Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra kybernetiky

BAKALÁŘSKÁ PRÁCE

PLZEŇ 2013 **Červený Martin**

P R O H L Á Š E N Í

Předkládám tímto k posouzení a obhajobě diplomovou/bakalářskou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem bakalářskou/diplomovou práci vypracoval(a) samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne

.....

Anotace

Tato bakalářské práce se zabývá použitím metod vyhledávání klíčových slov v textu v českém jazyce. Jejím cílem je nalézt vhodnou metodu pro ohodnocení slov nějakou matematickou funkcí, a jako klíčová vybrat ta slova, která mají tuto funkci nejvyšší ze všech. Konkrétně se práce zabývá aplikací těchto metod ve dvou klasifikátorech. První určuje kategorie, druhý téma neznámého článku, přičemž druhý je schopen objevování nových témat. Jejím cílem je nalézt nejlepší možnou metodu pro oba tyto klasifikátory pro dostupnou množinu dat. Výsledky jsou poté porovnány s Bayesovským klasifikátorem a výsledky jiné, původně navrhované metody pro anglický jazyk.

Klíčová slova

klasifikace kategorií, detekce tématu, vyhledávání klíčových slov

Annotation

This work examines keyword extraction methods applied to a text written in Czech language. It's goal is to find a suitable method that evaluates words with some kind of mathematical function, and to pick as key those words, that have the highest value of the function of all the words. Specifically the work examines application of these methods in two classifiers. First one for category classification, second one for theme classification and exploration of new themes. It's goal is to find the best method for both of these classifiers given the availiable data. Results are then compared with a Bayesian classifier and results of other, originally suggested method for English language.

Key words

category classification, theme detection, keyword extraction

Obsah

1 Úvod

Klasifikace textu je poměrně složitý problém. Klasický přístup měření příznaků není příliš vhodný, neboť dimenze vektoru je příliš velká. Klasické metody selekce příznaků nejsou vhodné ze stejného důvodu. Obzvláště, pokud je hlavním kritériem rychlost klasifikace (případně i trénování klasifikátoru). Autoři David Bracewell a spol. (1) navrhli postup trénování a klasifikace pro krátké články. Také navrhli algoritmus pro objevování nových témat (tříd) a jejich okamžité začlenění do klasifikátoru během klasifikace, tedy po průběhu trénování. Úkolem této práce je implementovat všechny tyto postupy a prozkoumat jejich účinnost při použití různých metod hledání klíčových slov, které nahrazují klasickou selekci příznaků.

Metody vyhledávání klíčových slov mají za úkol vybrat z textu maximální množství informace při vybrání minimálního možného počtu slov. Touto oblastí se zabývá obor *Information Retrieval,* vyhledávání informací. V klasickém pojetí je snahou vybrat klíčová slova, která text shrnou, a důležité je aby se výsledky metody shodovaly se slovy vybranými učitelem – člověkem, jako například klíčová slova této práce. Tyto metody mají totiž člověka v této zdlouhavé práci zastoupit. Používají se také obvykle na delší texty. Pro práci bylo nutné vyhledat metody použitelné pro krátké texty, avšak na přístupnost člověku není přihlíženo, neboť metody jsou určeny pro klasifikátor - program.

Metody jsou porovnávány mezi sebou a s Bayesovským klasifikátorem. Byly provedeny tři experimenty. První experiment je založen na klasické množině dat při přiřazování do více tříd, bez prvku objevování nových tříd. Druhý je pro přiřazování do jedné třídy s objevováním nových. Třetí experiment se zabývá speciálně prvkem objevení nové třídy.

2 Formulace problému

2.1 Klasifikace

Klasifikace je schopnost rozpoznávání předmětů (či jevů), jejich zařazení do určité třídy. Jelikož předměty v reálném světě jsou příliš složité, a naše znalosti o nich jsou omezené, můžeme na nich pouze měřit určitý vektor příznaků $[x_1, x_2, \ldots, x_n]$ (příkladem příznaku může být třeba délka článku). Tento vektor příznaků se nazývá obraz (X) a jelikož popisuje konkrétní reálný předmět, používá se také termín rozpoznávání obrazů.

$$X = [x_1, x_2, \ldots, x_n] \tag{1}$$

Obraz X je potřeba zařadit do jedné nebo více z tříd $[w_1 \ldots w_k]$

$$w_x \in [w_1, w_2, \ldots, w_k] \tag{2}$$

Klasifikátor je pak funkce, která neznámému obrazu X přiřadí třídu w_x.

$$\gamma : X \rightarrow w_x \tag{3}$$

Klasifikátory jsou reprezentovány v této práci programy s koncovkou "Klas".

K vytváření klasifikátoru obecně existují dva přístupy: ruční psaní pravidel (která napíše nějaký expert) která mají charakter booleovských výrazů. Tak lze dojít ke klasifikátoru "přímo". Druhý přístup je užití technik strojového učení. Tato práce se zabývá tímto způsobem. Ke klasifikátoru lze dospět tak, že se na trénovací množinu T aplikuje trénovací metoda Γ:

$$\Gamma(T) = \gamma \tag{4}$$

Trénovací metody jsou v této práci reprezentovány programy s koncovkou "Tren".

Trénovací množina se běžně skládá z objektů:

$$T = ((w_a, X_a), (w_b, X_b) \ldots) \tag{5}$$

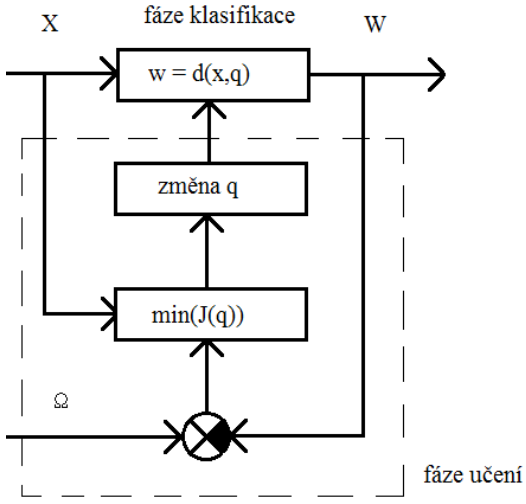
Které jsou tvořeny dvojicemi obraz X a třída w do které náleží. Tato množina je tedy vstupem trénovací metody Γ, jejímž výstupem je klasifikátor. Aplikací různých trénovacích množin na stejnou metodu lze dojít k rozdílně natrénovaným klasifikátorům.

2.1.1 Učení s učitelem

Toto je případ, kdy dvojice v trénovací množině jsou známé (připraveny učitelem - člověkem) a určují rozhodování klasifikátoru, tedy se jedná o učení s učitelem. Pokud je známá celá trénovací množina lze určit nastavení klasifikátoru analyticky. Příkladem takto nastavovaných klasifikátorů jsou: Bayesův klasifikátor, klasifikátor podle minimální vzdálenosti a klasifikátor podle (K) nejbližšího souseda.

Problémem je odhad apriorní pravděpodobnosti tříd, který je nutno odhadnout z (nezávisle vybrané) trénovací množiny.

Trénovací množina je obvykle rozsáhlá. Obvykle je tak výhodné předkládat dvojice klasifikátoru postupně. Takový systém se nazývá učící se klasifikátor:



Obrázek 1: Učící se klasifikátor

Učící se klasifikátor má dva vstupy - obraz (X) a informaci od učitele Ω (správná třída) a jeden výstup, třídu W (určenou k obrazu X). q je pak parametr a $J(q)$ je střední ztráta na množině všech Ω (všech možných tříd). Je třeba najít takový parametr q aby $J(q)$ byla minimální. Existují dvě fáze činnosti učícího se klasifikátoru:

1) Fáze učení – klasifikátoru jsou předkládány dvojice $[X_k, \Omega_k]$ z trénovací množiny, klasifikátor porovná výstup w s požadovaným chováním Ω a nastaví svůj parametr q tak aby pro k jdoucí k nekonečnu bylo q optimální (poskytuje minimální střední ztrátu $J(q)$).

