Univerzita Pardubice
Fakulta ekonomicko – správní

Srovnávací studie text miningových nástrojů

Lukáš Hrdlička

Diplomová práce

2009

Univerzita Pardubice
Fakulta ekonomicko-správní
Ústav systémového inženýrství a informatiky
Akademický rok: **2008/2009**

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Lukáš HRDLIČKA**

Studijní program: **M6209 Systémové inženýrství a informatika**

Studijní obor: **Informatika ve veřejné správě**

Název tématu: **Srovnávací studie text miningových nástrojů**

Zásady pro vypracování:

1) text mining (popis hlavních pojmů a principů data miningu, text miningu)
2) nástroje text miningu (analýza nástrojů, jenž budou srovnávány)
3) srovnání text mining nástrojů (porovnání těchto nástrojů, jejich výhody a nevýhody)

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

[1] FELDMAN, Ronen, SANGER, James. Text mining handbook: advanced approaches in analyzing unstructured data. Cambridge University Press, 2007. 410 s. ISBN 978-0-521-83657-9.
[2] BERRY, Michael J. A., Linoff, Gordon S. Data Mining Techniques: for marketing, sales and customer relationship. Wiley: Indianapolis, 2004. 454 s. ISBN 0-471-47064-3.
[3] WITTEN, Ian H., FRANK, Eibe. Data mining: practical machine learning tools and techniques, Morgan Kaufmann: San Francisco, 2005. 525 s. ISBN 0-12-088407-0.

Vedoucí diplomové práce: **doc. Ing. Pavel Petr, Ph.D.**
Ústav systémového inženýrství a informatiky

Datum zadání diplomové práce: **6. října 2008**

Termín odevzdání diplomové práce: **1. května 2009**

L.S.

doc. Ing. Renáta Myšková, Ph.D.
děkanka

doc. Ing. Jiří Křupka, Ph.D.
vedoucí ústavu

V Pardubicích dne 6. října 2008

Prohlašuji:
Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně Univerzity Pardubice.

V Pardubicích dne 30. 4. 2009

Lukáš Hrdlička

PODĚKOVÁNÍ

Touto cestou bych rád poděkoval svému vedoucímu práce doc. Ing. Pavlu Petrovi Ph.D za odborné vedení mé diplomové práce, cenné rady a připomínky k diplomové práci.

Také bych rád poděkoval svým rodičům za celoživotní podporu a za to, že mi umožnili studovat.

SOUHRN

Diplomová práce se zabývá analýzou text miningových nástrojů a následným porovnáním včetně určení jejich výhod a nevýhod. Analýza text miningových nástrojů bude soustředěna do procesu extrakce informací, jakožto hlavního využití oblasti text miningu. Použitelnost těchto nástrojů bude ověřena na příkladě webovské stránky a textového souboru. Hlavním cílem práce je najít optimální alternativu. K porovnání alternativ byl zvolen program Criterium Decision Plus.

KLÍČOVÁ SLOVA

Text mining, extrakce informací, vyhledávání informací, zpracování přirozeného jazyka, Clementine, GATE, RapidMiner, AHP

TITLE

Comparative study of text mining tools

ABSTRACT

This diploma thesis deals with analysing text mining tools with comparision of them and specifying their advantages and disadvantages. Analysing of text mining tools will be concerned to information extraction, as main usage of text mining. Application of these tools will be tested on web page and text file. The aim of the thesis is to find optimal alternative. For comparison of alternatives was chosen software called Criterium Decision Plus.

KEYWORDS

Text mining, Information extraction, Information retrieval, Natural language processing, Clementine, GATE, RapidMiner, AHP

Obsah

Úvod

Svět zažívá v posledních letech explozi informací. Množství informací v textové formě roste přímo exponenciálně. Může za to převážně rozvoj Internetu. Množství digitálních knihoven, e-mailů a podnikových textových záznamů neustále vzrůstá. Nastává tedy čas pro využívání text miningu, který byl vyvinut pro objevování nových informací z velké kolekce textových dat. Pomocí text miningu lze z textu odkrýt skryté vztahy, vzorce a trendy.

Nejvýznamnější oblastí text miningu je extrakce informací. Právě na tento proces bude v této práci zaměřena největší pozornost. Provede se analýza extrakce informací těchto systémů: Clementine, RapidMiner a GATE. Clementine zastupuje komerční data miningový nástroj. RapidMiner je také data miningový nástroj, který je ale poskytován zdarma. Oba zmíněné nástroje pro potřeby text miningu potřebují doinstalování doplňku. Dalším nástrojem je GATE, což je text miningový nástroj poskytovaný zdarma.

Diplomová práce se skládá z pěti hlavních částí. V první části budou vysvětleny základní pojmy text miningu, které jsou důležité pro pochopení tématu. Mezi tyto pojmy patří - data mining, extrakce informací, kategorizace textu, získávání informací a zpracování přirozeného jazyka. V dalších třech částech budou postupně představeny jednotlivé text miningové nástroje. Tyto nástroje budou analyzovány na webovské stránce a prostém textovém souboru. Poté se shrnou jejich výhody a nevýhody. V páté kapitole budou každému nástroji přiřazena kritéria a provede se jejich vzájemné porovnání. Porovnání bude provedeno metodou analytického hierarchického procesu (AHP), pomocí programu Criterium Decision Plus (CDP).

