theme Configurator uživateli usnadní proces vytváření potomka šablony a je velice užitečný při vytváření takovéto šablony z již upravené nadřazené šablony. Uživatel pouze zadá jméno nové šablony, název šablony nadřazené a označí soubory, které již v nadřazené šabloně změnil. Doplněk se o zbytek postará sám.

Druhou variantou je vytvoření zcela nové šablony. Tento proces je sice komplikovanější než postup předchozí, avšak ne příliš složitý. Nově vytvořená šablona musí minimálně obsahovat soubor style.css a index.php [23]. Druhý zmiňovaný soubor je jeden z hlavních šablonových souborů. Může být např. použit k zahrnutí všech odkazů na hlavičku, postranní panel, zápatí, obsah, kategorie, archivy, vyhledávání, chyby a jakoukoli jinou stránku vytvořenou v aplikaci WordPress. Tyto části však mohou být také zahrnuty jako samostatné soubory. Značnou výhodou WP je existence výchozích šablonových souborů, které jsou použity v případě, že použitá šablony potřebný soubor neobsahuje. Uživatel si však může definovat vlastní šablonové soubory s určitou funkcionalitou. K tomu stačí pouze vytvořit PHP soubor ve složce templates a na začátek souboru připojit hlavičku obsahující název.

Vytváření šablon je poměrně rozsáhlé téma a dnes je možné na internetu najít podrobné návody celého procesu. Šablony přinášejí několik výhod [23]. Ve WP jsou soubory šablony odděleny od souborů systémových což znamená, že aktualizace CMS jako takového nepromítne žádné změny do prezentační vrstvy. Jelikož je šablona napojená přímo na administrátorské rozhraní WP, není nutné pro úpravy, např. pro změnu barvy písma, umět CSS, HTML či PHP. Navíc umožňují během vývoje stránky velice jednoduše měnit jejich podobu.

4.1.2 Doplňky pro WordPress

Doplněk pro WP je prostředek pro jednoduché přizpůsobení, úpravu či vylepšení WP stránky. Jedná se o program nebo sadu jedné nebo více funkcí, napsaných v jazyce PHP, který WP stránku rozšiřuje sadou funkcionalit nebo služeb [25]. Doplněk je možné do stránky integrovat za využití přístupových bodů a metod WordPress API [5].

Stejně jako v případě šablon, i doplňků bylo vytvořeno, ze strany komunity, spousta. Existují doplňky pro vytváření samotných stránek, které uživateli umožní vytvořit si profesionálně vypadají stránky bez napsání řádku kódu. Jiné doplňky umožňují sestavit formulář jaký uživatel potřebuje, opět pouze klikáním a umisťováním komponent na dané místo. Samozřejmě existují doplňky pro propojení webové stránky se sociálními sítěmi. Ty umožňují zobrazovat obsah sociálních sítí na webové stránce či odeslat naopak obsah webové stránky na sociální síť. Velmi užitečné jsou doplňky pro správu uživatelských rolí, správu přihlašování uživatelů či výše zmíněný doplněk pro vytvoření potomku šablony.

WP také ihned po instalaci nabízí dva výchozí doplňky – Akismet a Hello Dolly [25]. Akismet umožňuje kontrolu komentářů u příspěvků, zda se nejedná o spam. Hello Dolly byl prvním

[5] API, celým názvem Application Programming Interface, je sada protokolů a nástrojů pro vytváření softwarových aplikací. V podstatě rozhraní API určuje, jak by měly softwarové komponenty interagovat.

WordPress doplňkem vůbec a zároveň je uváděn jako vzor pro vytváření zdrojových kódů nových doplňků. Všechny nainstalované doplňky uživatel nalezne ve složce wp-content\plugins. Zde se ukládá i nově vyvíjený doplněk, který je možné, ihned po přidání do složky, vidět v administrační stránce WP.

Nově vytvořený doplněk musí splňovat několik požadavků [25]:

- Doplněk musí mít unikátní název.
- Hlavní PHP skript musí mít unikátní jméno v celém repozitáři (většinou totožné s názvem doplňku). V případě uložení doplňků ve složce, musí mít i ta unikátní jméno.
- Doplněk musí obsahovat minimálně jeden PHP skript (může obsahovat také JavaScripty, obrázky, kaskádové styly apod.).
- případě publikování doplňku musí složka doplňku obsahovat soubor readme.txt ve standardizovaném formátu.

Dále je doporučeno vytvořit pro doplněk webovou stránku popisující funkcionality doplňku, jeho instalaci a další náležitosti. Hlavní PHP skript doplňku musí, stejně jako soubor style.css u šablon, obsahovat hlavičku s informacemi jako název doplňku, verze, popis apod. a navíc informaci o licenci, se kterou je doplněk distribuován [25]. Po uvedení hlavičky může již skript obsahovat samotnou funkčnost doplňku. Aby mohly být funkce doplňku využívány, musí být registrovány, a to využitím funkce add_action ($tag, $nazev_fc), tedy např. add_action('init', 'hello_world'). Pokud je doplněk aktivován, takto registrovaná funkce je automaticky zavolána. Funkčnost doplňku pak může být otestována jednoduchým skriptem:

```php
<?php
    if(function_exists('hello_world')) {
        hello_world();
    }
?>
```

Výpis 2: Otestování funkčnosti doplňku (PHP)

WP také umožňuje do menu své administrační stránky přidat záložku s vytvořeným doplňkem, která může obsahovat možnosti jeho nastavení. Všechny možnosti doplňků, návody, rady a odkazy je možno najít v článku [25].

4.2 Joomla!

Joomla!, stejně jako WP, je CMS založený na skriptovacím jazyce PHP a databázi MySQL. Provozovat ji lze na webovém serveru s Apache nebo IIS. V novějších verzích však Joomla! Rozšířila podporu i na další databáze jako PostgreSQL, Oracle, SQLite apod.

Podle informací dostupných na českém portálu Joomla!, tento CMS podporuje indexaci stránek, RSS, tisknutelné verze stránek, blogy, diskusní fóra a mnoho dalšího. Výstupem je HTML, CSS kód a JavaScript. Joomla! byla uvedena na trh v roce 2005. Jejími zakladateli jsou bývalí vývojáři jiného CMS, s názvem Mambo, kteří se rozhodli, po neshodách s jeho komunitou, jít vlastní cestou.

Tento CMS je bohužel v porovnání s WP méně uživatelsky přívětivý a začátečníkovi trvá delší dobu, než se v rozhraní zorientuje. Na druhou stranu může být značnou výhodou možnost vývoje v češtině.

Příklady firem a organizací, jejíž webové stránky využívají Joomla! CMS:

- Linux.com,
- Gsas.hardvard.edu,
- PlayShakespere.com.

4.3 Drupal

Drupal byl vytvořen v roce 2001 holandským studentem Driesem Buytaertem. Nyní se o vývoj stará několik hlavních vývojářů a několik stovek přispěvatelů. Celý systém staví na modulech, přičemž Drupal dává možnost si vlastní modul vytvořit.

Jako předešlé dva systémy i Drupal pracuje s jazykem PHP a databází MySQL. Navíc přidává podporu PostgreSQL, SQLite, MariaDB, MongoDB ad.

Drupal má poměrně velké množství dostupných pluginů, především díky rozšířené komunitě vývojářů. WP je však na tom stále lépe. Výhodou je možnost úpravy stránek po blocích, avšak jejich editace může být pro začátečníka obtížnější.

Příklady webových stránek využívajících CMS Drupal:

- WhiteHouse.gov,
- London.gov.uk,
- Grammy.com.

5 Realizace prototypu

Tato část obsahuje kapitoly týkající se praktické části této práce, jejímž cílem je navrhnout a implementovat prototyp klasifikátoru textových dokumentů na základě jejich podobnosti s následným využitím crowdsourcingu za účelem zkvalitnění kategorizace. Kapitoly se budou postupně zabývat návrhem, architekturou a implementací řešení.

5.1 Návrh

Navrhovaný systém se skládá ze dvou částí – z klasifikátoru a rozhraní pro crowdsourcing. V následujících dvou kapitolách bude popsán návrh jednotlivých částí.

5.1.1 Klasifikátor

Z velkého množství algoritmů, určených pro kategorizaci textu v přirozeném jazyce, byl vybrán algoritmus využívající N-gramy. Výhodou tohoto algoritmu je poměrně snadná implementace, a především nezávislost na jazyce textu. Postup kategorizace pomocí N-gramů byl navržen na základě článku Williama B. Cavnara a spol. [8] (viz kapitola 1.5.1) a skládá se ze dvou hlavních fází – inicializační fáze a fáze samotné kategorizace. Účelem první fáze je nahrání a zpracování dat trénovací sady. Tato fáze zahrnuje následující kroky:

- Import trénovací sady textových dokumentů do databáze (dále jen DB) společně s definovanými kategoriemi
- Výpočet profilů jednotlivých kategorií:
  - Tokenizace textů všech dokumentů v kategorii s vynecháním stop slov, čísel a speciálních znaků.
  - Vytvoření N-gramů, z textů zpracovávané kategorie, s délkou dvě až pět.
  - Seřazení získaných N-gramů podle jejich četnosti.
  - Vytvoření profilu kategorie useknutím seznamu N-gramů na pozici 600.
  - Uložení vytvořeného profilu do DB.

První část druhé fáze probíhá podobně jako fáze inicializační s tím rozdílem, že proces je aplikován pouze na jeden textový dokument. Samotná kategorizace se tedy skládá z následujících kroků:

- Výpočet profilu dokumentu určeného ke kategorizaci:
  - Stejný postup jako u výpočtu profilu kategorie, pouze vztažený na jeden dokument.
- Výpočet vzdáleností profilu kategorizovaného dokumentu k profilu každé kategorie pomocí metody out-of-place.