2) Fáze klasifikace – při této fázi se nepoužívají části v obrázku ohraničené přerušovaným obdélníkem. Využívá se zkušeností z předchozí fáze uložených v parametru q a klasifikátor se chová jako jednoúčelový automat.

Funkce $J(q)$ obsahuje dvě pravděpodobnosti: apriorní pravděpodobnost třídy a podmíněnou pravděpodobnost zda obraz náleží do dané třídy. Obě je nutné odhadnout, přičemž apriorní pravděpodobnost tříd se odhaduje z trénovací množiny. Odhad podmíněné pravděpodobnosti, zda obraz náleží do dané třídy, dělí metody na parametrické a neparametrické.

Parametrické

Za předpokladu znalosti tvaru rozdělení pravděpodobnosti (u všech tříd je stejný), zda obraz náleží do dané třídy, tyto metody určují rozdílné parametry těchto rozdělení. Patří sem například metoda momentů a metoda maximální věrohodnosti.

Neparametrické

Tyto metody odhadují tvary rozdělení pravděpodobnosti pro každou třídu zvlášť. Příkladem je metoda histogramu a metody založené na přímé minimalizaci ztrát (Rosenblattův algoritmus, metoda konstantních přírůstků).

Příkladem učení s učitelem je klasifikace dokumentů v této práci. Obrazy *X* jsou dokumenty a třídy *w* jsou kategorie nebo témata.

2.1.2 Učení bez učitele

Obecně nemusí být známy třídy náležící obrazům v trénovací množině, dokonce ani počet možných tříd. Tento případ se nazývá učení bez učitele. Metody učení bez učitele naleznou shluky obrazů, které jsou vzájemně blízké. Metody se dělí na hiearchické a nehiearchické:

Nehiearchické metody

Hlavní skupinu těchto metod tvoří optimalizační metody podle nějakého kritéria. Nejčastěji se používá suma kvadrátů odchylek vzdáleností obrazů od středů shluků do nichž náleží. Patří mezi ně například K-průměrová metoda (nebo také MacQueenův algoritmus) a metoda Iterativní optimalizace.

Hiearchické metody

a) Aglomerativní

Tyto metody určí v prvním kroku jako shluky jednotlivé obrazy a postupně je dále shlukují. Příkladem může být metoda shlukové hladiny.

b) Divisní

Postupují opačně než aglomerativní metody – v prvním kroku jsou všechny obrazy v jednom shluku a ten se postupně rozděluje. Příkladem těchto metod jsou: jednoprůchodový heuristický algoritmus hledání shluků, metoda řetězové mapy a metody rovnoměrného nerovnoměrného binárního dělení.

Příkladem použití učení bez učitele může být hledání klíčových slov, kde obrazy *X* jsou dokumenty a třídy *w* jsou klíčová slova.

2.2 Klasifikace textu

Klasifikaci textu (nebo také kategorizaci textu či klasifikaci témat, v této práci je používán termín klasifikace textu) lze rozlišit na manuální a strojovou. Manuální spočívá v použití lidské inteligence pro zařazení jednotlivých dokumentů do určitých kategorií, například knihovník dává knihy do určitých oddělení. Ve strojové klasifikaci místo lidské inteligence provádí stroj (počítač), čili se jedná o doménu oboru umělá inteligence, konkrétně odvětví strojového učení. Tato práce se zabývá strojovou klasifikací.

Úlohou klasifikace textu je k nějakému dokumentu D :

$$D=\left[x_{1,}x_{2,}\ldots,x_{n}\right] \tag{6}$$

Přiřazení jedné nebo více tříd (kategorií, témat) c:

$$c\in\left(c_{1,}c_{2,}\ldots,c_{k}\right) \tag{7}$$

Konkrétně pro tuto práci se v případě témat jedná o právě jednu třídu, v případě kategorií o více tříd, přičemž nemusí být také klasifikátorem vybrána žádná.

Problémem klasifikace obecně je volba vhodných příznaků. Pokud zvolíme všechny příznaky pro dokumenty stejné, je nutné použít celý slovník, včetně všech tvarů možných slov a také cizích slov. Dimenze takového problému by byla příliš velká a klasifikace by vedla k příliš velké časové náročnosti, z tohoto důvodu se v klasifikaci textu používá předzpracování, které například převede slova na předem daný tvar nebo na jejich kořen. I tak je ale dimenze stále velká.

V této práci se tento problém řeší tak, že každý dokument má uložený v počítači vektor pouze několika klíčových slov, přičemž tento vektor je jeho reprezentací (při zachování příznakového prostoru). Smyslem je vybrat z článku správná slova, obsahující hodně informace, a ignorovat nevýznamná slova, jako například spojky. Metody zabývající se tímto problémem se nazývají metody hledání klíčových slov (viz

následující kapitola).

Klasifikace textu také úzce souvisí s vlastnostmi textu. Například pokud klasifikátor využívá jazykový přístup (metody hledání klíčových slov založené na gramatických pravidlech), funguje pouze pro jeden jazyk. Oproti tomu pokud žádné jazykové znalosti nepoužívá, a metody jsou čistě statistické, lze jej pro různé trénovací množiny aplikovat na různé jazyky. Také obvyklá délka textu hraje významnou roli, kupříkladu v této práci bylo nutné zavrhnout některé metody hledání klíčových slov, vzhledem k jejich zaměření na dlouhé texty. Všechny metody byly trénovány a testovány na jedné množině dat, a to v českém jazyce.

Příkladem použití klasifikace textu v praxi je například označování spamu v e-mailové schránce. Uživatel, jakožto učitel, označuje klasifikátoru zprávy, které považuje za spam, a klasifikátor poté při příjmu nových zpráv určuje, jestli patří do kategorie "spam" nebo nepatří, a pokud ano, odkládá je do speciální složky.

2.3 Metody hledání klíčových slov

Metody hledání klíčových slov (dále metody hledání KS) jsou metody, které z textu vyberou určitý počet slov a určí je jako nejlepší shrnutí článku, tedy když si je čtenář přečte, měl by mít představu čím se článek zabývá. Kupříkladu každá moderní bakalářská práce má klíčová slova (včetně této). Jako klíčová slova by tak měla být vybrána slova s největším obsahem informace a rozhodně by tam neměla být slova typu "a" nebo "například", která se mohou vyskytnout v článku zabývajícím se jakoukoli problematikou. V případě bakalářských prací budou slova nejspíše v drtivé většině vybírána ručně samotnými autory. Problém však lze řešit strojově a ušetřit tak čas a lidské úsilí. Existuje několik přístupů k tomuto problému:

Statistický přístup

Je poměrně jednoduchý a nevyžaduje metodu hledání KS trénovat, což je velmi

výhodné. Využívá hlavně statistických informací, jako je počet slov v článku (článcích), jejich pozice, společný výskyt slov ve větách TF, IDF (viz dále) a podobně. Jsou rychlé a lze je použít pro různé jazyky a nevyžadují expertní znalost jazyka, proto jsou metody v této práci většinou statistického rázu.

Jazykový přístup

Vyžaduje expertní znalost jazyka, využívá větné skladby a vztahů větných členů ve větách, nebo dokonce vztahů vět v dokumentech. Spadají sem také různé jiné postupy jako použití *stoplistu*, seznamu často používaných slov (typického pro jazyk), které se tak jeho použitím odstraní. Tento přístup se hlavně používá pro zlepšení statistických metod, jako například v metodě použité původně v klasifikátorech implementovaných v této práci (ve verzi pro anglický jazyk) a také při předzpracování (skloňování, lemmatizace).

Přistup strojového učení

Klasická doména oboru umělá inteligence (viz kapitola 1.1), obrazy jsou dokumenty, třídy jsou klíčová slova, přičemž lze tříd přiřadit více. Tento přístup má tu výhodu, že může v případě učení s učitelem článku přiřadit klíčové slovo, které se v článku vůbec nevyskytuje. Což může být na druhou stranu také nevýhoda, pokud je přiřazeno slovo špatné. Statistický přístup ale takovéto věci není vůbec schopen. Další nevýhoda je nutnost trénování metody, a velký počet tříd (celá encyklopedie nebo slovník).

