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

Diplomová práce

Automatická extrakce příspěvků z diskusních fór

Plzeň 2017 Jakub Sido

Prohlášení

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

V Plzni dne 15. května 2017

Jakub Sido

Abstract

Internet is very quickly growing medium. It is becoming more requisite to process data contained therein automatically. This work deals with extraction of informations from web sources, especially from the web discussion forums. It discuss this discipline and examines existing systems. Then is this knowledge aplicated and is designed a system, which does this job without a human intervation. There are also used methods of machine learning and of analysis of natural language to designation of meaning of acquired data

Abstrakt

Internet je velice rychle rostoucí médium. Stává se více žádané data na něm obsažená zpracovávat automaticky. Tato práce se zabývá extrakcí informací z webových zdrojů, především z webových diskuzních fór. Pojednává o tomto oboru a zkoumá existující systémy. Následně jsou tyto poznatky aplikovány a je navrhnut systém, který tento úkol plní bez zásahu člověka. Dále jsou použity metody strojového učení a analýzy přirozeného jazyka k označení významu získaných dat.

Poděkování

Rád bych poděkoval Ing Miloslavu Konopíkovi Ph.D. za jeho trpělivost, dobrý přístup a cenné poznámky, které mi pomohly při zpracování mé práce. Dále pak svým kamarádům a blízkým, za jejich pohled na mou práci z jiné perspektivy a odstupem.

Obsah

1 Úvod **1**

2 Studie oblasti získávání informací **2**
2.1 Způsoby prezentace informací na webu 2
2.2 Strukturovanost webu 2
2.3 Document Object Model 3
2.4 XPath 4
2.5 Šablona 5
2.6 Datové schéma 6
2.7 Model vytváření stránky 7
2.8 Automatická extrakce 7
2.9 Algoritmus bez učitele 9
2.9.1 Neshoda řetězce 9
2.9.2 Neshoda TAGu 9
2.9.3 Nepovinné bloky dat 9
2.9.4 Dynamické seznamy 10
2.10 Reprezentace úlohy 10

3 Strojové učení a zpracování přirozeného jazyka v oblasti extrakce internetových fór **12**
3.1 Klasifikace 12
3.2 Entropie 13
3.3 Bag-of-words 13
3.4 N-gramy 14
3.5 Ověření úspěšnosti klasifikátoru 16
3.6 Matice záměn 17
3.7 Křížová validace 18

4 Existující systémy pro extrakci informací **19**
4.1 RoadRunner 19

4.2 TISP . . . 19
4.3 IePAD . . . 20
4.4 ExAlg . . . 20
4.5 DeLa . . . 20
4.6 FeedExtract . . . 21
4.6.1 Trénovací data . . . 21
4.6.2 Rozpoznání příspěvků fóra . . . 22
4.7 Project Extraction of Web Discussion . . . 22

5 Návrh nového systému 24
5.1 Architektura systému . . . 25
5.1.1 Preprocesor . . . 25
5.1.2 Dávkové zpracování RR . . . 27
5.1.3 LabeledPage . . . 27
5.1.4 DataProvider . . . 27
5.1.5 ForumModel . . . 27
5.1.6 Validator . . . 27
5.2 Křížová validace . . . 28
5.3 Rozšíření trénovacího korpusu . . . 28
5.4 Extrakce dynamických dat z webového fóra . . . 29
5.5 Preprocesing . . . 29
5.6 Příznaky . . . 30
5.6.1 Bag-of-words a Bag-of-character-n-grams . . . 30
5.6.2 Délka klasifikovaného textu . . . 30
5.6.3 Poměrné zastoupení číslic v textu . . . 31
5.6.4 Shoda části řetězce s regulárním výrazem . . . 31
5.6.5 Klíčová slova . . . 31
5.7 Přehled nad procesem extrakce . . . 31

6 Úspěšnost systému 34
6.1 Experiment oddělitelnosti 1 . . . 35
6.2 Experiment oddělitelnosti 2 . . . 36
6.3 Experiment párování výstupu RR s označenými daty 1 . . . 38
6.4 Experiment párování výstupu RR s označenými daty 2 . . . 39
6.5 Experiment velikost dávky zpracovávané RR . . . 42
6.6 Experiment vliv jednotlivých příznaků . . . 44
6.7 Experiment nastavení prahové hranice Bag of words . . . 46
6.8 Experiment stálosti třídy datových bloků . . . 47
6.9 Experiment korekce bloků podle šablony . . . 48

6.10 Známé nedostatky programu 50
6.10.1 ROAD RUNNER 50
6.10.2 Trénovací data 52
6.10.3 Správné spojení údajů 52

7 Závěr **53**

Příloha a - uživatelská dokumentace **56**

1 Úvod

Internet a celý web je velice rychle rostoucí médium. Stává se více žádané data na něm obsažená zpracovávat automaticky. Většina informací, zobrazovaných na webu, je však určena pro lidi a při jejím uveřejňování většinou není myšleno na pozdější strojové zpracovávání. Tato nestrukturovanost velice ztěžuje získávání informací.

Největším cílem této oblasti a zároveň důvodem, proč se zabýváme touto problematikou, je zjednodušení přístupu k informacím v textu pro budoucí zpracování. Tato práce se věnuje extrakci dat z webových diskuzí.

Tato data mohou být použita různými způsoby, od cílených reklam přes analýzu názorů po vyhledávání nevhodných činností ve virtuálním světě, jako je například šikana, zneužívání dětí nebo extrémistické chování.

Existují způsoby, jak vytvořit nástroj, který bude extrahovat žádaná data z konkrétních webových stránek. Je však vždy potřeba optimalizovat systém pro určitý zdroj. Cílem této práce je však vytvořit prostředek, kterým bude možné automaticky získávat data z velkého množství malých webových diskuzí.

Nejprve bude provedena studie oblasti automatické extrakce informací z webových stránek. Následně budou tyto obecné postupy aplikovány konkrétně v oblasti webových diskuzí a fór. Dále budou rozebrány výsledky této práce a konstruktivně popsány objevené nedostatky a naznačeny směry, kterými by se mohl další výzkum v této oblasti ubírat.

2 Studie oblasti získávání informací

Tato kapitola se zabývá studií oblasti získávání informací. Obecně definuje webovou stránku, její vznik a popisuje metodiku vyhledávání informací.

2.1 Způsoby prezentace informací na webu

Je několik způsobů, jak informace na webu zveřejňovat. Lze je rozdělit do dvou skupin podle toho, jestli jsou data zveřejňována **pomocí nějaké šablony** a jsou dále děleny, at' už více či méně, do struktur[1], nebo se jedná o **prostý text** obsahující informace.

Použití šablony Příkladem prvního z uvedených může být například internetový obchod, knihovna nebo třeba výsledky sportovních utkání. Stránka je vždy generována pomocí stejné šablony. v tomto případě se mění informace, ale významy jednotlivých datových bloků zůstávají zachovány.

Prostý text V druhém případě není žádná unifikovaná forma, jak data prezentovat, a to ani v rámci jednoho webového portálu. Do této skupiny patří například zpravodajské servery nebo blogy.

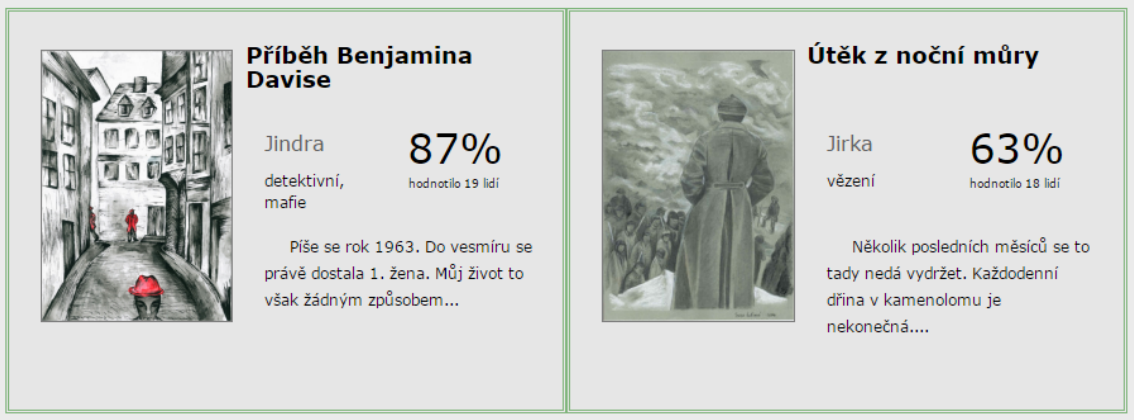
Tato práce se zabývá extrakcí příspěvků diskuzních fór. Ta jsou vždy plněna z datového úložiště. v rámci jednoho serveru mají proto všechny příspěvky stejnou datovou strukturu. Jedná se tedy o první zmíněný příklad - použití šablony.

2.2 Strukturovanost webu

Na webu je široká škála různých stránek používající k prezentaci dat šablony a každá přistupuje jinak k interpretaci informací, které sděluje. Některé stránky jsou vhodnější pro hledání informací a jejich následnou interpretaci. Například stránky nabízející širokou škálu produktů – knihovny, internetové obchody a podobné (viz obr. 2.1

[1] Strukturou je myšlen celek, který obaluje informace, které spolu souvisí

str. 3). Používají stejnou šablonu pro interpretaci všech objektů a informace často strukturují. Cílem získávání informací není hledat význam dat, ale pouze vyhledat atributy stejného významu.(viz tab. 2.1 str. 3)



Obrázek 2.1: Ukázka z knihovny gamebooků[2]

stránka	a	B	C	D	E	...
1	Příběh Benjamina Davise	Jindra	87%	19	Píše se rok 1963...	...
2	Útěk z noční můry	Jirka	63%	18	Několik posledních ...	...
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.

Tabulka 2.1: Tabulka atributů stránek

2.3 Document Object Model

Dokument Object Model (DOM) je na platformě a programovacím jazyku nezávislé rozhraní, které dovoluje programům a skriptům dynamicky přistupovat a upravovat obsah, strukturu a styl dokumentu. [5]

Element Element je definován jako vše od začínající značky až po značku ukončující (včetně obou těchto značek). Například :

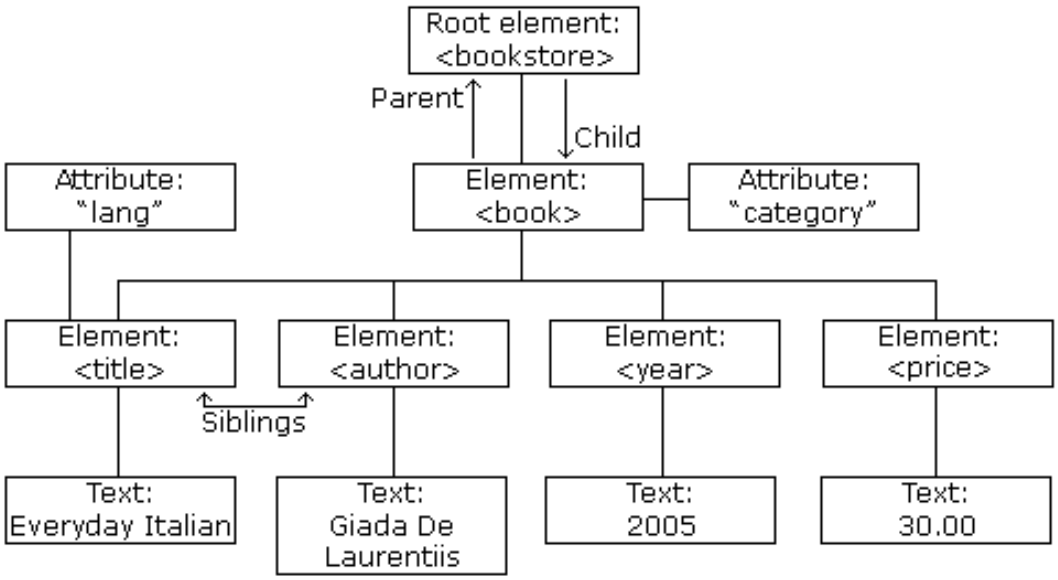
<cena>29.99</cena>

Element může obsahovat:

- text,
- atributy,
- jiné elementy.

V DOM jsou definovány tyto vztahy mezi jednotlivými elementy:

- rodič (parent),
- potomek (child),
- sourozenec (sibling).



Obrázek 2.2: Ukázka struktury DOM [5]

2.4 XPath

Zkratka z XML Path Language. Je to jazyk, jímž lze adresovat části XML dokumentu. Pomocí tohoto jazyka lze z XML dokumentu vybírat jednotlivé elementy,

atributy a hodnoty. Lze adresovat jak od samotného kořenu stromu, tak vybírat libovolné podstromy odpovídající kritériím.

Příkladem může být například tyto XPath[4].

//div[@class=”author”]//a/

Tato XPath vybere všechny elementy **a**, které jsou kdekoli v podstromu potomků elementu **div** kdekoli v dokumentu, který má atribut **class** roven hodnotě **author**.

//bookstore/book[price>35.00]

Tato XPath vybere všechny elementy **book** mající atribut **price** větší než 35, a které jsou přímými potomky elementu **bookstore** kdekoli v dokumentu.

Tento jazyk má velice silnou vyjadřovací schopnost. Je několik knihoven umožňující pracovat s tímto jazykem. Například Xsoup [6], která je založena na knihovně JSoup [13]. Tato knihovna podporuje všechnu syntaxi standardního jazyka XPath. Navíc je přidáno několik možností navíc. Například hledání regulárních výrazů. To umožnilo přidat do XSoup navíc další možnosti vyhledávání. Definuje nové operátory.

- ! = (se nerovná hodnotě)
- = (začíná hodnotou)
- $ = (končí hodnotou)
- ∗ = (obsahuje hodnotu)

2.5 Šablona

Jak bylo řečeno šablona se používá k interpretaci datového schématu. Je tedy nezávislá na datech, která zobrazuje a je pro všechny stránky stejného typu identická. Patří do ní nejen elementy, které obsah dělí do logických celků ale také texty. v předchozím případě je součástí šablony také například: **hodnotilo** nebo **lidí**.

2.6 Datové schéma

Datové schéma obecně popisuje atributy zobrazovaného objektu a je jej možné definovat, neboli konstruovat pomocí několika základních typových konstruktorů[7]:

- Typ T - obecně jeden z následujících.
- Základní typ B - řetězec tokenů (HTML i text).
- N-tice typů $< T_1, T_2, T_3 >$ - uspořádaná n-tice po sobě jdoucích typů.
- Množina typů $\{T\}$.