- Zařazení dokumentu do kategorie s nejmenší vzdáleností.

Na rozdíl od článku [8], využívá navržený klasifikátor redukci počtu slov dokumentu pomocí odstranění stop slov. Díky této redukce není potřeba začínat profil kategorie na pozici 300, jak popisuje článek [8], ale je možné jej brát od začátku seznamu. Oproti článku využívá navržený klasifikátor také delší profily, a to především kvůli plánované kategorizaci psychologických textů, jejichž kategorie mají mezi sebou mnohdy velmi tenkou hranici. Lze tedy předpokládat, že na rozpoznání jednotlivých kategorií bude za potřebí více N-gramů.

Vedle kategorizace bude druhou funkcí aplikace také výpočet klíčových slov daných kategorií. Tato klíčová slova budou následně zobrazena vybraným uživatelům s prosbou o jejich využití v příspěvku. Zobrazením klíčových slov pouze některým uživatelům budou vytvořeny dvě skupiny uživatelů, které budou navzájem sloužit jako referenční skupiny pro ověření následující hypotézy:

- Kategorizace bude dosahovat lepších výsledků, pokud budou příspěvky, určené ke kategorizaci, obsahovat předem stanovená klíčová slova.

Výpočet klíčových slov bude realizován pomocí algoritmu TF-IDF uzpůsobeného, podle článku [28], účelu kategorií. Vzorce pro výpočet jednotlivých složek budou tedy vypadat následovně:

$$TF_{ij} = \frac{n_{ij}}{\sum_k n_{kj}}, \tag{5.1}$$

kde n_{ij} je četnost výskytu termu i v dokumentech kategorie j.

$$IDF_i = log(\frac{|D|}{|\{d : t_i \in d\}|}), \tag{5.2}$$

kde t_i je term a D je sada všech kategorií.

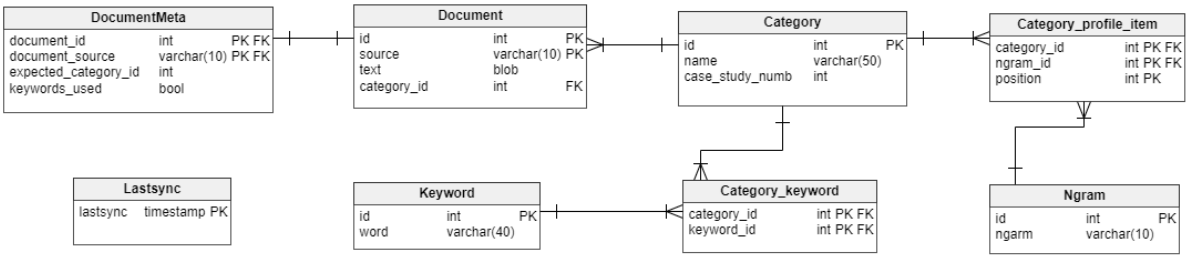
TF_{ij} je tedy podíl četnosti výskytu termu n_{ij} k počtu všech slov v dokumentech dané kategorie a IDF_i jako logaritmus podílu počtu kategorií ku počtu kategorií, ve kterých se nachází term t_i.

Z takto vypočítaných klíčových slov je pět slov s nejvyšší vahou pro každou kategorii uloženo do databáze.

Jak bylo naznačeno v popisu inicializační fáze, klasifikátor ukládá data trénovací sady a profilů kategorií do DB. Pro tyto účely byla vybrána databáze MySQL a to především s ohledem na pozdější plánované využití redakčního systému WordPress, který také s tuto databází pracuje. Využitím jednoho typu databáze pro oba systémy odpadne nutnost instalace a administrace více druhů databází na serveru, na kterém budou obě aplikace spuštěny.

Celá DB se skládá z osmi tabulek. Tabulky Document, Category, Ngram a vazební tabulka Category_profile_item jsou tabulky vytvořené čistě pro účely klasifikátoru. Tyto tabulky obsahují data trénovací sady jako jsou texty dokumentů, zdroj dokumentu (crowdsourcing nebo původní trénovací sada), potřebné kategorie či profily jednotlivých kategorií. Jelikož mohou být v databázi v jednu chvíli uložena data z více zdroju, obsahuje každá kategorie, kromě svého ID a názvu, také položku case_study_numb, která odlišuje jednotlivé případové studie. Ostatní tabulky – DocumentMeta, Lastsync, Keyword a Category_keyword – uchovávají data, která souvisejí s rozhraním pro crowdsourcing (viz kapitola 5.1.2). Tabulka Lastsync je využívána při synchronizaci dat mezi WP a klasifikátorem. Data v této tabulce zabraňují vytvoření duplicitních dat v databázi klasifikátoru a to tak, že mezi klasifikátorem a WP jsou přenášena pouze data starší, než je datum poslední synchronizace. Tabulka Keyword, společně s vazební tabulkou Category_keyword, obsahuje klíčová slova kategorií, která jsou dále využívána v crowsourcingovém rozhraní (viz kapitola 5.1.2). Tabulka DocumentMeta slouží k uchování metadat dokumentu získaných z rozhraní pro crowdsourcing. Tato data jsou ukládána pro účely vyhodnocení úspěšnosti klasifikace.

Relační model využité databáze je znázorněn na obrázku 9.



Obrázek 9: Relační model databáze klasifikátoru

5.1.2 Rozhraní pro crowdsourcing

Součástí této práce je ověření a zvýšení úspěšnosti kategorizace pomocí crowdsourcingu. Za tímto účelem byla navrhnuta a vytvořena webová stránka, jejíž funkce budou popsány v této kapitole. Tato práce by měla zároveň sloužit jako základ pro systém vyvíjený v rámci připravovaného projektu mezi Ostravskou univerzitou v Ostravě a Vysokou školou Báňskou - Technickou univerzitou. Na základě tohoto projektu bylo stanoveno téma webu potažmo cílové kategorie pro klasifikátor. Klíčovým tématem je život neformálních pečujících v závislosti na poskytovanou péči. Popis jednotlivých kategorií a využitého davu obsahuje kapitola 6.2.

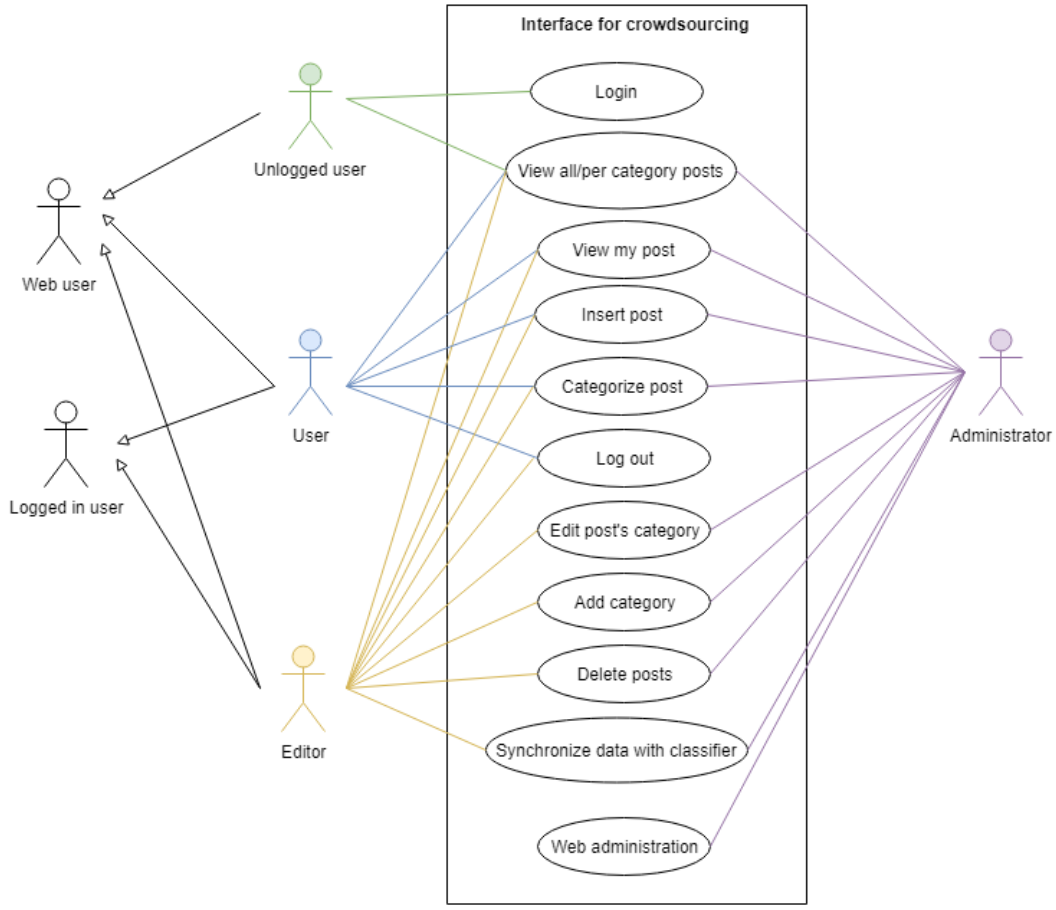
Jak již bylo naznačeno výše, k vytvoření rozhraní pro crowdsourcing byl využit redakční systém WordPress. Tento CMS byl vybrán především kvůli jeho popularitě a jednoduchosti. Z hlediska zapojení práce do projektu je možné předpokládat, že ne všichni zůčastětní budou mít hlubší informatické vzdělání, jednoduchost administrace webového rozhraní je zde tedy zásadní.

Navrhovaný web by měl obsahovat stručný popis jeho účelu a popis práce s celým rozhraním. Dále by měl obsahovat formulář pro vkládání příspěvků, možnost kategorizace jednotlivých příspěvků, administraci příspěvků a možnost spuštění synchronizce dat.