Cílem diplomové práce je vymezení základních pojmů text miningu, analýza text miningových nástrojů, srovnání těchto nástrojů, určení výhod a nevýhod, a hodnocení použitých nástrojů.

1 Základní pojmy

Pro lepší pochopení následujících kapitol a celé problematiky následuje vymezení jednotlivých pojmů, týkajících se dané problematiky. Text mining vychází z oblasti data miningu. Nejprve bude tedy vysvětlen pojem data mining. Poté budou následovat jednotlivé pojmy z oblasti text miningu.

1.1 Data mining

Řada firem během posledních desetiletí či let vytvořila a nyní spravuje rozsáhlé informační databáze a datové sklady. Bylo odhadnuto, že množství dat se ve světových databázích každých 20 měsíců zdvojnásobuje [28]. Svět se tak stává stále složitější a zahlcuje nás daty, které vytváří. Výskyt přemíry dat a prudký rozvoj databází přivádí do popředí oblast data miningu. Inteligentně analyzovaná data jsou cenným zdrojem, který může vést k novým pohledům a v obchodním prostředí ke konkurenční výhodě.

Data mining lze charakterizovat jako proces extrakce relevantních, předem neznámých nebo nedefinovaných informací z velmi rozsáhlých databází. Fayyad definoval data mining takto: „Data mining je netriviální proces zjišťování platných, neznámých, potenciálně užitečných a snadno pochopitelných závislostí v datech.“ [7]

Data mining je tedy definován jako proces odhalování vzorů v datech. Proces musí být automatický, nebo alespoň jako je tomu ve většině případů poloautomatický. Objevené vzory musí být významné v tom, že vedou k určitým výhodám, které jsou převážně ekonomické. Data mining tedy pojednává o extrakci nebo dobývání znalostí z velkého množství dat. [28]

V oblasti data miningu se setkáváme s celou řadou úloh v řadě odvětví. Data mining využívá širokou škálu matematických a statistických technik, například shlukovou analýzu, klasifikaci, rozhodovací stromy, neuronové sítě, genetické algoritmy a jiné. V ekonomické oblasti se data mining používá například při analýze a predikci úvěrového rizika, predikci rizika při vydávání kreditních karet a podobně. Členění data miningových úloh je názorně zobrazeno v tabulce 1.

Tabulka 1: Úlohy a metody data miningu. Zdroj: [3]

Úloha	Metoda
Klasifikace	Diskriminační analýza
	Logistická regresní analýza
	Klasifikační (rozhodovací) stromy
	Neuronové sítě (algoritmus "back propagation")
Odhady hodnot vysvětlované proměnné	Lineární regresní analýza
	Nelineární regresní analýza
	Neuronové sítě (RBF "radial basis function")
Segmentace (shlukování)	Shluková analýza
	Genetické algoritmy
	Neuronové shlukování (Kohonenovy mapy)
Analýza vztahů	Asociační algoritmus pro odvozování pravidel typu If X, then Y
Predikce v časových řadách	Boxova-Jenkinsova metodologie
	Neuronové sítě ("recurrent back propagation")
Detekce odchylek	Vizualizace
	Statistické postupy

1.2 Text mining

Data lze najít v různých formách. Některé z nich jsou vhodnější pro automatickou analýzu dat a tedy snazší ke zpracování, jiné zase složitější. Obvyklé metody datové analýzy předpokládají, že data jsou uložena v tabulkách uspořádaných do polí, které mají nadefinovaný rozsah možných hodnot. Zde vyvstává otázka, co lze dělat, pokud jsou data uložena v textové formě, tedy když nemáme žádné záznamy – máme jen text. Nyní již máme metody, pomocí nichž manipulujeme s textem, za účelem získání poznatků z těchto dat. Touto problematikou se zabývá oblast text miningu, čili dobývání znalostí z textů. [19]

Text mining je poměrně nový obor využívající počítačové technologie, který odhaluje dosud neznámé informace automatickou extrakcí informací z textových zdrojů. Klíčovým prvkem je spojování extrahovaných informací do podoby nových faktů nebo nových hypotéz. [27]

Definice text miningu není jednoznačně dána, Nahm a Mooney například popisují text mining jako hledání zákonitostí v nestrukturovaném textu [2] nebo modifikovaná verze definice data miningu dle Fayyada zní: „Text mining je netriviální proces zjišťování platných, neznámých, potenciálně užitečných a snadno pochopitelných závislostí v textech.“ [7] V současnosti text mining představuje specifickou oblast zahrnující různé nástroje pro klasifikaci, filtrování textu, shlukování, extrakci informací, sumarizaci a další. Text mining se zabývá zpracováním nestrukturovaných dat. To jsou data, která nemají předem danou strukturu, například délky slov, přesto však mohou být použita ve strukturách jako věty, odstavce, slovní spojení. Na

rozdíl od zpracování číselných dat je však text mining plně závislý na národních zvyklostech, tedy jazyku, ve kterém je text k dispozici, struktuře textu, použité gramatice, která se například i u jednoho jazyka vzhledem k zeměpisné oblasti může lišit. To velmi znesnadňuje rychlé šíření efektivních nástrojů pro jazyky málo používané v globálním měřítku. [18]