V našem příkladu jsou zobrazovány informace o knize. Kniha má vždy **titulek** a **jednoho nebo více autorů**. Dále je jí přiřazen **jeden nebo více žánrů**. Zobrazuje se také informace o **hodnocení čtenářů** v procentech a **počet hodnocení**. v poslední řadě je pak zařazena krátká **ukázka** z knihy.

$$S_1 =< B, \{B\}_{T_1}, \{B\}_{T_2}, B, B, B >_{T_3}$$

Kde T_1 a T_2 jsou množinové konstruktory a T_3 je n-tice typů. Instancí takového schámatu je například:

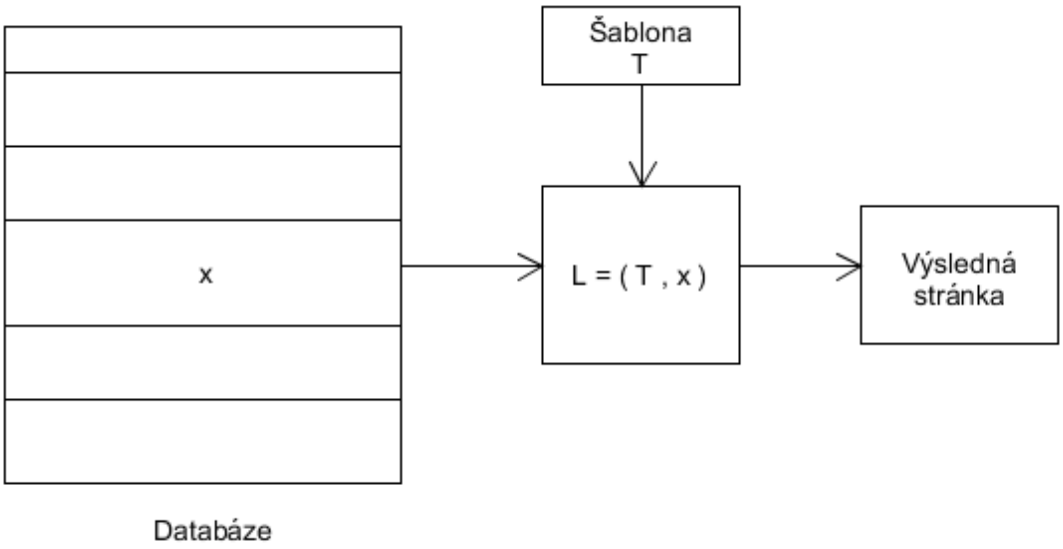
$$S_1 =< t, \{a_1\}, \{z_1, z_2\}, h, p, u >$$

Kde:

- t - titulek,
- a_1 - autor,
- z_1, z_2 -žánr,
- h - hodnocení,
- p - počet hodnocení,
- u - ukázka.

2.7 Model vytváření stránky

Model vytváření samotné stránky (viz obr. 2.3 str. 7) zobrazuje data z databáze objektů pomocí funkce $L(T, x)$, jež využívá šablonu T. Pomocí dříve definovaného datového schématu je možné následně stránku popsat stromovou strukturou (viz obr. 2.4 str. 8), kde je vždy možné podstrom celého schématu chápat jako podschéma [7].

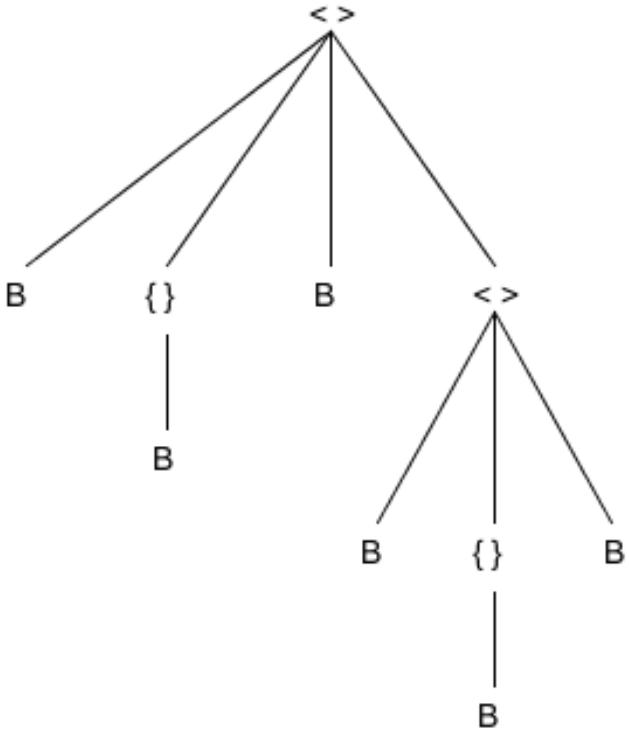


Obrázek 2.3: Schéma vytváření stránky pomocí šablony a dat

Šablona T pro schéma S je definována jako funkce, která mapuje každý typový konstruktor T_k schématu S na seřazenou množinu řetězců .

2.8 Automatická extrakce

Cílem automatické extrakce dat je vytvořit generátor šablon, který dokáže z webových stránek stejného typu vyhledat šablonu, pomocí níž jsou stránky plněny daty.



Obrázek 2.4: Příklad stromové struktury

Požadavky na systém

- Systém by měl být naprosto samostatný a neměl by vyžadovat žádnou interakci s uživatelem.
- Tento úkol by měl zvládat bez předchozí znalosti struktury webové stránky, z níž jsou data extrahována.
- Systém bude odvozovat šablonu z několika webových stránek, které byly vygenerovány pomocí téže šablony.

V předchozí kapitole (viz část 2.7 str. 7) bylo vysvětleno, jak se data uložená v databázi pomocí šablony plní do předem připravené struktury. Cílem navrhovaného systému je opačný proces. Tedy z vygenerovaných stránek zpětně vyhledat šablonu a identifikovat data.

Z požadavků je zřejmé, že systém bude založen na hledání vzájemné podobnosti dvou webových stránek a na tomto základě identifikuje dynamická data.

2.9 Algoritmus bez učitele

Základní myšlenka je z webových stránek, vygenerovaných pomocí stejné neznámé šablony, tuto šablonu zjistit a následně identifikovat dynamická data. Tento přístup je založen na porovnávání zdrojových kódů webových stránek, hledání dynamických dat a vytváření generalizované šablony, která dokáže zahrnout všechny vygenerované varianty.

2.9.1 Neshoda řetězce

Předpoklad je takový, že pokud je ve skupině stránek text na některých částech stejný, nenese žádnou dynamickou informaci. Lze tedy jednoznačně zařadit do statické části šablony.

Pokud se řetězec napříč skupinou stránek liší, jedná se o dynamická data, která chceme extrahovat.

Základní myšlenka je prostá, avšak při podrobnějším zkoumání a navrhování lze narazit na několik komplikací.

2.9.2 Neshoda TAGu

Pokud se neshodují dvě stránky v HTML TAGu, může to být kvůli jednomu z několika následujících důvodů. v případě tohoto případu neshody se nejdříve pokusí vyhledat dynamický seznam (viz část 2.9.4 str. 10). v případě této neshody, je ověřována nepovinnost bloku dat.

2.9.3 Nepovinné bloky dat

Protože předem neznáme strukturu stránky ani dat, které zobrazuje, je možné, že na některých z webových stránek bude zobrazována informace, která na ostatních není.

Může to být například nepovinný atribut struktury. v takovém případě není možné prosté porovnávání dvou stránek, ale je zapotřebí zvolit sofistikovanější přístup. Abychom poznali skutečně úplnou formu šablony, bude potřeba více než jen dvě stránky, aby se zajistilo, že se ve vzorku objeví všechny možné variability.

Pokud se ověřuje podezření na nepovinný blok dat, v obou stránkách se pokusíme pokračovat ve vyhledávání shod po přeskočení právě vyhodnocovaného bloku. Pokud jsme dále schopni pokračovat po přeskočení neshodujícího se tagu, generalizujeme šablonu tak, aby obsahovala tento blok jako nepovinný.

2.9.4 Dynamické seznamy

Na webových stránkách se často objevují dynamické seznamy v mnoha podobách. To jsou všechna data, která se opakují v nějaké formě seznamu v ekvivalentním tvaru, jednotlivé položky tedy mají stejnou strukturu, je ale proměnná jeho délka. v takovém případě také není možné použít jednoduchý přístup.

Pokud se narazí na neshodu tagu, pokusíme se předchozí shodující se značku najít znovu dále. Pokud se nám to podaří, zpětně ověříme, zda sedí šablona na vše mezi těmito značkami.

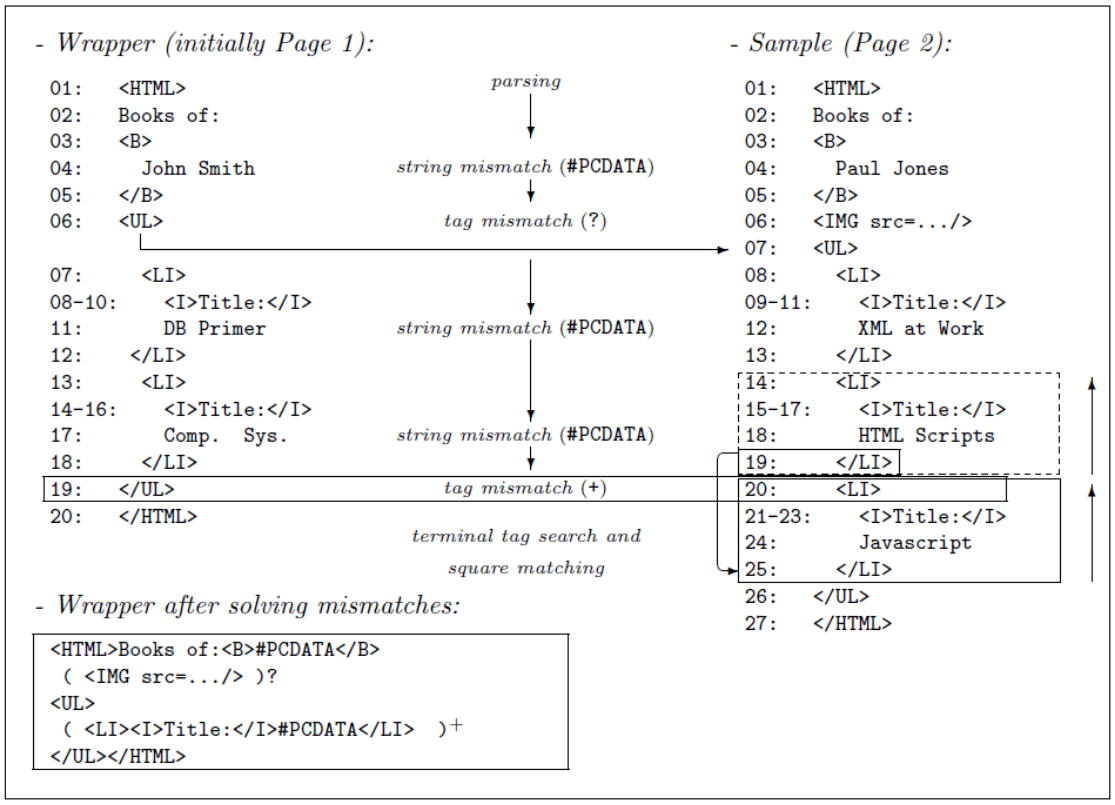
Na řádkách 19 - 20 (viz obr. 2.5 str. 11) dojde k neshodě tagu. Následně se ověřuje přítomnost předchozí značky dále ve zdrojovém kódu stránky. Tento výskyt je potvrzen na řádce 25. Následně se zpětně ověřuje shoda. Tedy v dokumentu vpravo od 25. řádky do 20. se ověřuje shoda k řádkám 18 až 13 v levém dokumentu.

Oba tyto problémy je možné řešit odhalením důvodu neshody stránek a vhodnou generalizací šablony.

Obrázek(viz obr. 2.5 str. 11) ukazuje, jak se s těmito problémy vypořádává algoritmus ROAD RUNNER(viz část 4.1 str. 19). V prvním případě neshody TAGu se jedná o nepovinnou část. v druhém případě je detekován seznam.

2.10 Reprezentace úlohy

Je několik způsobů, jak k problému přistupovat. Základní myšlenka je vždy velice podobná. Jelikož webová stránka má sama formu stromové struktury, je možné po-



Obrázek 2.5: Ukázka průchodu algoritmem ROAD RUNNER[11].

užít pro reprezentaci šablony AND–OR stromy. Mají dobrou vyjadřovací schopnost, která dokáže obsáhnout vše potřebné.

3 Strojové učení a zpracování přirozeného jazyka v oblasti extrakce internetových fór

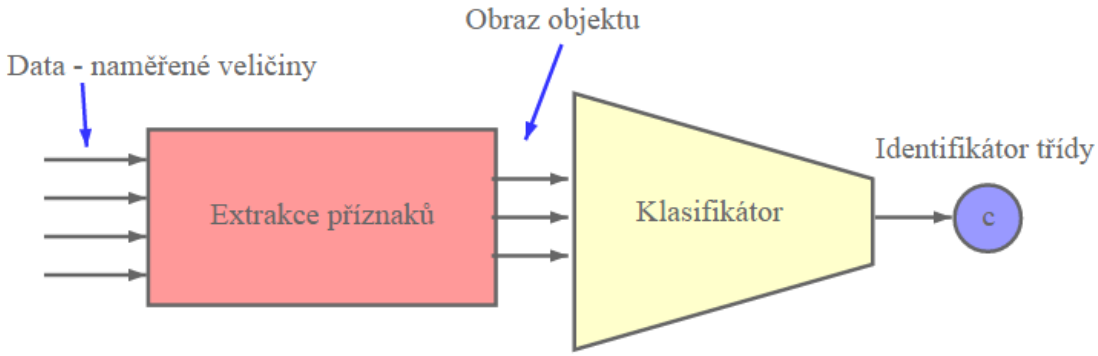
Jelikož tato práce se soustředí hlavně na extrakci obsahu diskuzních fór, budeme chtít nějakou formou strojového zpracování automaticky vyhledávat, jaká část vy-extrahovaných příznaků nás zajímá vzhledem ke specializaci vyvíjeného softwaru.

Nejprve bude popsána obecná úloha klasifikace. Dále princip entropie a její význam v této úloze a následně možný způsob reprezentace textů a dokumentů a jazykových modelů v této oblasti. Na konec budou zahrnuty možnosti měření úspěšnosti.

3.1 Klasifikace

Obecná úloha klasifikace se zabývá extrakcí příznaků z naměřených veličin. Tímto vzniká obraz naměřeného objektu. Následně je tento obraz předložen klasifikátoru, který přiřadí k objektu identifikátor třídy (viz obr. 3.1 str. 12).

Klasifikátoru jsou předkládány objekty, u kterých je známá třída. Ten se z těchto předložených označených objektů naučí, jaké hodnoty příznaků jsou charakteristické pro jednotlivé třídy. Podle těchto závislostí následně zařazuje neznámé objekty. Konkrétní způsoby, formy a metody použité klasifikátorem se mohou lišit.



Obrázek 3.1: Obecná úloha klasifikace

3.2 Entropie

Entropie označuje míru neuspořádanosti nebo neurčitosti. Čím je systém více predikovatelný, je hodnota entropie nižší. Entropie je maximální u rovnoměrného rozdělení. Je možné ji vyjádřit následujícím vztahem (viz vzorec 3.1 str. 13).

$$H(S) = -\sum_{i=1}^{n} P(s_i) log_2 P(S_i) \tag{3.1}$$

Maximum entropy klasifikace Maximum entropy klasifikátor je pravděpodobnostní klasifikátor. Na rozdíl od naivního bayesovského klasifikátoru nepředpokládá nezávislost jednotlivých příznaků. Maximum entropy klasifikátor je založený na principu maximalizace entropie, při čemž ze všech možných modelů, které odpovídají trénovacím datům je vybrán ten, který má největší entropii. Tento typ klasifikátoru může být velice dobře využit při klasifikačních úlohách textu, jako je například detekce jazyka, klasifikace tématu, analýza sentimentu a další. [16][3]

3.3 Bag-of-words

Jedná se o způsob reprezentace dokumentu, založeném na přítomnosti slov ze slovníku v konkrétním textu. Často se k takové reprezentaci používá vektor velikosti slovníku, ve kterém je text napsán, a poté je vytvořen prostým zobrazením přítomnosti či nepřítomnosti jednotlivých slov. Jedničkou v konkrétní složce, pokud slovo slovníku je v konkrétním textu zastoupeno. Nulou, pokud v textu slovo použito není. [9] Následně je možné tento vektor použít ke klasifikaci textů. [17]

Pro zlepšení výsledků je možné model modifikovat tak, že kromě zachycení pouhé existence uloží také počet výskytů.

