Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE

Jan Šebesta

Dolování dat z příchozích zpráv elektronické pošty

Katedra softwarového inženýrství

Vedoucí diplomové práce: RNDr. Michal Žemlička, Ph.D.
Katedra softwarového inženýrství

Studijní program: Informatika, softwarové systémy

2009

Děkuji panu doktoru Žemličkovi za jeho cenné rady, motivaci a podporu při překonávání obtížných situací.

Prohlašuji, že jsem svou diplomovou práci napsal(a) samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne 7.8.2009 Jan Šebesta

Obsah

Název práce: Dolování dat z příchozích zpráv elektronické pošty
Autor: Katedra (ústav): Katedra softwarového inženýrství
Vedoucí bakalářské práce: RNDr. Michal Žemlička, Ph.D.
e-mail vedoucího: michal.zemlicka@mff.cuni.cz

Abstrakt: V předložené práci studujeme možnosti automatického třídění příchozí emailové komunikace. Naším hlavním cílem je rozpoznání informací o nadcházejících workshopech a konferencích, nabídkách práce a vydávaných knihách. Snažíme se vyvinout nástroj, který informace vydoluje z dat získaných z oborových konferencí. Nabídky v konferencích přicházejí ve formě html, rtf, nebo prostého textu, ale informace v nich je zapsána v běžném jazyce. Text–miningovými metodami získáváme informace z běžného textu a ukládáme je ve strukturované formě, kterou je možné jednoduše strojově zpracovávat. Zkoumáme způsob zpracování pošty člověkem a následně tyto poznatky aplikujeme při tvorbě systému. V průběhu práce řešíme problémy se samotným získáním zpráv, rozpoznáním jazyka a kódování a rozpoznáním typu zprávy. Informace, kterou ze zprávy potřebujeme získat se různí v závislosti na typu zprávy a události, které se týká. Teprve po rozpoznání nosné informace ve zprávě jsme schopni vydolovat data pro zjištěný typ události. Na závěr ukládáme získané znalosti do databáze, která umožňuje rychlou interakci s uživatelem.

Klíčová slova: e–mail, workshop, text–mining, třídění, automatizace, parsování

Title: Data mining from incoming e–mail messages
Author: Jan Šebesta
Department: Department of software engineering
Supervisor: RNDr. Michal Žemlička, Ph.D.
Supervisor's e-mail address: michal.zemlicka@mff.cuni.cz

Abstract: In the present work we study possibilities of automatic sorting of incoming email communication. Our primary goal is to distinguish information about oncoming workshops and conferences, job offers and published books. We are trying to develop tool to mine the information from data from professional mailing lists. Offers in the mailing lists come in html, rtf or plain text format, but the information in it is written in common spoken language. We are developing the system so it will use text mining methods to extract the information and save it structured form. Than we will be able to work with it. We are examining the handling of the mails by user and apply the knowledge in the development. We solve the problems with obtaining of the messages, distinguishing language and encoding and estimating the type of message. After recognition of the bearing information we are able to mine data. In the end we save the mined information to the database, which allows us to display it in well–arranged way, sort and search according to the user needs.

Keywords: e–mail, workshop, text–mining, automatization, extraction

Kapitola 1

Úvod

Aby byl člověk informovaný o aktuálním dění na akademickém poli a neunikly mu žádné příležitosti k publikacím a návštěvám odborných seminářů, může odebírat zprávy z různých mailových konferencí, které zprostředkovávají informace o aktuálně pořádaných událostech. Množství zpráv přicházejících z mailing listů takové konference je obrovské a jedná se o několik desítek zpráv denně. Pokud je člověk přihlášen do více konferencí, není už v jeho silách zkontrolovat všechny zprávy každý den. Může proto promeškat nějaké termíny, nebo strávit neadekvátní dobu procházením příchozí pošty.

Aby počítač mohl práci ulehčit, je třeba roztřídit poštu z konferencí nějakým sofistikovaným způsobem. Je třeba poznat, kdy se jedná o relevantní zprávu, kdy tato zpráva obsahuje důležité informace, a tyto informace z mailu získat. Uživatel potom může zajímavá data vidět v nějaké rozumné podobě, která mu urychlí a zpřehlední orientaci, například přehledný barevně rozlišený kalendář se seznamem termínů. Jeden pohled do kalendáře, který obsahuje události řekněme za měsíc zkrátí dobu potřebnou ke zpracování šesti set mailů, které při počtu dvacet mailů denně stihnou za tento měsíc přijít, z desítek minut na pár vteřin. Naším cílem v této práci bude vytvořit tak sofistikovaný systém, který by byl schopen zpracovat maximální množství pošty a na vlastním přečtení nechal jen minimum zpráv.

Kapitola 2

Motivace

2.1 Data mining

Intuitivně lze odhadnout, že data mining neboli dolování dat znamená nějaké vyhledávání informací. Přesněji to znamená získávání znalostí (knowledge) z informací. Problém s dnešním internetem je totiž v množství informací, které je se na něm vyskytují. Znalost se od informace liší v tom, že ji umíme nějakým způsobem využít. Můžeme mít nepřeberné množství informací, ale vybrat mezi nimi ty, které nám pomohou v dosažení našich cílů nemusí být triviální záležitostí. Jako příklad můžeme vzít expertní systémy, které pro základ činnosti využívají znalostí expertů. Informace, které experti mají, jsou stejné i pro ostatní systémy, ale experti mají navíc znalosti, které z těchto informací dokáží nebo dokázali ze zkušeností získat. Stejné to je i s elektronickou poštou. Zpráv přichází velké množství a udržet v nich přehled je velmi obtížné. Proto je hledání znalostí důležité pro rozhodování, co se kterou zprávou udělat. Pokud toto rozhodování bude provádět člověk, ze kterého se po několika stovkách zpráv stane expert, tak ze zkušenosti bude vědět, které zprávy už podle předmětu zahodit, které číst, které zpracovat důkladněji a které například archivovat pro pozdější nahlédnutí. Člověku to ale trvá velmi dlouho a bylo by tedy dobré vytvořit program, který to provede za něj.

Nejprve si musíme uvědomit, co je vlastně cílem našeho programu. Data mining je získávání nových poznatků z velkého a nepřehledného množství strukturovaných dat, text mining je to samé, ale znalosti získáváme z textových záznamů, nikoliv z předem připravené databáze. Jak ale pohlížet na náš problém, je tohle přesně to, co my chceme? My nezískáváme nové poznatky, my se snažíme získat známá data z textu zprávy. Nesnažíme se o lingvistickou či syntaktickou analýzu zprávy. Není naším cílem umět porozumět psanému textu. Naším cílem je spíše specializovaná forma univerzálního vyhledávače či indexovače / crawleru. Nechceme vytvořit univerzální nástroj pro získávání obecných dat, ale konkrétních informací týkajících se konkrétního typu události. Tomu by odpovídalo označení vyhledávač, ale to také není přesné. Pod pojmem vyhledávač si dnes představíme internetový vyhledávač, který má ve svém indexu uloženy seznamy stránek a snaží se v nich najít hledaný

řetězec. To ovšem také neodpovídá tomu, co požadujeme my. Proto se podíváme na náš problém tak, že v procházejícím velkém množství dat (přicházející emailové zprávy) hledáme smysluplnou informaci. Tuto informaci je třeba ze zpráv vydolovat. Tedy nepřehlédnout, když prochází a posléze vyčíst detaily. O tom ale už můžeme prohlásit, že je dolování dat z textu, neboli text mining. Můžeme tedy říci, že naší snahou opravdu je text-miningový nástroj na prohledávání pošty.

2.2 Požadavky

Náš systém by měl vykonávat většinu práce za uživatele tím, že poštu dokáže sám roztřídit a vybrat důležité informace. Na rozhodování pro uživatele by měl nechat jen minimální, v optimálním případě nulové množství zpráv, které nedokáže sám identifikovat. Ostatní zprávy je možné zpracovat kladným nebo záporným způsobem. Kladný způsob znamená, že zpráva je relevantní, systém vybere data, zaarchivuje zprávu a pro uživatele připraví shrnutí informací. Záporný způsob označuje rozpoznaný spam, který je smazán. V některých případech je možné výsledek rozhodnutí propagovat až do poštovní schránky, která je systémem spravována. Pokud je schránka dostupná přes protokol IMAP, je možné zpracované zprávy přesouvat do složek přímo na serveru, přes protokol POP3 je možné je pouze mazat.

Kapitola 3

Analýza

3.1 Stávající produkty

Prohledáváním pošty se dostáváme na pole několika významných hráčů. Prvním z nich jsou systémy automatického zpracování pošty. Jejich úkolem je něco podobného, o co se snažíme my. Roztřídit zprávu uživatelům do složek tak, aby při jejich zpracování měli co nejméně práce. Navíc ještě poštu indexovat, najít klíčová slova a udělat výtahy. Tím umožní uživateli všímat si pouze vybrané pošty ve vybraných složkách, případně si rozložit práci tak, aby nemusel reagovat na všechny zprávy ve stejnou dobu. Základní způsob zpracování umožňují prakticky všechny mailové klientské programy. Tím nemyslíme jen lokální programy, ale i webová rozhraní mailserverů. Zpravidla jsou využity takové funkce, jako dělení za pomoci adresy nebo části adresy odesílatele, klíčová slova v předmětu, klíčová slova v těle dokumentu, části adresy příjemce, datum odeslání, velikost, názvy a počty příloh. Tato základní dělení ovšem pouze přibližně odhadují, co daná pošta bude obsahovat, vůbec se nedokáží věnovat obsahu zprávy.

Složitější nástroje umějí pracovat s obsahem zprávy. Komerční nástroje Advanced Email Parser [4] a Email2DB [5] jsou určeny k automatizaci procesů za pomoci emailové komunikace. Umějí nejen procházet emaily a zpracovávat jejich hlavičky, ale také na ně automatickým způsobem odpovídat a vyvolávat navazující business procesy. Důležitým rozdílem oproti standardnímu zpracování hlaviček je ale způsob práce s tělem zprávy. Je možné předem zadat regulární výrazy, či řádkové operace, či vyhledávání, které naleznou odpovídající výrazy a použijí je jako proměnné pro další zpracování. Je tedy možné ze standardizovaných formulářových dat získat hodnoty jednotlivých polí a poté je uložit do databáze, použít v odpovědi, nebo navázat na nějakou následnou řídící činnost. Všechny skripty, regulární výrazy a reakce na ně je možné (nutné) ručně upravovat, takže tyto nástroje mohou být velmi silné. [4] umí vyhledávat řádky odpovídající zadaným řetězcům, vyhledávat celé bloky, nebo vyhledávat regulární výrazy z celého textu. Je možné nadefinovat podmínky a akce v závislosti na existenci shody daného výrazu, nebo přímo na nalezeném obsahu. Navíc pro definici podmínek je k dispozici uživatelsky přívětivé rozhraní.