Uživatelé webu jsou rozděleni na dvě skupiny - přihlášení a nepřihlášení. Jedinou funkcí, dostupnou nepříhlášeným uživatelům, je zobrazení přidaných příspěvků. Přihlášeným uživatelům je dále přidělena jedna ze tří základních rolí, podle které je určena dostupná funkčnost. Role jsou následující:

- uživatel,
- editor,
- administrátor.

Každému registrovanému uživateli je explicitně přiřazena role uživatele. Roli editora může uživateli přiřadit administrátor webu. Následující use case diagram zobrazuje funkce dostupné jednotlivým uživatelům.

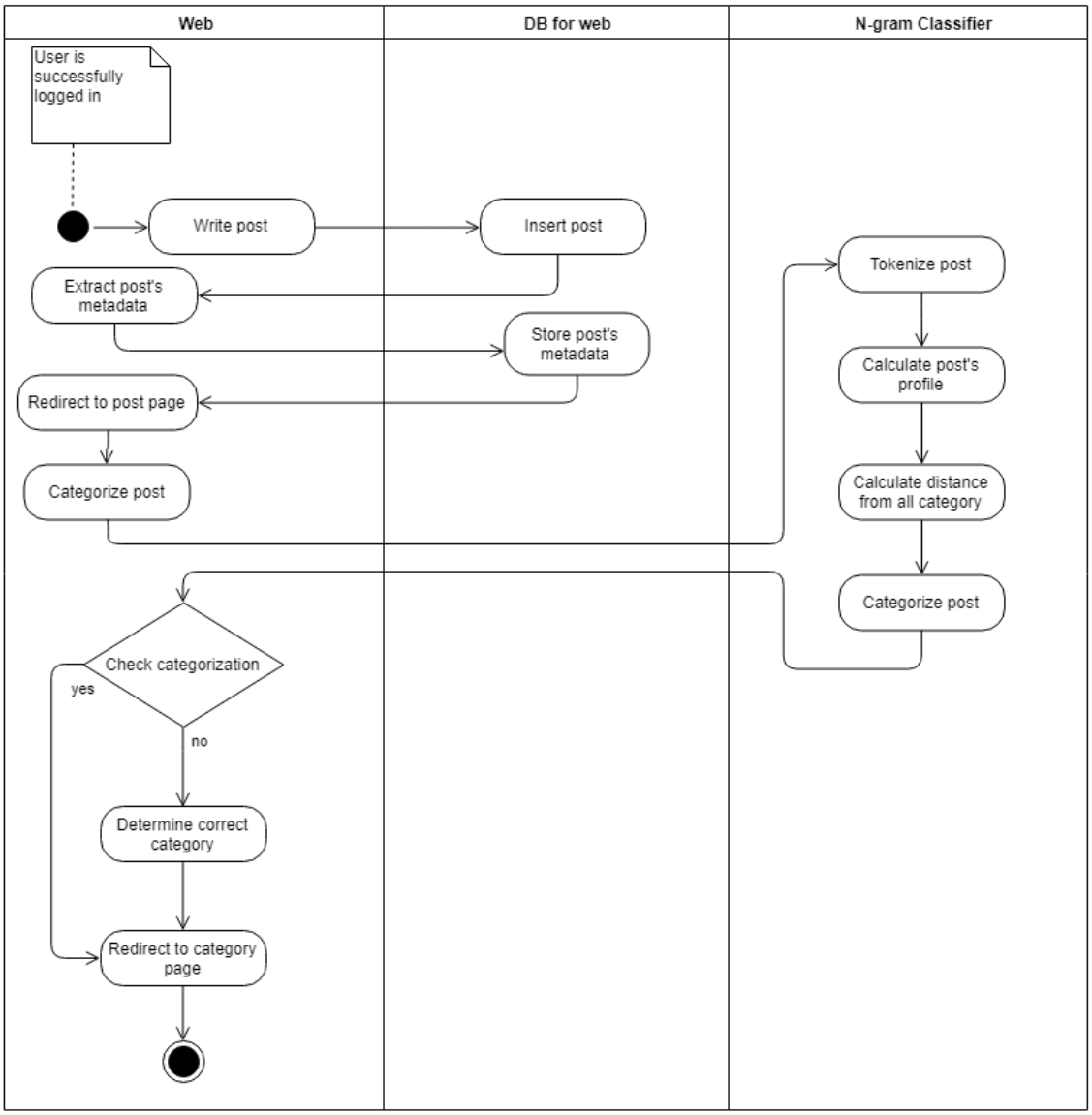


Obrázek 10: Use case diagram dostupných funkcí rozhraní pro crowdsourcing podle role uživatele

Vytvořený web bude obsahovat několik klíčových funkcí. Všechny tyto funkce budou přístupné pouze přihlášeným uživatelům a poslední z nich pouze uživatelům s oprávněním editora či administrátora:

- vkládání příspěvků,
- kategorizace příspěvků,
- synchronizace dat s klasifikátorem.

Proces vkládání příspěvků a jejich následná kategorizace je popsán na následujícím diagramu aktivit. Jeho jednotlivé fáze budou rozebrány v textu níže.



Obrázek 11: Diagram aktivit pro proces vkládání příspěvku a jeho kategorizace

Každý přihlášený uživatel má možnost vložit na web příspěvek týkající se jedné z definovaných kategorií. V této fázi práce je každý přihlášený uživatel před vkládání příspěvku vyzván, aby napsal příspěvek na konkrétní přiřazené téma. Zároveň, jak již bylo naznačeno výše, je každému uživateli přiřazen příznak, zda mu při vkládání příspěvku budou či nebudou poskytnuta klíčová slova kategorií, která by měl v příspěvku použít.

Cílem přiřazení uživatele ke konkrétní kategorii je rovnoměrné otestování všech kategorií. Pokud uživatel vloží příspěvek na požadované téma, bude mu přiřazeno téma další. Takto může uživatel přidat příspěvky do všech kategorií. Účelem rozdělení uživatelů, na skupinu s klíčovými slovy a bez nich, je potvrzení hypotézy o zvýšení kvality kategorizace při využití sady klíčových slov.

Po vložení příspěvku jsou extrahována jeho metadata jako např. kategorie, do které by měl být příspěvek přiřazen nebo údaj o tom, zda uživateli byla poskytnuta klíčová slova. Tato metadata by měla sloužit k následné kontrole a zhodnocení kategorizace.

Druhou fází je kategorizace příspěvku. Uživatel má možnost spustit pro daný příspěvek kategorizaci a následně její výsledek vyhodnotit. Po zaslání požadavku na kategorizaci je text příspěvku zaslán do klasifikátoru, který vrátí stanovenou kategorii. Následně má uživatel dvě možnosti - označit kategorizaci za správnou nebo za chybnou. V případě, že je kategorizace označena jako správná, uživatel je přesměrován na stránku, která obsahuje všechny příspěvky dané kategorie. Pokud však uživatel označí kategorizaci za chybnou je vyzván, aby určil, podle něj, kategorii správnou. Takto opravená kategorie je následně příspěvku přiřazena a zanesena do databáze. Následně je uživatel, stejně jako v případě správné kategorizace, přesměrován na stránku s příspěvky dané kategorie.

Součástí této práce je využití crowdsourcingu pro zkvalitnění kategorizace. Pro tento účel jsou všechny příspěvky, u kterých byla kategorizace označena za chybnou, zařazeny do trénovací sady klasifikátoru. Díky tomu se bude trénovací sada rozšiřovat, což je základní předpoklad pro vyšší úspěšnost kategorizace. Do trénovací sady budou dále zařazeny příspěvky vložené uživateli s oprávněním administrátora a editora a také příspěvky, jejiž kategorie byla administrátorem či editorem upravena.

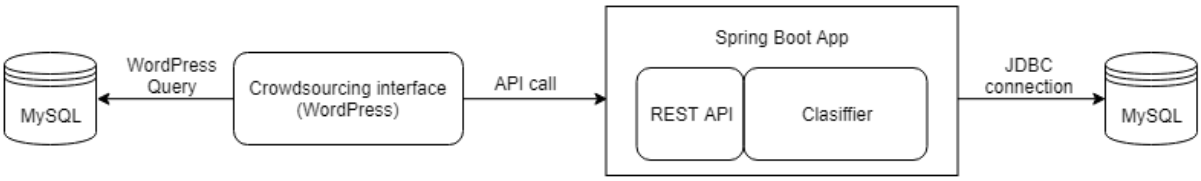
Rozšiřování trénovací sady klasifikátoru je zajištěno synchronizací dat mezi webem a klasifikátorem. Data určená k synchronizaci jsou poslána klasifikátoru, který na jejich základě přepočítá profily a klíčová slova kategorií a ovlivní tak kategorizaci následující. Synchronizace bude spouštěna v pravidelných intervalech. Zároveň bude mít administrátor a editor možnost spustit ji ručně.

5.2 Architektura a implementace

Navrhovaný prototyp se skládá ze dvou částí - klasifikátoru a rozhraní pro crowdsourcing. Každá část představuje samostatnou aplikaci, přičemž každá je implementována v jiném jazyce. Crowdsourcingové rozhraní je založeno na CMS Wordpress a je tedy implementováno v jazyce PHP.

Klasifikátor je, oproti tomu, implementován v jazyce Java a je spuštěn jako Spring Boot aplikace. Zároveň má každá část systému vlastní datové uložiště - MySQL databázi.

Obě tyto části spolu komunikují přes REST rozhraní, které poskytuje klasifikátor. Tato komunikace je vyvolána vždy ze strany rozhraní, které posílá dotazy na klasifikátor za využití cURL[6].



Obrázek 12: Architektura systému pro kategorizaci a její hodnocení pomocí crowdsourcingu

5.2.1 Klasifikátor

Jak již bylo napsáno výše, klasifikátor je implementován v jazyce Java. Kromě již zmíněného frameworku Spring, využívá také knihovnu Apache Lucene, konkrétně její stemmer, a knihovnu Apache POI pro čtení .DOCX souborů.