Pro demonstraci je uveden příklad.

Mějme tyto věty:
1 : *Alice ráda sleduje filmy v televizi.*
2 : *Bob rád chodí na filmy do kina.*

Slovník utvořený z této množiny by vypadal následovně (pro názornost budou ve slovníku použity základní tvary slov).

$w = \{Alice, mít\ rád, sledovat, filmy, v, televize, Bob, chodit, na, do, kino\}$

Bag of words reprezentace těchto dvou vět je znázorněna v následující tabulce (viz tab. 3.1 str. 14).

n	Alice	mít rád	sledovat	filmy	v	televize	bob	chodit	na	do	kino
1	1	1	1	1	1	1	0	0	0	0	0
2	0	1	0	1	0	0	1	1	1	1	1

Tabulka 3.1: Ukázka interpretace vět bag of words

S těmito vektory je možné následně pracovat. Například je možné na tomto principu vytvořit klasifikátor textů. Pokud bychom například měli větší množství takovýchto vět, které by byly přiřazeny do kategorií, bylo by možné pomocí nich natrénovat klasifikátor pro pozdější zařazování neznámých textů do těchto tříd.

Dimenze vektoru je určena slovníkem. v tomto příkladě byl použit slovník vytvořený z jednotlivých slov obsažených v množině. Proto název bag of words. Je však možné tento přístup modifikovat a používat místo slov slovní nebo znakové n-gramy (viz část 3.4 str. 15).

Často se pro reprezentaci vektoru používají metody, které dovolují ukládat pouze nenulové dimenze konkrétních vektorů. Může to být například mapa, kde klíčem je slovo ve slovníku. Tento přístup přinese značnou úsporu paměti.

Samozřejmě poměrná jednoduchost této metody je na druhé straně vyvažována několika nevýhodami. Například přicházíme touto reprezentací o původní pořadí slov, avšak díky této abstrakci je možné s dokumenty dále pracovat a v některých úlohách tento nedostatek není příliš relevantní.

3.4 N-gramy

N-gram je obecně sled po sobě jdoucích složek posloupnosti. Pokud se jedná o n-gramy textu, je možné elementární položku definovat několika způsoby. Mezi nejvíce používané patří znakové a slovní n-gramy.

N-gramy je možné použít pro vytvoření klasifikátoru textu. Pokud bude slovní zásoba a skladba věty dostatečně odlišná, například, jsou-li texty z jiných domén

nebo je použita rozdílná forma jazyka [1].

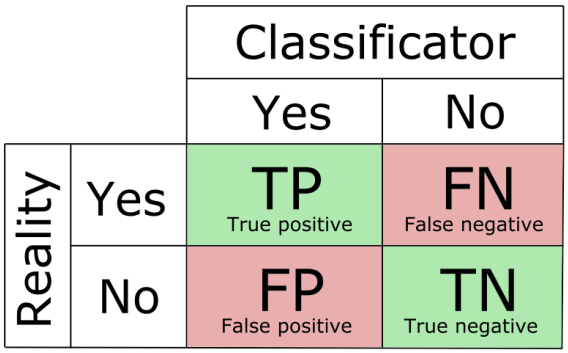
Slovní n-gramy Jsou tvořeny po sobě jdoucí sekvencí slov. Jako oddělovač slov je možné použít například mezery nebo čárky ve větách, jako oddělovač vět tečky. I tyto znaky však mohou být zahrnuty do n-gramů. Mohou přidat informaci navíc. Často je také na začátek a konec vět přidáván speciální symbol, který umožní zachytit slova stojící na začátku a konci vět.

Znakové n-gramy Pokud se jedná o znakové n-gramy, nejmenší jednotkou je znak. Model tedy následně pracuje se n-ticemi znaků. Velikost n-tice může být různá. v takovém případě se často nepoužívá mezera jako oddělovač slov, ale je součástí n-gramu jako ostatní znaky. Takový model dobře zachytí začátky a konce slov. Je tedy často vidět opakující se předpony, koncovky a přípony.

[1] Například spisovná nebo hovorová čeština.

3.5 Ověření úspěšnosti klasifikátoru

Shodu klasifikátoru s očekávanými hodnotami lze rozdělit do čtyř skupin (viz obr. 3.2 str. 16). Zelené kvadranty představují shodu klasifikátoru s očekávanými hodnotami. Červené potom neshodu. v levém spodním kvadrantu jsou případy, které klasifikátor označil jako příslušné do třídy avšak nemělo tomu tak být (FP). v pravém horním kvadrantu jsou pak případy, které klasifikátor neodhalil, ale měl (FN). [8][12]



Obrázek 3.2: Kontingenční tabulka klasifikace a očekávaných hodnot

Accuracy Accuracy vyjadřuje, jaká část predikce byla provedena správně. Lze ji vypočítat pomocí následujícího vztahu (viz vzorec 3.2 str. 16).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.2}$$

Precision Preccision, česky přesnost, vyjadřuje, jaká část objektů, pozitivně označených klasifikátorem, byla označena správně. Lze ji vypočítat pomocí následujícího vztahu(viz vzorec 3.3 str. 16).

$$Preccision = \frac{TP}{TP + FP} \tag{3.3}$$

Recall Recall, česky úplnost, vyjadřuje, jaká část objektů patřící do třídy byla zachycena klasifikátorem. Lze ji vypočítat pomocí následujícího vzorce(viz vzorec 3.4 str. 17).

Protože český ekvivalent pro anglické **Accuracy** a **Precision** je stejný (přesnost), bude v textu dále používáno anglických označení.

$$Recal = \frac{TP}{TP + FN} \tag{3.4}$$

F-míra Protože lze dosáhnout velmi vysoké precision, pokud bychom vybrali jen malý zlomek případů, u kterých jsme si opravdu jistí, popřípadě velmi vysoké recall, pokud bychom označili všechny objekty jako patřící do třídy, je nevhodné orientovat se pouze podle jedné z nich. Oba tyto hraniční případy nejsou žádané. Pro hodnocení úspěšnosti je často potřeba pouze jedno konkrétní číslo, aby bylo možné podle něj provádět optimalizace. Pro tyto účely se používá tak zvaná F-míra, která je harmonickým průměrem Recall a Precision (viz vzorec 3.5 str. 17).

$$F_{maesure} = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{3.5}$$

Klasifikace do více tříd Pokud je třeba počítat tyto míry pro klasifikaci do více tříd, je možno použít jednu z následujících metod.

Mikro průměr Mikro průměr jakékoli z uvedených metrik úspěšnosti lze vypočítat z agregované konfidenční matice pomocí následujícího vztahu (viz vzorec 3.6 str. 17).

$$B_{micro} = B\left(\sum_{i=1}^{q} tp_i, \sum_{i=1}^{q} tn_i, \sum_{i=1}^{q} fp_i, \sum_{i=1}^{q} fn_i\right) \tag{3.6}$$

Makro průměr Makro průměr jakékoli z uvedených metrik úspěšnosti lze vypočítat z dílčích úspěšností klasifikací do jednotlivých tříd dle následujícího vztahu (viz vzorec 3.7 str. 17).

$$B_{macro} = \frac{1}{q} \cdot \sum_{i=1}^{q} B\left(tp_i, tn_i, fp_i, fn_i\right) \tag{3.7}$$

3.6 Matice záměn

Pokud je klasifikováno do několika tříd, je vhodné používat matici záměn, která nemá agregované četnosti. z této matice lze vypočítat nejrůznější typy metrik. Touto

formou uložení není ztracena žádná informace, která by mohla vznikat například při agregaci true-negative klasifikace (viz tab. 3.2 str. 18).

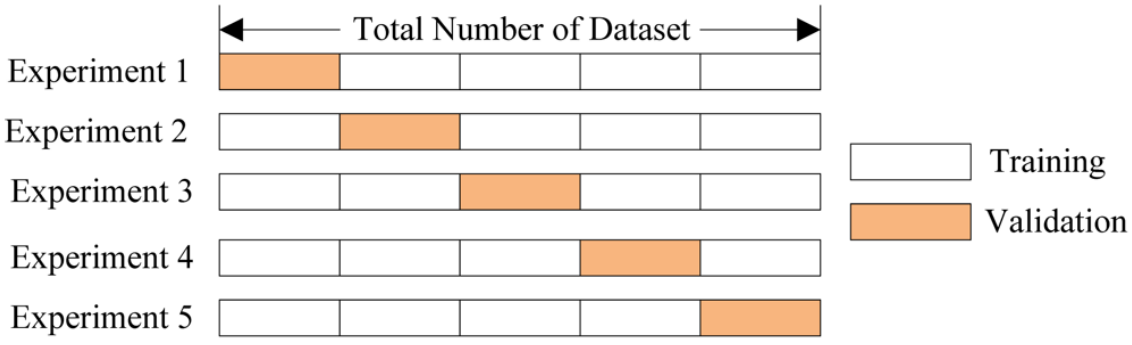
	Classificator			
	NICK	DATE	COMM	NORM
NICK	1826	0	293	31
DATE	0	2016	0	223
COMM	142	0	1781	79
NORM	20	144	139	3190

Tabulka 3.2: Ukázka matice záměn

3.7 Křížová validace

Aby bylo jisté, že není žádným způsobem ovlivňován výsledek vyčíslení úspěšnosti, musí být zajištěno, že jsou striktně oddělena trénovací data od testovacích.

Aby bylo možné použít všechna data jak pro trénování tak pro testování, často se používá technika křížové validace, při které se vždy systém natrénuje na části dat, následně se ověří na datech, která byla odložena právě pro tento účel. Následně se proces opakuje, avšak je vybrána jiná část dat. Takovýmto způsobem je možné použít postupně pro testování všechna data. (viz obr. 3.3 str. 18)



Obrázek 3.3: Ilustrace křížové validace [1]

4 Existující systémy pro extrakci informací

V následující kapitole jsou popsány existující projekty zabývající se problematikou extrakce dynamických dat[1] z webových stránek a následně systémy, které jsou navrženy pro získávání informací z diskuzních fór.

4.1 RoadRunner

Tato metoda extrahuje data z webových stránek bez jakéhokoli zásahu člověka. Používá algoritmus, který na základě shodných částí webových stránek rozpozná, co je součástí šablony, kterou poté postupně generalizuje. Na jejím základě potom dokáže tvořit datový soubor obsahující všechna dynamicky zobrazovaná data v množině předložených stránek. Jeho nevýhodou je, že počítá s validním xHTML zdrojovým kódem[11].

4.2 TISP

Tato metoda vytváří šablonu porovnáváním podobností a rozdílností tabulek. Podobnosti slouží k tvorbě popisků tříd a rozdílnosti jsou brány jako obsah tabulky. Pracuje na principu porovnávání stromů tvořící strukturu Document Object Modelu. Pomocí několika předdefinovaných tabulkových šablon a regulárních výrazů se snaží TISP vpasovat tabulku do těchto šablon.

Zvládá vnořené tabulky stejně tak dobře jako variabilitu v tabulkové struktuře, například nepovinné buňky. Celý proces je automatický. Uvádí se, že je schopen pracovat s 94.5 F-measure na testovacích datech. TISP je ale systém výhradně pracující pouze s tabulkami [15].

[1] To jsou data, která jsou zobrazována šablonou

4.3 IePAD

IePAD - Information Extraction based on PAttern Discovery Tato metoda je jedna z prvních mezi těmi, které se zabývají extrakcí informace. Využívá toho, že webové stránky obsahují rozdílná data. Jsou však vypisována použitím stejné šablony pro přehledné zobrazení. Může být tedy odhalen opakující se vzor, který povede k získání šablony. IePAD používá datovou strukturu zvanou PAT strom, který je binární suffixový strom, aby odhalil opakující se vzory na webové stránce[15][10].

4.4 ExAlg

Metoda založená na extrakci šablony z výsledné stránky. Probíhá ve dvou krocích. První je ECGM - Equivalence Class Generation Stage, která hledá třídy ekvivalence. To je množina tokenů mající stejnou frekvenci výskytu v každé stránce. Stará se o to modul FindEquiv. Může být nalezeno spousta tříd ekvivalence, ExAlg zahrne ale jen ty, které se vyskytují v převážné většině stránek.

Takové třídy ekvivalence jsou zvány Large and Frequently Occurring Equivalence Classes (LFEQs). Ty jsou generovány stejným typovým konstruktorem v šabloně[15] [7].

4.5 DeLa

Tato metoda je rozšíření IePAD. Metoda rovněž pracuje bez zásahu do procesu extrakce, generalizuje pravidla a pracuje s vnořenými objekty. Vytváření šablony v DeLa probíhá ve dvou krocích. DSE - Data-rich section Extraction je navržen tak, aby vyhledal sekce, které obsahují jakákoli data. Tento proces je založen na porovnávání dvou DOM stromů pro dvě webové stránky ze stejného webového serveru a odstraňování elementů se stejným obsahem podstromu.

V druhé části se odhalují opakující se vzory za pomoci suffixových stromů. Také obsahuje několik heuristik snažících se vytvořit popisky pro data založené na maximálním prefixu a maximálním suffixu stejném pro všechny atributy objektů[15].

4.6 FeedExtract

V návaznosti na tento projekt, který probíhal na Západočeské univerzitě v Plzni v minulých letech, byla zadána má práce. Tento projekt pracuje s příznaky extrahovanými z jednotlivých částí fóra, na které později používá MaxEnt klasifikátor [2]. Obsahuje trénovací korpus, který vznikl rovněž na ZČU. Jeho velikost je však velice malá. Obsahuje pouze něco málo přes 2000 označených trénovacích stránek.

4.6.1 Trénovací data

Projekt byl vyvíjen několika skupinami v delším časovém období, obsahuje proto velmi rozdílné přístupy k trénovacím datům. Terminologicky jsou v projektu děleny na StaticPages a DynamicPages.

StaticPages je reprezentace označkovaných dat na velice nízké úrovni. Je vytvořena samostatná třída pro jednotlivé webové stránky, která obsahuje konkrétní indexy jednotlivých elementů. Každá taková třída je svázána právě s jedním souborem, který reprezentuje právě jednu webovou stránku (viz obr. 4.1 str. 21). Obsahují tedy řádově maximálně desítky příspěvků. Takových tříd je v projektu 30.

```
public class ForumGentooLong extends TrainingPageStatic {
    @Override
    public int[] setCommentIndex() {
        int[] indecis = {156, 216, 296, 360, 417, 475, 552, 623, 691, 774, 852, 934,
                1007, 1070, 1120, 1210, 1265, 1364, 1414, 1496, 1546, 1607, 1661, 1711, 1770};
        return indecis;
    }
    @Override
    public int[][] setFinalIndex() {
        int[][] indecis = {{4, 22, 31}, {4, 23, 32}, {4, 22, 31}, {4, 23, 32}, {4, 23, 32}, {4, 22, 31},
                {4, 22, 31}, {4, 23, 32}, {4, 22, 31}, {4, 23, 32}, {4, 22, 31}, {4, 23, 32}, {4, 22, 31},
                {4, 22, 31}, {4, 23, 32}, {4, 23, 32}, {4, 22, 31}, {4, 22, 31}, {4, 23, 32}, {4, 22, 31},
                {4, 22, 31}, {4, 22, 31}, {4, 22, 31}, {4, 23, 32}, {4, 22, 31}};
        return indecis;
    }
    @Override
    public String setFile() { return "forumGentoo-long.txt"; }

    @Override
    public String getType() { return "gentoo"; }
}
```

Obrázek 4.1: Ukázka třídy, která používá metodu StaticPages

[2] Tento typ klasifikátoru byl popsán v předchozí kapitole(viz část 3.2 str. 13)

DynamicPages Tento novější přístup používá skripty v jazyku Perl. Takto načítaná označená data představují převážnou většinu všech trénovacích dat. Celkem je zahrnuto tímto způsobem 2033 webových stránek z 57 webových serverů.