Kombinace přístupů

Téměř naprostá většina metod používá nějakou kombinaci výše zmíněných přístupů. Metody v této práci nejsou žádnou výjimkou, využívají předzpracování (jazykový přístup), i když jsou hlavně statistické.

Tato oblast se stále vyvíjí, jak ukazuje odkaz (6), který byl dokončen v průběhu vývoje této práce (2012). V dnešní době existují programy zabývající se touto problematikou, hlavně pro anglický jazyk, vyvinuté týmy expertů které si lze koupit a

využívat, jako například AlchemyAPI. Existuje i open-source (volně přístupný veřejnosti) projekt TexLexAn.

Metody použité v práci

Klíčovým kritériem pro vybírání metod je fakt, že nejsou určeny pro shrnutí informací pro člověka, ale pro stroj (klasifikátor). Z tohoto důvodu (a také z důvodu trénování metody hledání KS) byl zavržen přístup strojového učení, neboť stroj klíčová slova nevyskytující se v článku pro správné určení třídy nepotřebuje, naopak nesprávně určená by měla katastrofální důsledky. Učení bez učitele je však použito, každému slovu je přiřazena kriteriální funkce a vybírá se *k* slov s nejvyšší hodnotou.

Dalším kritériem je rychlost. To odrazuje od použití jazykového přístupu, jehož prostředky jsou obvykle pomalé. Další kritický důvod je že projekty z této oblasti obvykle vyžadují spolupráci více expertů a tak jeho použití přesahuje rámec možností této bakalářské práce.

Statistický přístup se tak jeví jako nejlepší možný, neboť je (relativně) rychlý a je vhodný pro určení podobnosti dokumentů a znalosti autora mu zcela postačují.

Následuje matematický popis metod jak jsou implementovány v práci:

Každá z metod *F* použitých v práci ohodnotí každé slovo $x_1....x_n$ z dokumentu *D* nějakou hodnotou $a_1...a_n$:

$$D=(x_{1....}\,x_n) \tag{8}$$

$$F(D)=(a_{1....}\,a_n) \tag{9}$$

Poté se provede výběr *k* slov s nejvyšším ohodnocením, přičemž s rostoucím *k* značně rostou výpočetní nároky, zatímco s klesajícím *k* klesá úspěšnost použitých klasifikátorů. V práci je vybrán kompromis mezi rychlostí a přesností pro $k = 30$ u všech metod. Například u metody TF-IDF došlo k nárůstu úspěšnosti klasifikace o 10% při $k = |D|$ (délka dokumentu), ale výpočetní doba se zvýšila více než dvacetkrát (z řádu minut do řádu hodin).

Většina metod nevyhledává klíčová slova pro dokumenty, ale pro víc dokumentů (danou třídu (kategorii, téma)). V tom případě vyhledají *k* klíčových slov pro danou množinu dokumentů.

2.3.1 TF

TF - *Term frequency*, neboli četnost výrazu, je nejjednodušší možná statistická metoda hledání klíčových slov. Počítá se u každého článku zvlášť. Pro každé slovo článku se spočte jeho výskyt. Nejčastější slovo se pak bere jako to s největším významem. Hlavním úskalím této metody je, že nejčastější slova jsou obvykle spojky a předložky jako například "a", "i", "z" a podobně, dále také tvary slovesa být jako "se" nebo "je" a také zájmena. Hodnotu je také vhodné normalizovat (zde délkou dokumentu). Vypočítá se tak jako:

$$TF(t,d)=\frac{f(t,d)}{|d|} \tag{10}$$

d je dokument (jmenovatel je jeho délka), *t* je slovo, a *f(t,d)* je četnost slova *t* v dokumentu *d*. Metoda přiřadí každému článku *k* klíčových slov, tedy jednotlivým třídám jich přiřadí nejméně *k* (obvykle více).

2.3.2 TF-IDF

TF je zkratka pro *term frequency* (viz 1.2.1), IDF pro *inverse document frequency,* převrácená četnost v dokumentech. Složka *Inverse document frequency* slouží k tomu, aby odstranila slabinu druhé složky, *term frequency*, častá slova. *Inverse document frequency* se počítá se pro určitou množinu dokumentů (množina *D*). Jestliže se nějaké slovo vyskytuje ve všech článcích v *D*, pak se téměř jistě jedná o slovo s nízkou informační hodnotou (třeba spojku) a je potřeba mu přiřadit nízkou váhu.

$$IDF(t)=\log\left(\frac{|D|}{|\{d\in D: t\in d\}|}\right) \tag{11}$$

Čitatel ve vzorci je počet všech dokumentů v množině, jmenovatel počet dokumentů v množině, kde se slovo *t* vyskytlo (nemůže být nula pro existující *t*). Celý zlomek je jedna (pokud bylo slovo *t* ve všech článcích), nebo více než jedna. Logaritmus jedné je nula, nehledě na jeho základ, který je tak volitelný. V této práci je použit základ deset.

Kombinace TF a IDF pak je součin obou hodnot:

$$TF-IDF(t,d,D)=TF(t,d)\cdot IDF(t,D) \tag{12}$$

TF-IDF je nejčastěji používaná statistická metoda v oboru vyhledávání informací. Lze ji také použít pro vektor příznaků jednotlivých dokumentů při klasifikaci nebo srovnávání dvou dokumentů. Její největší nevýhoda je závislost na množině *D*. Pokud v ní budou totiž dokumenty z jedné třídy (kategorie), je možné, že se ve všech (nebo téměř ve všech) bude vyskytovat nějaké slovo typické pro danou třídu (například "počítač") a nebude to nedůležitá spojka a podobně. Pak bude mít nízkou (nulovou) hodnotu IDF, a tudíž i TF-IDF. Z tohoto důvodu je obecně lepší používat jako *D* dokumenty z více různých tříd, nejlépe všech. Dalším aspektem množiny *D* je její velikost. Čím je větší, tím je lepší IDF.

2.3.3 Chí-kvadrát (χ^2)

Tuto běžnou statistickou metodu pro určení nezávislosti lze v problematice hledání klíčových slov použít dvěma způsoby. První možností je aplikace na pár kategorie-slovo (použito v této práci), druhá na pár slovo-slovo. Druhá možnost, včetně důvodu jejího zamítnutí, je rozebrána na konci této podkapitoly. Hodnota χ^2 se vypočítá následovně:

$$\chi^2(t,c)=\frac{N\cdot(N_{t,c}\cdot N_{\neg t,\neg c}-N_{t,\neg c}\cdot N_{\neg t,c})}{(N_{t,c}+N_{\neg t,c})\cdot(N_{t,c}+N_{t,\neg c})\cdot(N_{t,\neg c}+N_{\neg t,\neg c})\cdot(N_{\neg t,c}+N_{\neg t,\neg c})} \tag{13}$$

Kde:

t je dané slovo

c je daná kategorie

N je počet všech dokumentů v množině (trénovací)

$N_{t,c}$ je počet dokumentů z *N*, patřících do kategorie *c* kde se vyskytuje slovo *t*.

$N_{\neg t,c}$ je počet dokumentů patřících do kategorie *c,* a nevyskytuje se v nich slovo *t*.

$N_{t,\neg c}$ je počet dokumentů patřících do jiné kategorie než *c*, a vyskytuje se v nich slovo *t*.

$N_{\neg t,\neg c}$ je počet dokumentů nepatřících do *c* a nevyskytuje se v nich slovo *t*.

Z toho vyplývá, že pro trénování jsou potřeba i příklady jiných tříd (případně negativní příklady). Také nemůže být trénováno pro každou třídu zvlášť, ale je nutné trénovat pro více najednou. Nejlepší je logicky trénovat pro všechny najednou, neboť tak se vyzdvihnou slova, kterými se jednotlivé třídy navzájem liší. Takto je také tato metoda implementována v práci. Teoreticky by bylo možné počítat tuto hodnotu po dvojicích (případně trojicích a tak dále) vybraných třeba náhodně, a byla by tak rychlejší, avšak nedávala by tak dobré výsledky.