Jako příklad základní funkčnosti obou nástrojů můžeme uvést právě emailovou konferenci. Vyhledáním řádku *subscribe* nebo *unsubscribe* v těle mailu můžeme určit, jestli se má daná adresa přidat nebo odebrat ze seznamu příjemců této konference.

Nevýhodou obou řešení je závislost na pevném formátu přijímané zprávy. Taktéž je problém s kódováním a cizími jazyky. Navíc oba programy jsou základními řešeními, ke kterým je třeba ještě doprogramovat podmínky a skripty, které mají zprávy zpracovávat. Pokud by se podařilo vytvořit dobré regulární výrazy a kombinace podmínek a následných akcí, je možné, že by bylo možné tyto produkty použít k řešení našeho problému. Ceny obou řešení se pohybují mezi $400 až $1000 podle zvolené verze.

Jako nástroj na proudové zpracování přicházející a potenciálně i procházející pošty se dostáváme do oblasti, kde jsou ještě další konkurenti. Těmi jsou tajné služby. Nejedna vláda vynakládá velké částky do ochrany svého kybernetického prostoru. Za příklad můžeme vzít Spojené státy americké, které mají speciální armádní oddělení zaměřené na kybernetický prostor. Mluví o tom sekretář Lynn [7] a ve svém článku také bývalý důstojník Astore [6]. Přestože nejsou zmíněny žádné konkrétní způsoby sledování, je vysoce pravděpodobné, že monitorování e–mailové komunikace je jedním ze základních způsobů dohledu nad populací. Všichni tedy mohou počítat s tím, že emaily do Severní Ameriky budou procházet nějakou automatickou kontrolou obsahu. Všichni, kdo nejsou občané Spojených Států jsou potencionální nepřátelé, není tedy důvod dodržovat nějaké listovní tajemství. Navíc normy týkající se listovního tajemství jsou v různých státech na různé úrovni a ohledně emailové komunikace je situace dost často nevyřešená. O kontrole emailové komunikace se nikde moc nemluví, nikdo nenabízí nástroje, a výsledky výzkumu nejsou nikde jednoduše dohledatelné. Přestože jsou vypisovány granty a požadavky na zpracování tohoto problému, množství dostupných výsledků neodpovídá. Teorie je taková, že všechny výsledky v této oblasti jsou uschovány a použity tajnými službami různých států. Je možné, že některé služby těchto technologií nevyužívají, nebo opravdu dodržují listovní tajemství. Vzhledem k mediální masáži a stupňujícímu se tlaku na kybernetické frontě, jak říká Klimánek[8], je dodržování emailového tajemství velmi zpochybněno.

Systémy pro dolování dat ze streamovaných dat ale nejsou jen doménou tajných služeb, ale také zaměstnavatelů. Přestože se teoreticky jedná o porušování zákonů, některé firmy prohlížejí svým zaměstnancům poštu, aby v práci neřešili osobní záležitosti. Firmy se ohrazují tím, že pracovní email na pracovním serveru je majetek firmy, na kterém má zaměstnanec pracovat. Tudíž není pod ochranou zákona, jako osobní korespondence.

3.2 Vlastnosti doručené pošty

Obecně můžeme říci, že elektronická pošta je velmi nesourodá a příchozí zprávy mohou obsahovat téměř cokoliv, navíc v libovolném jazyce. Po chvíli ale přijdeme na to,

že to v našem případě není tak úplně pravda. Předpokládámeli běžnou schránku, nalezneme v ní osobní korespondenci, data z různých formulářů a webových stránek, obchodní korespondenci, reklamní sdělení (spam) a relevantní zprávy z konferencí.

Osobní korespondence Tato skupina zpráv může obsahovat skoro cokoliv. Běžný text s oslovením, bez oslovení, přílohy, obrázky, html formátování, odkazy. O této skupině opravdu můžeme říci, že se jedná o zprávy bez jakéhokoliv obecně uchopitelného formátu. Na druhou stranu ale můžeme říci, že počet lidí, od kterých chodí osobní korespondence není zase tak velký. Takže osobní korespondenci lze odlišit podle toho, od koho přichází. Problémovou se stává situace, kdy blízký člověk je zároveň spolupracovníkem a chodí od něj osobní i pracovní komunikace.

Obchodní korespondence Pracovní zprávy zpravidla obsahují hlavičky a zápatí firem, které spolu navzájem komunikují. Zprávy bývají na závěr podepsány a začínají nějakým standardizovaným formálním úvodem použitého jazyka.

Formulářová data Touto skupinou zpráv myslíme zprávy oznamující ztracená hesla, potvrzení objednávek z obchodů, oznámení o registraci, automatické zprávy ze serverů, logy z vlastních serverů a další. Tyto zprávy mají velmi stabilní formát a dají se dobře rozpoznávat za pomoci regulárních výrazů či klíčových slov. Zpracování těchto zpráv by mohlo být prováděno automatickým systémem.

Spam Tato velmi obsáhlá kategorie obsahuje nepřeberné množství sdělení, která něco opravdu nabízejí, tváří se, že něco nabízejí, nebo jen obsahují viry a trojské koně a tváří se jako osobní sdělení. Velkou část z nich blokuje sám o sobě poštovní server, část pak je možné zablokovat jednoduchými filty založenými na výskytu některých klíčových slov. Obecně je problém spamu velmi rozsáhlý a nějak tuto kategorii automaticky rozpoznávat je velmi obtížné. Ani profesionální nástroje pro detekci spamu nejsou stoprocentní. Z této pošty žádné důležité informace nechceme získávat, ale bylo by dobré, aby nám nezanášela chybné údaje do systému a příliš jej nezatěžovala. Obsah spamu v poště je totiž velmi vysoký.

Poštovní konference Poslední skupina zpráv, která nás nejvíc zajímá. Data z konferencí jsou odspamována již na konferenčním serveru. Zprávy obsahují nějaké opravdové sdělení a nepředpokládají odpověď na sebe samu. Zprávy také nejsou odpověďmi, nebo zprávami přeposlanými dále (reply, forwarded message). Obsahem je oznámení o nějaké události. Už to, že se jedná o oznámení, s sebou nese určité nároky na obsah. Znamená to, že zpráva musí být čitelná a srozumitelná pro příjemce. Nemůže to být zmatečná nepřehledná hromada informací, ale musí mít nějakou strukturu. Úvod, který říká, o co se jedná. Upřesnění data. Seznam témat, nebo alespoň rozdělení zprávy na odstavce. Rozpoznání zpráv z konferencí je velmi jendoduché. Konferenci rozesílá jeden

konferenční server. Tento server nedělá nic jiného, takže zprávy od jednoho konferenčního serveru jsou pouze zprávami dané konference. Když tedy vybereme zprávy pouze od určených odesílatelů (konferenčních serverů), máme všechny zprávy z konferencí a nic navíc. V dalším textu se již budeme zabývat pouze zprávami z konferenčních serverů.

Maily z poštovních konferencí obsahují informace o blížících se konferencích, workshopech, sympoziích, knihách, časopisech, nebo nabídky práce. Každá z těchto oblastí má specifický slovník a pro každou z těchto oblastí je třeba získat jiná data. V rámci jedné oblasti jsou však informace velmi podobné. Oznámení o nějaké konané události musí být srozumitelné, proto není možné každou zprávu psát nějakým novým způsobem. Můžeme tedy říci, že většina zpráv týkajících se jednoho typu události si bude velmi podobná.

Kvůli čitelnosti není pro autora zprávy možné, aby všechny informace napsal do jednoho odstavce, jako kdyby psal osobní dopis. Proto mají všechny důležité dopisy nějaký rozumný způsob úpravy. Obsahují rozumně vypsané termíny do nějakého seznamu, seznam témat, autorů, tabulku s termíny, nebo jsou alespoň rozděleny na odstavce s jednotlivými informacemi. Občas pro čitelnost pomůže také nějaké standardní zarovnání nebo odsazení. To všechno nám může pomoci při automatickém prohledávání textu. Navíc zprávy přicházejí přes standardní poštovní protokoly (MIME), které umožňují přidávat důležité informace do záhlaví. Zprávy posílané ve formátu MIME mohou mít uživatelsky definované hlavičky. Tyto hlavičky navíc musí mít protokolem daný formát, takže číst z nich data je jednoduché.

Některé konference toho využívají a přímo do MIME hlaviček zapisují relevantní informace. Například hlavička X-Date může obsahovat datum konání akce. Problém zde ovšem nastává například v případě pravidelných konferencí. Stane se totiž, že autor zprávy vezme loňskou pozvánku a přepíše data v textu. Zapomene ovšem provést tuto úpravu i v hlavičkách, takže hlavičky odkazují na loňskou událost, kdežto tělo zprávy láká na aktuální dění. Proto je třeba každou informaci vážit a porovnávat s ostatními indiciemi.

Dalším z problémů, které mohou automat překvapit už v počátku zpracování zprávy jsou různé odpovědi a zprávy předávané dál. Například zpráva, která byla zaslána do konference, kterou nesledujeme, přišla někomu zajímavá, a tak ji poslal do námi sledované konference. Poslal ji ovšem jako přeposlanou zprávu, aby nebral autorovi zásluhy, takže celý předmět bude začínat "FW:". Navíc může před každý řádek původní zprávy přidat nějaký uvozovací znak, například »". Taková zpráva najednou může vypadat velmi zajímavě, jestli ji ještě nějaké mail—servery cestou rozsekají na kratší řádky, než jak je označil odesílatel.

Cílem automatického systému by mělo být vyhledávat informace, aby se tím uživatel nemusel zabývat. Podívejme se nejprve na způsob, jakým uživatel zprávy čte. Jako první zpracuje MIME formát, aby pro něj byla zpráva jednoduše čitelná. Tuto činnost za něj provádí poštovní klient, kterým zprávy prochází. Zpravidla tento klient rozparsuje zprávu a zobrazí uživateli pouze vybrané části hlavičky jako předmět,

datum přijetí a jméno odesílatele. Podle této informace uživatel většinou pozná, jestli je napsaná jazykem, kterému rozumí, a jestli je ochoten zprávu dále číst. V předmětu je většinou napsáno, k jaké události se zpráva váže. Potom otevře tělo a z prvních pár řádků pozná detaily této události, jako jsou datum a místo konání. Nakonec přečte i zbytek zprávy, aby poznal detaily.