4.6.2 Rozpoznání příspěvků fóra

Je definováno několik příznaků, které jsou brány v potaz při rozpoznávání jednotlivých příspěvků a jejich částí. Například:

- počet shodných atributů se sourozenci,
- počet sourozenců,
- existenci atributu ID,
- existenci atributu class,
- existenci atributu ID u rodiče,
- celkový počet podřadných elementů v DOM,
- ověření shody jména elementu se sourozenci,
- shodu textu s konkrétními tvary data pomocí regulárních výrazů.

4.7 Project Extraction of Web Discussion

Projekt z košické univerzity z oddělení kybernetiky a umělé inteligence. Jedná se o pravidlově řízený přístup[14]. V tomto přístupu jsou člověkem vytvářena pravidla, která vychází ze znalosti charakteru příspěvků webových diskuzí. Pomocí těchto pravidel je poté vytvořen koeficient (viz vzorec 4.1 str. 22), který by měl vypovídat o pravděpodobnosti, že element patří do diskuzního fóra.

$$k = \frac{Z \cdot I \cdot B \cdot T}{L + P} \tag{4.1}$$

Pracuje s atributy :

- počet symbolů ve zkoumané oblasti – Z,
- počet vykřičníků ve zkoumané oblasti – I,
- počet elementů
 ve zkoumané oblasti – B,
- počet časových značek ve zkoumané oblasti – T,
- počet odkazů ve zkoumané oblasti – L,
- počet elementů <P>ve zkoumané oblasti – P.

5 Návrh nového systému

Jelikož neexistuje velké množství systémů, které by se zabývaly jen a pouze extrakcí textů z webových diskuzí, je mnoho neprozkoumaných oblastí. Předchozí kapitola byla věnována také rozboru existujících systémů pro extrakci obecných dat z webových stránek. Také bylo zmíněno několik faktů, která se webových fór týkají.

Pokud máme vylepšit stávající systém, který byl navržen na ZČU, je třeba se zamyslet nad tím, jak pracuje a jaké by mohly být jeho nedostatky.

Architektura Vzhledem k tomu, že zkoumaný systém nemá ucelenou architekturu a vznikal v několika po sobě jdoucích letech několika vývojovými týmy, je vhodné začít s vývojem od počátku s jednotnou architekturou. Bude vytvořen návrh architektury a podle něj bude následně systém vyvinut.

Trénovací data Jelikož v původním projektu je velice malá a mnohdy špatně označená databáze dat, bylo by vhodné tuto trénovací množinu rozšířit, aby bylo možné efektivně natrénovat klasifikátor.

Vzhledem k navrženému přístupu extrakce dynamických dat, není vyhovující, že z jednotlivých webových fór je k dispozici pouze velice omezené množství stránek, řádově několik jednotek, maximálně desítek. Objevují se však také webové servery, ze kterých je poskytnuta pouze jedna stránka, což je pro analýzu šablony stránky naprosto nedostačující.

Způsob použitý v projektu také není příliš vhodný a nezapadá do systému jako celku. Součástí nového systému tedy bude mnohem větší korpus dat, která budou získána automatickým procesem z veřejných diskuzí, následným označením požadovaných dat formou XPath. To umožní vytvořit řádově rozsáhlejší korpus, než jaký byl v původním projektu.

Význam textu příspěvku V referenčním systému je používána jistá forma analýzy textu. Nejsou však žádné reference o tom, jaké metody analýzy textu byly použity, ani proč byly zvoleny zrovna tyto konkrétní metody a nastavení. Systém není prakticky popsán, až na starou dokumentaci. Program byl však poté dále upravován a změny již dokumentovány nebyly. Zmíněná dokumentace poskytuje tedy

pouze částečný pohled na aplikaci. Také komentáře v kódu programu jsou velice sporé a neposkytují příliš dobrý pohled na to, jak funguje systém jako celek, jaké analýzy provádí a proč. Navržený systém tedy bude vytvořen tak, aby umožnil rozsáhlejší analýzu a testy. Tyto experimenty budou dobře zdokumentovány.

Je moderním trendem, že všechny podobné úlohy jsou řešeny statistickými modely. v tak komplexní úloze, jako je tato, je sice možné vymyslet pravidlově řízený systém, ale je nemožné, aby fungoval univerzálně na celý web nezávisle na doméně, strukturách konkrétní stránky nebo jazyku, ve kterém je diskuze vedena.

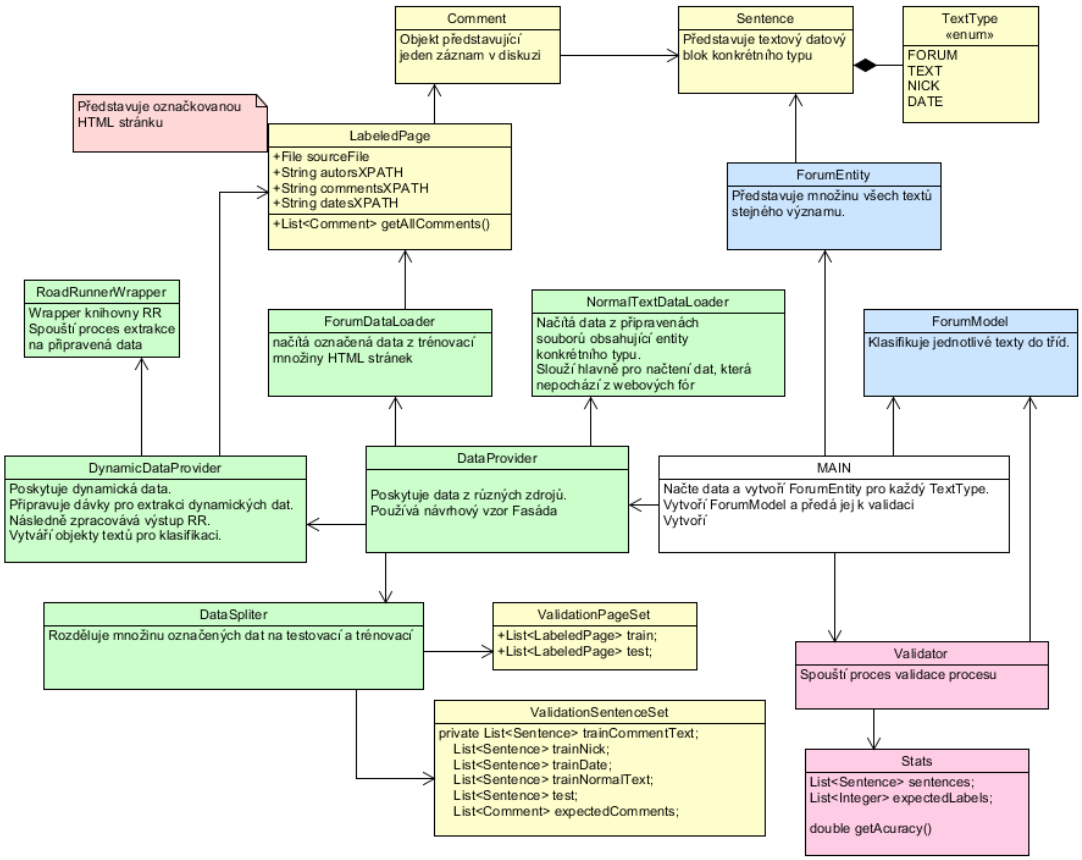
Největší úspěšnosti bude podle mého odhadu docíleno použitím statistických metod v celém procesu. Pokud bude program navržen s použitím těchto metod, výsledek bude přesný v závislosti na trénovacích datech, která takovému systému poskytneme.

5.1 Architektura systému

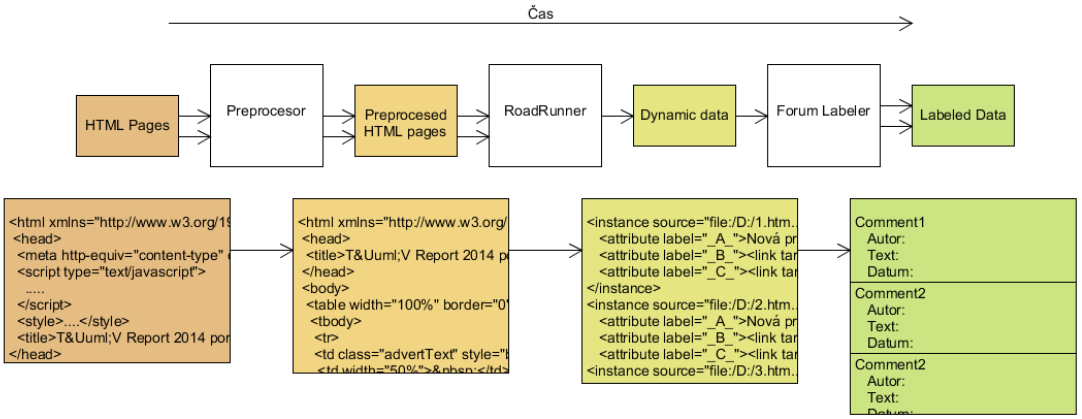
Proces extrakce je možné rozdělit do několika fází. Předpokládejme, že máme soubor webových stránek stažených z webu ve formátu HTML. Bude třeba z takového souboru **extrahovat všechen text, který má význam analyzovat** pro podezření, že se jedná o cenná data. Po získání takových dat je bude potřeba **otestovat po významové stránce**. Dá se předpokládat využití všech dostupných informací, a proto budeme chtít rozeznávat nejen samotné texty webových fór, ale také **identifikovat autora příspěvku, či datum jeho vytvoření**. a následně tyto označené informace **uložit do souboru** pro pozdější zpracování. Na obrázku (viz obr. 5.1 str. 26) je znázorněn návrh architektury systému. Na obrázku (viz obr. 5.2 str. 26) je pak zobrazen tok dat a jejich postupné zpracovávání.

5.1.1 Preprocesor

Preprocesor odstraňuje z webových stránek elementy, které nejsou vhodné pro zpracování knihovnou RoadRuner. Jsou to hlavně skripty a kaskádové styly umístěné ve zdrojovém kódu.



Obrázek 5.1: Návrh architektury



Obrázek 5.2: DataFlow

5.1.2 Dávkové zpracování RR

Třída **DynamicDataProvider** vytváří dávky stažených souborů webových diskuzí. Spustí proces preprocesingu a následně volá **RoadRunnerWrapper**, která je používána pro extrakci dynamických dat[1]. Ta používá knihovnu RoadRunner. Na předem připravené a předzpracované dávky souborů spouští proces extrakce.

5.1.3 LabeledPage

Tato třída reprezentuje označenou webovou stránku. Je v ní uložena cesta k souboru stažené stránky a také XPath ke všem označeným datům. Obsahuje metody pro extrakci komentářů a ostatních textů. Pro načtení **LabeledPage** slouží třída **DataProvider**

5.1.4 DataProvider

Třída poskytující data. Přistupuje k souborům na disku a vytváří objekty reprezentující entity na webových fórech. Je konfigurována soubory obsaženými v *etc/dataloader/*. Pomocí třídy **DataSpliter** vytváří množiny trénovacích a testovacích dat. Také implementuje návrhový vzor iterátor a iteruje jednotlivými množinami křížové validace. Používá třídu **DataSpliter**, která obsahuje několik metod pro rozdělení množin dat na trénovací a testovací.

5.1.5 ForumModel

Tato třída se stará o samotnou klasifikaci, používá připravené příznaky a trénuje klasifikátor. Ke klasifikaci je použit maximum entropy klasifikátor z knihovny Brainy
.

5.1.6 Validator

Tato třída se stará o ověření správnosti procesu. Ověřuje schopnost systému spárovat dynamická data získána RR s označenými trénovacími daty. Také ukládá po-

[1]Data, která jsou zobrazována pomocí šablony na webové stránce

drobné informace o párování do souborů (viz část 5.7 str. 31). Používá **ForumModel** k označení testovacích dat a následně porovnává s očekávanými hodnotami. Výsledky klasifikace předává do konstruktoru třídy **Stats**. Ta vytvoří konfidenční matici a následně umožňuje svými metodami počítat metriky(accuracy, recall, precision, f-maesure) umožňující lépe zkoumat, jak je proces extrakce úspěšný.

5.2 Křížová validace

Jelikož rozpoznávání příspěvků na fórech není závislé jen na konkrétních textech, ale také na formě interpretace dat, například tvaru data, či jedna množina uživatelských jmen v rámci jednoho webového serveru, je nutné testovat na úrovni rozdílných fór.

Bude tedy počítána úspěšnost správného rozpoznání konkrétních příspěvků na konkrétním webovém serveru za předpokladu, že systém je natrénován na ostatních. Postupně se takto otestují fóra z jednotlivých webových serverů.

5.3 Rozšíření trénovacího korpusu

Toto bylo jednou z navrhovaných úprav. Bude vytvořen nástroj pro snadnou správu trénovacích dat. Formou konfiguračního souboru budou přidávány cesty k jednotlivým složkám, které budou obsahovat webové stránky z jednotlivých webových serverů. Součástí konfiguračního souboru budou také XPath k jednotlivým příspěvkům a jejich konkrétním částem.

V souboru **etc/dataLoader/config.txt** bude uložena cesta ke složce se staženými stránkami. v této složce pak budou jednotlivé složky s konkrétními soubory webových stránek z jednotlivých webových serverů. Ve složce **etc/dataLoader/sites** pak bude vždy soubor odpovídajícího jména s XPath k jednotlivým datovým blokům.

XPath v comments označuje jednotlivé komentáře. Následují relativní cesty k jednotlivým informacím.

```
comments=//li[@class="received"]
text=//div[@class="post"]/div/allText()
author=//h5/a/allText()
date=//li[@class="date"]//allText()
```

Obrázek 5.3: Ukázka z konfiguračního souboru konkrétního webového serveru.

5.4 Extrakce dynamických dat z webového fóra

Pro extrakci dynamických dat z webového fóra je možné použít jakýkoli funkční a vhodný systém pro extrakci z webových stránek. Převážná většina je založena na porovnávání několika stránek z jednoho serveru a na základě rozdílů mezi nimi vytváří šablonu a extrahuje data. Toto je vlastně statistický přístup k problému. a univerzálně bude fungovat napříč všemi webovými servery. Vzhledem k předpokládanému využití navrhovaného systému je kladen velký důraz na rychlost celého procesu. Nejlepší volbou, pokud vezmeme v úvahu všechny potřebné parametry, bude použít ROAD RUNNER (dále jen RR). Je mnohem rychlejší při zpracování velkého množství stránek, než jemu podobné systémy, a jeho nedostatky nejsou relevantní pro konkrétní aplikaci na webová fóra. Bude tedy použit tento systém pro extrakci dynamických dat.

Dále bude implementována funkcionalita, která umožní označit typ textu, který extrakce dynamických textů poskytne. v případě webových fór jsou nejvíce žádané informace o uživatelském jménu, datu a textu komentáře. Bude tedy vytvořen klasifikátor, který umožní rozpoznat, zda se jedná o nějakou z těchto informací a pokud ano, označí, o kterou jde.