Úloha text miningu spočívá v získávání a rozkrývání užitečných informací obsažených v textu, které ani nemusí být na první pohled zjevné. Cílem text miningu je tedy objevit dosud neznámé informace. Získávání informací probíhá velmi často ne z jednoho textu, ale ze souboru textů. Text mining sestává ze dvou částí, předzpracování a vyhledávání znalostí. [18]

Předzpracování může být složeno z mnoha různých činností především v závislosti na dostupných činnostech pro daný jazyk textu. Prvním krokem je extrakce samotného textu z dokumentu a tím očištění od dalších dat jako obrázky, značky či jiné netextové informace. Odstraněny jsou informace o fontech, velikosti, barvě a dalších atributech písma. Naopak struktura textu muže být zachována pro další analýzu, je-li to užitečné a v závislosti na vybraném nástroji. Nyní je možné text rozdělit na termíny a slova, která je možné vyjmout z dalšího zpracování vzhledem k jejich četnosti či nedůležitosti (především předložky, spojky). Tím lze získat tvar textu určený pro samotné získávání znalostí. Získávání znalostí pak probíhá již pouze na tomto výběru dle požadovaného typu zpracování. [18]

Text mining využívá mnohé z přístupů data miningu. I proto lze nalézt takové data a text miningové systémy, které vykazují určité podobnosti. Oba typy systémů například požadují proces předzpracování, algoritmy odhalující vzory v datech a prvky prezentační vrstvy, jakými jsou vizualizační nástroje. [8]

Data mining předpokládá, že data jsou již uložena ve strukturovaném formátu a předzpracování se tedy skládá jen z těchto úloh: čištění dat, normalizace dat a vytvoření značného počtu tabulek. Text miningové systémy se oproti data miningu ve fázi předzpracování zaměřují na identifikaci a extrakci typických znaků dokumentu přirozeného jazyka. Tyto operace předzpracování přeměňují nestrukturovaná data do strukturovaného tvaru, který ale není vhodný pro většinu data miningových systémů. [8]

Rozdíl mezi data miningem a text miningem je v tom, že v text miningu jsou vzory extrahovány z přirozeného jazykového textu, zatímco v případě data miningu ze strukturovaných databází. Databáze jsou navrženy pro programy k automatickému zpracování, zatímco text je psán pro lidi ke čtení. Nemáme takové programy, které dokáží

přečíst text jako lidé. Mnoho vědců si myslí, že pro naprogramování aplikací, které budou schopny číst text jako lidé, bude potřeba dokonalé simulace postupu lidského myšlení. [27]

Primární problém správy těchto dat je ten, že nemáme standardní pravidla při psaní textu, takže počítač tomu nemůže porozumět. Jazyk, a v důsledku i význam, se mění pro každý dokument i pro každý kousek textu. Jediný přístup k přesnému získání a organizování takových nestrukturovaných dat je analýza jazyka a odhalení jeho významu. K extrakci konceptů z nestrukturovaných dat existuje několik automatizovaných přístupů. Tyto přístupy lze rozdělit na dva druhy: lingvistické a nelingvistické. [23]

Některé organizace zkoušejí použít automatizované nelingvistické řešení založené na statistice a neuronových sítích. Těmito řešeními s použitím počítačové technologie lze skenovat a kategorizovat klíčové koncepty rychleji než lidé. Přesnost takových řešení je bohužel velmi nízká. Většina systémů založených na statistice zjistí počet vyextrahovaných termínů a vypočítají statistickou proximitu příslušných výrazů. Produkují ale mnoho irelevantních výsledků, včetně šumu, který je nutný odstranit. To je možné zajistit jak pomocí filtrování slov s příliš vysokou četností v textu, tak i pomocí slovníku termínů. [23]

Naopak text mining založený na lingvistickém přístupu využívá principy zpracování přirozeného jazyka (NLP) – počítačem podporovaná analýza lidského jazyka k analýze slov, frází, syntaxe a struktury textu. Systém, který v sobě začleňuje NLP, může inteligentně vyextrahovat termíny včetně frází. Znalost použitého jazyka navíc dovoluje klasifikaci výrazů do podobných skupin. [23]

Pro porozumění textu text mining založený na lingvistickém pojetí hledá význam textu rozpoznáním tvarů slov, které mají podobný význam, a analýzou struktury vět. Tento přístup nabízí velkou rychlost a efektivitu nákladů. Systémy také nabízí daleko větší přesnost při menší intervenci lidí. [23]