Z frameworku Spring je, kromě anotací a spouštění aplikace pomocí SpringBoot, využíván také JdbcTemplate k připojení k databázi klasifikátoru. Instance JdbcTemplate je vytvořena jako Spring bean[7] a následně je do každé DAO (data access object) třídy[8] injektována.

Knihovna Apache Lucene poskytuje mnoho funkcí (viz kapitola 2.1.1). Pro tento klasifikátor byla vybrána kvůli jejímu stemmeru, který je využíván při výpočtu klíčových slov pro crowdsourcingové rozhraní. Jednou z výhod Lucene stemmeru je také výchozí sada stop slov, která však může být zaměněna za sadu vlastní. Lucene také umožňuje aplikaci filtrů, jako např. odstranění speciálních znaků.

Stemming textu za využití Lucene je velice jednoduchý a představuje, v Javě, několik málo řádků kódu. Kód pro zpracování textu v implementovaném klasifikátoru vypadá následovně:

```
private static List<String> process(CzechAnalyzer analyzer, String text) throws
    IOException {
      List<String> tokens = Lists.newArrayList();

      TokenStream tokenStream = analyzer.tokenStream("doc", text);
      tokenStream = new LowerCaseFilter(tokenStream);
      tokenStream = new StopFilter(tokenStream, CharArraySet.copy(analyzer.
          getStopwordSet()));
      tokenStream = new PatternReplaceFilter(tokenStream, Pattern.compile("
          [0-9]*"), "", true);
```

[6] cURL je nástroj příkazové řádky a softwarová knihovna umožňující transfér dat pomocí různých protokolů.
[7] Spring bean je objekt, který je instancován a spravován Spring kontejnerem.
[8] DAO třídy představují rozhraní poskytující přístup k databázi nebo jiného datovému uložišti.

```
        CharTermAttribute charTermAttribute = tokenStream.getAttribute(
            CharTermAttribute.class);

        tokenStream.reset();
        while(tokenStream.incrementToken()){
            String token = charTermAttribute.toString();

            if(!token.isEmpty()) {
                tokens.add(token);
            }
        }

        tokenStream.end();
        tokenStream.close();

        return tokens;
    }
```

Výpis 3: Zpracování textu pomocí Lucene (Java)

V tomto kódu je zpracovávaný text převeden na malá písmena, jsou z něj odstraněna definovaná stop slova a čísla a následně je celý text tokenizován a stemován. Výstupem je tedy seznam stemovaných slov.

Knihovna Apache POI je využívána v inicializační fázi kategorizace při načítání dat z datové struktury (struktura dat viz kapitola 6.1) do klasifikátoru a jeho DB. Tato knihovna umožňuje zpracovat dokumenty ve formátu .DOCX a .DOC. Kromě těchto formátů je klasifikátor připraven také na zpracování formátu .TXT.

Vstupním bodem klasifikátoru je REST rozhraní, které poskytuje funkce jako je kategorizace dokumentu, vytažení klíčových slov kategorií, import dokumentů v rámci inicializační fáze klasifikace a synchronizace dat mezi oběma částmi systému.

Pro kategorizaci je vstupem do rozhraní číslo případové studie a pole bytů představující text a jeho výstupem JSON[9] obsahující název definované kategorie. Vstupní text je nejprve tokenizován přičemž jsou odstraněna čísla, interpunkční znaménka, speciální znaky a jména a názvy, které jsou detekovány pomocí regulárního výrazu. Nakonec jsou odstraněna česká stop slova z definovaného seznamu. Dalším krokem je vygenerování všech N-gramů o délce dva až pět. Každému N-gramu je následně vypočítána jeho četnost ve vstupním textu. Výsledný seznam N-gramů s jejich četnostmi, seřazený od nejčetnějšího a zkrácený na pozici 600, tvoří profil dokumentu. Mezi takto vypočítaným profilem a každou kategorii dané případově studie je následně

[9]JSON (JavaScript Object Notation) je datový formát určený pro přenos dat ve formě polí či objektů. Jeho výhodou je platformní nezávislost.

vypočítána vzdálenost metodou "out of place"(viz kapitola 1.5.1). Dokument je následně zařazen do kategorie s nejenší vzdáleností.

Získání klíčových je reprezentováno GET metodou, jejíž vstupem je pouze číslo případové studie a výstupem JSON se seznamem klíčových slov. Zavoláním této metody jsou již vypočítaná klíčová slova pro všechny kategorie, dané případové studie, pouze vytažena z databáze. Výpočet klíčových slov probíhá v inicializační fázi kategorizace a při synchronizaci dat a je prováděn upraveným algoritmem TF-IDF (viz kapitola 5.1.1). Do DB je následně uloženo pět klíčových slov s nevyššími hodnotami pro každou kategorii.

Import dokumentů představuje inicializační fázi kategorizace. Jejím vstupem je číslo případové studie, příznak, zda mají být či nemají být vypočítávána klíčová slova a cesta k souborům s trénovací sadou. Při importu dokumentů jsou prováděny dva nebo tři základní kroky:

- načtení textů a jejich kategorií do DB,
- výpočet N-gramů a profilů všech kategorií dané případové studie a jejich uložení do DB,
- pokud je vyžádán výpočet klíčových slov, jsou vypočítána klíčová slova pro všechny kategorie dané případové studie a jsou uložena do DB.

Jak již bylo naznačeno výše, synchronizace dat mezi klasifikátorem a rozhraním je prováděna kvůli rozšiřování trénovací sady a je spouštěna v pravidelných tříhodinových intervalech.

REST rozhraní obsahuje dvě funkce vztahující se k synchronizaci. První je zjištění data poslední synchronizace. Toto datum je uloženo v DB klasifikátoru. Pokud v DB žádné datum není, jako odpověď na dotaz je posláno datum 1970-01-01, tedy výchozí datum. Druhou dostupnou metodou je samotná synchronizace dat.

Celý proces synchronizace probíhá ve třech krocích:

- zjištění data poslední synchronizace,
- vytažení požadovaných dat z WP databáze a formátování do požadovaného tvaru,
- synchronizace dat.

Požadovanými daty jsou všechny veřejné příspěvky, starší než datum poslední synchronizace, které byly vloženy či upraveny administrátorem nebo editorem a dále příspěvky, které mají v DB příznak chybné kategorizace. Tato data jsou v JSON formátu poslána POST metodou do klasifikátoru, kde jsou následně zpracována a uložena do databáze. Poslední krokem je přepočet dat potřebných ke kategorizaci (profily kategorií, klíčová slova), čímž bude ovlivněn výsledek následující kategorizace. Tento proces je spuštěn na pozadí a zároveň je ošetřeno, že v jednu chvíli může být spuštěna pouze jedna synchronizace.

5.2.2 Rozhraní pro crowdsourcing

Jak již bylo zmíněno, crowdsourcingové rozhraní je založeno na redakčním systému WordPress. Při implementaci rozhraní tedy byly využity všechny klíčové vlastnosti WP jako jsou potomci šablon či doplňky.

Celé rozhraní je postaveno na šabloně Freddo, ze které byl vytvořen potomek. Do vytvořeného potomka byla následně přidána potřebná funkcionalita jako je formulář pro vkládání příspěvků uživatelem, tlačítka pro spuštění kategorizace příspěvků či synchronizace dat či funkce volání REST rozhraní klasifikátoru pomocí cURL. Většina jednotlivých stránek rozhraní je založena na implementovaných templatech potomka šablony, které představují danou potřebnou funkcionalitu.

Rozhraní také využívá několik doplňků vytvořených WP komunitou. Použité doplňky jsou:

- Theme My Login - zajišťuje přihlašování, odhlašování, registraci, zapomenutá hesla uživatelů.
- User Role Editor - umožňuje vytvoření rolí uživatelů a následnou úpravu jejich práv.
- Nav Menu Roles - umožňuje zobrazovat uričeé položky menu pouze přihlášeným/odhlášeným uživatelům či uživatelům s určitým oprávněním.
- WP Crontrol - zajišťuje administraci a spouštění cronů[10] v uživatelem stanovených intervalech v rámci WP-Cron.

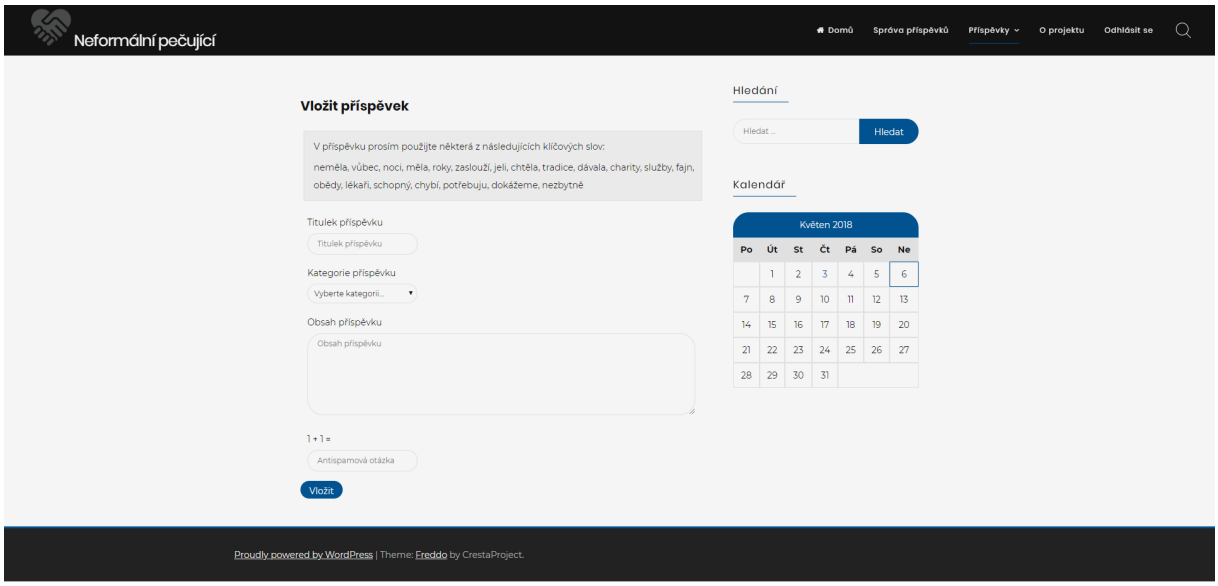
Poslední jmenovaný plugin je využíván pro automatické spouštění synchronizace dat mezi klasifikátorem a rozhraním, přičemž interval spouštění je nastaven na tři hodiny.