5.5 Preprocesing

V současné době není zobrazovaná stránka pouze čisté HTML, na kterém by bylo možné rovnou provádět výše uvedené postupy. Mnohdy bývá použito spousta rozšiřujících mechanismů, které dokáží prohlížeče webových stránek využívat. Mohou to být například do stránky vložené styly upravující zobrazení dat nebo různé skripty spouštěné v prohlížeči. Všechny tyto součásti jsou nežádoucí pro zpracování. Je sice teoretická možnost například využívat kaskádové styly pro lepší orientaci na stránce, ale to není případ námi zvoleného postupu. Bude tedy implementován mechanismus, který bude provádět preprocesing na datech, která budou následně zpracovávána

RR.

5.6 Příznaky

Jak již bylo zmíněno v návrhu systému, bude zkoumán význam textu poskytnutý analýzou dynamických dat z webového fóra. Následuje výčet použitých příznaků, které byly navrženy právě za tímto účelem.

5.6.1 Bag-of-words a Bag-of-character-n-grams

Pro zachycení obsahu jednotlivých datových bloků lze dobře použít bag of words (viz část 3.3 str. 13) a bag of character n-grams(viz část 3.4 str. 15). Prostředí webového fóra je však velice rozmanité, například velice problematická by mohla být odlišnost uživatelských jmen. Pokud by byl tvořen vektor pro klasifikaci datového bloku, uživatelská jména by zapříčinila prudký nárůst velikosti slovníku. Proto bude vhodné navíc použít prahovou hranici počtu, pod kterou slovo nebo n-gram nebudou brány v potaz. To zabrání nepoměrně rostoucím paměťovým nárokům, také nebude hrozit nebezpečí přetrénování klasifikátoru.

Budou tedy implementovány příznaky, které se postarají o extrakci vektorů z textu. Následně budou použita pro klasifikaci.

K datu je třeba přistupovat odlišným způsobem. Pokud bychom nechali datum při klasifikaci nezměněné, vznikalo by několik problémů. Jedním z hlavních by bylo to, že by klasifikátor založený na takovém přístupu nezachycoval základní vlastnosti data. Neklasifikoval by datum podle jeho obecného tvaru, ale podle konkrétních tvarů, které byly předloženy při trénování. s velkou pravděpodobností by docházelo k přetrénování. Je tedy vhodné nejdříve předzpracovat text, než z něj jsou extrahovány znakové n-gramy. Pokud se zamění jednotlivé číslice zástupným znakem, n-gramy zachytí obecný tvar data, nikoli konkrétní případy.

5.6.2 Délka klasifikovaného textu

Délka neznámého textu může přispět k lepší oddělitelnosti tříd. Některá uživatelská jména mohou mít podobný charakter, co se týče skladby textu, uživatelská jména

budou však povětšinou kratší. Bude tedy implementován příznak, který bude zachycovat délku textu.

5.6.3 Poměrné zastoupení číslic v textu

Myšlenka je taková, že některé datové bloky, které se objevují, lze rozpoznat podle poměru počtu číslic k celkovému počtu znaků bloku. Konkrétně datum bude z velké části složeno z číslic oproti například běžnému textu nebo uživatelskému jménu.

5.6.4 Shoda části řetězce s regulárním výrazem

Pokud chceme rozpoznávat, zda je klasifikovaný text datum, můžeme definovat příznak složený ze shod částí řetězce s regulárním výrazem. Množina používaných tvarů není příliš rozsáhlá, popřípadě je snadné ji doplnit v případě potřeby.

5.6.5 Klíčová slova

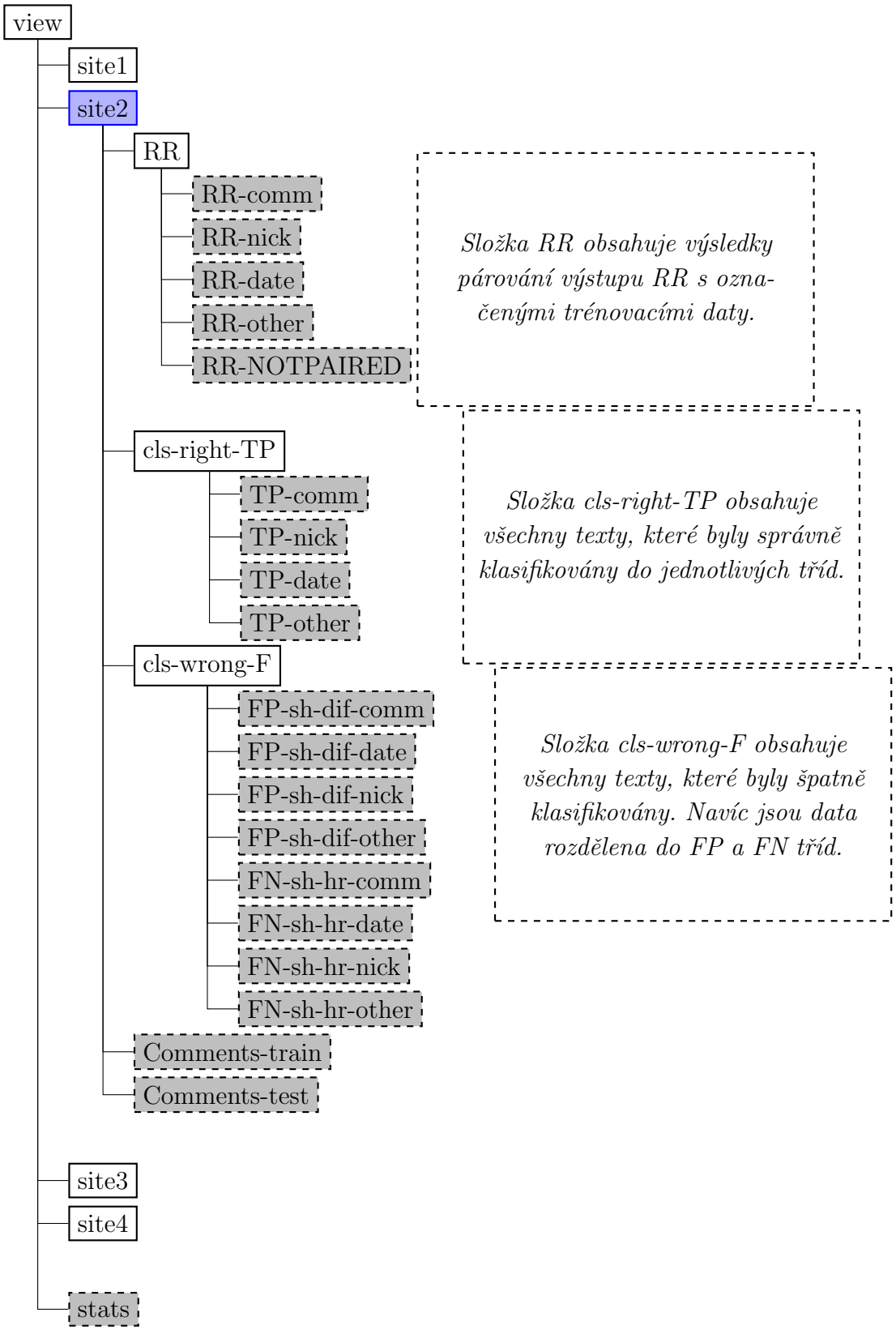
Ve webových fórech a obecně na internetu se ustálilo několik často používaných slov, které jsou často součástí některých bloků. Můžou to být například klíčová slova, které se na fórech stávají součástí elementů obsahující konkrétní údaje.

Například s datem mohou být spojená slova: **zveřejněno:**, **odesláno:**, **publikováno:**. S uživatelským jménem například slovo **Od:**. Jiná klíčová slova mohou být součástí stránkování nebo jiné dynamické navigace. Bude tedy implementován příznak, který bude zkoumat přítomnosti klíčových slov. Více je tato problematika rozepsána později (viz část 6.2 str. 36).

5.7 Přehled nad procesem extrakce

Aby bylo možné rozumným způsobem ladit program při jeho vyvíjení a následně nastavovat jednotlivé proměnné atributy extrakce, bude implementován mechanismus umožňující sledování jednotlivých fází a zpětnou analýzu. v adresářovém stromu bude vždy složka s jednotlivými částmi křížové validace. v ní budou uloženy informace z konkrétních kroků přehledně rozděleny do souborů. (viz obr. 5.4 str. 33)

Také je uchovávána matice záměn, pro následné počítání různých metrik úspěšnosti.

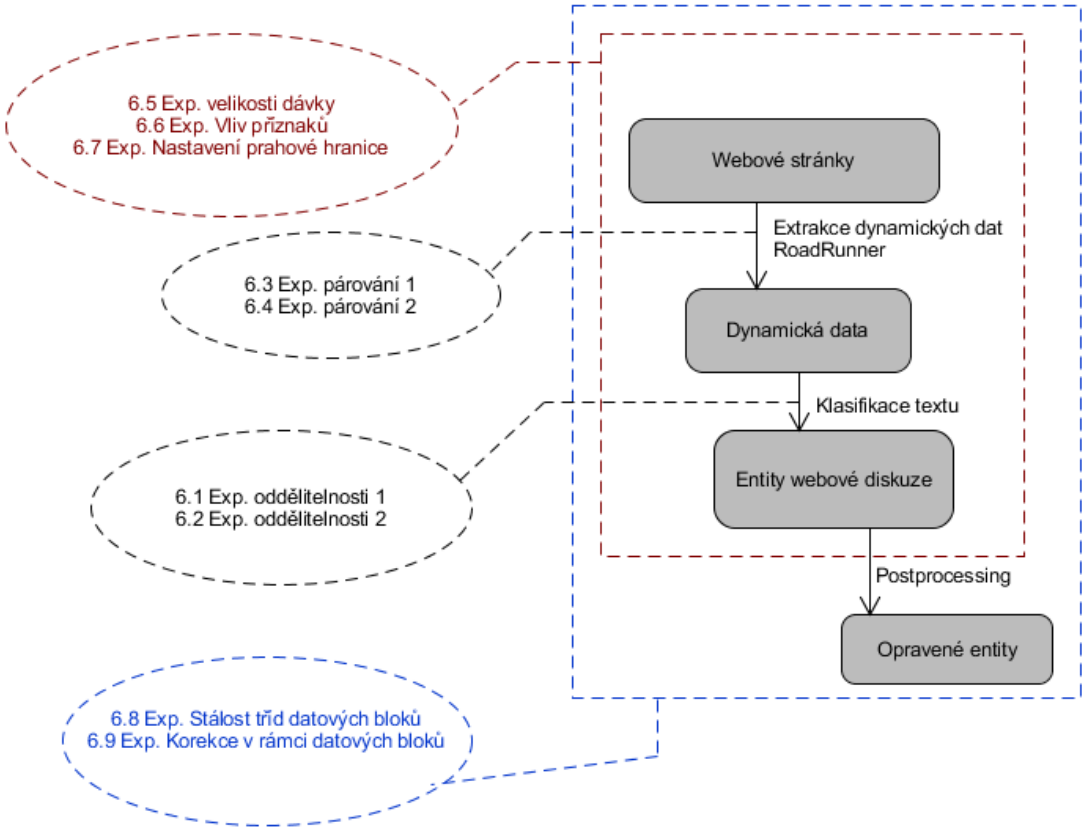


Obrázek 5.4: Schéma stromu zachycující průběh a úspěšnost procesu

6 Úspěšnost systému

Jak již bylo popsáno, navržený systém pracuje v několika krocích. Následující kapitola je věnována experimentování s jednotlivými dílčími operacemi, které jsou postupně s daty prováděny.

Následující obrázek (viz obr. 6.1 str. 34) zachycuje, které experimenty se týkají konkrétních operací.



Obrázek 6.1: Schéma experimentů

Tyto experimenty pomáhaly ověřit, zda je konkrétní elementární část procesu extrakce připravena k použití ve funkčním celku.

Následně je shrnuto, jakých výsledků bylo docíleno, a ze získaných statistik je popsáno, jak jednotlivé dílčí operace přispívají k chybě systému, a jak by bylo možné je vylepšit.

Data z webových diskuzí pro všechny experimenty poskytuje popsaný DataProvider. Poskytuje sice náhodně zamíchaná data ze všech webových serverů, avšak deterministickým způsobem. To tedy umožní tyto experimenty opakovat na stejných datech s různými nastaveními celého procesu extrakce.

6.1 Experiment oddělitelnosti 1

Aby bylo možné získat obecný přehled nad tím, zda je navrhovanými metodami možné oddělit informace typu uživatelské jméno, datum, text příspěvku na diskuzním fóru a běžný text, byl navržen následující experiment.

Popis experimentu Označené webové stránky jsou rozděleny na trénovací a testovací část. Následně je načten běžný text [1], který je rozdělen stejným způsobem. Následně jsou trénovací množiny spojeny a je na nich natrénován navržený klasifikátor. Tomu jsou následně předložena testovací data. Tabulka (viz tab. 6.1 str. 35) ukazuje výsledky experimentu.

	Klasifikátor			
	NICK	DATE	COMM	NORM
NICK	469	14	53	4
DATE	0	540	0	0
COMM	5	1	520	14
NORM	1	0	4	394

Tabulka 6.1: Tabulka záměn pro experiment
F-míra : 0.9533

Závěr experimentu Experiment ukázal (viz tab. 6.1 str. 35), že množiny navrženým postupem lze oddělit. Je však nutné podotknout, že se jednalo o experiment v ideálních podmínkách, protože pro běžný text nebyly použity texty ze stránek s diskuzemi. Dá se předpokládat, že při následném použití narazíme na další překážky, které budou dělat separaci obtížnější.

[1] Pro tento experiment byly použity texty z českých článků na wikipedii. Soubor, který byl použit je přiložen ve složce data/texts.

6.2 Experiment oddělitelnosti 2

První experiment ukázal, že čtveřice uživatelské jméno, datum, text příspěvku a běžný text, jsou od sebe dobře oddělitelné. Jednalo se však o experiment v ideálních podmínkách. Byl tedy navržen další experiment, který měl ověřit separaci na konkrétních webových serverech.

Popis experimentu Oproti prvnímu experimentu se liší v nahrazení textu z wikipedie za obsah z webových fór, který není označen. Všechny bloky, které RR vrátí jsou tedy opět párovány s trénovacími daty a zbytek je použit jako nezajímavý text. Při tomto experimentu byla použita křížová validace (viz část 3.7 str. 18). Více se tedy projevily následky dříve popisovaných problémů.

Průběh experimentu V průběhu experimentu se ukazovalo, že oproti běžnému textu nastává několik problémů. Tabulka záměn (viz tab. 6.2 str. 37) ukazuje výsledky experimentu. Došlo k záměně některých bloků při klasifikaci. Například některá fóra zobrazují částečně maskovanou ip adresu, nebo stránkování, které byly zaměňovány za ostatní třídy. Také byly zaznamenány problémy při záměně některých bloků obsahujících klíčová slova. Například:

Napsal: 23 pro 2014 21:32

Klíčové slovo **Napsal:** není součástí data, avšak je součástí klasifikovaného bloku. To je způsobeno příčinami rozebíranými již dříve.(viz část 6.3 str. 38)

Mezi další patří problém rozeznat správně například informace o uživateli **(1)**, stránkování a navigaci**(2)**, ochranu před roboty**(3)** atp. Například:

- Registrován: 18 zář 2013 21:02 Příspěvky: 184 Bydliště: BRNO **(1)**,
- Stránka 68 z 96 **(2)**,
- Spočítejte toto prosím: 2 + 4 **(3)**,
- 1 starší >**(2)**,
- předchozích 10 **(2)**,

- posledních 10 **(2)**,
- 10019 10018 10017 **(2)**.