3.3 Typy událostí

Workshop Základní události, na kterou konference upozorňují je seminář (workshop). Podívejme se důkladněji, jak probíhá pozvání na takový workshop a jaké informace potřebujeme, abychom se rozhodli pro případnou účast. Zaprvé musíme znát název semináře, který nám napoví jeho téma. Například *Semantic Web Services In Practice Workshop*, *VLDB Secure Data Management Workshop* nebo *International Workshop on Testing Database Systems*. Bohužel některé zprávy přijdou s předmětem *Call for Papers*, případně *TDS:Call for Papers*. Takže podle předmětu není nikdo schopen určit, k jaké události se daná zpráva vztahuje, případně velmi těžko podle udané zkratky. Jakmile víme alespoň rámcové téma workshopu, jsme schopni určit, jestli máme k danému tématu co říci. Teprve potom přemýšlíme o návštěvě. Dále musíme vědět, kdy a kde se koná, a do kdy je třeba poslat přihlášku a abstrakt. V některých případech je odesílatel sdílný a oznámí i termín, kdy budou vyhodnoceny příspěvky a vybrány ty, které se budou semináře účastnit.

Tyto všechny informace mohou být v jednom dopise, ale není tomu tak vždy, a nikdy to není jediný dopis, který přijde. Ke každému workhopu můžeme počítat s několika maily. První mail bude informace o tom, že se vůbec nějaký workshop pořádá. Tento první mail zpravidla obsahuje všechna důležitá data, nebo jejich alespoň přibližnou hodnotu. To znamená, že je již známo datum konání s přesností na měsíce, obsah, data zaslání abstraktů. Tento dopis se občas jmenuje *Preliminary Call For Papers*. Může se ale jednat jen o velmi přibližné údaje a nemusí obsahovat žádnou důležitou informaci. Tento dopis někdy ani nemusí vůbec přijít a místo něj přichází rovnou *Call for papers*. Naopak někdo může už do úvodního dopisu vložit kompletní informaci o všech datech a některé deadline. U všech dat se může stát i to, že jsou to termíny už po konání workshopu. Mohou to být například data vydání sborníku. Z toho plyne, že nemůžeme jednoduše nahlédnout všechny časové údaje ve zprávě a řídit se podle nich. Je třeba je porovnat a vzít do souvislosti.

Hlavním svolávacím dopisem na workshop je *Call for papers*. Tento dopis obsahuje seznam témat, kterých se mají příspěvky na semináři týkat. Hlavním údajem v něm je ale přesné datum, do kdy je třeba příspěvky odevzdat. Těchto dopisů zpravidla přichází několik preliminary, first, second, last, ... nebývá jich více jak čtyři. Může ale ještě přijít několik upomínek na to, že se blíží uzávěrka odevzdání *Deadline approaching*. Většinou přijde i jedna až

dvě změny data uzávěrky. Pořadatelé workshopů nemající dostatek příspěvků prodlužují termíny odevzdání, aby vyburcovali opozdilce.

Někdy v průběhu také dochází k upřesnění data konání samotného workshopu. Poté přicházejí upozornění na revize příspěvků, termíny odevzdání revizí a poté samotný workshop a upozornění na něj. Tento dopis se jmenuje "Call for participation", výzva k účasti na workshopu. Zpravidla už obsahuje kompletní rozvrh, protože přichází po uzávěrce a vybrání příspěvků. Může se ale stát, že tento mail je odeslán už někdy předem, bez rozvrhu, ještě dříve, než jsou vybrány příspěvky. Pro jeden workshop tedy přichází kolem pěti emailů. Tyto emaily mají různé předměty a někdy i překlepy v názvech. I pro člověka to je někdy matoucí, ale ten překlepy převážně přehlédne a pochopí obsah zprávy. Pro počítač to ale není jednoduchý úkol. Některé zprávy navíc chodí ve formátech html nebo rtf. Pozvánka na stejnou akci může také přijít v několika světových jazycích, pokud se jedná o větší událost.

Aby nebylo rozpoznání zprávy z workshopu tak jednoduché, existuje ještě jeden typ zprávy, který se týká worshopu. Jedná se o zprávu typu *Call for Workshop*. Ta předchází (a nemusí na nich vůbec záviset) všechny výše zmiňované zprávy. Tuto zprávu zasílá zpravidla organizátor nějaké větší události (konference, symposia, výstavy). Ten vyjadřuje záměr uspořádat workshop v rámci dané akce a očekává nabídky pořadatelů nebo účastníků, jaký workshop by se jim hodil, jaké by mělo být jeho téma, případně kdo by ho chtěl pořádat a v jakém termínu. Typický dotaz v tomto mailu zní: "V termínu T pořádáme konferenci, pošlete návrhy na workshop, který by se v rámci konference mohl uspořádat. Odpovězte ihned."Občas také přijde další zpráva typu "Stále ještě máme na konferenci volné místo na workshop". Pokud bychom jenom vyhledávali slova *workshop*, tento mail bychom s největší pravděpodobností našli. Musíme si tedy dávat pozor na nadpis tohoto mailu. Ten bude s největší pravděpodobností nadepsán jménem pořádané konference a samotný workshop nemá jméno, protože se teprve pořádá výběrové řízení na jeho konání. Nesmíme proto zaměnit názvy konference a workshopu, musíme si dávat pozor, co které velmi podobně vypadající názvy znamenají.

Konference Konference může sdružovat několik workshopů a několik přednášek. V rámci zprávy upozorňující na konferenci může být i *call for workshop* požadavek. Ten ale může přijít zvlášť. Nepříjemnou vlastností zpráv upozorňující na konference je konání několika workshopů a informace o nich ve stejné zprávě. Navíc všechny workshopy se konají v rozmezí několika dnů, překrývají se, jsou ve stejných budovách a místnostech. Jakýkoliv automatický systém tedy může mít velké problémy se správným přiřazením jednotlivých datových údajů. Pokud je konference větší, může navíc jejím výsledkem být více sborníků. Může se jednat o sborník ke každému workshopu zvlášť, nebo několik sborníků vždy obsahujících sloučené podobně zaměřené workshopy. V rámci odborného symposia nebo multi–konference může dokonce existovat více kon-

ferencí s vlastními názvy a vlastními workshopy. Upozornění na tuto událost může být velmi obsáhlé a může odkazovat na další zprávy, které přijdou zvlášť ke každé přidružené konferenci. Navíc přidružené konference mohou každá ve své zprávě odkazovat na hlavní multi–konferenci. Každá z přidružených konferencí se může konat v jiném místě poblíž hlavního centra symposia. Stejně tak workshopy patřící pod jednotlivé konference. Rozpoznat a správně oklasifikovat zprávu obsahující takovéto údaje už není jednoduchou záležitostí, přestože rozpoznání, že se jedná o zprávu obsahují informace o konferenci, nemusí být složité.

Zajímavým oříškem je také informace o konferenci v rámci *Call for Participation* zprávy. Samotná zpráva může být nadepsána jako výzva k účasti na semináři, ale obsah může být o konferenci, v rámci které se tento seminář koná. Tedy na první pohled bez zkoumání obsahu každý vidí, že ve zprávě nalezne informace o workshopu, avšak o něm je zde pouze zmínka, hlavní zpráva je o chystané konferenci.

Kniha Pod částí zabývající se knihami můžeme chápat dvě různé události. Jednou je vydání knihy. To je prosté oznámení, že kniha vyšla. Neočekává žádnou odpověď a víceméně by se dalo zařadit do marketingové pošty. Mnohem zajímavější je ale požadavek na zaslání kapitoly, neboli *Call for Chapters*. Tato výzva obsahuje popis knihy, která se má vydávat a její přibližný obsah. Každý, kdo si myslí, že by jeho příspěvek mohl být cenným pro danou knihu, může zaslat svou kapitolu a editor poté vybere nejlepší z příspěvků, složí je dohromady a knihu vydá. Získávat nějakou stručnou informaci z tohoto mailu není jednoduché, protože obsahuje většinou několik odstavců obsahujících slovní popis pojednávající o dané knize. Obsahuje informace o autorovi, důvodech proč knihu vydat, očekávaném publiku, případně také jména autorů, kteří již přislíbili účast. To všechno, aby více autorů bylo motivováno k zaslání svých kapitol. Z tohoto mailu je tedy zajímavé získat informace o datu vydání, zastřešujícím editorovi a jeho organizaci a témata, případně název knihy, pokud je předem znám. V tomto případě je ale cenná celá zpráva, protože pomůže objasnit důvody k tvorbě kapitoly.

Zaměstnání / doktorandská pozice Přes konference se naposílají standardní nabídky zaměstnání, jako v nějaké personální agentuře, ale nabízejí se specializovaná místa vyžadující oborovou znalost a zkušenost. Většinou se jedná o doktorské nebo doktorandské posty na některé univerzitě, nebo o účast na nějakém výzkumu. Do této kategorie také spadají zprávy obsahující nabídky postgraduálního studia, které ale jsou velmi vyjímečné, protože moderátoři konference se ji snaží nezahltit reklamou a nabídnout pouze kvalitní příspěvky. U nabídky práce nebo studia je důležité kde se práce nachází, kdo ji nabízí, jaké jsou požadavky, počátek, trvání případně další podrobnosti.

Pokud odebíráme zprávy z konferencí v různých jazycích, může to automatickému systému značně přitížit. V některých případech mohou zprávy přicházet s přílohami, historií diskuse nebo jako denní digest. Takové zprávy jsou tedy velmi dlouhé, obsahují několik samostatných zpráv a je třeba určit, která z nich je ještě relevantní. Ve výsledku tedy rozpoznat relevantní zprávu a určit k jaké události patří není triviální.

Když se podaří rozpoznat událost, je třeba z několika emailů k ní vybrat informace, které jsou důležité, a určit, jestli jsou aktuálně platné. Některé informace se v emailech opakují a jsou použitelné k rozpoznání shodné události, jiné mohou událost doplnit o podrobnosti. Nemůžeme ani vyloučit několikanásobné doručení jedné a té samé zprávy. V případě, že událost už známe, měli bychom být schopni jednoznačně určit, zda se jedná o tu samou událost. Například každoroční setkání může mít stejný název, stejné místo konání, stejný obsah, jen datum se liší. A jak rozpoznat, že ta jediná z informací, která se liší, není změna data konání, ale definuje novou událost?

Pokud navíc předpokládáme, že zprávy získáváme z běžné e–mailové schránky, musíme počítat s velkým množstvím irelevantních zpráv. Těmi může být spam i běžná komunikace, která přes danou schránku probíhá. Přestože schránka může být dedikována pouze pro příjem zpráv z konferencí, stejně se tam obchodní sdělení dostanou a je tedy třeba počítat s tím, že ne všechna pošta bude zpracovatelná.