Reprezentace dokumentu

Klíčovým elementem text miningu je kolekce dokumentů. Tato kolekce se skládá z textů. Většina text miningových řešení je zaměřena na odhalování vzorů z těchto rozsáhlých kolekcí dokumentů. Počet dokumentů v těchto kolekcích může dosáhnout desítek i tisíců. [8]

Kolekce dokumentů mohou být statické, kdy počet dokumentů zůstává nezměněn nebo dynamické, ve kterých se počet dokumentů v průběhu času mění. Při extrémně velkých kolekcích dokumentů nebo kolekcích, kde dochází k rychlé obměně dokumentů, se mohou

vyskytovat problémy při optimalizaci výkonu komponent těchto text miningových systémů. [8]

V souvislosti se zpracováním textů vymezíme kromě dokumentu další základní pojem, a to termín. Dokument chápeme jako jakýkoliv souvislý úsek textu, který může být považován za samostatnou jednotku (kniha, článek, e-mailová zpráva apod.). Termínem je nejčastěji slovo, ale může jít také o dvojici slov nebo ustálené víceslovné spojení. Termínem může být i e-mailová adresa či nějaká skupina znaků, která se vyskytuje v textu. Dokument je v nejjednodušším případě vektor termínů. [8]

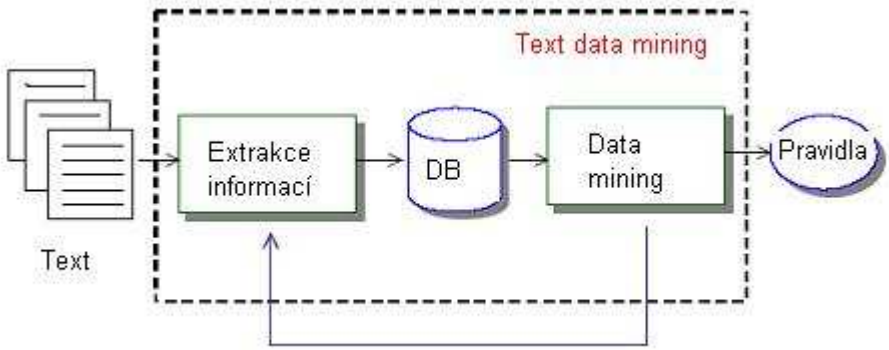
1.2.1 Extrakce informací

Information extraction (IE) neboli extrakce informací je proces automatického získávání strukturovaných dat z nestrukturovaného dokumentu přirozeného jazyka. Systémy IE spoléhají na data generovaná systémy NLP. Úlohy, které mohou být provedeny systémy IE jsou [11]:

1. analýza termínů, která identifikuje termíny v dokumentu, kde tyto termíny se mohou skládat z jednoho či více slov. Tohle je vhodné především pro dokumenty s mnoha složenými termíny, jako např. vědecké studie.
2. rozpoznání jmenných entit, které identifikují názvy v dokumentech, jako např. jména lidí či názvy organizací. Některé systémy jsou schopny rozpoznat datumy, množstevní jednotky, procenta apod.
3. extrakce faktů, která z dokumentů identifikují a extrahují fakta. Taková fakta mohou být vztahy mezi entitami a událostmi.

Extrakce informací hraje v oblasti text miningu velmi důležitou roli. Rozdíl mezi data a text miningem při extrakci informací je v tom, že data mining striktně vyžaduje strukturovaná data, ale text je přirozeně nestrukturovaný. Text lze strukturalizovat zobrazením, při němž je vypočítán prostý výskyt slov. Extrakce informací je oblastí text miningu, která se snaží dostat text mining na stejnou úroveň jako strukturovaný data mining. Pokud máme nestrukturované informace, které se běžně nachází v dokumentech, potom potřebujeme oddělený proces pro extrakci dat z nestrukturované formy. Cílem je vzít nestrukturovaný dokument a automaticky doplnit hodnoty do tabulky. Databáze, která je organizována pomocí polí a tabulek, je tedy strukturovaná. Schéma extrakce informací je znázorněno na následujícím obrázku 1. [26]

Na proces extrakce informací lze nahlížet jako na systém za sebe seřazených modulů. Modul požaduje vstup z modulu předchozího a výstup je napojen na modul další. Do prvního modulu vstupuje celá kolekce dat, výstupem jsou strukturovaná data, znalosti.



Obrázek 1: Proces extrakce informací. Zdroj: vlastní – upraveno na základě [24]

Na obrázku je vidět schéma systému extrakce informací. Na vstupu do tohoto procesu jsou dokumenty, ze kterých je potřeba extrahovat informace. Nejprve se manuální anotací menšího počtu dokumentů s informacemi, které jsou potřeba vyextrahovat, může relativně přesný systém IE tímto menším korpusem natrénovat. Poté tento proces může být aplikován na větší korpus textů a tím vytvořit databázi. Přesnost současných systémů IE je omezená, a tedy automaticky extrahovaná databáze bude nevyhnutelně obsahovat značné počty chyb. Zde vyvstává důležitá otázka, zda znalost odhalená z této „nečisté“ databáze je významně méně spolehlivá než znalost odhalená z normální (čisté) databáze. Na tuto databázi se dále aplikují standardní data miningové algoritmy. [24]