	Classificator			
	NICK	DATE	COMM	NORM
NICK	1687	0	179	0
DATE	0	1391	480	0
COMM	80	0	1722	0
NORM	825	233	1465	0

Tabulka 6.2: Tabulka záměn experimentu
F-míra : 0.5407

Závěr experimentu Pokud bychom měli pro trénování klasifikátoru více dat, je pravděpodobné, že by zachytil datové bloky, které jsou součástí některých fór. Problém je v tom, že v trénovací množině není nikdy označen dostatečně podobný text jako nezajímavý datový blok.

Dle mého úsudku, pokud by byla množina trénovacích dat dostatečně veliká, klasifikátor by byl na základě použitých příznaků schopen tento text označit správně. Otázkou zůstává, jak velká by tato množina musela být. Je pravděpodobné, že je konečné množství způsobů, jak řešit například stránkování. Aby klasifikátor vždy správně rozpoznal, že se jedná o stránkování a označil tuto část jako nezajímavý text, bylo by třeba mít v trénovacích datech množinu stránek, která se vzájemně doplňuje právě v těchto způsobech použitého stránkování.

Stejně tak množinu ostatních datových bloků, jako je například ověření robotů, nebo informace o uživatelích, by bylo třeba dostatečně zastoupit v trénovacích datech.

Tyto skupiny představují přibližně 66 % špatně klasifikovaných případů, které byly ze třídy ostatní text.

6.3 Experiment párování výstupu RR s označenými daty 1

Aby bylo možné ověřit úspěšnost systému jako celku, je třeba spárovat výstup RR s trénovacími daty. Teprve poté bude možné zkoumat, zda je klasifikace úspěšná. Byla definována **chyba párování W** (viz vzorec 6.1 str. 38), která zachycuje, jakou část z označených dat se nepodařilo spárovat s výstupem RR.

$$W = \frac{T_{np}}{T_a} \tag{6.1}$$

Kde **W** je definovaná chyba, T_{np} je počet nespárovaných textů a T_a je počet všech textů.

Popis experimentu Dynamická data, která byla extrahována RR jsou postupně porovnávána s označenými prvky na webové stránce a párována.

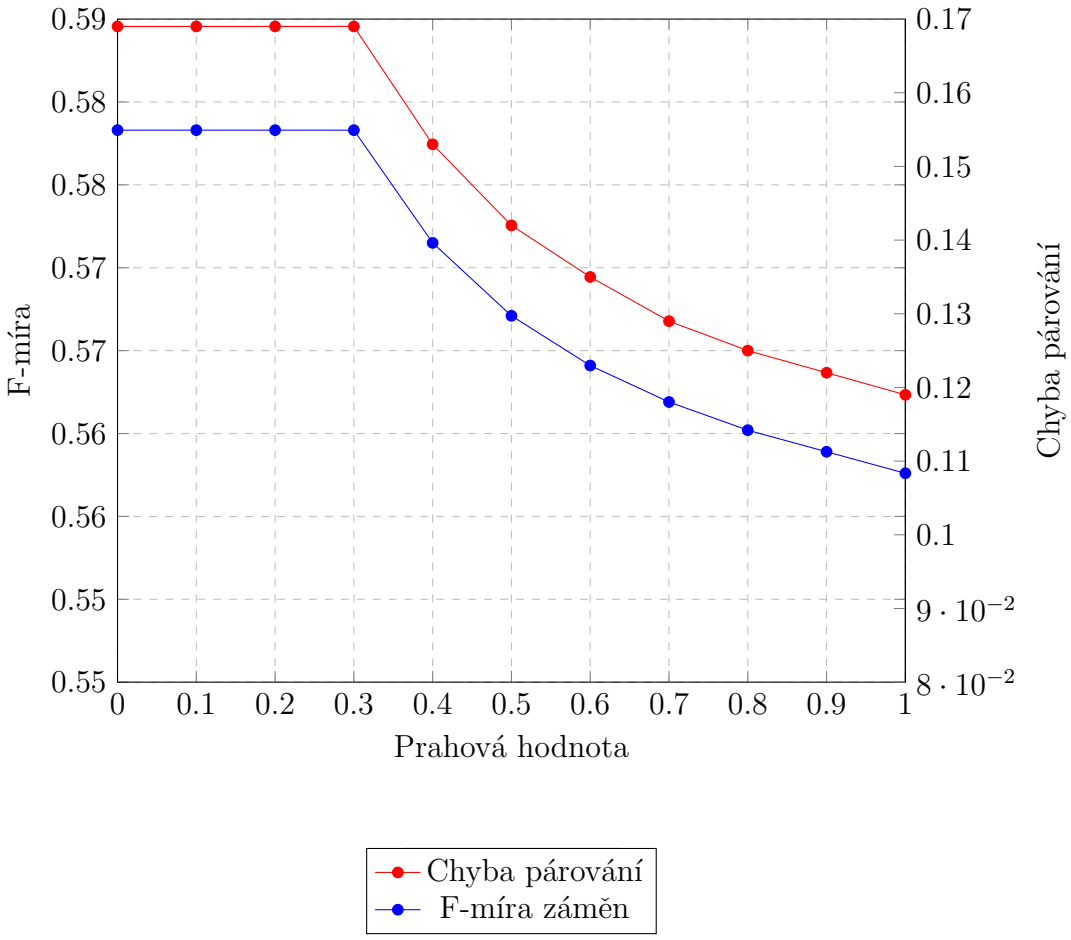
Průběh experimentu Postupně bylo nutné přidat sofistikovanější přístup, než pouhé porovnávání řetězců. Často se totiž stane, že RR rozdělí prvky stránky jiným způsobem, než by bylo třeba. Není však možné tomu zabránit. Je to zapříčiněno hlavně nevhodným návrhem webových stránek (viz část 6.4 str. 40).

Byla tedy použita metrika, která počítá vzdálenost porovnávaných řetězců. Je povoleno pouze odstraňování znaků. Například operace záměny znaku nebo jeho přidání není z logiky věci přípustná. Výsledná vzdálenost je tedy počet znaků, které se musí z řetězce odebrat, aby byly řetězce stejné.

Je možné nastavit prahovou hodnotu, která symbolizuje, kolik procent znaků lze z řetězce odebrat. Následující graf (viz obr. 6.2 str. 39) ukazuje závislost této hranice na úspěšnost systému.

Závěr experimentu V experimentu se ukázalo, že výsledná úspěšnost je přímo závislá na chybě, která vzniká při párování testovacích dat. Prudká změna u hodnoty 0.3 vzniká při překročení hranice, která povolí spárování následující dvojice a jí typově podobných.

Napsal: *23 pro 2014 11:48*



Obrázek 6.2: Závislost f-míry na chybě párování

23 pro 2014 11:48

Závislost úspěšnosti klasifikace na chybě vzniklé při párování se objevuje kvůli nepřítomnosti slova **Napsal:** v trénovacích datech. Pokud by v trénovacích datech bylo toto klíčové slovo obsaženo, dalo by se předpokládat, že tato závislost zmizí.

6.4 Experiment párování výstupu RR s označenými daty 2

Protože předchozí experiment odhalil, že proces extrakce je velice závislý na tom, které HTML tagy jsou použity pro oddělení datových bloků, byl navržen experiment,

který odhalí, které tagy v množině způsobují chybu.

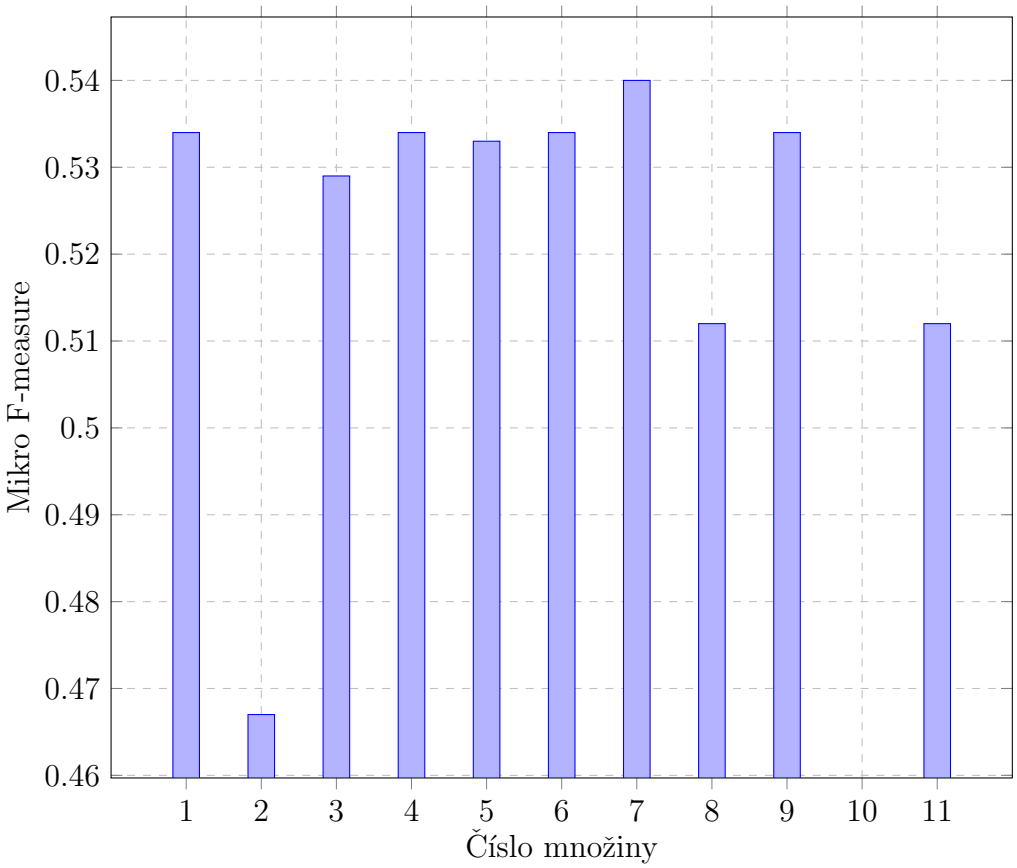
Popis experimentu Postupně bude spouštěn proces extrakce s různou konfigurací RR. Sledovaná bude úspěšnost při párování s jednotlivými označenými prvky.

Průběh experimentu Následující tabulka (viz tab. 6.3 str. 40) a graf (viz obr. 6.3 str. 41) ukazují, že jednotlivé HTML tagy mají opravdu vliv na úspěšnost procesu extrakce. v prvním řádku jsou výsledky extrakce, při které je použita celá testovaná množina tagů a postupně je každý jeden z množiny odebírán. v některých případech se úspěšnost zvýšila, v jiných se markantně zhoršila. v řádku deset, kdy byl odstraněn tag
, byl proces extrakce tak výpočetně náročný, že po více jak několika desetinásobném prodloužení času, který stále neprodukoval výsledek, byl experiment ukončen.

číslo množiny	a	img	font	b	i	strong	p	em	br	span	**F-míra**
1	•	•	•	•	•	•	•	•	•	•	**0.534**
2		•	•	•	•	•	•	•	•	•	**0.467**
3	•		•	•	•	•	•	•	•	•	**0.529**
4	•	•		•	•	•	•	•	•	•	**0.534**
5	•	•	•		•	•	•	•	•	•	**0.533**
6	•	•	•	•		•	•	•	•	•	**0.534**
7	•	•	•	•	•		•	•	•	•	**0.540**
8	•	•	•	•	•	•		•	•	•	**0.512**
9	•	•	•	•	•	•	•		•	•	**0.534**
10	•	•	•	•	•	•	•	•		•	**-**
11	•	•	•	•	•	•	•	•	•		**0.512**

Tabulka 6.3: Tabulka množin použitých HTML tagů

Závěr experimentu Experiment ukázal, které HTML tagy přispívají k lepším výsledkům, a které naopak k chybě. Nutno však podotknout, že jde o výsledky, které se vztahují k jedné konkrétní testované množině. Pokud by se uvažovalo o budoucím rozšiřování trénovací množiny, bylo by nutné experimenty provést znovu. Chyba je důsledkem hlavně nevhodného návrhu webových stránek. Nezřídka jsou



Obrázek 6.3: Naměřené hodnoty f-míry při použití různých množin tagů při extrakci dynamických dat

od sebe významově odlišné informace rozlišeny nebo odděleny pouze formátovacími elementy (<b><i> aj.), nebo jinými tagy, které nejsou určeny k významovému rozdělování dat (např. <img><a>), místo toho, aby byly logicky odlišeny významově ve stromové struktuře.

V důsledku právě popsaného problému je také možné, že bude docházet ke kolizím. Konfigurace, která by byla vhodná na jedné stránce, nemusí už tak dobře fungovat na stránce jiné. Bylo by možné implementovat mechanismus, který by přistupoval k tomuto problému ve více fázích a pokud by extrakce vedla na příliš velké rozdíly v analýze jednotlivých webových serverů, mohl by v průběhu nastavení měnit. To by vyžadovalo mnohem sofistikovanější přístup k tomuto problému. Jelikož se jedná v testovací množině o poměrně malé množství stránek, dále se v této práci tato problematika neřešila.

6.5 Experiment velikost dávky zpracovávané RR

RoadRunner dokáže zpracovávat libovolně velké množiny souborů najednou. Aby bylo možné optimalizovat, jak velké dávky je vhodné předávat k extrakci, byl navržen následující experiment.

Popis experimentu Označené webové stránky budou rozděleny do dávek různých velikostí. Následně budou předkládány k extrakci. Bude měřena úspěšnost jednotlivých konfigurací na stejných datech.

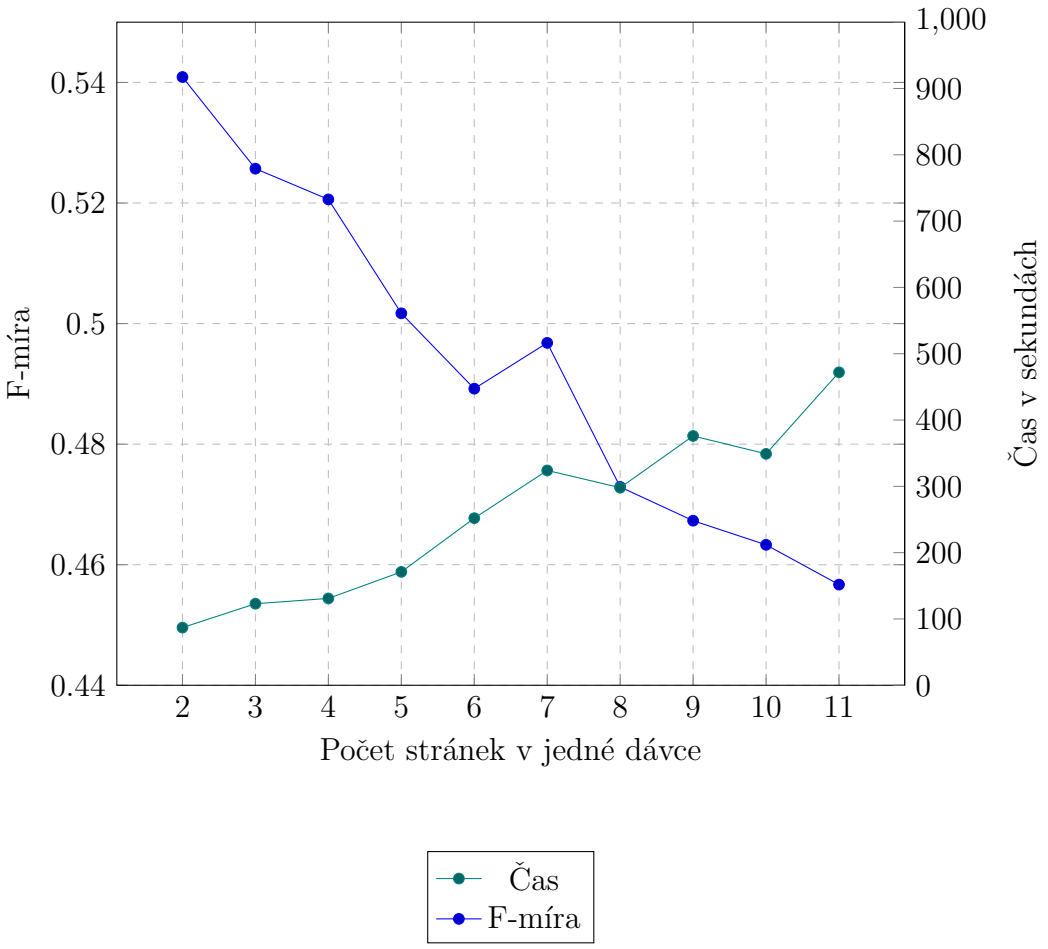
Průběh experimentu Následující tabulka(viz tab. 6.4 str. 42) a graf (viz obr. 6.4 str. 43) ukazují, jaká je závislost mezi úspěšností a časovou náročností celého procesu a velikosti dávky pro RR.

velikost dávky	F-míra	čas [ms]
2	0.5409	87
3	0.5257	123
4	0.5206	131
5	0.5017	171
6	0.4892	252
7	0.4968	324
8	0.4729	298
9	0.4673	376
10	0.4633	349
11	0.4567	472

Tabulka 6.4: Závislost úspěšnosti a času na velikosti dávky

Závěr experimentu Ukázalo se, že velikost dávky má vliv na záměnu textu, který je součástí šablony. Pokud je dávka příliš veliká, více se projeví anomálie na stránce. Například pokud je na diskuzi používaný tzv. softdelete [2] s oznámením,

[2]Způsob mazání, kdy se článek pouze odstraní z veřejného fóra, ale dále zůstává v databázi. Místo něj bývá často zobrazeno oznámení, že příspěvek byl smazán. Tato zpráva má však jinou strukturu, než běžný příspěvek.