Největší nevýhodou této metody je to, že ji nelze použít pro vyhledávání klíčových slov u neznámých článků (není známo *c,* články jsou brány po jednom). Z toho vyplývá nutnost jejího zastoupení (v této práci metodou TF).

Co se dvojice slovo-slovo týče, zabývají se jím autoři odkazu (2). Metoda popisovaná v článku využívá vzájemného výskytu dvojic slov ve větách jako hlavní informace pro určení klíčových slov. Stačí jí informace z jednoho dokumentu (na rozdíl od dvojice kategorie-slovo nebo TF-IDF) a častých slov se zbavuje pomocí *stoplistu* - seznamu často používaných slov (například spojek *"a" "nebo"* a podobně). Její nevýhoda je však u krátkých textů, kde slova jednoduše nemají ani dost výskytů na to, aby se objevila více než třikrát spolu s dalším slovem. Úkolem této práce je nalézt metody hledání klíčových slov pro novinové články, které jsou většinou krátké (méně než dva tisíce slov), a proto bylo od implementace této metody po krátké analýze (většina článků neobsahuje dvojice více než třikrát) upuštěno.

2.3.4 Vzájemná informace (mutual information)

Tato metoda je podobná metodě χ^2 (viz 1.2.3), v tom, že je nutné ji aplikovat na celou trénovací množinu najednou. Stejně jako χ^2 ji lze použít na dvojice slovo-slovo nebo kategorie-slovo. Opět je použita pouze dvojice kategorie-slovo, ze stejných důvodů jako u χ^2 . Vzorec pro výpočet ohodnocení slov článku je následující:

$$MI(t,c)=\frac{N\cdot N_{t,c}}{(N_{t,c}+N_{\neg t,c})\cdot(N_{t,c}+N_{t,\neg c})} \tag{14}$$

Pro význam jednotlivých zkratek viz 1.2.3 .

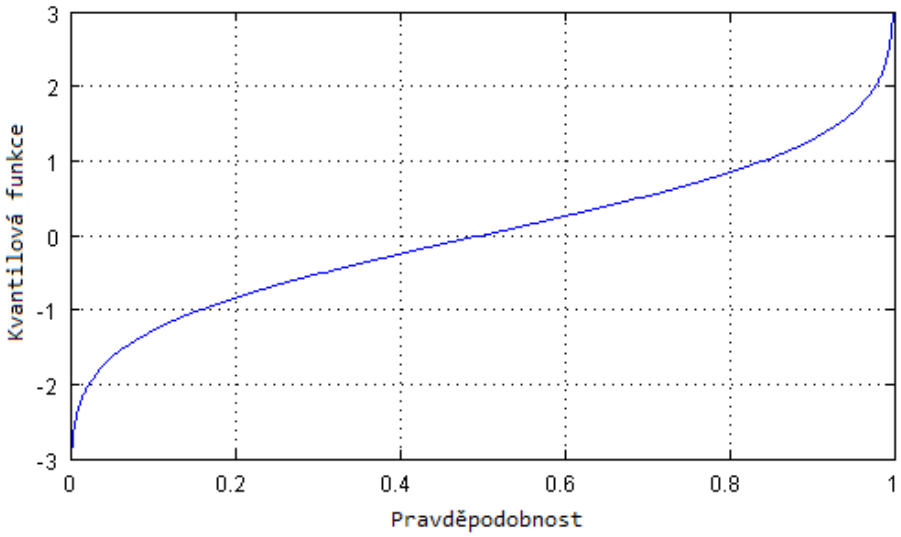
2.3.5 Normální rozdělení (bi normal separation)

Jak napovídá název, normální rozdělení, konkrétně jeho kvantilová funkce (inverzní distribuční funkce), může být použito pro ohodnocení slov článku. Touto metodou se podrobně zabývá autor odkazu (3). Metoda má stejné úskalí jako χ^2 nebo MI, je ji opět nutné aplikovat na celou trénovací množinu najednou a na dvojici kategorie-slovo. Má však další nevýhodu, konkrétně kvantilovou funkci. Tu totiž nelze vyjádřit analyticky, je nutné ji aproximovat (součtem polynomů). Následuje vzorec pro výpočet ohodnocení slov:

$$biNormal(t,c)=\left|F^{-1}\left(\frac{N_{t,c}}{N_{t,c}+N_{t,\neg c}}\right)-F^{-1}\left(\frac{N_{\neg t,c}}{N_{\neg t,c}+N_{\neg t,\neg c}}\right)\right| \tag{15}$$

Pro význam zkratek N viz 1.2.3 . F^{-1} je kvantilová funkce normálního rozdělení (při střední hodnotě $\mu = 0$ a rozptylu $\sigma^2 = 1$ čili standardní).

Hodnoty funkce v nule a v jedné jsou rovny mínus nekonečno respektive plus nekonečno, řešení spočívá v tom, že hodnoty vstupující do funkce jsou ohraničeny hodnotami: [0,0005 ; 0,9995]. Algoritmus pro výpočet této funkce byl převzat z odkazu (4), který využívá kombinaci čtyř polynomů různých stupňů.



Obrázek 2: Vykreslení aproximace kvantilové funkce

2.3.6 Morfologická metoda

Tato metoda je podrobně popisována v odkazu (5). Používá velmi silně předzpracování a rozdělení slov na *noun phrases,* jmenné fráze (dále NP) a jejich vyčlenění z textu. NP je podstatné nebo přídavné jméno a větné členy, které jsou k němu vázané. Obecně může jít o jedno nebo více slov. Podle odkazu (5) obsahují NP nejvíce informace v textu (odkaz pracuje s anglickým jazykem).

Po vybrání NP z dokumentu se ke každé vypočítá ohodnocení podle následujícího vztahu:

$$score(NP) = \frac{UF(NP) \cdot NPF(NP)}{|NP|} \tag{16}$$

$|NP|$ je počet slov v dané NP, *NPF noun phrase frequency* představuje množství výskytů dané NP v článku, a *UF unigram frequency* se vypočte podle následujícího vztahu:

$$UF = \sum_{i=1}^{|NP|} TF(ti, d) \tag{17}$$

TF(ti,d) ve vzorci jsou *term frequency* jednotlivých slov z NP pro daný

dokument *d*. Poté co jsou NP ohodnoceny dojde k jejich pseudoshlukování. S metodami shlukování *obrazů* z oboru umělá inteligence tato metoda nemá nic společného, jde totiž o shlukování *NP* (v rámci jednoho dokumentu). Pro jednoduchost je ale v této kapitole používán termín shluk. Myšlenka je prostá: NP které mají společné slovo jsou umístěny do stejného shluku. Provedení probíhá ve třech etapách:

1. Jednoslovné NP jsou umístěny do samostatných shluků
2. Víceslovné NP které lze zařadit jsou přidány do příslušných shluků
3. Zbylé NP jsou umístěny do samostaných shluků

Následuje výběr N shluků s nejvyšším ohodnocením, ohodnocení shluku se vypočte takto:

$$score(shluk) = \frac{\sum_{i=1}^{|shluk|} score(NPi)}{|shluk|} \tag{18}$$

|*shluk*| ve vzorci je počet NP v daném shluku a suma je součet jejich příslušných ohodnocení. Z těchto N shluků je vybrán jeden představitel, a to nejkratší slovo, jenž je vybráno jako klíčové. Dojde tak k výběru N klíčových slov.

Metoda je zde uváděna proto, že i když není implementována v této práci, byla použita v anglické verzi metod v této práci implementovaných (1), a to jako jediná. I když pseudoshlukování je možné naprogramovat relativně snadno, extrakce NP v českém jazyce je velmi obtížná jazykovědná překážka, přesahující rámec této práce.

2.3.7 Experimentální metoda

Různé experimenty v průběhu vypracování práce daly většinou vzniknout nepříliš dobrým metodám. Tato podala nejlepší výsledky v rozumném čase.