Implementované rozhraní je dostupné na adrese `https://cmppecujici.cs.vsb.cz`. Úvodní strana rozhraní, pro příhlášeného uživatele s oprávněním administrátora či editora, je zobrazena na obrázku 13.

[10] Cron je nástroj, který v předem stanoveném čase/intervalech spouští stanovený proces.

Obrázek 13: Úvodní strana crowdsourcingového rozhraní

Formulář pro vkládání příspěvku pro uživatele, který má dostupná klíčová slova, vypadá následovně:



Obrázek 14: Formulář pro vkládání příspěvku s klíčovými slovy crowdsourcingového rozhraní

5.2.3 Použité technologie

Následující seznam obsahuje všechny použité technologie, jak v klasifikátoru, tak v rozhraní pro croqdsouring.

- Java - implementace klasifikátoru (`https://java.com/en/`)

- Spring framework - Spring boot, anotace, JdbcTemplate (`https://spring.io/`)
- Apache Lucene - stemmer pro český jazyk (`https://lucene.apache.org/`)
- Apache POI - čtení .DOCX a .DOC souborů (`https://poi.apache.org/`)
- JSON - přenos dat mezi klasifikátorem a rozhraním pro crowdsourcing (`https://www.json.org/`)
- Maven - build a dependency management (`https://maven.apache.org/`)
- PHP - implementace crowdsourcingového rozhraní (`http://php.net/`)
- cURL - volání REST rozhraní klasifikátoru (`https://curl.haxx.se/`)

Ke správě zdrojových kódů byl použit nástroj Git. Všechny kódy jsou přístupné na adrese `https://gitlab.com/BarboraCigankova/dp`.

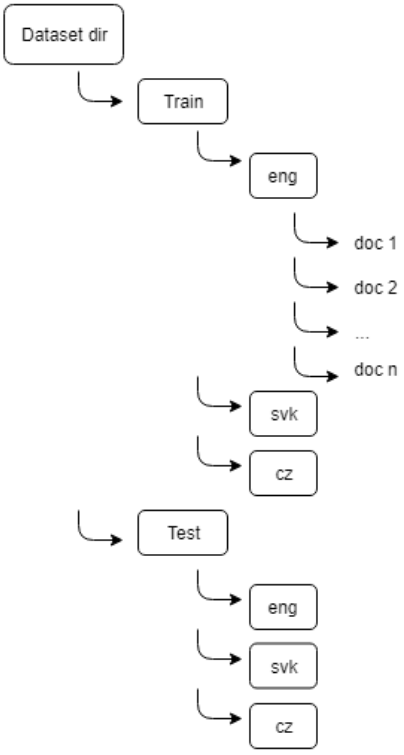
6 Ověření klasifikátoru

Jak již bylo naznačeno v Úvodu, funkčnost klasifikátoru byla nejprve ověřena na dvou datových sadách a posléze za využití crowdsourcingu. Rozhraní pro komunikaci s davem bylo vytvořeno pomocí CMS WordPress. Následující kapitoly popíší oba využité způsoby ověření.

6.1 Datové sady

Obě použité datové sady obsahovaly X textových dokumentů zařazených do Y kategorií. Každá sada byla následně rozdělena na dvě části, a to na sadu trénovací a sadu testovací. Z textů sady trénovací byly vypočítávány profily kategorií, tak jak je popsáno v kapitole předešlé, a sloužily tedy jako referenční sada pro dokumenty testovací. Ty byly následně, jeden po druhém, použity jako vstup do klasifikátoru, jehož výstup byl porovnán s definovanou kategorií. Po zpracování celé sady byla vyhodnocena úspěšnost její kategorizace v procentech.

Pro jednodušší zpracování dat byly obě sady uloženy do specifické struktury složek. Ta vypadá následovně:



Obrázek 15: Struktura složek datových sad

Hlavní složka datové sady obsahuje dvě složky – train a test. Jak je již z názvů složek jasné, první z nich obsahuje trénovací data, druhá data testovací. Obě tyto složky obsahují tolik složek, kolik je definováno kategorií, přičemž jejich názvy jsou použity jako názvy kategorií. Tyto složky již obsahují jednotlivé textové dokumenty.

Následující dvě kapitoly budou obsahovat podrobnější popis jednotlivých datových sad a výsledků kategorizace.

6.1.1 Jazyková sada

První datovou sadou je soubor textů v různých jazycích. Cílem použití této sady bylo ověření nezávislosti klasifikátoru na jazyce textového dokumentu.

Texty v této sadě jsou rozděleny do tří kategorií – anglicky, česky, slovensky – podle toho, v jakém jazyce jsou napsány. Každá kategorie obsahuje 40 textů o rozsahu 60 až 200 slov. Z celé sady vznikla trénovací a testovací sada, každá s 20 texty v každé kategorii.

Zdrojem dat jsou sborníky konference DATAKON z let 2010, 2012, 2013 a 2014. Jedná se tedy o texty technického rázu. Tyto texty byly ke zpracování poskytnuty vedoucím práce.

Ze sborníků jednotlivých ročníků konference bylo náhodně vybráno několik článku pouze na základě použitého jazyka. Z celého článku pak byly, opět náhodně, použity úryvky textu, které byly následně zařazeny do datové sady. Data, vybrána z [26], vypadají například takto:

„Podrobnější komentář k hodnocení metodik lze najít v [24]. Je třeba ale uvést, že rozsah i zaměření hodnocených metodik se liší. Např. MELODA poskytuje hlavně metodu, jak hodnotit datové sady z hlediska snadnosti jejich dalšího využití. Metodika nedefinuje proces publikace otevřených dat a ani to není jejím cílem. Hodnocení je tak třeba chápat jako porovnání oblastí a problémů, kterým se jednotlivé metodiky věnují. Hodnocení nevystihuje vhodnost či nevhodnost metodiky pro určité použití."

Přehled výsledků klasifikátoru, na popisované datové sadě, je uveden v následující tabulce:

Tabulka 2: Úspěšnost kategorizace u jazykové sady

Kategorie	*Úspěšnost*
Anglický jazyk	20/20
Slovenský jazyk	20/20
Český jazyk	20/20
Celkem	100 %

Z dostupné datové sady klasifikátor správně zařadil všech 60 dokumentů v testovací datové sadě. Bezproblémovost zařazení anglických textů byla předpokládaná, avšak u dalších dvou kategorií bylo očekáváno jejich zaměňování.

Nezávislost kategorizace dokumentů pomocí N-gramů na jazyce se tedy potvrdila. Dalším možným postupem by bylo rozšíření sady o jazyky jako polština či ruština, které jsou češtině a slovenštině poměrně podobné. Jelikož je datová sada poměrně malá, dalším vhodným krokem by bylo její rozšíření, a tedy ověření výsledků na více datech.

6.1.2 Sada s psychologickými texty

Oproti sadě první, tato sada obsahuje výrazně nevyvážený počet textů v jednotlivých kategoriích. Cílem jejího použití bylo zjistit, jak si klasifikátor poradí s ne příliš kvalitně strukturovanou datovou sadou. Sada navíc obsahuje texty s psychologickou tématikou. Texty jsou tedy zařazeny do kategorií, jejichž hranice nejsou tak striktní jako například u předešlé datové sady. Takovéto texty jsou mnohdy i člověkem těžko zaraditelné. Předpokládaná úspěšnost kategorizace těchto dat tedy nebyla příliš vysoká.

Popisovaná sada obsahuje 84 dokumentů zařazených do následujících třech kategorií:

- osobní problémy, nemoc aj.,
- práce, finance, škola,
- vztahy partnerské, rodinné, na pracovišti.

První zmíněná kategorie obsahuje nejvíce dat – 63 dokumentů. Druhá zmíněná obsahuje pouze 8 a poslední kategorie 14 dokumentů. Z každé z nich bylo náhodně vybráno 80 % dokumentů do trénovací sady a 20 % do sady testovací.

Data, s již definovaným zařazením, byla poskytnuta doc. RNDr. Martinem Malčíkem, Ph.D. k účelu implementace strojové kategorizace již v rámci dřívější práce. Správná kategorizace byla tedy stanovena odborníkem.

Výsledky kategorizace dané datové sady jsou znázorněny v následující tabulce:

Tabulka 3: Úspěšnost kategorizace psychologických textů

Kategorie	*Úspěšnost*
Osobní problémy, nemoc aj.	12/13
Práce, finance, škola	1/2
Vztahy partnerské, rodinné, na pracovišti	2/3
Celkem	69 %

V každé kategorii byl tedy jeden dokument zařazen nesprávně. První z nich, dokument z nejpočetnější kategorie, klasifikátor vyhodnotil jako text patřící do skupiny vztahy partnerské, rodinné, na pracovišti. Vypočítané vzdálenosti profilů naznačují, že tato chyba může být zapříčiněna délkou textu, který se týká přímo tématu. Celý text čítá 150 slov, avšak úvod textu je věnován tématu zcela jinému. To znevýhodňuje klasifikátor, jelikož nemusí mít dostatek N-gramů k porovnání s profily kategorií, což může zapříčinit chybnou kategorizaci.