Obrázek 6.4: Nameřené hodnoty při zpracování 60 stránek

který má jinou strukturu než běžný příspěvek, je to vnímáno jako narušení šablony a RR označí část jako proměnlivý podstrom. Součástí takového podstromu jsou ale také texty, které by byly normálně zařazeny do šablony. Proto narůstá s velikostí dávky také chybovost hledání šablony.

Byl však zaznamenán také problém, který vzniká při malé velikosti dávky. Pokud jsou v dávce jen dvě stránky, může se stát, že na jedné z nich chybí nějaká informace, která by byla jinak očekávaná. Při výzkumu se objevilo několik případů, kdy je na diskuzním fóru zobrazen příspěvek, ale není uveden autor, ačkoli v jiných případech zobrazen je. v takovém případě rovněž dochází k neshodě šablon. Proces extrakce neselže, avšak existující informace o autorovi, která nemá odpovídající informaci v druhé stránce, bude zanedbána. Řešením by bylo dávky zvětšit. Při testování k tomu docházelo však zřídka a nedostatky větších dávek se více projevily.

Větší dávka má také za následek prodloužení času nutného k nalezení šablony.

6.6 Experiment vliv jednotlivých příznaků

Jelikož bylo navrženo několik příznaků, při tvorbě systému, byl navržen následující experiment, který testuje vhodnost použití těchto příznaků.

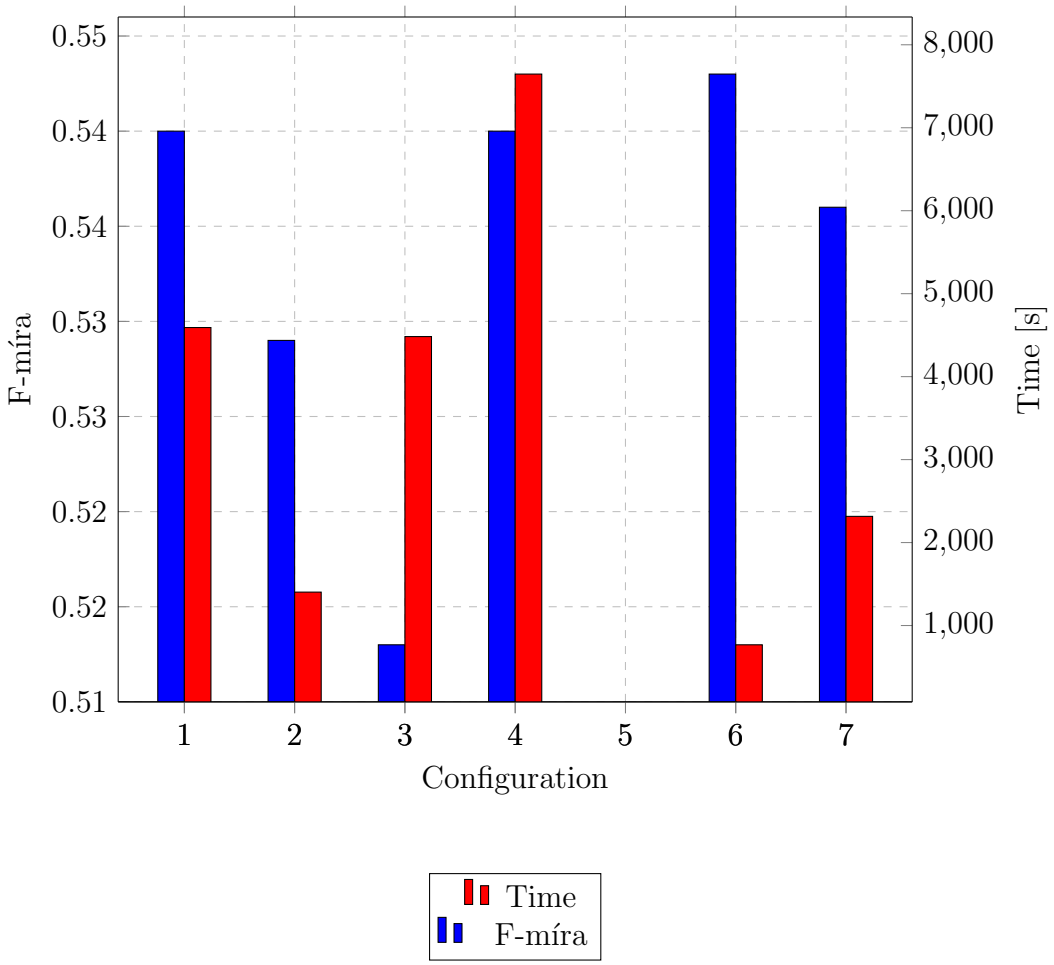
Popis experimentu Postupně bude spouštěn proces extrakce s různými konfiguracemi. Budou porovnávány úspěšnosti při použití jednotlivých příznaků.

Průběh experimentu Následující tabulka (viz tab. 6.5 str. 44) a graf (viz obr. 6.5 str. 45) ukazují, jak přispívají jednotlivé navržené příznaky k celkové úspěšnosti systému. v první konfiguraci jsou použity všechny navržené příznaky, postupně je z množiny každý jeden odebrán a proces extrakce je opakován.

Conf.	Bag of Word	Bag of Character n-grams	Numbers coun	Length	Date	Key Words	**Čas**	**F-míra**
1	•	•	•	•	•	•	**4592**	**0.540**
2		•	•	•	•	•	**1403**	**0.529**
3	•		•	•	•	•	**4483**	**0.513**
4	•	•		•	•	•	**7647**	**0.540**
5	•	•	•		•	•		
6	•	•	•	•		•	**768**	**0.543**
7	•	•	•	•	•		**2316**	**0.536**

Tabulka 6.5: Příspěvky příznaků k úspěšnosti klasifikace

Závěr experimentu Z výsledků je možné postřehnout několik užitečných informací. Například mezi konfigurací 1 a 4 není rozdíl v úspěšnosti, avšak v časové náročnosti je rozdíl markantní. z tohoto můžeme dedukovat závěr, že příznak používající poměrné zastoupení číslic v textu nepřidává na úspěšnosti, avšak jeho použitím se velice snižuje časová náročnost procesu.



Obrázek 6.5: Graf závislosti použitých příznaků na času a úspěšnosti

Experiment ukázal, že některé příznaky jsou velice důležité pro správnou klasifikaci.

Je zde ale také vidět jeden příznak, který do procesu vnáší chybu a jeho nepoužitím proces nejen zlepší svou úspěšnost, ale také se řádově urychlí. Konkrétně se jedná o příznak zachycující přítomnost regulárních výrazů v textu. Myšlenka tohoto příznaku byla jasná, tedy lépe zachytit známé tvary kalendářních údajů. Jasně se ale ukázalo, že myšlenka byla mylná. Příznaky, které byly navrženy pro podobný účel, například poměr zastoupení číslic v textu ve spojení s maskovanými znakovými n-gramy, splní cíl velice dobře s podstatně nižšími časovými nároky. v konfiguraci č. 5 proces trénování klasifikátoru použitou knihovnou nebyl dokončen.

6.7 Experiment nastavení prahové hranice Bag of words

Protože součástí navrženého systému byl příznak Bag-of-words s prahovou hranicí, byl navržen následující experiment, aby odhalil, jaká hranice bude vhodná.

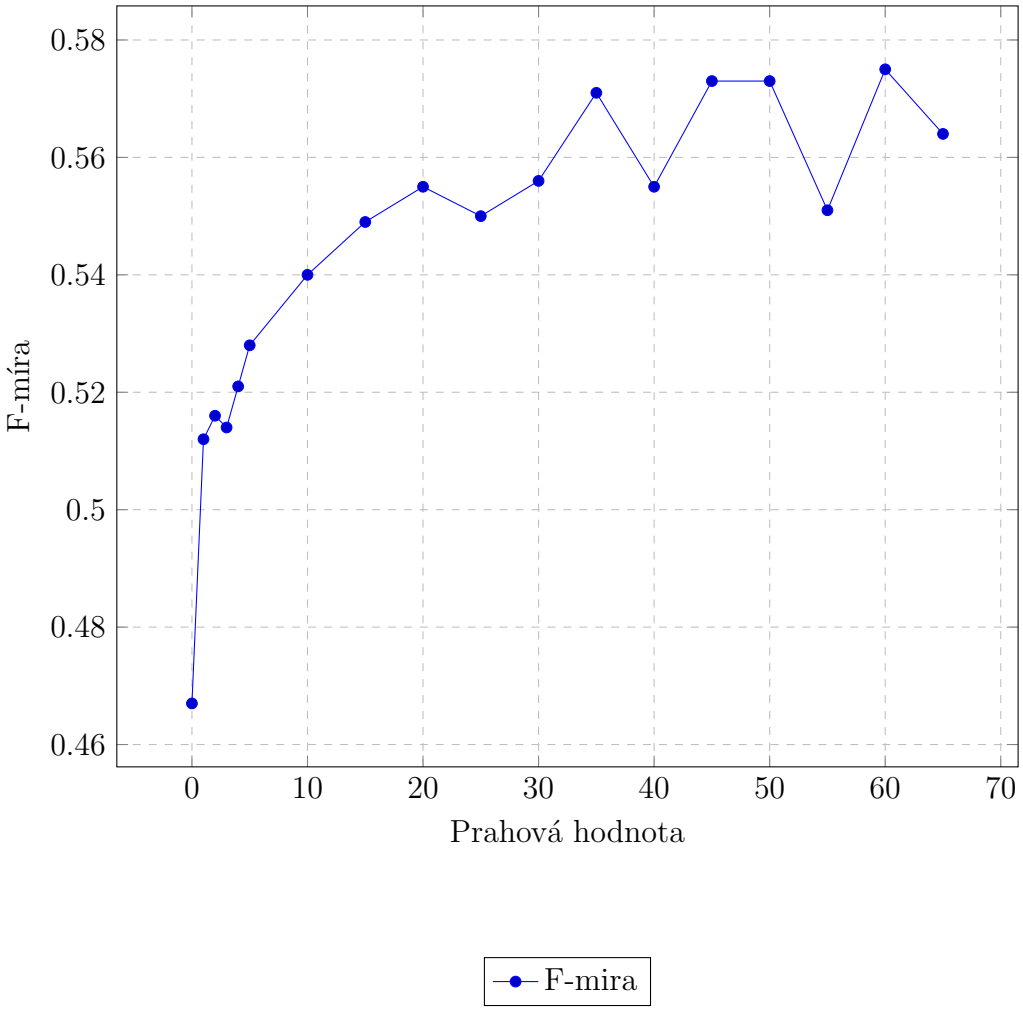
Popis experimentu Postupně je proces extrakce spouštěn s různými nastaveními prahové hranice, bude zkoumána závislost úspěšnosti.

Průběh experimentu Následující tabulka (viz tab. 6.6 str. 46) a graf (viz obr. 6.6 str. 47) zachycují míru úspěšnosti v závislosti na nastavení prahové hranice.

práh BoW	F-míra	práh BoW	F-míra
0	0.467	25	0.550
1	0.512	30	0.556
2	0.516	35	0.571
3	0.514	40	0.555
4	0.521	45	0.573
5	0.528	50	0.573
10	0.540	55	0.551
15	0.549	60	0.575
20	0.555	65	0.564

Tabulka 6.6: Závislost úspěšnosti na použité prahové hodnotě.

Závěr experimentu Je znatelná konstantně se zvyšující míra úspěšnosti s postupně rostoucí prahovou hodnotou do určité hodnoty. Od této hodnoty dále úspěšnost osciluje což zapříčiňuje momentální převaha lepších, respektive horších slov odříznutých prahovou hranicí.



Obrázek 6.6: Naměřené hodnoty při zpracování 60 stránek

6.8 Experiment stálosti třídy datových bloků

Myšlenkou je ověřit, zda nedochází k chybě vlivem špatné klasifikace konkrétního textu. z principu toho, jak webová stránka s diskuzním fórem vzniká, tedy použitím šablony pro zobrazení dat z databáze příspěvků, je možné otestovat, jestli datové bloky se stejnou adresou v rámci DOM, mají i stejnou třídu. Byl navržen následující experiment.

Popis experimentu Bude provedena analýza četnosti příslušníků jednotlivých tříd napříč stránkami v rámci jednoho webového serveru se stejnou adresou v rámci DOM.

Průběh experimentu Objevilo se několik případů, kdy byl například komentář, který měl jinak charakter uživatelského jména chybně klasifikován. Jednalo se v převážné většině o krátké odpovědi. Jako například:

- hmmm...
- Smudla Pudla
- Dr.Haus

Závěr experimentu Předběžný experiment, který měl vyhodnotit, zda k takovýmto záměnám dochází prokázal, že tyto případy nastávají. Některé příspěvky mohou být bez kontextu opravdu zaměněny například za jména uživatelů. Může samozřejmě docházet i k jiným záměnám podobného charakteru.

Předběžný experiment ukázal, že k takové záměně dojde přibližně v 5,5% případů.