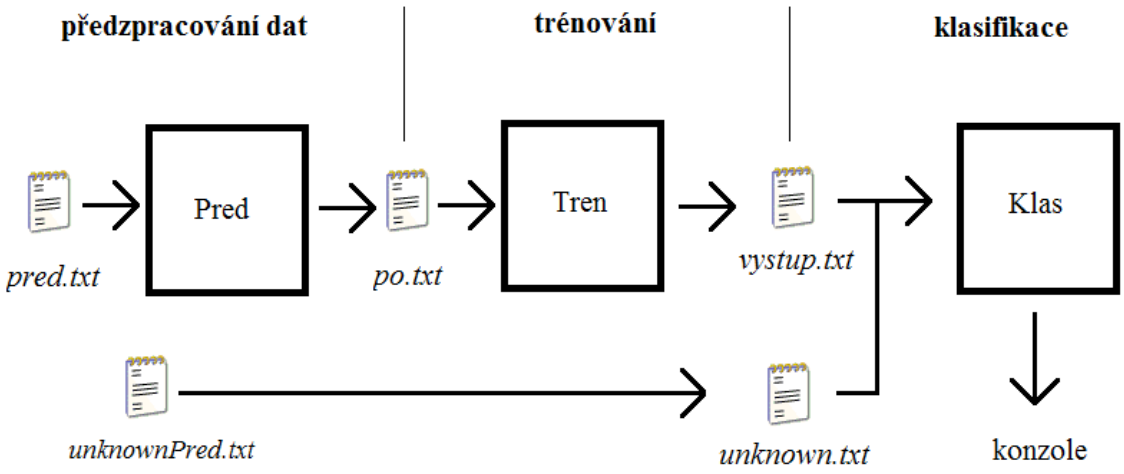
Jedná se o ohodnocení slov v rámci třídy metodou TF-IDF, ale na rozdíl od klasické metody je aplikována postupně na jednotlivé třídy a složka *inverse document*

frequency je počítána pro každou třídu zvlášť. Metoda pro každý článek vybere *k* (30) slov, takže ve výsledku pro třídu vybráno více než *k* klíčových slov (jako u TF). Poté je nahrazena TF v případě neznámých článků. Její největší nevýhodou je použití TF u neznámých článků i v trénovacích výpočtech – to má za následek špatný výsledek při objevování nových témat.

3 Navržený způsob řešení

3.1 Struktura programu

Pro trénování a klasifikaci se používají oddělené programy. Programy obsahující ve svém názvu slovo "Tren" slouží k trénování, programy obsahující slovo "Klas" ke klasifikaci. Po natrénování klasifikátoru tak není potřeba trénovací program opětovně spouštět a klasifikátor lze samostatně používat na články, u nichž má určit třídu (případně více tříd). Vstupní data obou programů je také potřeba předzpracovávat (např. odstranit velká písmena, tečky a čárky apod.). Výsledky klasifikace se vypisují na konzoli. Struktura programů je znázorněna na následujícím schématu:



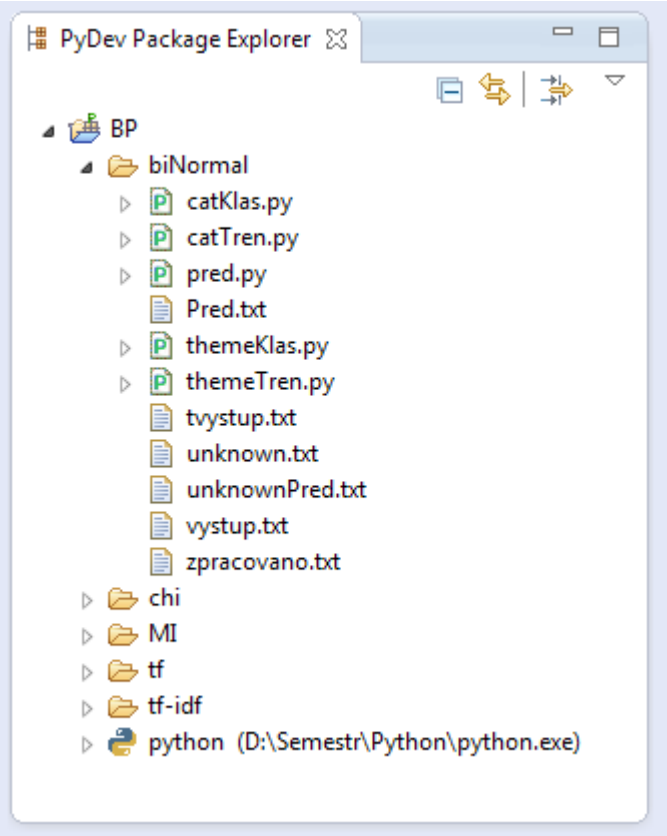
Obrázek 3: Struktura programů

3.1.1 Předzpracování dat

Předzpracování spočívá v odstranění slov s délkou jedna. Tak se odstraní například slovo "a" a také čárky, tečky a podobně. Všechna písmena se také převedou na malá (lowercase).

3.1.2 Použitý programovací jazyk a organizace adresářů

Použitý programovací jazyk je Python, který je open-source (volně přístupný veřejnosti). Použitá verze je 2.7. Adresáře jsou strukturovány tak, že každá metoda má svůj vlastní adresář s vlastním klasifikátorem pro témata a kategorie, předzpracováním atd. (viz 3.1).



Obrázek 4: Struktura adresářů (ve vývojovém prostředí Eclipse)

3.2 Metoda pro klasifikaci kategorií

V odkazu (1), článku Category Classification and Topic Discovery in Japaneese and English News Articles je metoda popisována. Tato metoda přiřadí článku nula až n tříd (kategorií), kde n je počet kategorií obsažených v trénovací množině.

3.2.1 Trénování klasifikátoru kategorií

Trénovací množina obsahuje dvojice (článek, kategorie), přiřazené učitelem. Pro každou kategorii se trénuje zvlášť. Z trénovací množiny se vyberou články s jednou kategorií a provedou se s nimi následující operace:

1. Vyberou se klíčová slova článků kategorie pomocí nějaké metody.
2. Všechna klíčová slova kategorie se dají do jednoho vektoru.
3. Do dalšího vektoru stejné délky se uloží číslo, u kolika článků bylo toto klíčové slovo určeno.
4. Uloží se počet článků v trénovací množině pro tuto kategorii a její jméno.

Pokud metoda přiřazuje klíčová slova pro celou kategorii, použije se v bodu 3 místo počtu článků, kde bylo slovo určeno jako klíčové, počet článků ve kterých se klíčové slovo vyskytuje – jako u metody trénování témat (viz 3.3.1).

Program provádějící tyto operace má název "cKlas.py". Jeho vstupem je textový soubor "zpracovano.txt" (viz 3.1.1). Data se ukládají do souboru "vystup.txt". Ten má následující strukturu:

vystup.txt:

```
psi kočky                       - první řádek – vektor všech kategorií
1 1                             - druhý řádek – jejich počet: psi 1, kočky 1
pes má čtyři tlapy a            - klíčová slova kategorie psi
1 1 1 1 1                       - počet klíčových slov kategorie psi
kočka má ocas a dvě             - klíčová slova kategorie kočky
1 1 1 1 1                       - počet klíčových slov kategorie kočky
```

3.2.2 Klasifikace kategorií

Po načtení dat ze souboru "vystup.txt" a vytvoření příslušných vektorů v paměti počítače se načtou neznámé články ze souboru "unknown.txt". Ten má stejnou strukturu jako vstup klasifikátoru, dvojice článek – kategorie:

```
unknown.txt:
pes vrtí ocasem                    - první neznamý článek -
psi                                - jeho kategorie -
kočka mňouká                       - druhý neznamý článek -
kočky                              - jeho kategorie -
```

Kategorie zde slouží jen pro zobrazení výsledků (určení přesnosti), při klasifikaci nejsou vůbec používány, a nemusí být vůbec vyplněny (místo nich bude prázdný řádek).