Proces extrakce informací probíhá v těchto fázích. Nejprve se vybere ta část dokumentu, která bude použita v procesu extrakce informací. Zde se uplatňují metody známé z vyhledávání informací. Následně se provede lexikální analýza. V této analýze se dokument rozdělí do vět, věty následně do tokenů. Tokenizace je tedy proces, při němž je text rozložen na základní prvky, se kterými bude dále pracováno. Prvky mohou být slova, čísla, interpunkce, spojení slov. Ve slovníku proces extrakce informací vyhledá tokeny a volitelně jim přiřadí syntaktické kategorie. Po lexikální analýze se provede stemizace, což je proces pro převod slova do základního tvaru (na kořen slova). Nyní může proběhnout rozpoznání jmen – osob, geografických lokací, společností, organizací apod.

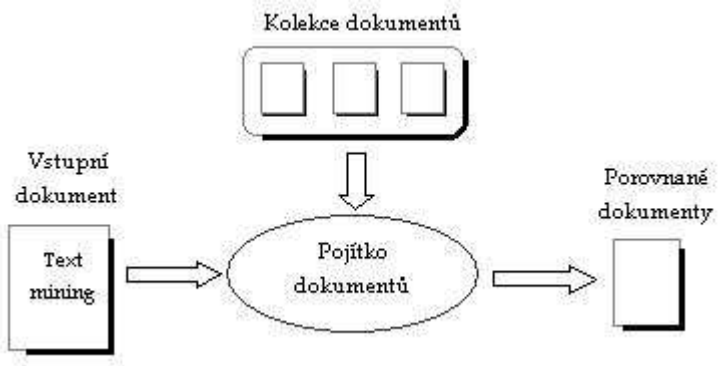
Zde je výčet čtyř základních prvků, které mohou být z textu extrahovány [26]:

- **Entity.** Entity jsou základními stavebními bloky, které lze najít v textových dokumentech. Entity zahrnují lidi, organizace apod.
- **Atributy.** Atributy jsou znaky extrahovaných entit**.** Příklady atributů jsou název osoby, věk osoby a typ organizace.
- **Fakta.** Fakta jsou relace, které existují mezi entitami. Např. zaměstnanecký vztah mezi zaměstnancem a zaměstnavatelem.
- **Události.** Událostí je aktivita nebo výskyt zájmu, na kterém entity participují. Např. narozeniny, výplata apod.

1.2.2 Získávání informací

Cílem information retrieval (IR) neboli vyhledávání informací je přesně a rychle získat dokumenty, které obsahují informace užitečné pro uživatele. Počet relevantních dokumentů je závislý na rozhodnutí uživatele. Získané dokumenty jsou seřazeny podle určitého kritéria (oblíbeným kritériem je skóre podobnosti mezi vektorem dokumentu a vektorem dotazu). [26]

IR je téma obvykle asociováno s on-line dokumenty. Obecná úloha IR je znázorněna na obrázku 2. K dokumentům, které chceme získat z kolekce dokumentů, přiřadíme určité pojítko. Dokumenty porovnány s těmito pojítky jsou odpověďmi na naše dotazy. [26]



Obrázek 2: Proces vyhledávání informací. Zdroj: vlastní – upraveno na základě [26]

Vyvstává zde otázka, co jsou zmíněná pojítka a jak jsou použita k získání relevantních dokumentů? Těmito pojítky rozumíme slova, která pomáhají určit význam uložených dat. Při vyhledávání na Internetu jsou u vyhledávacího stroje hledaná slova připojena k uloženým dokumentům. Jako odezvy jsou poskytnuty nejpřesnější spojení. Tento proces lze generalizovat na dokument, kde místo těchto slov, pojítek, je rozuměn celý dokument. Vstupní dokument je tedy připojen ke všem uloženým dokumentům a jako výsledek jsou získány nejvhodnější dokumenty. [26]

Základním konceptem vyhledávání informací je měření podobnosti. Srovnává se podobnost dvou dokumentů. I jakkoli malý počet slov na vstupu vyhledávacího stroje je považován za zmíněné pojítko, které je porovnáno s ostatními dokumenty. Měření podobnosti[1] je spojené s prediktivními metodami pro učení a klasifikaci nazvanými metody nejbližšího souseda. Měření podobnosti a různé obměny této metody jsou základem pro vyhledávání informací. [26]

1.2.3 Kategorizace textů

Kategorizace textů je úkol, při němž jsou dokumenty automaticky zařazovány do předem dané množiny předdefinovaných tříd. Ty mohou být kategorizovány podle obsahu (tématu, klíčových slov, názvů, aj.) nebo žánru, autora, apod. Dokumenty, pro něž může být použita kategorizace textů, jsou články, e-maily, zprávy, webové stránky a jiné.