Další dva dokumenty byly naopak zařazeny do kategorie osobní problémy, nemoc aj. Obsah dokumentu z kategorie práce, finance, škola se pohybuje na pomezí dvou témat (škola a osobní problémy), což naznačuje i malý rozdíl ve vzdálenostech profilu dokumentu a profilů obou kategorií. Tento dokument je možné označit právě jako těžce zaraditelný i člověkem.

Poslední dokument se opět kontextem pohybuje na hranici. Tady je však rozdíl ve vzdálenostech překvapivě výrazný, přičemž správná kategorie je od profilu dokumentu vzdálena nejvíce.

Možné vylepšení výsledků kategorizace by spočívalo především v lepší strukturovanosti datové sady.

6.2 Crowdsourcing

Úspěšnost klasifikátoru byla ověřena, kromě datových sad, také pomocí implementovaného crowdsouringového rozhraní popsaného v kapitole 5.2.2.

Jelikož má tato práce tvořit základ pro připravovaný projekt mezi OSU a VŠB, kategorie textů i složení davu byly tímto ovlivněny.

Téma příspěvků vkládaných do rozhraní bylo stanoveno na život neformálních pečujících a jeho ovlivnění v důsledku péče. Vzhledem k tomuto tématu byly vytvořeny čtyři kategorie textů:

- motivace k péči,
- dopady a přínosy péče,
- podpora pečujících,
- potřeby pečujících.

Texty první kategorie obsahují podtémata jako např. co neformálně pečujícího vede k péči a starání se o opečovávaného člověka nebo co mu nedovolí dát ho do ústavní péče. Druhá kategorie se zabývá texty týkající se změn života neformálního pečujího v důsledku péče o opečovávaného. O tom, co by neformálnímu pečujícímu pomohlo nebo co mu pomáhá ve vykonávání péče, jsou texty kategorie třetí. Poslední kategorie se týká strádání neformálního pečujícího vlivem péče. Patří zde texty týkající se toho co by neformální pečující potřeboval změnit nebo čeho má nedostatek.

Trénovací sada textů klasifikátoru byla poskytnuta Ostravskou univerzitou. Tato sada obsahuje 180 jedno či dvouvětných úryvků textu zařazených do čtyř definovaných kategorií.

Dav vytvářející texty ke kategorizaci a zároveň hodnotící správnost kategorizace byl v této práci tvořen převážně studenty Lékařské fakulty OSU. Vkládání příspěvků probíhalo pomocí jednoduchého formuláře se vstupy pro text příspěvku a název příspěvku. Hodnocení kategorizace bylo prováděno neprodleně po samotné kategorizaci, kdy uživatel sám určil správnost kategorizace a v případě chyby také definoval kategorii správnou. V takovémto případě byl fakt o chybě zanesen do DB.

Kromě samotné kategorizace bylo rovněž cílem práce navrhnout postup pro zvýšení kvality kategorizace pomocí crowdsourcingu. Postup zkvalitnění kategorizace byl navrhnut pomocí rozšiřování trénovací sady klasifikátoru. Sada byla rozšiřována převážně chybně zkategorizovanými příspěvky.

Dosud bylo přes rozhraní přidáno osm příspěvků - dva příspěvky do každé kategorie. Autorem všech příspěvků je však jeden uživatel a tudíž použitý slovník všech příspěvků je podobný.

Texty byly přidány a zpracovány ve dvou fázích. V první fázi byl vložen jeden příspěvek do každé kategorie. Tyto příspěvky byly zkategorizovány s nulovou úspěšností, což je připisováno zejména odlišné povaze textů vložených příspěvků od textů v trénovací sadě, především ve smyslu jejich rozsahu. Po synchronizaci dat byla provedena druhá fáze, při níž byl opět přidán jeden příspěvek do každé kategorie. U těchto příspěvků se již úspěšnost kategorizace zvýšila na 50 %. I když je zde znatelné zlepšení kategorizace a předpokládané učení se klasifikátoru, není možné z tak malého množství dat vyvozovat konkrétní závěry. Další překážkou je také jediný autor všech přidaných příspěvků. Při použití odlišné slovní zásoby je možné předpokládat pokles úspěšnosti. Z těchto důvodů je tedy velice důležité trénovací sadu co nejvíce rozšířit, jak do množství, tak do různorodosti slovní zásoby.

Během práce byla rovněž stanovena hypotéza o zvýšení kvality kategorizace v případě použití definovaných klíčových slov. Pro potvrzení hypotézy byli uživatelé rozhraní rozděleni do dvou skupin - s klíčovými slovy a bez nich. Jelikož byly všechny nasbírané texty vloženy jediným uživatelem, nelze na základě dosud nasbíraných dat hypotézu potvrdit ani vyvrátit.

Studenti byli instruováni jak s rozhraním pracovat pomocí sepsaného návodu. Všechny pokyny byly rovněž přidány přímo do rozhraní pod záložkou O projektu. Instrukce poskytnuté studentům jsou přiloženy k této práci v příloze B.

Závěr

Cílem této práce bylo navrhnout a implementovat prototyp klasifikátoru textových dokumentů na základě jejich podobnosti s následným využitím crowdsourcingu za účelem zkvalitnění kategorizace.

Po analýze algoritmů určených pro kategorizaci textových dokumentů a nastudování metody spolupráce davu, byl navrhnut a implementován klasifikátor textových dokumentů založený na principu N-gramů. Tento klasifikátor byl vybrát především kvůli jeho nezávislosti na jazyce analyzovaného textu a také kvůli poměrně jednoduché implementaci. Klasifikátor byl následně propojen s vytvořeným crowdsourcingovým rozhraním přes REST rozhraní klasifikátoru, které je crowdsourcingovým rozhraním dotazováno pomocí technologie cURL. Jádrem crowdsourcingového rozhraní je redakční systém WordPress.

Efektivita vytvořeného klasifikátoru byla ověřena na dvou datových sadách a následně pomocí vytvořeného rozhraní. Na první datové sadě dosáhl klasifikátor výborných výsledků, čímž byla potvrzena nezávislost klasifikátoru na použitém jazyce. U druhé sady, sady s psychologickými texty, dosáhla kategorizace rovněž uspokojivých výsledků, přičemž nižší úspěšnost byla přičítána především tenkým hranicím mezi jednotlivými kategoriemi.

Ověření správnosti kategorizace přes implementované rozhraní bylo ponecháno na samotném uživateli, který ji sám určuje. Pro zkvalitnění kategorizace byl navrhnut postup rozšiřování trénovací sady klasifikátoru a to konkrétně o příspěvky, u kterých byla kategorizace označena za chybnou.

S ohledem na malé množství získaných příspěvků není možné vyvodit konečné závěry o úspěšnosti klasifikátoru. Nicméně i na taktovémto malém vzorku dat je vidět rostoucí trend úspěšnosti. Tyto výsledky však nelze zobecňovat i vzhledem k faktu, že autorem textů je pouze jeden uživatel a tudíž použitá slovní zásoby je ve všech příspěvcích stejná.

V práci byla rovněž stanovena hypotéza o zvýšení úspěšnosti kategorizace v případě využití klíčových slov vypočítaných z dostupných dat. Vzhledem k nízkému množství doposud získaných příspěvků a především vzhledem k jedinému autorovi všech příspěvků, nelze momentálně tuto hypotézu potvrdit ani vyvrátit.

Vytvořený klasifikátor i crowdsourcingové rozhraní by mělo také sloužit jako základ pro připravovaný projekt mezi Ostravskou univerzitou v Ostravě a Vysokou školou báňskou - Technickou univerzitou.

Další možný postup v práci by spočíval především v potvrzení či vyvrácení stanovené hypotézy o klíčových slovech a ověření stability úspěšnosti kategorizace při využití odlišné slovní zásoby. Systém je také možné obohatit např. o realtime synchronizaci dat, která je nyní prováděna v tříhodinových intervalech.

Literatura

[1] KARMAN, S. Senthamarai; RAMARAJ, N. Similarity-Based Techniques for Text Document Classification. Int. J. SoftComput, 2008, 3.1: 58-62.

[2] OPITKA, P.; ŠMAJSTRLA, V. „PRAVDĚPODOBNOST A STATISTIKA," 2013. [Online]. Available: `https://homen.vsb.cz/~oti73/cdpast1/KAP02/PRAV2.HTM`. [Přístup získán 4. 3. 2018].

[3] „Tf-idf :: A Single-Page Tutorial - Information Retrieval and Text Mining," [Online]. Available: `http://www.tfidf.com/`. [Přístup získán 25. 12. 2017].

[4] LANDAUER, Thomas K.; FOLTZ, Peter W.; LAHAM, Darrell. An introduction to latent semantic analysis. Discourse processes, 1998, 25.2-3: 259-284.

[5] HÁJEK, Petr, et al. Možnosti využití přístupu indexování latentní sémantiky při předpovídání finančních krizí. POLITICKÁ EKONOMIE, 2009, 6: 755.

[6] „Support Vector Machines (SVM)," TIBCO Software Inc, [Online]. Available: `http://www.statsoft.com/Textbook/Support-Vector-Machines`. [Přístup získán 28. 12. 2017].

[7] ŽIŽKA, J. „Studijní materiály předmětu FI:PA034," [Online]. Available: `https://is.muni.cz/el/1433/podzim2006/PA034/09_SVM.pdf`. [Přístup získán 29. 12. 2017].

[8] CAVNAR, William B., et al. N-gram-based text categorization. Ann arbor mi, 1994, 48113.2: 161-175.

[9] HABROVSKÁ, P. „Vybrané kapitoly z počítačového zpracování přirozeného jazyka," 2010. [Online]. Available: `http://www.inflow.cz/kratce-o-zpracovani-prirozeneho-jazyka`.