3.4 Lidské zpracování

Když dostane zprávu běžný uživatel, tak jako první provede rozparsování zprávy podle MIME formátu. Tuto činnost samozřejmě neprovádí ručně čtením kódu, který mu přijde od odesílatele. Toto za něj provede klienský software, který ke čtení zpráv používá. I výchozí webové rozhraní všech komerčně dostupných emailových serverů provádí tento rozklad zprávy. Jako první rozloží hlavičku na části a uživateli zobrazí seznam dostupných zpráv podle data, předmětu a odesílatele. Člověk provede první rozpoznání údajů zprávy. Když pozná jazyk jako jeden z těch, které ovládá, pokračuje dál, jinak zprávu s největší pravděpodobností rovnou smaže. Vyjímkou je, pokud je to zpráva od někoho známého nebo důležitého, potom zprávu ještě podrobí druhé kontrole. Poté se podívá na předmět a podle něj se rozhodne, zda zprávu číst, nechat na později, nebo smazat. Podle hlavičky je tedy velmi často možné odhalit spam nebo poznat úmysly celé zprávy.

Když známe téma zprávy, podíváme se do těla a přečteme si detaily. Při prvním čtení zjistíme, jestli je pravda to, co nám napověděla hlavička. Zvýšenou pozornost věnujeme prvním několika řádkům, které by nás měly informovat o hlavních bodech zprávy. Při každé větě, kterou si přečteme, náš mozek automaticky porovnává nově získanou informaci se všemi informacemi, které již známe. Asociativně si přiřazuje události k sobě, případně je chronologicky řadí. Navíc do hry vstupují emoce, které ovlivňují celou činnost uživatele. Při velkém zájmu může později dojít k dalším

čtením zprávy za účelem vypsání si klíčových bodů - zejména termínů. Tento způsob zpracování ale počítači úplně nevyhovuje, tak si zkusíme představit jinou situaci. A totiž člověka se špatnou pamětí, nebo sekretářku. Ta nemá informaci o tom, co by jejímu nadřízenému vyhovovalo. Proto přečte celou zprávu, pochopí, o co se jedná, a vypíše si klíčové informace. Potom se podívá do kalendáře nebo předá tyto informace nadřízenému, který provede jejich rychlé začlenění do svého kalendáře. A tento přístup je ten, který jsme schopni v počítači nasimulovat. Po odhadnutí, co by zpráva asi tak mohla znamenat, můžeme pro odhadovaný typ zprávy vyhledat v textu maximum relevantních informací. Když je máme, můžeme zkonzultovat nové znalosti se stávajícími.

Kapitola 4

Návrh

Nejsme schopni se na zprávu podívat a všechny informace najednou vědět. Rozložíme tedy práci programu podobným způsobem, jakým zpracovává informaci člověk. Rozložíme systém do několika částí (skupin modulů) a těmito částmi necháme každou zprávu postupně procházet. Vyzkoušíme si, jestli jsou jednotlivé části schopny autonomně provádět rozpoznávací činnosti, které by normálně člověku zabraly pár vteřin. Ve výsledku bychom měli mít oklasifikovanou zprávu s vydolovanými informacemi. Každá část se bude skládat z jednoho nebo několika modulů (podprogramy, dynamické knihovny, skripty...), které příslušným způsobem zpracovávají procházející zprávu. Tyto moduly mohou být přidávány do programu postupně. Při přidání nového modulu bude třeba připravit arbitra dané části, který z výsledků souběžně běžících modulů bude umět vybrat nejpravděpodobněji správný výsledek. Činnost částí bude řídit hlavní program, který bude interpretovat výsledky a umožňovat interakci uživatele.

Typy zpráv, které chceme rozpoznávat:

- Call for Papers - výzva k zaslání článků pro workshop/konferenci
- Nabídka studia, letní školy, laboratoří, nového studijního programu... může také být nazýváno Call for Applications
- Deadline extension - posunutí termínu odevzdání nebo notice - připomenutí, že se termín blíží
- Book announcement, call for books, journal announcement, ...

Jedna z částí systému by tedy měla klasifikovat, o kterou informaci v daném emailu jde. Podle této informace se potom použijí příslušné podprogramy pro získání relevantní informace.

Každá skupina modulů by měla mít jednotnou formu, aby byl systém rozumně škálovatelný a udržovatelný. Předpokládáme tedy, že každá součást je zařaditelná do postupného zpracování a má vstupní a výstupní porty. V první fázi budeme k přenosu používat XML soubory. Použití souborů umožní kontrolu funkčnosti jednotlivých modulů, protože bude možné soubor ručně prohlédnout. A pro účely testování

je možné i tyto XML soubory ručně připravit a testovat každý jednotlivý modul zvlášť. Každý modul by měl mít nějaký administrační vstup, kterým se upravují nastavení tohoto modulu. Protože předpokládáme, že modulem mohou být programy, knihovny nebo logické celky v programu, nelze přesně stanovit interface, který bude administrační část modulu mít. Může se jednat o vstupní množinu podmínek, množinu regulárních výrazů, konfigurační soubor, předem naučenou neuronovou síť, nebo jen volací parametry programu. Vše závisí na použitých modulech. Proto budeme používat jednotnou konfiguraci pro nastavení pořadí modulů, ale konfigurační soubory jednotlivých modulů necháme na nastavení každého modulu zvlášť.

Musíme ovšem zavést standard pro interface vstupní a výstupní formou XML. Předpokládáme, že moduly nejsou vždy schopny práci provést, ale také nejsou vždy schopny poznat, že ji nedokážou provést. Proto každý modul bude mít možnost ukončit provádění a nechat rozhodnutí problému na uživateli. Bude tedy existovat opět standardizovaný výstup, který se předloží vyššímu arbitrovi (uživateli), který rozhodne, jak se má s daným problémem nakládat dále. Po tomto autoritativním rozhodnutí by měl modul dostat problém zpět k řešení, tentokrát ale už se znalostí správného řešení. Když opět narazí na problém, použije znalost řešení k jeho vyřešení a případně využije tuto informaci k naučení se na příště. Učení systému by mělo probíhat s učitelem, nikoliv samovolně. Naším cílem není vytvořit samostatně myslící stroj, ale dostatečně spolehlivý automat, který je možné vylepšovat a přidávat mu nové vlastnosti. Ideálem by bylo, kdyby se přidáním několika málo příkladů dala přidat nová oblast zpracování, kterou dosud systém nezpracovával (například rozpoznat nabídky z obchodu, úkoly ze zaměstnání, ...).

Pro prohlížení práce systému a pro ruční klasifikaci pošty musí systém mít učící část, která umožní uživatelský zásah do zpracovávané pošty. Uživatel tak bude mít možnost pomoci systému s obtížnými zprávami. Systém by měl s postupem učení uživateli přenechávat čím dál méně zpráv. V ideálním případě by se měl dostat na sto procent zpracovaných zpráv.

4.1 Celkový vzhled systému

Jako první je použita logika na ochranu proti spamu, předpokládáme, že je implementována v mail serveru. Poté je příchozí mail duplikován a jedna kopie standardním způsobem uložena do schránky, druhá kopie je zanalyzována.

- Prvním pracovním modulem je vytěžovač informací z hlavičky. Hlavička emailu je relativně jednoduše zpracovatelná, protože její formát je pevně daný a pokud není správně formovaná, email ani není doručen. Cílem rozparsování podle hlavičky je získat dvojice klíč-hodnota obsahující důležité informace. To znamená:
    - Odesílatel — Jedná se o osobu, která zprávu odesílá, v případě konference tedy člověk, který zasílá zprávu do konference. Toto pole vyplňuje

odesílatel zprávy jako pole *From*.

- Příjemce — Osoba, která zprávu má dostat (uvedena jako příjemce — pole *To*). Typicky adresa konferenčního serveru.
- Adresa odesílatele — Nemusí se shodovat s odesílatelem, jedná se o adresu, která je uvedená v emailovém poli *Sender*. V případě poštovní konference se jedná o adresu konferenčního serveru. Tuto položku nevyplňuje odesílatel ale poštovní server v závislosti na tom, kdo se opravdu k serveru přihlašuje a zprávu odesílá.
- Adresa příjemce — Měla by obsahovat adresu, na kterou byla zpráva skutečně doručena, čili vlastní adresu našeho účtu. Toto pole se velmi často vůbec nevyskytuje a skutečného příjemce je poznat jen z toho, že víme, do kterého účtu byla zpráva doručena.
- Přenosy — Seznam serverů, přes které byla zpráva přeposílána, včetně časů, kdy na ně byla doručena.
- Předmět — Nadpis zprávy, který zadal odesílatel. Obsahuje důležité informace, které rozhodují o tom, jestli příjemce vůbec bude zprávu číst, nebo se ji rozhodne ignorovat či rovnou smazat.
- Datum odeslání — Datum a čas, kdy odesílatel označil zprávu jako připravenou k odeslání.
- Datum přijetí — Datum a čas, kdy byla zpráva doručena do cílové schránky.

Pokud má tělo mailu nějakou vlastní ručně psanou hlavičku nějakým výrazným způsobem oddělenou od zbytku textu, můžeme i rozparsování tohoto přidat do činnosti prvního modulu. Předpokládáme ale pouze jednoduché rozpoznání za pomoci regulárního výrazu. V rámci prvního zpracování hlavičky bude také určeno, o jaký formát dat se jedná, zpráva totiž může být ve formátu prostého textu, RTF nebo HTML. Jazykové a další moduly založené na pravděpodobnostních metodách mají problém s prací s rozšířenými formáty, proto je důležité získat z mailu čistý text. Tento text můžeme předat jako jednu ze získaných hodnot. Zprávu a získané pole hodnot poté předáme k dalšímu zpracování jazykovému modulu.

- Jazykový modul by měl rozpoznat jazyk zprávy. Pokud bude jazyk určený, můžeme omezit množství dalších operací, které se zprávou budeme provádět, protože můžeme mít další klasifikátory přesnější a chytřejší, když budou připraveny pro konkrétní jazyk. Informace o jazyku může být nejednoznačná, proto můžeme použít klasifikátory pro více jazyků. Třetí modul provede primární rozdělení kategorie. To je ve velkém množství případů možné provést už ze znalosti informací z hlavičky a předmětu vydolovaných prvním modulem. Určení kategorie může být nejednoznačné, proto bude kategorií označeno více a v následujícím zpracování se bude s mailem nakládat, jako kdyby patřil do všech z nich.

- Další skupina modulů se bude skládat z více modulů, které řeší každý jinou oblast, obecně je však nazveme zdobením zprávy. Půjde totiž o to, doopatřit jednotlivé části textu, které mají nějaký speciální význam. Toto zdobení předpokládejme ve dvou částech. Za prvé hrubá struktura, tj. určení řádků, rozpoznání seznamů, tabulek, nadpisů a ostatních formátovacích útvarů a za druhé sémantické detaily. Rozpoznání hrubé struktury může být jednodušší například při html formátovaných mailech, ale měli bychom být schopni rozpoznat zajímavé útvary i v prostém textu. Problémovým začíná být už dělení řádek, protože některé mailové servery automaticky řádky zalamují a my musíme potom vymýšlet, kde končí věta/odstavec a kde došlo k automatickému zalomení. Při rozpoznávání seznamů zase můžeme dojít k tomu, že sloučíme dva za sebou položené seznamy, nebo nerozpoznáme dvouúrovňový seznam. V tabulkách mohou být prázdné buňky a tím se nám celá tabulka posune, když to nezdetekujeme. Samotné rozpoznání struktury textu není jednoduché a může být potřeba porozumět syntaxi celku.