Pokud jsou stanoveny předem klasifikační třídy a jednotlivé dokumenty jsou k těmto jednoznačně přiřazeny, jde o učení s učitelem. V tomto rámci jsou konstruovány modely klasifikující příští dokumenty, u nichž není známa třída. Pokud klasifikační třídy nejsou předem známy, hovoříme o učení bez učitele. [12]

Cílem kategorizace textu je tedy umístit dokumenty do příslušných tříd. Tyto třídy jsou vytvořeny na základě znalosti struktury dokumentu a očekávaných témat. Častější je ale situace, kdy je struktura dokumentu neznámá. Společnost například pomocí help desku přijímá a zaznamenává volání uživatelů. Společnost v tomto případě bude chtít znát kategorie stížností. Jsou dány dokumenty a je požadováno je zařadit do takových tříd, které sdružují podobné dokumenty. [26]

Stále více dokumentů je dostupných on-line a použitelnost této metody se stále rozšiřuje. Tyto dokumenty se vztahují převážně k e-mailu, jako např. přeposílání e-mailu příslušnému oddělení společnosti nebo detekování spamu.

Kategorizaci lze provádět těmito metodami [18]:

- metoda nejbližšího souseda,
- bayesův naivní klasifikátor,
- kosinová podobnost.

[1] Metody měření podobnosti jsou vysvětleny v kapitole 1.2.3 Kategorizace textů

Metoda nejbližšího souseda spočívá v hledání nejpodobnějších dokumentů vůči dokumentu testovanému. Měření probíhá pomocí výpočtu Euklidovské vzdálenosti [18]:

$$d_e(x_j, x_k) = \sqrt{\sum_{i=1}^{n} (x_{ij} - x_{ik})^2}\,, \tag{1}$$

kde:

- x_{ij} je hodnota znaku i pro objekt j,
- x_{ik} je hodnota znaku i pro objekt k,
- n je celkový počet znaků.

Při této metodě je nutné jednak stanovit počet nejpodobnějších dokumentů, kdy tato hodnota významně ovlivňuje kvalitu zpracování dokumentů, dále pak určit způsob ohodnocení blízkosti dokumentů.

Bayesův naivní klasifikátor je založen na pravděpodobnostním modelu a i přes své nedostatky dosahuje poměrně dobrých výsledků. Mezi jeho přednosti patří schopnost zařazení i neúplně popsaných případů, rychlost zpracování díky relativní nenáročnosti výpočtu. Naopak jeho použití nedosahuje optima při málo početných třídách, reprezentace znalostí pomocí pravděpodobností je méně srozumitelná

U textových dokumentů je jednou z často používaných funkcí kosinová podobnost. Kosinová podobnost je definována podle vztahu [12]:

$$d_{\cos}(D_j, D_k) = \frac{\sum_{i=1}^{n} d_{ij} d_{ik}}{\sqrt{\sum_{i=1}^{n} d_{ij}^2} \sqrt{\sum_{i=1}^{n} d_{ik}^2}}\,, \tag{2}$$

kde:

- D_j, D_k jsou normalizované vektory
- d_{ij} je váha i-tého termínu v j-tém vektoru,
- d_{ik} je váha i-tého termínu v k-tém vektoru,
- n je celkový počet termínů.

D_j je první dokument a D_k je druhý dokument. Termíny v kolekci jsou očíslovány 1 až n.

Jedná se o skalární součin normalizovaných vektorů D_i a D_j, jenž je ekvivalentní kosinu úhlu svíraného mezi oběma vektory. Totožné vektory a vektory stejného směru mají maximální podobnost, minimální podobnost je mezi vektory na sebe kolmými. Zkušenosti potvrzují, že kosinová podobnost je efektivnější než např. euklidovská podobnost. [12]

1.2.4 Zpracování přirozeného jazyka

Natural language processing (NLP) neboli zpracování přirozeného jazyka je hlavní oblastí fáze předzpracování dokumentu. Techniky NLP zahrnují statistické přístupy a strojové učení, rozšířené o přístupy umělé inteligence a informační teorie. [22]

Výzkum zpracování přirozeného jazyka se snaží zjistit, jak dále přeměňovat text zadaný do počítače v přirozeném jazyce do podoby vhodnější k dalšímu zpracování. Je spjatý s oblastí porozumění přirozeného jazyka, která zahrnuje problematiku strojového překladu, odpovídání na otázky v přirozeném jazyce, rozpoznávání řeči atd. [1]

V tomto ohledu je nutné zmínit, že přirozený jazyk lze zpracovávat umělou inteligencí na různých úrovních, a to na úrovni fonetické, morfologické, lexikální, syntaktické, sémantické a pragmatické, přičemž největší problémy představují poslední dvě úrovně. Sémantická se snaží o přiřazení významu určitým součástem věty, přičemž používá tezaurus, nicméně nemůže fungovat bez analýzy na pragmatické úrovni, která usiluje o posouzení významu slov a slovních spojení podle okolních slov a vět, kontextu. Systémy, které by toto dokázaly, ještě nebyly vyvinuty, protože by si musely pod určitými slovy, binárními kódy, představovat určité věci a vztahy, což není dost dobře možné. Počítač umí maximálně zobrazit text zadaný v binárním kódu, ale neumí jej zpracovat po významové stránce. [1]