[10] SCAGLIARINI, L.; VARONE, M. „Natural language processing and text mining," 11 Duben 2016. [Online]. Available: `http://www.expertsystem.com/natural-language-processing-and-text-mining/`. [Přístup získán 15. 12. 2017].

[11] KODIMALA, Savitha. Study of stemming algorithms. 2010.

[12] RISUENO, T. „The difference between lemmatization and stemming," 28. 1. 2018. [Online]. Available: `https://blog.bitext.com/what-is-the-difference-between-stemming-and-lemmatization/`. [Přístup získán 4. 3. 2018].

[13] ŠMERK, P.; RYCHLÝ, P. „Majka – rychlý morfologický analyzátor," 2009. [Online]. Available: `https://www.muni.cz/vyzkum/publikace/935762`. [Přístup získán 15. 12. 2017].

[14] ESTELLÉS-AROLAS, Enrique; GONZÁLEZ-LADRÓN-DE-GUEVARA, Fernando. Towards an integrated crowdsourcing definition. Journal of Information science, 2012, 38.2: 189-200.

[15] SCHENK, Eric; GUITTARD, Claude. Crowdsourcing: What can be Outsourced to the Crowd, and Why. In: Workshop on Open Source Innovation, Strasbourg, France. 2009.

[16] AITAMURTO, Tanja; LEIPONEN, Aija; TEE, Richard. The promise of idea crowdsourcing–benefits, contexts, limitations. Nokia Ideasproject White Paper, 2011, 1: 1-30.

[17] KALSI, M. „Crowdsourcing through Knowledge Marketplace,“ 3. 3. 2009. [Online]. Available: `http://blog.spinact.com/knowledge_as_a_service/2009/03/crowdsourcing-through-knowledge-marketplace-.html`. [Přístup získán 2018 3. 4.].

[18] KAUFMANN, Nicolas; SCHULZE, Thimo; VEIT, Daniel. More than fun and money. Worker Motivation in Crowdsourcing-A Study on Mechanical Turk. In: AMCIS. 2011. p. 1-11.

[19] KEARNS, K. „9 Great Examples of Crowdsourcing in the Age of Empowered Consumers,“ 10. 7. 2015. [Online]. Available: `http://tweakyourbiz.com/marketing/2015/07/10/9-great-examples-crowdsourcing-age-empowered-consumers/`. [Přístup získán 10. 3. 2018].

[20] CIGÁNKOVÁ, B. Publikování obsahu webových stránek na Facebooku. Ostrava, 2016. Bakalářská práce. Ostravská univerzita v Ostravě.

[21] ROUSE, M. „Content management system (CMS),“ 6. 2016. [Online]. Available: `http://searchcontentmanagement.techtarget.com/definition/content-management-system-CMS`. [Přístup získán 11. 3. 2018].

[22] „Usage of content management systems for websites,“ W3Techs.com, 10. 3. 2018. [Online]. Available: `https://w3techs.com/technologies/overview/content_management/all`. [Přístup získán 11. 3. 2018].

[23] „Theme Development « WordPress Codex,“ [Online]. Available: `https://codex.wordpress.org/Theme_Development`. [Přístup získán 2. 4. 2018].

[24] „ Child Themes « WordPress Codex,“ [Online]. Available: `https://codex.wordpress.org/Child_Themes`. [Přístup získán 2. 4. 2018].

[25] „Writing a Plugin « WordPress Codex,“ [Online]. Available: `https://codex.wordpress.org/Writing_a_Plugin`. [Přístup získán 2. 4. 2018].

[26] CHLAPEK, D.; KLÍMEK, J.; KUČERA, J.; NEČASKÝ, M. „Otevřená a propojitelná data – metodiky, postupy, nástroje a praxe,“ DATAKON 2014, pp. 17-37.

[27] HAŠEK, R. „LINEÁRNÍ ALGEBRA A GEOMETRIE - KMA/LA2,“ 15 2. 2018. [Online]. Available: http://home.pf.jcu.cz/~hasek/LA2/P11/ObecnaRovniceNadroviny.pdf. [Přístup získán 10. 3. 2018].

[28] VRL, NICTA. An unsupervised approach to domain-specific term extraction. In: Australasian Language Technology Association Workshop 2009. 2009. p. 94.

A Adresářová struktura přiloženého disku

- text_dp_CIG0032 - text diplomové práce
- zdrojove_kody_CIG0032 - složka obsahující zdrojové kódy klasifikátoru a crowdsouringového rozhraní

B Instrukce k práci s crowdsourcingovým rozhraním poskytnuté studentům OSU

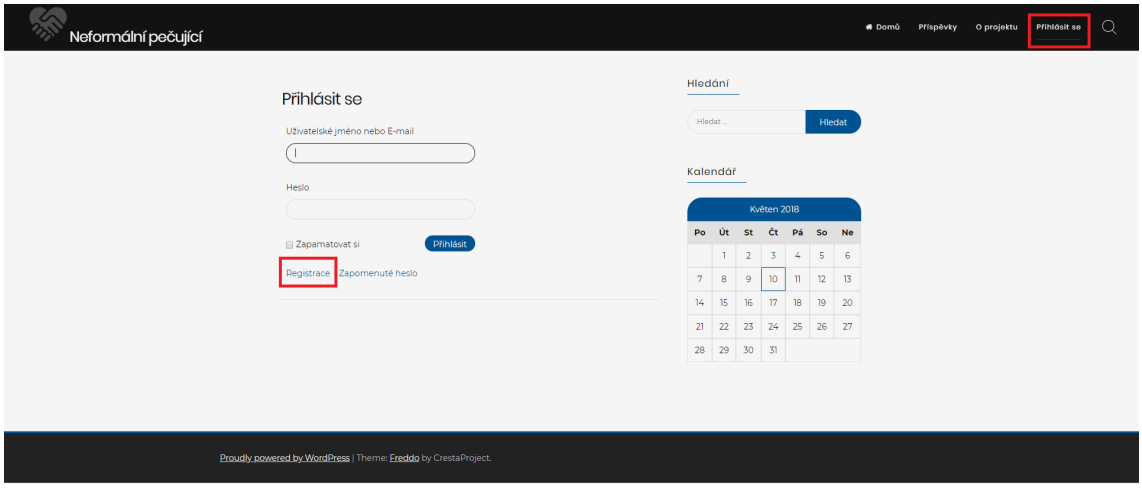
V rámci připravovaného projektu mezi OSU a VŠB a přidružené diplomové práce byla vytvořena webová stránka určená neformálním pečujícím. Tato webová stránka umožňuje vkládat uživatelům příspěvky na daná témata, s možností zobrazit příspěvky podobné. Určování podobnosti příspěvků je řešeno pomocí strojové kategorizace, která byla implementována jako strojové učení v rámci zmíněné diplomové práce. Jelikož se jedná o strojovou kategorizaci je potřeba mít, pro dostatečnou úspěšnost kategorizace, co možná nejširší sadu textů, s kterými jsou následně vložené příspěvky při kategorizaci porovnávány. Z tohoto důvodu bychom Vás rádi požádali o pomoc při rozšiřování datové sady (textů) určené pro učení klasifikátoru, a také při kontrole správnosti kategorizace. Co je považováno za správnou a co za špatnou kategorizaci bude popsáno dále v textu. V této fázi budou texty zařazovány do čtyřech kategorií:

- dopady a přínosy péče na neformální pečující,
- motivace neformálních pečujících k péči,
- podpora neformálních pečujících,
- potřeby neformálních pečujících.

Následující text také obsahuje instrukce pro práci s webem. Web se nachází na adrese cmppecujici.cs.vsb.cz. Pro vkládání příspěvku je nutné, aby se uživatel registroval, čímž chceme zamezit anonymním uživatelům vkládat anonymní texty. Pro registraci je potřeba zadat uživatelské jméno, e-mail a heslo. Titulní strana webu vypadá následovně:

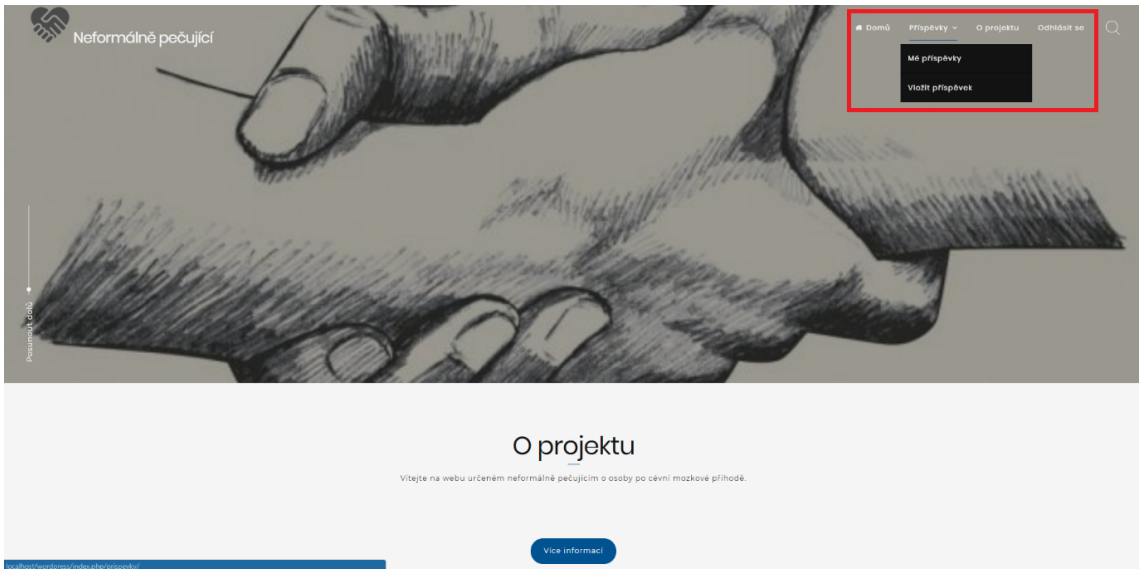
Obrázek 16: Titulní stránka webu

Na obrázku jsou znázorněné dvě části webu. Označení vpravo nahoře znázorňuje menu webu, kde se nachází i odkaz k přihlášení. Stránka s přihlášením obsahuje také možnost registrace či obnovení hesla (viz další obrázek). Označení spodní znázorňuje část O projektu, která obsahuje zkrácené informace o projektu a instrukce k práci s webem. Kompletní instrukce jsou přístupné přes tlačítko Více informací nebo přes záložku O projektu v menu stránky.