  Sémantickými detaily myslíme rozpoznání datumu, států, měst, míst, jmen, a dalších. Toto je jedna z nejsložitějších částí, protože k rozpoznaným částem musíme většinou znát i kontext, abychom z nich později byli schopni vytěžit nějakou informaci.

  Pro ukázku obtížnosti si ukažme následující větu z jedné reálné zprávy: *Conference will be held from June 29th to July 1st at the Villa Gualino Convention Center, on the hills overlooking Torino.*

  Z této věty musíme být schopni získat datum, kde si rok musíme doplnit podle data odeslání zprávy, název centra a název města. Stát si musíme odvodit ze znalosti pozice města Torino, pokud jej chceme uvádět a nestačí nám uživateli ukázat pouze město. Musíme si dát pozor na nejednoznačnost vlastních jmen. Město se stejným názvem může existovat i v různých světadílech, proto musíme hledat kompletní informaci. Vlastní názvy místních označení mohou také být zavádějící, například v předchozím příkladě může název místa vypadat jako určení budovy Villa Gualino v rámci Turínského konferenčního centra místo názvu celého tohoto centra. Po označení jednotlivých částí spočítáme výskyty a předáme dál.

- Tuto kompletní informaci další modul musí oklasifikovat, může k tomu použít výskytů slov, klíčových slov, i informace z primární kategorizace. Výsledkem je už konečná informace o tom, čeho se daný email týká a co je z něj třeba získat. Poté se provede ještě jemnější dozdobení v závislosti na typu zprávy. Označí se informace o kontaktních osobách, přihláškách, nástupech, rozšíří se okolí označených klíčových slov. Na text se spustí vlastní dolovače příslušné k danému typu zprávy. Tyto dolovače již hledají v označeném informaci, kterou potřebují, nikoliv opačně, že by hledaly v textu nějakou relevantní informaci.

- Výsledný soubor informací o události je poté zpracován posledním modulem,

který má za úkol porovnávat a třídit informace v databázi. Tento modul informaci zatřídí do databáze, rozpozná, jestli už tam náhodou není a případně upraví stávající událost podle nově získaných dat. Tento modul by měl zároveň zajistit, aby se s původním emailem ve schránce provedla akce příslušná k danému typu zprávy. To znamená třeba odstranit, aby s ní už uživatel nemusel zabývat, přesunout do vybrané složky, označit jako přečtené, nebo něco úplně jiného. To už musí záležet na nastavení uživatele používajícího systém.

Většina modulů umožňuje výstup do univerzálního učícího interface, přes který je možné systému pomoci s identifikací zpráv, se kterými si neumí poradit. Tento interface by měl zachytit všechny zprávy, které systém nedokázal zpracovat. Uživatel může přistoupit k učitelskému rozhraní a oklasifikovat zprávy ručně tak, jak by to měly jednotlivé moduly udělat automaticky. Pokud moduly jsou schopné učení, měly by při příštím zpracování již využít informace, kterou uživatel ručně zadal. V tomto rozhraní může uživatel prohlížet filtrované ty zprávy, které systém nezpracoval, a ručně rozhodnout o jejich dalším osudu ve schránce.

4.2 Řídící program

Hlavní program, který se stará o spouštění jednotlivých součástí musí být konfigurovatelný bez nutnosti překompilovávat program. Proto jsme v úvahu brali tyto možnosti:

- Shell script/dos command
  - Umožňuje dynamicky naprogramovat celé řízení systému
  - Je třeba pevně přizpůsobit operačnímu systému, nad kterým běží
  - Jednotlivé součásti musí být samostatně spustitelné programy
  - Předávání informací mezi součástmi je možné zřetězením vstupů a výstupů pomocí roury, nebo ukládáním souborů a předáváním odkazů v parametrech
  - Automatické spouštění je možné za pomoci démonů a automatických časovačů
  - Nejjednodušší na implementaci - není vlastně potřeba žádný řídící systém, je řízeno přímo skriptem
  - Obtížnější ladění
- Jediný řídící program
  - Řídící program je robustní a umožňuje jednodušší konfiguraci za pomoci pevně dané množiny příkazů.

- Podprogramy mohou být definovány jedním standardním rozhraním, které je možné implementovat v libovolných dynamických knihovnách. Přidání nového modulu znamená jen jeho vytvoření a zapsání do řídícího konfiguračního souboru.
- Mezi moduly je možné předávat přímo vnitřní struktury programu - struktury mailu, načtené xml soubory. Dá se bez úprav systému jen v jednom místě nastavit, jakým způsobem se mají informace mezi moduly předávat.
- Samotný program může být spuštěn jako služba na pozadí a sám si řídit volání a spouštění podprocesů.
- Úplná kontrola nad vlákny a množství prováděných operací.
- Umožňuje využívat jak hotových programů, tak dll.
- Možnost využívat grafického rozhraní programu.

• Samostatně řízené moduly

- Každá součást se řídí sama a je kompletním samostatným celkem.
- Spouštění je postupné kdy buď každý modul ví, které moduly mají následovat, a umí je zavolat, nebo jsou všechny součásti spuštěny zároveň a předávají si zprávy za pomoci signálů nebo souborů.
- Každý modul si musí načíst data pro svou práci. Není možné předávat kusy zpráv mezi moduly.
- Větší zátěž systému, více zároveň spuštěných procsů, složitá komunikace a synchronizace.
- Lepší možnost rozdělit práci mezi více strojů, každý modul může být spuštěn na jiném stroji.
- Nezávislé součásti, při chybě a pádu jedné z nich nejsou ostatní ovlivněny.
- Není centrální arbitr omezující výkon systému.

Pro implementaci jsme vybrali možnost jednoho řídícího modulu, který sám hlídá množství vláken zpracovávajících požadavky. Umožňuje nám hlavně rychlý přechod od ladící fáze, kdy se informace předávají v souborech, do konečné fáze, kdy se zprávy předávají přímo mezi podprogramy. Navíc umožňuje při použití modulů na jednotné platformě přenášet mezi nimi nejen textová data, ale i celé komplexní struktury, respektive pouze odkazy na ně, což výrazně sníží výpočetní nároky.

Kapitola 5

Dílčí problémy

5.1 Načtení pošty

5.1.1 Problém

Aby systém mohl vůbec nějakou poštu zpracovávat, musí se k ní nějakým způsobem dostat. První částí systému tedy musí být získávání zpráv. Předpokládáme offline zpracování zpráv, proto není třeba zprávy zachytávat v průběhu jejich doručování, nebo nějakým způsobem odklánět jejich cestu k serveru. Za důležité považujeme tyto parametry modulu:

Zátěž Systém by také neměl přespříliš zatěžovat poštovní server nebo jeho komunikační linky. Nesmí dojít k omezení funkčnosti nebo výkonu poštovního serveru.

Nastavení Nemělo by být příliš složité, nemělo by obtěžovat administrátora poštovního serveru. Ideálním řešením je možnost nastavení individuálně uživatelem.

Selekce Není třeba zachytávat a zkoumat veškerou poštu, už první modul by měl nějakým způsobem filtrovat a přijímat pouze relevantní zprávy.

Odezva Systém musí reagovat v rozumné době od doručení zprávy, nemusí se jednat o online zpracování, ale zpráva by měla přijít ke zpracování co nejdříve po doručení.

5.1.2 Analýza

Pro zachytávání pošty v průběhu jejího doručování je možné použít proxy server, který se postaví před vlastní poštovní server, ale toto řešení jsme zavrhli kvůli pracnosti pro administrátora serveru. Navíc bychom museli zpracovávat veškerou poštu, která na server přichází, včetně spamu, který později odfiltruje sám poštovní server.

Druhým řešením by byl zásuvný modul do poštovního serveru. Vývoj takového modulu by ovšem byl velmi náročný a vzhledem k množství aktuálně používaných

komerčních poštovních serverů by musel být vyvíjen pro každý server nový modul. Obě řešení se zachytáváním veškeré pošty serveru jsou nevhodná z hlediska požadavku na selekci. V úvahu proto přichází pouze metody práce s individuální schránkou.

Zde je možnost při doručení zprávy nastavit provedení různých akcí. Většinou se jedná o přeposlání zprávy, odeslání oznámení, nebo uložení na disk. Pokud bychom zprávu přeposílali, tak musíme zbytek našeho systému vytvořit jako vlastní mailserver, který je schopen přijímat emailovou komunikaci. Přeposílání zpráv ale umí každý poštovní server, takže by nebyl problém na straně původní schránky, ale mohli bychom se dostat do problémů s konektivitou. Když by náš server nebyl dostupný, nemuselo by se podařit zprávu doručit a náš systém by nevěděl, že taková zpráva přišla.

Pokud bychom využili přímého uložení na disk, může systém hned začít pracovat na uložené kopii zprávy, ale znamenalo by to, že náš systém musí běžet přímo na poštovním serveru. Jakákoliv správa programu by tedy byla obtížná a individuální nastavení uživatelů taktéž. Program by navíc vyžadoval určitou spolupráci s administrátorem.

Jako nejrozumnější se jeví využití protokolů pro vybírání poštovní schránky běžným klientem. Jedná se o protokoly POP3 a IMAP. Použití těchto protokolů přenáší odpovědnost za doručení zprávy na náš systém, který může běžet kdekoliv, odkud má konektivitu na poštovní server. Pokud se bude systém dotazovat poštovního serveru v pravidelných intervalech, zda nedorazily nové zprávy, jako to dělají běžné poštovní klientské programy, nezvýší se nijak výrazně zátěž serveru, a nejsou třeba žádné úpravy v jeho nastavení. Pokud bude interval rozumný, zachová se dostatečná odezva, aniž by se přetížila komunikační linka. Systém se vždy dotáže na nové zprávy a pokud nějaké budou, stáhne si jejich obsah na lokální disk. Navíc při dotazu na nové zprávy je obdržena i informace o jejich velikosti, nemusíme proto stahovat zprávy, které svou velikostí neodpovídají tomu, co očekáváme. Například můžeme stahovat jen zprávy menší než 200KB. U těch se dá předpokládat, že neobsahují reklamní letáky a jiné nepotřebné přílohy, ale jsou to opravdu jen texty z konferencí.