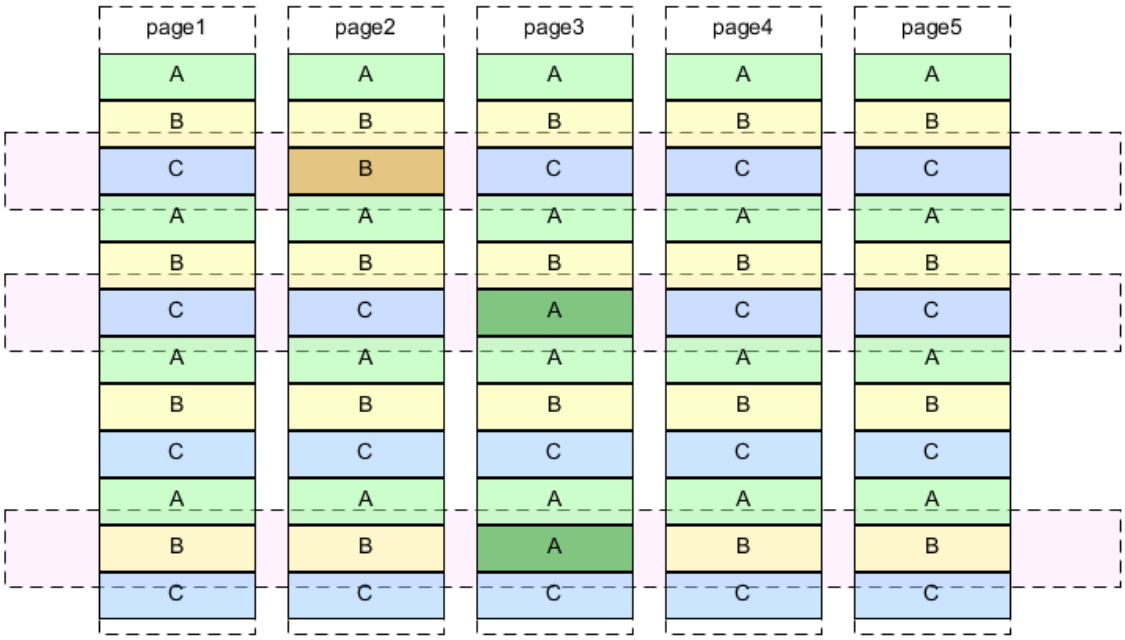
6.9 Experiment korekce bloků podle šablony

Předchozí experiment odhalil, že může docházet k záměnám tříd. Byl tedy navržen následující experiment, který ověří, zda je možné takovéto záměny odfiltrovat a opravit.

Na obrázku (viz obr. 6.7 str. 49) je názorně ukázáno, jak dochází k záměně tříd.

Popis experimentu Již označený výstup klasifikátoru bude dále podroben zkoumání. Provede se statistická analýza datových bloků následujícím způsobem. Budou sečteny objekty v jednotlivých třídách se stejnou adresou v dokumentu napříč všemi stránkami jednotlivých serverů. Protože z principu způsobu vznikání stránky pomocí šablony musí mít tato data stejný význam, bude provedena korekce menšinových tříd tak, aby odpovídali třídě většiny klasifikovaných objektů.

Průběh experimentu Následující tabulky záměn ukazují, jak proces korekce přispívá k snižování chyby. Tabulka (viz tab. 6.7 str. 49) zobrazuje stav před nasazením opravy, tabulka (viz tab. 6.8 str. 49) po ní.



Obrázek 6.7: Grafické znázornění vznikajících chyb a jejich detekce

	Classificator			
	NICK	DATE	COMM	NORM
NICK	4746	0	85	0
DATE	0	2109	0	0
COMM	125	0	3850	0
NORM	2095	914	6020	0

Tabulka 6.7: Tabulka záměn před použitím korekce
F-míra : 0.5515

	Classificator			
	NICK	DATE	COMM	NORM
NICK	4821	0	10	0
DATE	0	2109	0	0
COMM	20	0	3955	0
NORM	2048	914	6067	0

Tabulka 6.8: Tabulka záměn při použití korekce
F-míra : 0.5595

Závěr experimentu Korekce, která byla navržena, opravdu opravila část textů. v testovacím vzorku bylo touto metodou zpětně opraveno 85% špatně klasifikovaných textů ze tříd NICK, DATE a COMM. Protože je ale oprava prováděna pomocí statistiky, tato korekce nedokázala opravit chybné klasifikace poslední třídy. Tato chybovost je popisována v předchozím experimentu (viz část 6.2 str. 36).

6.10 Známé nedostatky programu

V následující části budou shrnuty a rozepsány známé nedostatky, které přispívají k chybovosti celého procesu. Zároveň se jedná o oblasti, které by měly být lépe prozkoumány a vylepšeny při pokračování ve vývoji tohoto systému.

6.10.1 ROAD RUNNER

Nalezení šablony Při vývoji jsem zaznamenal několik problémů, které se obecně týkají extrakce dynamických dat z webových stránek, konkrétně z webových diskuzních fór. Pokud bychom brali v úvahu jen jednu podobu příspěvku, běžně používané metody by neměly problém se správným vyhledáním příspěvků v diskuzi. v praxi ale narazíme na drobné problémy.

V diskuzi konkrétně, kterou představuje vlastně seznam jednotlivých příspěvků chronologicky seřazenou, jsou možné výskyty typově jiných datových bloků. Takovým příkladem můžou být například reklamy mezi příspěvky nebo soft-delete (viz část 6.5 str. 42) na některých fórech. Pokud by reklama byla vždy na stejném místě, což často bývá, nepředstavovalo by to veliký problém. Stala by se jednoduše součástí šablony a následovalo by její vyřazení. Pokud je ale náhodně umísťována mezi příspěvky, znemožní to celý proces získání šablony. Jak bylo ale zmíněno, tento přístup není příliš častý.

Druhý uvedený problém je více znatelný, pokud například administrátor znemožní zobrazení příspěvku a místo něj se zobrazí informace o tomto v nějaké formě. Například větu : příspěvek byl odstraněn administrátorem, opět je do stránky vnesen prvek, který ztíží vyhledání šablony. Tento případ je více častý než první uvedený.

Bylo by možné jej řešit několika způsoby. Jedním z nich by mohlo být například přidání tohoto předpokladu do samotného nástroje pro extrakci dynamických dat. Je to však problém velice komplexní. To, jestli je šablona diskuze nalezena správně

námi použitým systémem závisí na několika faktorech a liší se případ od případu.

Dynamické seznamy Road Runner je dobrý nástroj pro získávání šablon webových stránek. Na stránky s relativně jednoduchou strukturou(internetové obchody, katalogy aj.) funguje velice dobře. Dá se dobře použít i na extrakci ze složitějších struktur, jako jsou právě webová fóra. Avšak, jak již bylo zmíněno, pokud je do šablony zanesena i malá míra náhody (náhodně umísťované reklamy, smazané komentáře), nebo pokud jsou na fórech používány vnořené citace, nastávají problémy s nalezením šablony. Dynamické texty jsou nalezeny. Někdy jsou ale označovány jako dynamické podstromy. Pokud se tak stane, přicházíme o výhodu, kterou by přinesl rozpoznaný dynamický seznam. v takovém případě z výstupu RR nelze poznat, jaké datové bloky mají stejný význam. V ideálním případě by byl detekován dynamický seznam komentářů, to by znamenalo, že bychom měli množiny datových bloků, které, už podle šablony, mají stejný význam. k tomu však často nedochází právě kvůli výše popsaným problémům. Také přispívá, že ve webových fórech jsou zobrazovány příspěvky se stejným stránkováním v rámci jednoho webového serveru, tudíž je detekce dynamického seznamu o to obtížnější. RR v takovém případě dynamický seznam nepozná vůbec, protože se mu povede šablonu odhalit bez jeho použití a priorita ověřování těchto prvků je jasně dána z principu jeho fungování (viz část 2.9 str. 9).

Pro účely extrakce z webových fór by bylo třeba do nástroje pro extrakci dynamických dat zasáhnout a přidat možnost vynucení hledání dynamických seznamů, což není triviální operace.

Alternativní řešení Alternativním řešením by bylo přidat další vrstvu do vzniklého nástroje pro extrakci a rozpoznání datových bloků. Tato vrstva by měla za úkol projít původní zdrojový kód stránky a jednotlivé, již klasifikované a označené texty, přiřadit k elementům stromu dokumentu[3]. Následně by našla nejmenší obalující element, s největším počtem výskytů entit z webového fóra. Idea je taková, že takový element by obsahoval pouze jednotlivé bloky komentářů. Následně by se opět statistickým modelem vytvořily skupiny s podobnou XPath. Byla by to vlastně obdoba dynamického seznamu, který měl vytvořit RR. Avšak v tomto kroku by již byly k dispozici predikované třídy první části klasifikace. Statisticky by nemělo být složité detekovat typ množiny textů. Vzniklé chyby by byly opraveny pomocí okolních textů stejné významové množiny.

[3]Posobný přístup využívá již provedený experiment (viz část 6.9 str. 48).

Toto řešení však není příliš vhodné. Jedná se vlastně o vytvoření další vrstvy, která by měla opravovat chyby, které vznikly v jiné části procesu. Vhodnější by bylo upravit proces, ve kterém k chybě skutečně vzniká.

6.10.2 Trénovací data

Jedním z velkých nedostatků je stále množství trénovacích dat. Tvorba nebo získávání těchto dat není triviální. Aby bylo možné data vhodně použít pro trénování, je důležité, aby byla označována stejnou formou. Například aby byl označen vždy nejhlubší element, který obsahuje danou informaci. Často ale právě kvůli popsaným problémům (viz část 6.4 str. 39) není možné vhodně označit oblast podle předem nadefinovaných pravidel. Právě tato skutečnost přináší požadavky na velkou množinu trénovacích dat.

Součástí vytvořeného systému je způsob, jak relativně jednoduchou cestou, bez zásahu do programu či tvorby nového pro označování dat, přidávat další trénovací webové stránky (viz část 5.3 str. 28).

S rozšiřující se množinou trénovacích dat bude markantně stoupat také úspěšnost systému (viz část 6.2 str. 36) (viz část 6.3 str. 38).

6.10.3 Správné spojení údajů

Jak již bylo zmíněno dříve, RR měl problém s identifikací množiny komentářů jako opakujícím se vzorem. Bylo naznačeno také možné alternativní řešení tohoto problému(viz část 6.10.1 str. 51). Pokud by byla aplikována jedna či druhá možnost, neměl by být následně problém identifikovat, například formou metriky vzdálenosti, které údaje patří k sobě. Výsledkem by byly objekty komentářů, které by mohly být použité k přesnějším analýzám.

7 Závěr

Teoretická část práce se věnuje průzkumu oblasti získávání dat z webových stránek a obecně pojednává o způsobech prezentace informací na tomto médiu.

Následně je zahrnuta studie již existujících systémů, které jsou nejvíce podobné tématu této práce. Bylo prozkoumáno několik takových, které se věnují extrakci dat z webových stránek obecně, i těch, které se zabývají konkrétně webovými diskuzemi. Tato práce kombinuje několik ověřených přístupů, avšak navrhuje a aplikuje v této oblasti nové postupy.

Byly použity statistické metody, strojové učení a analýza přirozeného jazyka na webové stránky obsahující zmíněná data. Také se objevilo několik nedostatků, které se týkají jednotlivých částí procesu. Především to byl problém s extrakcí dynamických dat pomocí šablony. v mnoha případech přesná šablona nebyla nalezena vůbec, což zapříčinily hlavně občasné bloky, které byly jiného typu než příspěvek do diskuze. Byly také navrženy postupy, jak by bylo možné postupovat v dalším vývoji a při řešení těchto chyb. Druhý problém je stále nedostatečně velká množina trénovacích dat.

Na straně druhé, bylo ověřeno několik metod, které se ukázaly jako velice dobře použitelné při řešení této úlohy. Navržený klasifikátor dobře funguje na označování textových entit na webových diskuzích. Rovněž koncept post processingu se ukázal jako vhodný.

Všechny cíle práce, tedy prozkoumat oblast extrakce dat z nestrukturovaných zdrojů a existující systémy, navrhnout nové automatické metody pro extrakci příspěvků z webových diskuzních fór automatickou cestou a následně konstruktivně zhodnotit výsledky, byly splněny.

Literatura

[1] *K-folds Cross-Validation for Smother LCs* [online]. 2016. [cit. 2017/02/27]. Dostupné z: `http://sdsawtelle.github.io/blog/output/week6-andrew-ng-machine-learning-with-python.html`.

[2] *MALPAL* [online]. 2016. [cit. 2017/02/27]. Dostupné z: `https://malpal.cz/cz/homepage/knihovna`.

[3] *Machine Learning Blog and Software Development News* [online]. [cit. 2017/02/27]. Dostupné z: `http://blog.datumbox.com/machine-learning-tutorial-the-max-entropy-text-classifier/`.

[4] *XPath Tutorial* [online]. [cit. 2017/02/27]. Dostupné z: `https://www.w3schools.com/xml/xpath_intro.asp`.

[5] *XML DOM Tutorial* [online]. [cit. 2017/02/20]. Dostupné z: `https://www.w3schools.com/xml/dom_intro.asp`.

[6] *Xsoup* [online]. [cit. 2017/02/27]. Dostupné z: `https://github.com/code4craft/xsoup`.

[7] ARASU, A. – GARCIA-MOLINA, H. Extracting structured data from web pages. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, s. 337–348. ACM, 2003.

[8] ASCH, V. V. Macro- and micro-averaged evaluation measures. September 9, 2013.

[9] BROWN, P. F. et al. Class-based n-gram models of natural language. *Computational linguistics*. 1992, 18, 4, s. 467–479.

[10] CHANG, C.-H. – HSU, C.-N. – LUI, S.-C. Automatic information extraction from semi-structured Web pages by pattern discovery. *Decision Support Systems*. 2003, 35, 1, s. 129 – 147. ISSN 0167-9236. doi: http://doi.org/10.1016/S0167-9236(02)00100-8. Dostupné z: `http://www.sciencedirect.com/science/article/pii/S0167923602001008`. Web Retrieval and Mining.

[11] CRESCENZI, V. et al. Roadrunner: Towards automatic data extraction from large web sites. In *VLDB*, 1, s. 109–118, 2001.

[12] FABRIZIO, S. Machine Learning in Automated Text Categorization. *ACM Computing Surveys*. March 2002, 34, 1.

[13] HEDLEY, J. *Jsoup: Java HTML Parser* [online]. [cit. 2017/02/27]. Dostupné z: `https://jsoup.org/`.

[14] MACHOVÁ, K. – PENZÉŠ, T. Extraction of web discussion texts for opinion analysis. In *Applied Machine Intelligence and Informatics (SAMI), 2012 IEEE 10th International Symposium on*, s. 31–35. IEEE, 2012.

[15] PANDIAN, S. L. – PUNITHA, R. Annotation for Query Result Records based on Domain-Specific Ontology. *International Journal on Natural Language Computing*. June 2012, 3, 3, s. 93–103.

[16] RAYCHAUDHURI, S. et al. Associating genes with gene ontology codes using a maximum entropy analysis of biomedical literature. *Genome Research*. 2002, 12, 1, s. 203–214.

[17] WU, L. – HOI, S. C. H. – YU, N. Semantics-Preserving Bag-of-Words Models and Applications. *IEEE Transactions on Image Processing*. July 2010, 19, 7, s. 1908–1920. ISSN 1057-7149. doi: 10.1109/TIP.2010.2045169.

Příloha A

Uživatelská dokumentace

Celý systém byl vyvíjen i zkoumán v programovacím jazyku JAVA, konkrétně 64 bitovou verzí Java 8.

Projekt byl zanechán ve stavu, kdy při spuštění provede poslední experiment (viz část 6.9 str. 48) již se zapnutou post processingovou korekcí.

Do adresářového stromu popsaného v dokumentu dříve (viz obr. 5.4 str. 33) uloží věchny statistiky a na standardní výstup vypíše sumarizaci výsledků a tabulku záměn jednotlivých tříd.

Pro úspěšné spuštění a dokončení experimentu je postačující 1,5 GB operační paměti. Podrobnosti jsou blíže popsány v README.txt na přiloženém médiu.