Nejprve jsou ze všech článků v neznámé množině vybrána klíčová slova. Metoda musí být (pokud je to možné) stejná jako u trénování. Klasifikace pak probíhá pro každý článek zvlášť. U každého článku jsou vypočteny následující hodnoty:

1. Pro každou kategorii se vypočte pravděpodobnost, že článku daná kategorie odpovídá:

$$Likelihood\left(C_j|A=\{k_1,k_2\ldots k_n\}\right)=-\sum_{i=1}^{n}P\left(k_i|C_j\right)\log\left(P\left(k_i|C_j\right)\right) \qquad (19)$$

Tento vzorec lze chápat jako: Pravděpodobnost (likelihood), že kategorie C_j odpovídá článku A, skládajícímu se z klíčových slov $k_1,k_2 \ldots k_n$ je rovna záporné sumě přes všechna klíčová slova článku ze součinu: pravděpodobnost, že dané slovo náleží do kategorie C_j násobeno logaritmem této pravděpodobnosti. Pravděpodobnost se vypočte následujícím způsobem:

$$P\left(ki|Cj\right)=\frac{kolikrát\ bylo\ ki\ určeno\ \ pro\ kategorii\ Cj}{počet\ dokumentů\ v\ trénovací\ množině\ \ s\ kategorií\ Cj} \qquad (20)$$

Obě čísla jsou snadno rozpoznatelná v ukázce ze souboru "vystup.txt". Pokud

dané slovo nebylo v kategorii vůbec určeno, pravděpodobnost je nula. Všechny pravděpodobnosti se uloží do vektoru pravděpodobností - *LikelihoodVect*.

2. Pro každý článek se vypočte *threshold* – hranice. Pokud pravděpodobnost některé kategorie překročí tuto hranici, bude přiřazena článku.

$$threshold = průměr(LikelihoodVect) + směrodatná\ odchylka(LikelihoodVect) \quad (21)$$

Vzorec pro výpočet průměru a směrodatné odchylky z vektoru je následující:

$$průměr(\bar{X}) = p(\bar{X}) = \frac{\sum_{i=1}^{|\bar{X}|} x_i}{|\bar{X}|} \quad (22)$$

$$odchylka(\bar{X}) = s(\bar{X}) = \sqrt{\frac{\sum_{i=1}^{|\bar{X}|} (\bar{X}(i) - p(\bar{X}))^2}{|\bar{X}|}} \quad (23)$$

$$Likelihood(Cj|A) > threshold \Rightarrow A \in Cj \quad (24)$$

Všechny kategorie, jejichž pravděpodobnost je větší než threshold (součet průměru a směrodatné odchylky z vektoru všech pravděpodobností), jsou přiřazeny článku. V případě, že jsou všechny pravděpodobnosti rovny nule, je použit fakt, že se jedná o ostrou nerovnost, a článku není přiřazena žádná kategorie. Pokud se alespoň jedna z přiřazených kategorií shoduje se správnou kategorií ze souboru "unknown.txt" je klasifikace považována za úspěšnou.

3.3 Metoda pro klasifikaci témat

Metoda je popisována v odkazu (1), článku Category Classification and Topic

Discovery in Japaneese and English News Articles. Tato metoda přiřadí článku právě jednu třídu (téma), přičemž tato třída se dokonce nemusí vyskytovat v trénovací množině.

3.3.1 Trénování klasifikátoru témat

Trénovací množina obsahuje dvojice (článek, téma), přiřazené učitelem. Pro každou kategorii se trénuje opět zvlášť. Z trénovací množiny se vyberou články s jedním tématem a provedou se s nimi následující operace:

1. Vyberou se klíčová slova článků kategorie pomocí nějaké metody.
2. Všechna klíčová slova kategorie se dají do jednoho vektoru.
3. Do dalšího vektoru stejné délky se uloží číslo, v kolika článcích se toto klíčové slovo vyskytuje – tedy nemusí být nutně určeno jako klíčové.
4. Uloží se počet článků v trénovací množině pro tuto kategorii a její jméno.

V bodu 3. je tedy rozdíl oproti kategoriím. Články celé kategorie se po určení klíčových slov znovu celé prohledají kvůli výskytu klíčových slov. Program provádějící tyto operace má název "tKlas.py". Jeho vstupem je textový soubor "zpracovano.txt" (viz 3.1.1). Data se ukládají do souboru "tvystup.txt". Ten má stejnou strukturu jako u metody klasifikace kategorií (viz 3.2.1).

3.2.2 Klasifikace témat

Po načtení dat ze souboru "tvystup.txt", který má stejnou strukturu jako soubor "vystup.txt" (viz 3.2.2), jen s tím drobným rozdílem, že místo kategorií jsou témata.
Ta opět slouží pouze pro zhodnocení výsledků.

Největší rozdíl oproti klasifikaci kategorií je, že článku je přiřazeno vždy právě jedno téma. V případě, že mu známá témata neodpovídají, je totiž vytvořeno nové.

Nejprve se však u všech článků s neznámým tématem určí klíčová slova. Metoda

musí být (pokud je to možné) stejná jako u trénování. Klasifikace pak probíhá u každého článku zvlášť:

1. Pravděpodobnost, že článku náleží dané téma se určí porovnáním vektorů pomocí kosinové nerovnosti:

$$\cos(\alpha(\bar{x},\bar{y}))=\frac{\bar{x}\circ\bar{y}}{|\bar{x}|\cdot|\bar{y}|} \tag{25}$$

Čitatel zlomku je skalární součin vektorů, který je definovaný jako:

$$\bar{x}\circ\bar{y}=x_1\cdot y_1+x_2\cdot y_2+...+x_n\cdot y_n \tag{26}$$

Přičemž $x_1...x_n$ jsou složky vektoru $\bar{x}$. Ve jmenovateli jsou pak absolutní hodnoty vektorů. V případě tohoto programu se používá eukleidovská norma:

$$|\bar{x}|_e=\sqrt{x_1^2\cdot+x_2^2+...+x_n^2} \tag{27}$$

Vektor tématu je načtený ze souboru "tvystup.txt", ještě ale projde vyhlazovací technikou známou jako *add one smoothing*. Její princip je následující: Mějme téma psi a jeho tři klíčová slova *a, ocas, pes* vyskytující se v 1, 2, 3 článcích této kategorie:

```
psi    8 (celkem 8 článků v trénovací množině)
a      1  ppst = 1/8
ocas   2  ppst = 2/8
pes    3  ppst = 3/8
```

Add one smoothing přidá ke každému výskytu jedna, a k celkovému počtu článků počet slov (sumu všech jedniček). Což nejlépe ukazuje příklad:

```
psi    8 (celkem 8 článků v trénovací množině)
a      1 + 1 = 2  ppst = 1/(8 + 3) = 1/11
ocas   2 + 1 = 3  ppst = 2/(8 + 3) = 2/11
pes    3 + 1 = 4  ppst = 3/(8 + 3) = 3/11
```

Čímž vznikne vektor tématu.

Vektor článku se vytvoří tak, že po nalezení klíčových slov článku se vypočte jejich skóre. To je rozebráno v (1).

$$score(NP) = \frac{UF(NP) \cdot NPF(NP)}{|NP|} = tf(x) \tag{28}$$

Zkratka *NP* představuje *noun phrase* (viz 2.3.6), obecně více slov. *UF(NP)* je suma výskytů jednotlivých slov z fráze *NP* v daném článku lomeno délka článku, čili suma *term frequency*, *NPF* je výskyt celé fráze *NP* v článku. Metody vyhledávání klíčových slov použité v práci vyhledávají slova po jednom, čili délka |*NP*| je vždy jedna, a *UF* se změní na výskyt daného slova v článku, čili TF. Tak se vytvoří vektor pro článek.