Některé poštovní servery umožňují třídění zpráv do složek. To je pro uživatele výhodou, protože může nastavit jednoduchý filtr, který bude do vybrané složky přesouvat jen ty zprávy, které přišly z konferenčních adres. Použitím protokolu IMAP můžeme dosáhnout i stahování pouze zpráv z těchto složek. Bohužel protokol POP3 toto neumí a umožňuje pouze obecné stažení zpráv ze serveru. Na serveru většinou jde ještě nastavit, zda má protokol POP3 stahovat zprávy jen z hlavní složky, nebo i z podsložek.

5.1.3 Návrh

Pro načtení pošty používáme protokoly POP3 a IMAP. Díky těmto protokolům jsme schopni napojit se na stávající schránky a nemusíme zatěžovat administrátory serveru tím, že by přesměrovávali svou poštu skrz nějaký speciální modul. Takovéto

použití umožní několik dalších využití tohoto modulu.

+ Můžeme definovat, ze kterých složek zprávy načítat. Nemusíme potom nechat systémem procházet jakoukoliv poštu, ale omezíme se na poštu od konferenčních serverů a tím systému ušetříme práci.

+ Umožníme uživateli nastavit maximální velikost stahovaných zpráv.

+ Po zpracování zprávy můžeme využít modulu k opětné komunikaci se serverem, kdy můžeme zprávy na serveru smazat přímo z uživatelovy schránky. Pokud server podporuje IMAP, nemusíme zprávy mazat, ale můžeme je přesouvat do různých složek. Můžeme si dokonce i vytvářet složky. Takže můžeme v rámci složky se zprávami z konference vytvořit podsložky udržující zprávy pouze k vybraným typům událostí, případně rozdělit složky s úspěšně zpracovanými, nezpracovanými, nebo odpadními zprávami. Pokud některé zprávy ohodnotíme jako význačné, můžeme je uživateli přesunout zpět do hlavní schránky, kde si je sám přečte, aniž by prohlížel výsledky našeho systému.

+ Máme na lokálním disku uložené kopie zpráv a můžeme je zpracovávat offline kdykoliv budeme chtít.

– Systém se snaží v pravidelných intervalech komunikovat se serverem a zjišťuje stav zpráv na serveru.

– Ukládáme kopie zpráv na lokálním disku a po jejich zpracování musíme řešit jejich mazání či zaarchivování.

5.2 Parsování hlavičky

5.2.1 Problém

Email chodí přes poštovní servery kódovaný ve formátu RFC822 nebo jeho rozšíření MIME. Rozdíl je v tom, že MIME formát podporuje i jiný obsah než jen textové zprávy. Podle tohoto protokolu obsahuje každá zpráva nějaké hlavičky, které jsou zpracovávány přímo poštovním serverem, tělo zprávy, které zobrazí klientský prohlížeč, a přílohy. Pokud zpráva obsahuje tělo ve formě prostého textu a jeho alternativu v HTML formátu, jsou obě tyto části zapsány jako přílohy. Protože náš systém nevyužívá žádný prohlížeč emailových zpráv, musíme provádět dekódování MIME formátu sami. Po dekódování hlaviček můžeme rovnou použít některé z jejích částí jako proměnné použité v našem systému.

5.2.2 Analýza

Formát zprávy podle doporučení RFC822 říká, že mailová zpráva obsahuje hlavičky a tělo. Hlavičky jsou jednotlivé řádky, které obsahují název hlavičky a její hodnotu.

Hodnota může být různým způsobem kódována, protože data jsou přenášena ve formátu ASCII, ale toto už řeší další rozšíření výchozího doporučení. Tělo zprávy podle RFC822 obsahuje prostý text této zprávy. Rozšíření na formát MIME definuje možnost přenosu jiných než textových informací v mailu. Jsou to různé přílohy typu obrázek, dokument, HTML formátovaný text, nebo vlastně jakýkoliv jiný soubor. Tělo tedy už neobsahuje jen text, ale může obsahovat další části, které opět mohou obsahovat různé hlavičky a vlastní těla, opět hierarchicky rozložitelná na další části. Rozparsování zprávy tedy není úplně triviální záležitostí.

Pro parsování hlavičky emailu využívá každý program svůj vlastní algoritmus. Toho využívají programátorské firmy, které vyrábějí doplňkové knihovny, které formáty mime načítají. Je dostupné velké množství placených i volně dostupných knihoven. Tyto knihovny zpracovávají formát mime každá po svém. Některé se snaží zpracovat všechnu poštu, která přijde, některé se snaží striktně dodržovat formát, ty ostatní něco mezi. Ty knihovny, které nekontrolují striktní dodržování protokolu, pomáhají šíření spamu, který občas má formát hlavičky poškozen.

5.2.3 Návrh

Pro parsování MIME formátu jsme využili PHP skript mimeparser[9], který nám ji rozebere do asociativního pole obsahujícího jednotlivé součásti. Pokud jsou části hlavičky navíc v různých kódováních, tento skript provede zpětné dekódování znaků v hlavičce, zakódovaných jako html entity, aby je bylo možné přenášet v prostém ASCII kódování.

Po rozparsování zprávy a načtení do paměti je třeba předat tuto zprávu dalším částem systému. Rozhodli jsme se pro formát XML. Tento formát můžeme uložit v kódování UTF–8, pokud byly hlavičky správně označny a dokážeme překódovat všechny znaky. Provádíme tedy první překódování dat z hlavičky. Předpokládáme, že kódování určené MIME formátem je správné, tudíž nezkoumáme platnost dekódovaných dat. Chybné hlavičky by vůbec neměly dorazit skrz poštovní servery, navíc nejsme schopni na několika znacích poznat správnost výsledku. Pokud hlavičky obsahují nějaké nepřekódovatelné znaky, budeme předpokládat, že celý mail je chybný.

Jiné je to s tělem mailu. Tělo mailu může mít označené kódování, které nemusí být ve všech případech správné. Mail servery a hlavně uživatelské klientské programy mohou s nestandardními kódováními zpráv mít problémy a občas jsou některé zprávy chybně označené. Proto v tomto kroku neprovádíme žádné dekódování dat těla zprávy a tělo zprávy a všechny přílohy ukládáme do souborů. Data z těchto souborů jsou zpracovávána v dalším kroku.

5.3 Rozpoznání jazyka

5.3.1 Problém

Rozpoznat jazyk psaného textu nemusí být jednoduchou záležitostí. Obecně člověk i počítač umí rozpoznat pouze ty jazyky, se kterými se již někdy setkal. A i tehdy nemusí být rozpoznání jisté, pokud daným jazykem člověk plynně nehovoří. V případě počítače je porozumění jazyku irelevantním pojmem, důležité je, jestli má někde uloženy významné znaky daného jazyka. Úkolem rozpoznávače jazyka je potom určit, že text, který mu byl předložen je napsán v určitém jazyce. V našem případě je spolu s rozpoznáním jazyka třeba určit i kódování, ve kterém je daný text uložen. K tomu navíc je potřeba určit, jestli daný text je prostý text, HTML kód, nebo RTF kód. Předpokládáme úplné rozpoznání, ke kterému nemáme žádné vstupní informace o kódování. Přestože tyto informace známe z hlavičky, nejsou vždy správné a rozpoznání kódování v tomto kroku může pouze posílit nebo oslabit informaci o důvěryhodnosti hlavičky.

5.3.2 Analýza

V kroku rozpoznání jazyka určujeme jazyk, kterým je tělo zprávy napsáno a zároveň kódování. Jaké kódování máme očekávat, to nám napovídá hlavička, ve které je kódování těla uvedeno. Předpokládáme však, že některé údaje jsou chybné, proto je radši ověřujeme přímo parserem.

Problematika rozpoznání jazyka neznámého textu je dostatečně zpracovávána lingvisty a je k dispozici velké množství skriptů a programů, které jazyk zprávy určí. Bohužel většina z nich je komerčních. Prozkoumali jsme však vlastnosti několika z nich.

XRCE Language Identifier [11] Online program firmy Xerox. Výsledek odhadu je bohužel autoritativní bez označení pravděpodobnosti správnosti. Systém má překvapivě velké problémy s textem, který obsahuje diakritiku. Naopak češtinu psanou bez diakritiky rozpoznává poměrně spolehlivě. To je způsobeno tím, že tento systém má pro češtinu pouze slovník v ASCII kódování a proto má tendenci kód zadaný v UTF8 rozpoznávat jako některý z jiných jazyků, které utf8 slovník mají. Při delším textu ale i tento systém rozpozná češtinu.

TextCat [12] TextCat je program napsaný v perlu. Je k dispozici s kompletním zdrojovým kódem a jednoduše dodefinovatelnými slovníky. Stačí přidat text napsaný v daném jazyce a program už si s ním poradí. Program funguje na základním principu frekvenční analýzy výskytu skupin znaků od jednotlivých znaků až po čtveřice. Program má problémy s rozpoznáním při příliš krátké větě, může se stát, že oznámí neúspěch. To je způsobeno optimalizací, která program zrychluje, ale ubírá mu na síle. Vzhledem k dostupnosti zdrojových kódů je možné toto vylepšení zrušit. Tento program se spouští z příkazové

řádky a dostávat výstup v textové podobě. Je k němu dostupné i jednoduché webové rozhraní. Nevýhodou je, že opět výsledkem je autoritativní rozhodnutí o jazyku bez označení pravděpodobných dalších možností.

NSTEIN [13] Profesionální společnost zabývající se dolováním textů. Samotný nástroj na detekci jazyků není dostupný. Jedná se o součást profesionálního textminingového řešení TME (Text Mining Engine). Toto řešení je schopné získávat informace z růýných typů dokumentů. Nástroj by měl sloužit k tvorbě sémantického webu a měl by být schopen provádět i celou funkčnost, jako náš systém. Původní systém Qué?, který byl čistě jazykovým modulem byl v provozu do roku 2007 a byl implantován do systému Nstein TME. Původní odkaz: `http://www.alis.com/en/services_que.html`

Translated Labs [14] Celá firma Translated a její výzkumná laboratoř Translated Labs se zabývají text miningem a rozpoznávač jazyka je jedna ze součástí jejich aplikací. Demo je dostupné pouze online. Češtinu rozpozná až při delších větách, zejména po přidání diakritiky. Bez diakritiky není schopné češtinu poznat. Rozpoznává ji to jako Fidži, Bengálštinu, Lingalštinu nebo Tahitštinu. Systém rozpoznává mnoho netradičních jazyků, ale potřebuje pro to mnohem přesnější zadání vstupu. Výsledek je opět formou autoritativního rozhodnutí.

Lextek [15] Lextek Language Identifier je program pro windows s velmi jednoduchým grafickým rozhraním. Firma by měla mít k dispozici SDK, které jednoduchým způsobem umožní implementaci tohoto programu do jakékoliv nové vyvíjené aplikace. Výrobce na stránkách uvádí, že program je velmi rychlý. Při zkušebních testech se nepodařilo prokázat, že by tomu tak nebylo. Při detekci češtiny bez diakritiky opět dochází k problémům a chybné interpretaci u kratších textů. U výsledku není uvedeno, jak si je program jistý výsledkem, ani jaké jsou alternativy, které přicházejí v úvahu.