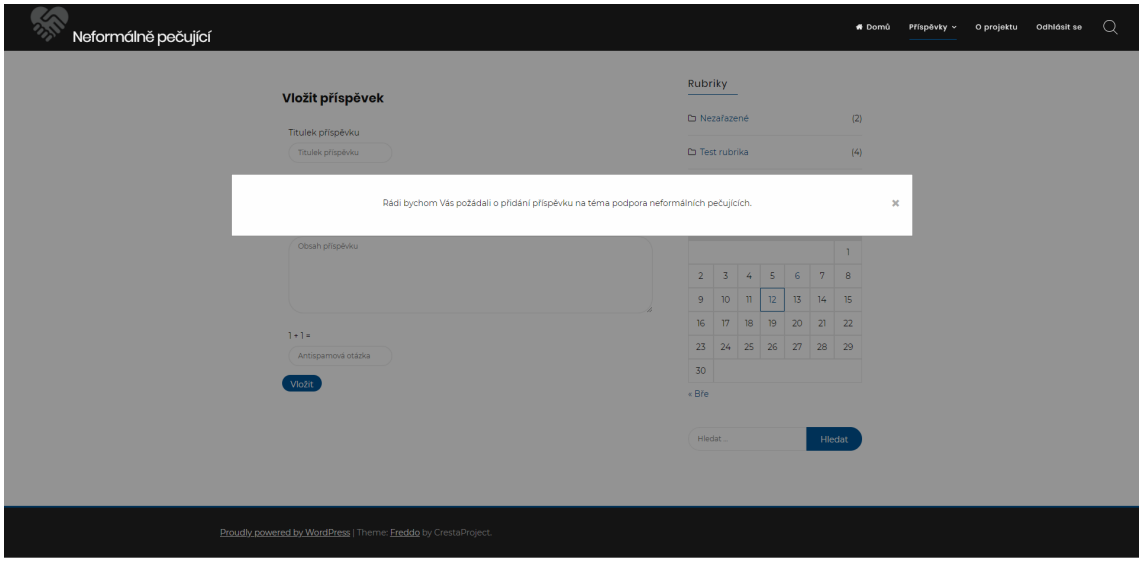


Obrázek 17: Registrace na web

Po přihlášení je již možné vkládat příspěvky. Abychom dosáhli co nejvyváženějšího počtu textů v jednotlivých kategoriích bude každému uživateli téma přiřazeno a oznámeno, společně s potřebnými instrukcemi, před každým vkládáním příspěvku. Tématem se myslí zaměření textu, které má být jeho dominantním sdělením/obsahem. Pokud uživatel vloží příspěvek na přiřazené téma a bude chtít přidat příspěvek další, bude mu přiděleno téma nové.

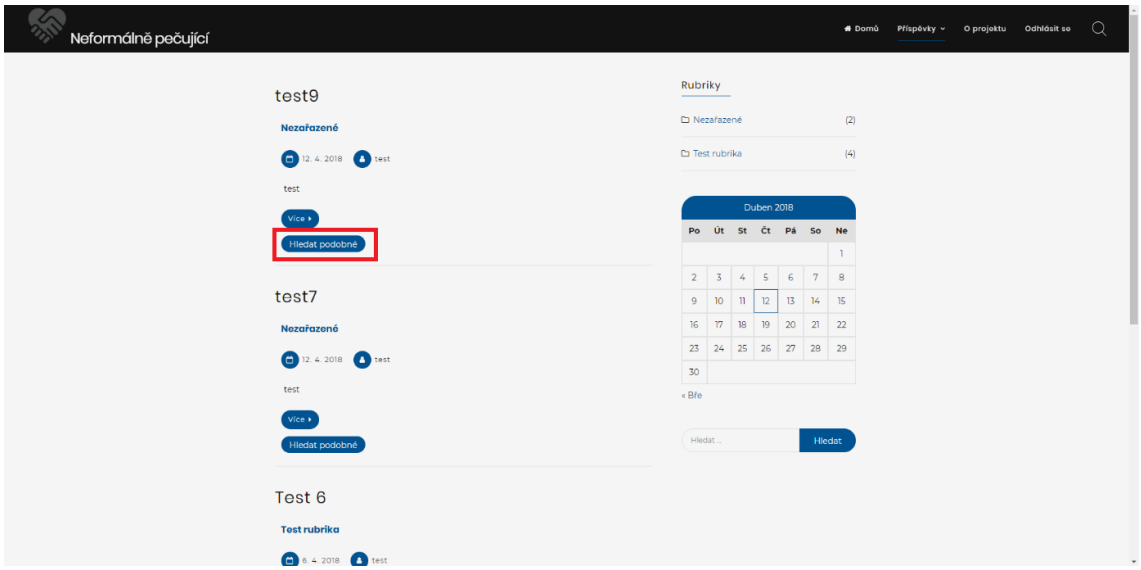


Obrázek 18: Navigace pro vkládání příspěvku



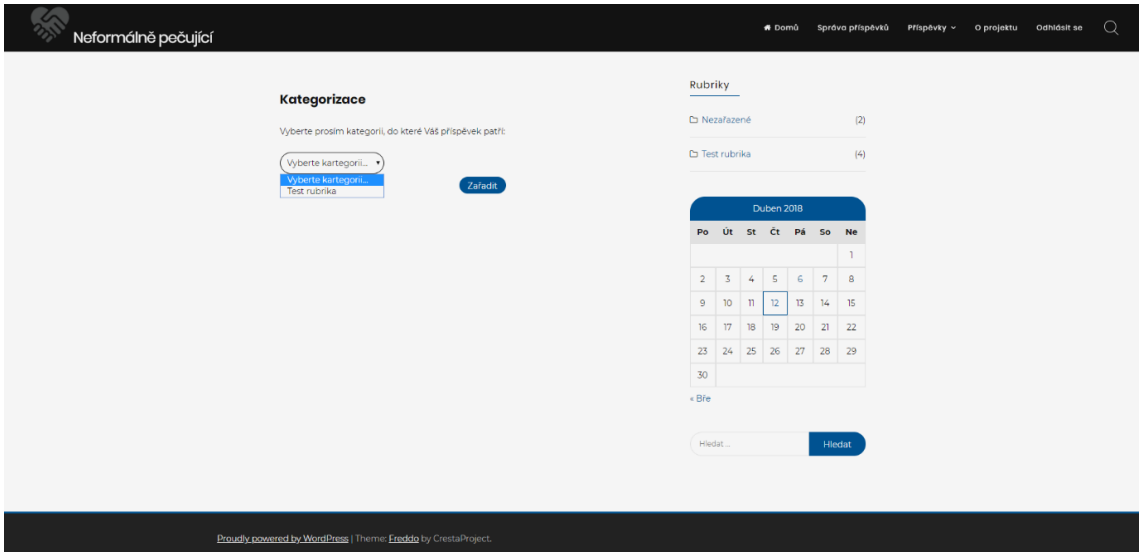
Obrázek 19: Přiřazení kategorie uživateli před vložením příspěvku

Po vložení příspěvku je uživatel přesměrován na stránku s přehledem příspěvků, kde nalezne tlačítko Hledat podobné (u nezkategorizovaných textů) nebo Zobrazit podobné (u již zkategorizovaných textů). Následující text Vás provede kategorizací, během které sám uživatel určí správnost či nesprávnost kategorizace. Správná kategorizace je taková, jejíž výsledná kategorie se shoduje s kategorií přiřazenou uživateli před začátkem vkládání příspěvku (viz obrázek výše). Pokud se výsledná kategorie a kategorie uživateli přiřazená nerovnají, může uživatel označit kategorizaci za nesprávnou a zároveň určit kategorii správnou. Takovéto, špatně zařazené, texty budou přidávány do sady textů, které klasifikátor používá k určení podobnosti, což by mělo postupně vést k vyšší úspěšnosti kategorizace.



Obrázek 20: Navigace pro spuštění kategorizace příspěvku

Kategorizace: Po kliknutí na tlačítko Hledat podobné bude spuštěna kategorizace a výsledná kategorie bude uživateli zobrazena. Uživatel určí, kliknutím na Ano/Ne, zda je určená kategorie správná či nikoliv. V případě, že je kategorizace potvrzena, je uživatel přesměrován na stránku s texty dané kategorie. V případě, že je kategorizace označena za nesprávnou, je uživatel vyzván k určení správné kategorie. Po tomto určení je uživatel opět přesměrován na stránku s texty kategorie.



Obrázek 21: Korekce kategorizace

Rádi bychom požádali všechny uživatele, aby ihned po vložení příspěvku spustili pro daný příspěvek kategorizaci a zároveň, aby spouštěli kategorizaci pouze u svých příspěvků, čímž bychom chtěli zamezit nesprávné korekci kategorizace.

Předem bychom Vám chtěli poděkovat za Váš čas a za Vaši pomoc. Výsledky úspěšnosti klasifikátoru Vám budou, po sesbírání dostatku dat, sděleny. Navíc bude naučený klasifikátor základem pro společný projekt OSU a VŠB podporující sdružení neformálních pečujících po CMP, ustaveném při LF OSU.