Ještě je však potřeba vyřešit (velmi častý) případ, kdy nemá článek a téma stejná klíčová slova ve vektoru. Řešením je vytvoření jednoho společného vektoru slov, přičemž hodnoty slov která tématu, respektive článku nenáleží se vyplní nulami (v příkladu *nebo*, *pes*):

```
 (před)                    (po)
psi                       psi
a ocas pes                a ocas pes nebo
1 2    3                  1 2    3   0
neznamy_clanek            neznamy_clanek
a nebo ocas               a ocas pes nebo
3 2    1                  3 1    0   2
```

Nyní je možné na oba vektory aplikovat kosinovou nerovnost a vypočítat

kosinus úhlu mezi nimi. Pokud se blíží úhel k nule, blíží se jeho kosinus k jedné a vektory jsou si podobné. Hodnoty v obou vektorech jsou také vždy větší než nula (což lze ve dvou dimenzích chápat jako pohyb v prvním kvadrantu), čili kosinus nemůže být záporný. Nejméně může být nula, tedy jej lze chápat jako pravděpodobnost, že téma patří danému článku. Aby bylo téma článku přiřazeno, je nutné však ještě ověřit dvě následující podmínky:

$$1.\cos(T,A) > 0.1 \ \wedge \cos(T,A) > newTsim(T,A) \tag{29}$$

$$2.oTemata > 10 \wedge \cos(T,A) > p(vektorCos) + s(vektorCos) \tag{30}$$

A ve vzorci je článek a *T* je téma. Funkce *newTsim* je vyjádření podobnosti mezi článkem a teoretickou kategorií, která je mu relativně podobná. Klíčová je hodnota 0.05, kterou autoři z (2) určili experimentálně. Pokud by se například používaly vektory hodnot TF-IDF místo předepsaného ohodnocení (vzorec 27), bylo by potřeba ji změnit (experimentálně nalézt).

$$newTsim(T,A) = \frac{(0.05 \cdot |T|) \cdot (p(A) - s(A)) \cdot p(T)}{(|A| \cdot (p(A)^2)) \cdot (|T| \cdot (p(T)^2))} \tag{31}$$

oTemata ve vzorci 29 pak reprezentuje počet klasifikátorem objevených témat (viz dále). *vektorCos* je vektor kosinových nerovností všech kategorií, ve vzorci je použit jeho průměr *p()* a směrodatná odchylka s*()* (viz 3.2.2).

Pouze pokud jsou obě podmínky splněny, je téma přiřazeno článku.

Pokud se tak nestane, vytvoří se nové téma, a proběhne trénování. Trénovací data jsou vektor článku, který se přidá do paměti počítače k ostatním vektorům témat. Název nového tématu je "new", následované číslovkou oTemata, počet objevených témat. Čili první objevené téma má název "new1", druhé "new2" a tak dále. Jakmile je téma objeveno, může být určeno u dalšího článku. Po ukončení programu dojde ke smazání dat objevených témat z paměti počítače.

4 Prezentace a zhodnocení výsledků

4.1 Použitá data

Pokusy byly prováděny na množině článků o průměrné délce přibližně dva a půl tisíce slov (polovina stránky A4), čili poměrně krátké články, podobné těm jež se denně vyskytují na internetových serverech, například novinky.cz. V dostupné množině je dva tisíce článků z dvaceti kategorií, sto pro každou. Do testovací množiny bylo zařazeno třicet článků z každé kategorie, do trénovací zbylých sedmdesát.

4.2 Způsob reprezentace výsledků

4.2.1 Precision

Precision, česky přesnost, je podíl správně určených obrazů ku všem určeným (správně i chybně) obrazům dané třídy při klasifikaci. Nejlepší možná hodnota přesnosti je 1 (100%), nejhorší nula.

Například pokud do klasifikátoru pro kategorii "terorismus" jsou jako vstup tři články (obrazy), všechny náležející do třídy terorismus, a jsou klasifikátorem určeny jako patřící do kategorie: "terorismus" (správně), "automoto" a "církve", je hodnota *precision* pro třídu terorismus:

$$precision(terorismus) = \frac{určené\ správně}{určené\ správně + určené\ chybně} = \frac{1}{1+0} = 100\% \qquad (32)$$

Pro třídu církve byl určen obraz jeden a to chybně tedy:

$$precision(cirkve) = \frac{0}{0+1} = 0\% \qquad (33)$$

4.2.2 Recall

Recall, česky v tomto případě nejlépe nejspíš dovolání, je podíl správně určených obrazů ku všem obrazům v testovací množině z této třídy. Představuje úspěšnost klasifikátoru. Nejlepší možná hodnota je jedna, nejhorší nula (stejně jako u přesnosti). Uvažujme stejný příklad jako v předchozí podkapitole (4.2.1), hodnota *recall* pro třídu "terorismus" pak bude:

$$recall = \frac{určené\ správně}{počet\ v\ trénovací\ množině} = \frac{1}{3} = 33.33\,\% \tag{34}$$

4.2.3 F-measure

Tato hodnota se vypočítá z hodnot *recall* a *precision* podle následujícího vztahu:

$$F = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{35}$$

Stejně jako obě předchozí nabývá hodnot od 0 do 1, a nejlepší možná hodnota je jedna.

4.3 Výsledky experimentů

Experiment byl následující: Z množiny dat o dvaceti třídách:

```
automoto
zdraví
církve
Euro
fotbal
historie
kriminalita
kultura
léky
```

```
počasí
právo
prezident
průmysl
soud
školy
terorismus
trh
věda
vláda
volby
```

Pro každou třídu bylo k dispozici sto exemplářů dokumentů (obrazů), z nichž pak bylo vybráno prvních sedmdesát dokumentů každé třídy jako trénovací množina (vstup pro programy "Tren") a posledních třicet jako testovací (vstup pro programy "Klas"). Těchto třicet klasifikátor určuje a z výsledků je počítána *recall, precision* respektive *F-measure*.

4.3.1 Experiment 1 - výsledky pro kategorie

Jedná se o určení *n* tříd klasifikátorem, kde *n* je větší rovno nule, a právě jedna je správná. Všechny tři hodnoty se počítají pro každou třídu zvlášť, a poté se z nich vypočte průměr, který byl zaokrouhlen na tři desetinná místa a vypsán ve formátu procent (všechny tři hodnoty nabývají od 0 do 1).

	TF	TF-IDF	experiment	BiNormal	chi	MI	Bayes
recall	87.9%	31.2%	78.1%	57.9%	86.3%	50.2%	85.4%
precision	39.9%	43.0%	36.3%	58.1%	49.9%	62.2%	33.6%
F-measure	54.8%	36.16	49.5%	58.0%	63.2%	55.5%	48.3%

Tabulka 1: Tabulka zobrazující precision, recall a F-measure metod pro experiment s kategoriemi.

Zhodnocení

Z tabulky je patrné, že nejlepších výsledků dosáhla metoda chí kvadrát. Jelikož se jedná o obvykle tři určené třídy, přičemž jedna je správná, je hodnota recall vysoká, a precision nižší. Metoda TF má například skvělý recall, ale nízkou precision, což je dáno tím, že přiřazuje obecně hodně kategorií (poměrně vzácně méně než tři). Experimentální metoda je na tom přibližně stejně jako TF. Jelikož ale nevybírala obvykle tolik kategorií, má v tomto případě o trochu nižší úspěšnost. Nejhorší je metoda TF-IDF, která pro tento algoritmus nejspíše nebude vhodná (což se potvrdilo v dalších experimentech). Všechny metody kromě TF-IDF v tomto případě dosáhly lepších výsledků než Bayesovský klasifikátor.

4.3.2 Experiment 2 - výsledky pro témata

Hodnoty *recall, precision* a *F-measure* jsou vypočtené stejně jako v předchozí podkapitole (4.3.1). Přibývá však ještě jedna hodnota a to je úspěšnost, neboť v případě tohoto algoritmu se detekují nová témata, a ta jsou přidána do celkového průměru. Úspěšnost je podíl správně klasifikovaných obrazů v testovací množině.

	TF	TF-IDF	experiment	BiNormal	chi	MI	Bayes
recall	29.7%	8.0%	49.2%	36.2%	54.1%	20.7%	66.4%
precision	48.2%	10.8%	53.8%	62.2%	59.1%	67.7%	64.2%
úspěšnost	29.7%	30.0%	49.2%	39.8%	65.3%	22.8%	64.2%
F-measure	36.6%	9.2%	51.3%	45.7%	56.5%	31.7%	65.3%

Tabulka 2: Tabulka zobrazující precision, recall a F-measure metod pro experiment s tématy.