Statistické rozpoznávání jazyka [10] Online verze jazykového rozpoznávače použitého naším systémem. Tento software s webovým rozhraním určí jazyk a graficky zobrazí stupeň jistoty. Je možné zobrazit informaci o jednom, pěti nejlépe hodnocených, nebo všech dostupných jazycích. Při krátkých textech je v tomto programu velmi znát příbuznost češtiny a slovenštiny. Jako jeden z mála je schopen pracovat a úspěšně rozpoznat češtinu s diakritikou i bez diakritiky. To se vzhledem k našemu využití jeví jako jedna z nejdůležitějších součástí systému, vzhledem k časté komunikaci programátorské komunity bez diakritiky. Program je kompletně dostupný včetně zdrojového kódu a dokumentace. Jedná se o bakalářskou práci Kamila Tomana, který svolil s využitím pro naše účely.

SILC [16] SILC (Systeme d'Identification de la Langue et du Codage) je rozpoznávačem jazyka univerzity v Montrealu. Demo je volně přístupné přes webové

rozhraní, ale systém je možné integrovat do libovolné aplikace pod mnoha platformami. Výhodou systému je, že určuje nejen jazyk dokumentu, ale i jeho kódování. Tento program je možné získat jako knihovnu a využít jej ve své aplikaci za použití dobře definovaného rozhraní. V demu neuvádí procentuelní váhu své jistoty, ale v programovém rozhraní je možné získat jednotlivé váhy pro každý jazyk. Opět má problémy s češtinou zadanou bez diakritiky při krátkém textu. S prodlužující se délkou se množství špatně ohodnocených textů blíží k nule. Systém se nepodařilo získat, protože jsme zjistili, že očekávané dll je obyčejný trojský kůň.

Talenknobbel [17] Zobrazuje seznam pravděpodobností všech jazyků, které zná a se kterými identifikoval jazyk, jeho hlavní nevýhodou je, že neumí češtinu. Zajímavé na něm je, že při rozpoznání jazyka není příliš velký rozdíl v pravděpodobnostech prvního a dalších jazyků. Tento rozdíl se zvětšuje až s rostoucí délkou textu, ale při jednom odstavci je rozdíl mezi prvním a druhým do pěti procent. Při porovnání s českým software z ufalu je na shodném textu jistota čtvrtinová. Systém není příliš spolehlivý. Když vezmeme příklad, který přímo na stránce uvádějí: "my name is john smith and i can speak many languages", tak tento rozpoznávač je velmi zmatený, určí jazyk jako indonézštinu a angličtinu zařadí až na 17. pozici s pouze 39,6

Polyglot 3000 [18] Polyglot 3000 je programem pro Windows, který má velmi jednoduché grafické rozhraní a velmi dobrou spěšnost. U delšího textu se prakticky nemýlí a u kratších také vykazuje dobrou úspěšnost. Navíc u každého výsledku zobrazuje procentuální jistotu svého odhadu. K rozpoznanému jazyku ještě ukáže seznam podobných jazyků. Výsledky ale nelze z programu získat nějakým rozumným způsobem, jedná se totiž o program pro windows, který vypisuje výsledky do svého okna, ze které je není možné ani zkopírovat.

Petamem [19] Tento web obsahuje řadu nástrojů pro práci s textem. Bohužel stránky většinou nefungují, přestože se jednou podařilo vyzkoušet funkční verzi systému. Jedná se o Českou firmu, která se zabývá prací s textem a nechává si za tuto práci platit. Stránky umožňují pouze zpracování krátkých částí textu, za každou delší část je třeba doplácet. Systém občas běží a je možné jej vyzkoušet, nepodařilo se jej však podrobit rozsáhlejšímu testování. Pro naše účely byl označen jako nevhodný.

U programů očekáváme označení možných jazyků textu plus nějaké určení jistoty odhadu. Jako ideálním výsledkem jsme si představovali výsledek:

```
Česky, UTF8, 95%
Cesky, UTF8, 80%
Slovensky, WINDOWS1250, 40%
English, ASCII, 5%
```

Z těchto dat náš systém vezme dostatečně pravděpodobné možnosti a zkusí další kroky v závislosti na jazycích, které zná.

5.3.3 Návrh

Použili jsme systém, který používá frekvenční analýzu výskytu různě dlouhých shluků znaků [10]. Potřebujeme proto pro rozpoznání jazyka dostatečně dlouhý text. Systém potřebuje být naučen. Umí rozpoznávat pouze ty jazyky a ta kódování, která jsme jej předem naučili. Z toho vyplývá, že je nutné připravit sadu testovacích učících dat, ze kterých se nástroj naučí požadované jazyky. Zde máme několik možností. Jednou z nich je připravit učící texty náhodně a na nich program naučit. Tuto možnost jsme vyzkoušeli a systém dokázal rozpoznávat jednotlivé jazyky. Lepších výsledků se ale dosáhne poté, co se pro učení využijí přímo doručené zprávy, které obsahují slova běžněji použitá ve zkoumaných konferencích.

5.4 Primární kategorizace

5.4.1 Problém

Úkolem primární kategorizace je na základě známých hodnot z hlavičky odhadnout obsah zprávy. K tomu můžeme využít různých regulárních výrazů a vyhledávání klíčových slov v předmětu nebo adresách odesílatele a příjemců. Také použijeme znalost jazyka, abychom určili k dalšímu zpracování pouze ty kategorie, které jsme schopni dále zpracovávat. K určení kategorie můžeme taktéž využít úvodní části těla zprávy, která občas obsahuje ručně vytvořené záhlaví popisující oznamovanou událost.

5.4.2 Analýza

Zprávy typu *Call for Papers* zpravidla obsahují znaky "CFP"přímo v předmětu zprávy. Zdá se tedy, že odhad kategorie tímto způsobem bude u těchto zpráv správný. Co ale zprávy, které obsahují v předmětu pouze název události? I tady se můžeme spolehnout na název, protože workshopy mívají přímo slovo workshop v názvu. Zde už opět bude záviset na různých kategoriích zpráv, které si nadefinujeme, jestli budeme schopni rozlišit detaily v *Call for Workshop*, *CFP: our Workshop* a dalších velmi podobných označeních.

V tomto kroku bude asi nutné vytvořit několik regulárních výrazů, které budou popisovat obvyklou strukturu předmětu zprávy pro dané téma. Pro větší jistotu by bylo dobré zkontrolovat i tělo zprávy. Zde ale zatím nemáme žádnou klasifikaci obsahu, žádné tagy určující obsah zprávy, můžeme tedy pouze kontrolovat výskyt klíčových slov.

5.4.3 Návrh

Využijeme načítací metodu a ke každému typu události předem vytvoříme seznam vlastností, které by zpráva měla mít, aby byla touto událostí. Poté sečteme vlastnosti a zjistíme, pro které typy události má zpráva dostatečné množství klíčových znaků. Jednotlivé typy si označíme takovou pravděpodobností, jakou je počet nalezených znaků ku celkovému počtu znaků dostupných pro daný typ zprávy. Tím získáme tabulku pravděpodobnosti typu zprávy.

Druhým krokem bude zpracování těla. Použijeme jednoduchý stavový automat, který projde tělo a spočítá výskyty klíčových slov. Klíčová slova budou opět uložena u jednotlivých typů zpráv. Opět spočítáme pravděpodobnost, že zpráva je daného typu. U této druhé metody bude asi třeba delšího ladění, než se podaří najít správné množiny slov. Výsledkem bude opět tabulka pravděpodobností.

Po dvojím zpracování pravděpodobnosti odhadu typu zprávy získáme lepší rozdělení a zůstane nám menší množina možných typů zprávy. V ideálním případě bude pouze jeden typ zprávy dostatečně pravděpodobný. V dalším kroku tedy budeme pokračovat pouze s těmi nástroji, které jsou určené pro odhadnuté typy zprávy. V tomto kroku by také mělo dojít k rozpoznání spamu, který se už nadále nebude zpracovávat žádnými nástroji, a provedou se odpovídající kroky. Bude záležet na uživateli, jestli bude spam rovnou mazat, přesouvat, nebo si jej nechá k individuálnímu přečtení.

5.5 Hrubé tagování

5.5.1 Problém

Úkolem tagovací části je rozpoznat důležité útvary v toku textu. Prvním úkolem je rozpoznat konce řádků a odstavců. Protože řádky v poště jsou automaticky zalámány na určitou délku při přenosu přes některé servery, je třeba rozpoznat, kdy začíná nový řádek a tím i nová věta, a kdy je zalomení řádku způsobeno technickými podmínkami přenosu. Zjistit správné řádkování a rozmístění bílých znaků je důležité při další činnosti modulu. Rozpoznávání seznamů a tabulek. Uživatelé, kteří píší zprávy ve formátu prostého textu, mají také potřeby zvýraznit určité části zprávy rozdělením na seznamy. Zpřehledňuje to zprávu a umožňuje příjemci rychle očima přecházet mezi jednotlivými body zprávy. V prostém textu je ale jakékoliv formátování zpráv velmi obtížné a je možné ho dosáhnout využitím bílých znaků jako jsou tabelátory, mezery a konce řádků, nebo nějakých běžně nepoužívaných znaků, například hvězdiček, svislých čar, křížků, ampersandů, zavináčů a dalších. Tyto sekvence je tedy třeba detekovat a určit, kdy se jedná o grafické útvary, které napomáhají porozumění textu. Zejména důležité jsou tabulky, protože nadpis sloupce říká hodně o obsahu tohoto sloupce, bez jehož znalosti nejsme schopni dešifrovat význam hodnoty o pár řádek níže.

Pomoci s rozdělením zprávy na logické celky může alternativa zprávy zaslaná

ve formátu HTML. V HTML je jednoduché najít tabulky a seznamy, neboť jsou uvozeny speciálními tagy. Proto je v počátku zkoumání zprávy třeba zjistit, jestli se ve zprávě náhodou nevyskytují HTML verze zpráva, případně jestli celá zpráva není jen v HTML. Je ale zvykem, pokud se posílá HTML, aby byla přiložena i verze zprávy ve formátu prostého textu.

5.6 Jemné tagování

5.6.1 Problém

Jemným tagováním myslíme rozpoznání a označení sémanticky významných útvarů. Těmito útvary jsou myšlena data, názvy, jména, klíčová slova, čísla. Pro jemné tagování je možné využít výsledků hrubého tagování, které určí jednotlivé textové bloky. Můžeme proto detaily označovat v menších částech textu. Úkolem je nalezené objekty buď označit uvnitř textu, nebo vytáhnout a uložit zvlášť, aby již byly přímo dostupné pro hlavní klasifikátor a dolovače.