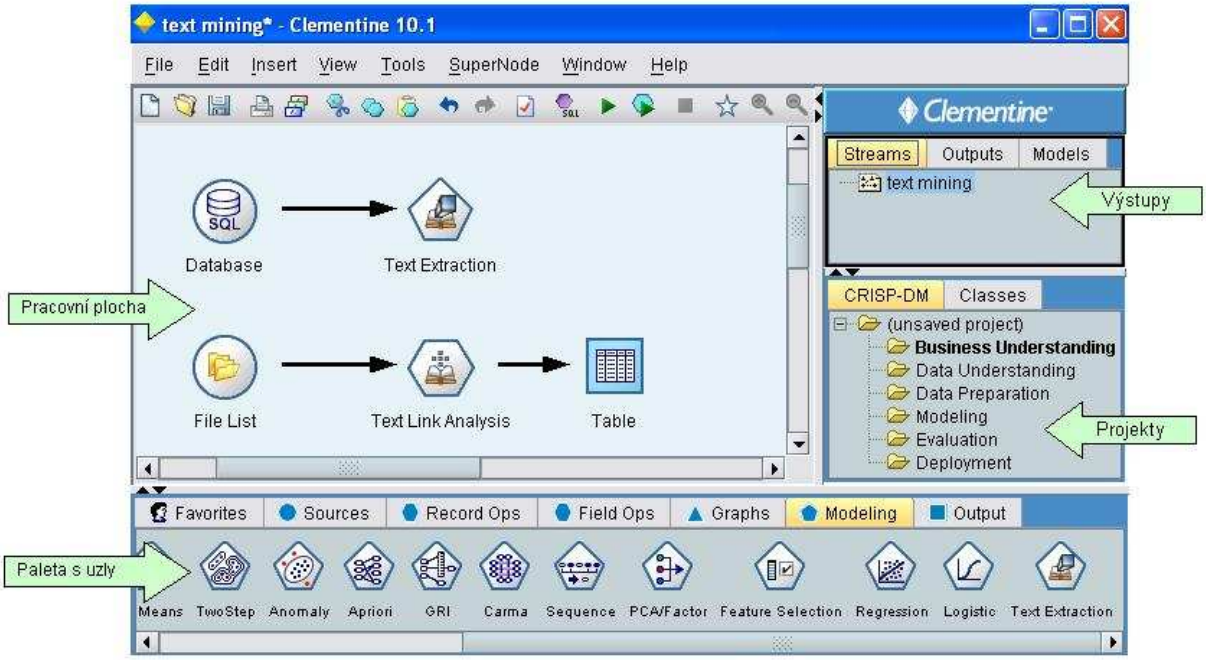
V praxi se nejčastěji setkáme s nestrukturovaným textem. To je text s celými gramatickými větami, které jsou zpracovávány právě metodami NLP. Vzory jsou získávány na základě jejich syntaktické a sémantické analýzy. [1]

Pojítko mezi text miningem a NLP je v tom, že dobývané informace z textu nutně vyžadují zobrazení alespoň některých lingvistických struktur textu. Tyto struktury mohou být buď místní jako výskyt vlastních jmen a odkazů na jiné dokumenty, nebo globální jako rozdělení dokumentu do témat a segmentů. Při objevení takových struktur, které jsou součástí procesu dolování, je má často smysl označit v původních dokumentech, např. hypertextovými odkazy. Tyto struktury potom mohou být vztaženy k záznamům jiných informačních zdrojů (např. použity k načtení nebo určení času záznamů). [10]

2 Clementine

Clementine je data miningový nástroj založený na vizuálním programování, jenž zahrnuje technologie strojového učení. Clementine byl vyvinut firmou Integral Solutions Ltd. (ISL)[2]. Poskytuje integrované prostředí data miningu pro koncové uživatele i vývojáře. Obsahuje četné data miningové algoritmy, které jsou začleněny do systému. Charakteristickým znakem systému Clementine je, že tento systém je objektově orientován a obsahuje rozhraní uzlů, kdy proces odhalování znalostí je v systému Clementine tvořen konstrukcí tzv. streamů neboli datových proudů, které se skládají z jednotlivých uzlů.

Prostředí Clementine se skládá ze čtyř oblastí (obrázek 3). Hlavní oblast, označená **Pracovní plocha**, je oblastí tvorby streamů. Další oblastí je **Paleta s uzly**, ve které jsou uzly seskupené podle jejich funkcí: zdroje dat, manipulace s daty (řádky nebo sloupci), vizualizace, modelovací techniky (data miningové algoritmy) a poslední skupinou jsou uzly pro výstup výsledků. Třetí oblast **Výstupy** obsahuje výstupy modelů vytvořených pomocí modelových streamů. Tyto modely mohou být opětovně použity s ostatními uzly nebo uloženy pro pozdější použití. Čtvrtá oblast **Projekty** definuje proces CRISP-DM a jeho jednotlivé fáze.