Zhodnocení

Při tomto experimentu se potvrdilo, že chí kvadrát je nejvhodnější metoda. Metody bi normálního rozdělení a mutual information mají nízkou recall, i když měly dobrou precision – je to dáno tím, že objevily dvě nová témata a mnoho článků do nich

nesprávně přiřadily. Proto jsou naprosto nevhodné pro tuto konkrétní metodu určování témat, pokud by se však jednalo o její modifikaci bez vzniku nových témat, byly by výsledky lepší. Metoda TF má poměrně špatné výsledky – což se od takto primitivní metody dá očekávat – ovšem v souvislosti s předchozím experimentem vzniká zajímavá souvislost, totiž že pouze v 29% případů z 90% kdy je úspěšná v prvním případě, má správná třída nejvyšší kriteriální funkci (kosinovou nerovnost). Ostatní třídy které mají kriteriální funkci kladnou tak často jsou přibližně na prvním až třetím místě. Tedy by TF mohla být dobrým ukazatelem pro nějaké složitější metody, což potvrzuje dobrý výsledek experimentální metody. Metoda TF-IDF má nejhorší výsledky, objevila totiž 55 nových témat, a je pro tuto metodu nevhodná. Jediná metoda která se může s Bayesovským klasifikátorem měřit je metoda chí kvadrát. Má dokonce vyšší úspěšnost než Bayesův klasifikátor, avšak má nižší precision a recall. Tento výsledek je dán objevením nových témat – Bayesovský klasifikátor nová témata neobjevuje a při zprůměrování tak může nové téma u chí kvadrát sehrát významnou roli, i když je úspěšnost přibližně stejná.

4.3.3 Experiment 3 - výsledky pro objevení nového tématu

Hodnoty jsou vypočtené stejně jako v předchozí podkapitole (4.3.2). Přibývá však ještě jedna hodnota a to je počet objevených témat v řádku "objeveno". Ubývá hodnota F-measure, neboť v tomto experimentu nemá příliš velký smysl, nejde o klasifikaci do více tříd jako v předchozím případě. Úspěšnost je hlavní ukazatel. Změnila se totiž trénovací a testovací množina. Trénovací množina obsahuje pouze dvě témata "léky" a "fotbal" o sto článcích (všech dostupných). Testovací množina obsahuje pouze příklady z jedné třídy, "školy" také o sto článcích. Všechny příklady by měly být určeny jako "new1" neboli nové neznámé téma, což se považuje jako úspěch. Smyslem experimentu je ukázat na schopnost metod použít techniku objevení nového tématu.

	TF	TF-IDF	experiment	BiNormal	chi	MI	Bayes
recall	0%	0%	0%	28.3%	1.6%	32.6%	-
precision	0%	0%	0%	33.3%	33.3%	33.3%	-
objeveno	0	80	0	1	1	1	-
úspěšnost	0%	0%	0%	85.0 %	5.0 %	98.0%	-

Tabulka 3: Tabulka výsledků metod pro třetí experiment s prvkem objevování nových témat.

Zhodnocení

Z tohoto pokusu je vidět, že metody MI a BiNormal mají v tomto případě vysokou úspěšnost. Jelikož v prvním případě dojde k objevení nového tématu, má metoda MI úspěšnost 100%. Slabinou těchto metod je, že nově objevená témata pokazí v obyčejném případě (experiment 2) klasifikaci a jsou přiřazována příliš často, tedy metody upřednostňují nové třídy. Metoda TF-IDF má jiný nedostatek – objevuje příliš mnoho témat. Metoda chi pak objeví jedno téma ale upřednostňuje staré třídy, avšak v rozumné míře, kdy se nové články musí razantně lišit od známých, aby došlo k objevení nového tématu. U TF a experimentální nedošlo k objevení nových témat, což znamená že upřednostňují staré třídy ještě více. Jde o jejich největší slabinu, neboť častá slova jsou ve všech článcích prakticky stejná. Porovnáním s ostatními je zřejmá vhodnost metody chi, kde *dojde* k objevení nového tématu, avšak není výrazně upřednostňováno. Z experimentu je také vidět, že konstanta 0.05 (viz 3.2.2 na konci) je zřejmě dobře ustanovena – nedošlo totiž u žádné metody kromě TF-IDF (která je extrémní) k objevení více než jednoho tématu. Bohužel Bayesovský klasifikátor objevování nových témat není schopen a není jej tedy možné s metodami porovnat.

5 Závěr

V počátku se práce zabývá nástinem klasifikace obecně. Poté nastiňuje speciální problém klasifikace textu a dále popisuje obecné přístupy k vytváření metod vyhledávání klíčových slov. Následně vysvětluje některé běžné metody hledání klíčových slov včetně jejich matematického popisu a použitých parametrů. Uvádí také jednu z možných variant implementace metod a popisuje obě metody, pro klasifikaci kategorií a pro klasifikaci témat.

Výsledky experimentů při porovnání metod mezi sebou ukazují, že metoda chí kvadrát značně převyšuje ostatní, a tak ji nejspíš jako jedinou je možné označit za vhodnou. Dosahuje stejných (v případě témat) nebo lepších (v případě kategorií a prvku objevování nových témat) výsledků než Bayesovský klasifikátor. Intuitivně lze očekávat, že metoda vybírající prvky, kterými se jednotlivé kategorie nejvíce liší, bude pro krátké články a objevování nových témat velmi vhodná.

První dva experimenty ukazují mnohem horší výsledky než původní metoda (kolem 90%). Bylo by možné hodnoty zlepšit, například použitím kvalitního předzpracování, lepší (nebo větší) trénovací množiny. Bylo by jistě zajímavé původní metodu prozkoumat, avšak její složitost přesahuje rámec této práce, vyvíjeli ji tři experti z oboru za použití pokročilého předzpracování. Masarykova univerzita v Brně se zabývá zpracováním přirozeného jazyka a vlastní program Synt, s jehož pomocí lze určit vztahy větných členů, a nejspíš by s jeho použitím bylo možné metodu částečně (bez předzpracování) implementovat.

Literatura

1 BRACEWELL, David a spol. *Category Classification and Topic Discovery of Japanese and English News Articles.* [online]. c2009 [cit. 2012-9-10]. Dostupné na World Wide Web:
<http://www.elsevier.com/journals/electronic-notes-in-theoretical-computer-science/1571-0661>.

2 MATSUO Y. , ISHIZUKA M. *International Journal on Articial Intelligence Tools : Keyword extraction from a single document using word co-occurrence statistical information,* Vol. 13, No. 1 World Scientic Publishing Company (2004). 12 s.

3 FORMAN G. *Journal of Machine Learning Research : An Extensive Empirical Study of Feature Selection Metrics for Text Classification,* [online]. c2003, last revision 27th of February 2009. [cit. 2013-1-4].
Dostupné na World Wide Web:
<http://jmlr.csail.mit.edu/papers/v3/>.

4 ACKLAM, John. *An algorithm for computing the inverse normal cumulative distribution function* [online]. c2004, last revision 27th of February 2009. [cit. 2012-12-10].Dostupné na World Wide Web:
<http://home.online.no/~pjacklam/notes/invnorm/>

5 BRACEWELL, David, REN, Fuji and KURIOWA, Shingo. *Multilingual Single Document Keyword Extraction for Information Retrieval,* Department of Information Science and Intelligent Systems, Faculty of Engineering, The University of Tokushima, Tokushima 770-0861, c2004.

6 LEE, Hong, ISA, Dino, CHOO, Wou Onn and CHUE, Wen Yen. *Expert Systems with Applications: An International Journal* : *High Relevance Keyword Extraction facility for Bayesian text classification on different domains of varying characteristic,*
Volume 39 Issue 1, Pergamon Press, Inc. Tarrytown, NY, USA (2012). 8 s.