5.6.2 Analýza

Protože v textu vyhledáváme mnoho různých útvarů, bude asi nejlepší pro každý typ útvaru mít vlastní rozpoznávač/tagovač. Všechny individualizované tagovače můžeme poté spouštět z jednoho řídícího taggeru, čímž je možné dosáhnout toho, aby se všechny tagovače dohromady tvářily jako jeden, který umí otagovat více věcí. V případě vyhledávání regulárních výrazů je možné vytvořit jeden nástroj, který bude umět nalézt všechny výrazy, které jsou zadefinovány. Problémem regulárních výrazů jsou minimální učící schopnosti stroje. Pro přidání nové funkčnosti je třeba ručně připravit nový regulární výraz. Systém může uživateli pomoci tím, že zkusí připravit výraz pro vybranou část textu, závěrečnou kontrolu správnosti ale vždy musí provádět uživatel.

Datum dokážeme rozpoznat za použití několika složitějších regulárních výrazů. Největší překážkou v rychlém nalezení je množství národních zvyklostí. Zde nám bude selhávat určení jazyka, protože přestože do konference je zasílán mail anglicky, použité označení data je lokální. Většina uživatelů zapíše datum tak, jak je zvyklá ve svém jazyce, aniž by přemýšleli nad tím, že píší anglicky. Samotným problémem zde bude už určení číselně zadaného dne a měsíce. 7.8. Může znamenat sedmého srpna, nebo osmého července. Zde naopak je třeba přihlédnout na jazyk, ve kterém je mail zapsán. Pomoci může také rok. 2009.07.08 již dost jasně napovídá, že se bude jednat o osmého července. Naopak 7.8.2009 v českém prostředí znamená sedmého srpna. Pro známé jazyky musíme také umět pracovat se slovně zadanými měsíci. Anglické měsíce a jejich zkratky jsou standardně používané i v jiných jazycích. Například data typu 2009 aug 7 jsou běžná, dokonce jsou občas použita i v hlavičkách zprávy. Přístupu k hlavičkám můžeme také využít a prozkoumat, jestli se nějaké z dat

náhodou nevyskytuje v již nalezených částech. To nám může pomoci s rozhodováním, jestli jsme datum našli nebo ne.

Názvy měst, států a míst můžeme odvozovat za prvé podle slovníku, za druhé podle velkých písmen na začátku slov. Rozpoznávání podle velkých písmen může vézt k nalezení vlastních jmen, ale také začátků vět. Slovník naopak nemusí obsahovat všechna města a státy, které se mohou ve zprávě vyskytovat. Vyhledávání proti slovníku také může být značně náročné, protože slovník měst bude obrovský.

Rozpoznávání vlastních jmen souvisí s vyhledáváním názvů. Názvy univerzit i měst mohou být víceslovné, s různě použitými velkými a malými písmeny. Proto se využití slovníku jeví jako lepší řešení, než vyhledávání podle velkých písmen. Problémem také může být správné přiřazení slov k sobě. Několik vlastních jmen po sobě může znamenat seznam míst, nebo dlouhý název jednoho místa o více slovech, nebo kombinaci několika názvů o několika slovech. Jako příklad si opět můžeme vzít *Villa Gualino Convention Center*. Při pohledu bez slovníku máme čtyři názvy s velkým písmenem, tudíž čtyři vlastní jména. V tomto případě neuděláme chybu, když vezmeme, že všechna vlastní jména neoddělená čárkou jsou jeden název. Když si ale představíme, že slovo *convention* bude s malým písmenem, už nám tato úvaha nepomůže.

Může nám ale pomoci označení klíčových slov. Klíčovými slovy se myslí právě slova typu *convention center, university, city, valley, research, group, laboratory...* Tato slova často doplňují názvy míst a pomáhají určit význam některých jmen.

5.7 Systém učení

5.7.1 Problém

Od systému učení požadujeme co největší variabilitu. Každý modul může mít problém, který nedokáže vyřešit. Výstup potom provede do pevně nadefinovaného formátu. Tento formát bude obsahovat veškeré informace potřebné k tomu, aby byl uživatel schopen rozhodnout, jak má systém zprávu zpracovat. Po uživatelově autoritativním rozhodnutí bude systém zpracovávat zprávu dál s tím, že použije uživatelem provedené rozhodnutí. Pokud bude systém v učícím režimu, nebo uživatel explicitně přikáže učící režim, systém zkusí ještě jednou provést stejnou úlohu, na které se zasekl, ale dodá provádějícímu modulu ještě uživatelem zadané informace o správném výsledku. Tento modul potom, pokud je toho schopen, využije informaci o správném výsledku k naučení se správného chodu. Tato definice je nepřesná, protože moduly mohou být slovníkové, které si přidají uživatelsky zadaná data do slovníku, regulární, které nedokáží ze správně nalezených výsledků odvodit regulární výraz, který k nim vedl, nebo třeba založené na principu neuronových sítí, které naopak potřebují množinu trénovacích dat, na kterých se naučí.

K systému učení by měl existovat uživatelsky přívětivý interface, který umožní uživateli rychle a efektivně zpracovat zprávy, které systém nezvládne. Tento sys-

tém vlastně nahradí běžný uživatelův prohlížeč zpráv, takže by měl být dostatečně komfortní na splnění této funkce.

5.7.2 Analýza

Mail vyhozený ze zpracování se uživateli zobrazí s tím, ve kterém modulu se zasekl, jaký byl výsledek/chyba, a umožní provést ruční rozhodování. Učitel musí být napsán přesně na míru každému modulu. Každý z modulů bude umět udělat standardizovaný výstup pro učitele. Tento výstup může obsahovat i libovolná individuální data daného modulu. Tato individuální data bude poté muset zpracovat i individualizovaný učitel. Zpráva se do učitele dostane i v případě, že systém běží v učícím režimu. V tomto případě pak zůstává na uživateli, aby pouze zkontroloval, že výsledek běhu byl správný.

Jedním z řešení tohoto systému je webové rozhraní k učiteli. Má tu výhodu, že je přístupné odkudkoliv a uživatel tak nebude vázán k přítomnosti u počítače se systémem. Stačí, aby systém byl dostupný přes web. Toto řešení může být sice trochu pomalejší, než standardní lokální aplikace, umožní však mnohem větší variabilitu. Aplikaci by bylo třeba psát pro každý operační systém, využití grafického rozhraní by mohlo vyžadovat instalaci dodatečných nadbytečných částí operačního systému.

Pro tvorbu webového rozhraní uvažujeme dva jazyky a to ASP.NET a PHP. ASP.NET běží na platformě .NET, nad kterou je napsaná hlavní část našeho systému. Je tedy možné využít některé komponenty systému, jako jsou načítání mailů, parsování, případně různé datové struktury. Systém automaticky podporuje nastavení jednotného vzhledu stránek a striktně dodržuje typovou kontrolu. PHP je otevřenější, existuje větší množství volně dostupných komponent a práce s ním je jednodušší díky minimální typové kontrole. Jednodušeji se vytváří výstup a je jednoduché psát výstup přímo v HTML.

Kapitola 6

Závěr

Na trhu jsou v současné době nástroje umožňující automatické zpracování e–mailové komunikace. Tyto nástroje vyžadují rozsáhlé vlastní úpravy nastavení, které vyžadují obsáhlé znalosti problematiky a programování s regulárními výrazy. Ani po vlastních nastaveních systému nemusí být úspěšnost dostatečná a je třeba opět zasahovat do nastavení nástrojů. Náš systém by měl uživatelům pomoci zpracovávat jejich příchozí poštu už ve výchozím nastavení. Nové úpravy či nastavení systému pro zpracování nových typů zpráv jsou jednoduché při využití stávajících modulů. Navíc systém umožňuje nahrazení libovolných modulů vlastními implementacemi a tím dosáhnout lepší výkonnosti, případně rychlosti. Testování úspěšnosti našeho systému nebylo provedeno, neboť systém není dokončen.

Literatura

[1] Ronen Feldman, James Sanger: *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press, 2006.

[2] Martin Pilát: *Získávání a správa údajů o konferencích a workshopech.* Bakalářská práce, MFF UK, Praha, 2007.

[3] Anne Kao, Steve R. Poteet (Eds.): *Natural Language Processing and Text Mining*, Springer, 2006.

[4] Tweak Marketing: *Advanced Email Parser,*`http://www.tweakmarketing.com/products/aep`. Praha, 2009.

[5] Parker Software: *Email2DB,* `http://www.email2db.com`. Stoke on Trent, Spojené Království, 2009.

[6] William J Astore: *Uncle Sam's cyber force wants you,* `http://www.atimes.com/atimes/Front_Page/JF28Aa01.html`, 2008.

[7] Deputy Secretary of Defense William J. Lynn, `http://www.defenselink.mil/speeches/speech.aspx?speechid=1365`, Center for Strategic and International Studies, Washington D.C, 15.6.2009.

[8] Oldřich Klimánek:*Hrozí světu kybernetická válka?,* `http://www.scinet.cz/hrozi-svetu-kyberneticka-valka.html`, Scinet, 2007.

[9] Manuel Lemos:*Mime Email Message Parser,* `http://www.freephpscripts.eu/scripts/149/Php-Mail-Scripts/MIME-E-mail-message-parser`.

[10] Josef Toman: *Statistické rozpoznávání jazyka* `http://quest.ms.mff.cuni.cz/~toman/recognition`. Bakalářská práce, MFF UK, Praha, 2007.

[11] Xerox Research Centre Europe (XRCE): *Language Identifier,* `http://www.xrce.xerox.com/competencies/content-analysis/tools/guesser.en.html`. Francie, 2009.

[12] Gertjan van Noord: *TextCat Language Guesser,* *http://odur.let.rug.nl/~vannoord/TextCat/Demo/textcat.html*. USA, 1994.

[13] Nstein Technologies: *TME Language Detector* *http://www.nstein.com/products/tme/modules/automatic-language-detection.html*. Montreal, Kanada, 2009.

[14] Translated Labs: *Language Identifier,* *http://labs.translated.net/language-identifier*. Řím, Itálie, 2009.

[15] Lextek International:*Lextek Language Identifier SDK,* *http://www.lextek.com/langid*. Provo, USA, 2000.

[16] Université de Montréal: *Syste'me d'Identification de la Langue et du Codage,* *http://rali.iro.umontreal.ca/Silc/index.jsp?lang=en*. Montréal, Kanada, 2009.

[17] Joost: *Talenknobbel,* *http://www.fuzzums.nl/~joost/talenknobbel/index.php*. Holandsko, 2009.

[18] Likasoft:*Polyglot 3000,* *http://www.polyglot3000.com/index.shtml*. Pervolia, Kypr, 2009.

[19] Petamem:*Language Identification,* *http://nlp.petamem.com/en/langident.cgi*. Praha, 2007. Stránky v současné době nefunkční.