Obrázek 3: Pracovní prostředí systému Clementine. [Zdroj: vlastní]

V dolním panelu jsou uzly seřazené podle funkčnosti. Kruhové uzly představují odkazy na datové zdroje a zakládají první uzel streamu. Šestiboké uzly manipulují s daty – operace na

[2] Později odkoupeno firmou SPSS

úrovni záznamů (např. výběry, agregace, sjednocení) nebo na úrovni položek (např. přetypování). Trojúhelníkové uzly poskytují vizualizaci. Jedná se o grafické výstupy dat (např. histogram, síťový graf). Tyto uzly slouží ke snadnějšímu pochopení dat. Pětiúhelníkové uzly jsou modelovací uzly, které při použití data miningových algoritmů slouží k identifikaci vzorů v datech. Tvoří jádro celého procesu. Poslední skupinou jsou výstupní čtvercové uzly, které jsou prostředkem pro získání informací z dat a modelů.

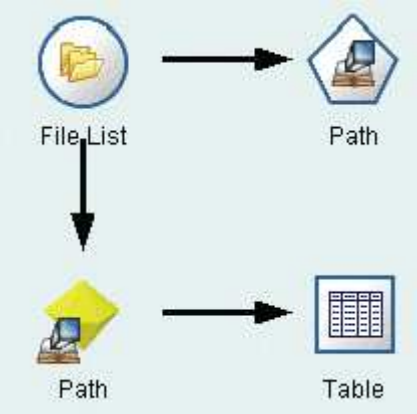
K tvorbě modelu analytik jednoduše vybere a spojí jednotlivé uzly z palety Records Operators, Fields Options, Modeling, Graph a Output. Tímto spojováním jsou vytvořeny popsané streamy.

Pro účely text miningu jsou určeny uzly Text Extraction a Text Link Analysis. Pro načtení textů ke zpracování těmito uzly je potřeba ke streamu připojit zdrojový uzel File List.

2.1 Uzel File List

Data pro text mining mohou být uložena v jakémkoliv standardním formátu podporovaném systémem Clementine. Mezi tyto zdroje patří databáze, které představují data v řádcích a sloupcích, nebo pro Clementine nestrukturované formáty dokumentů jako Microsoft Word, PDF a HTML.

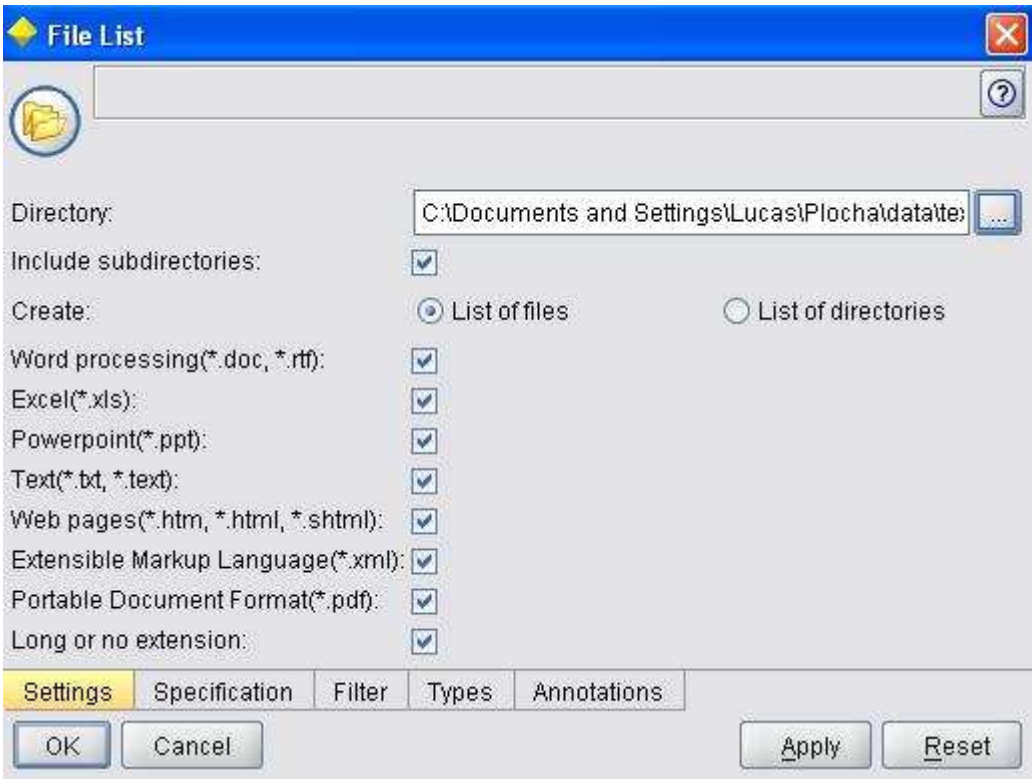
Pro načtení databáze lze použít jakýkoliv zdrojový uzel. K načtení textu z nestrukturovaných dokumentů se musí použít uzel File List. Tento uzel slouží k vygenerování seznamu dokumentů nebo složek, které vstupují do procesu text miningu. Zdrojový uzel File List si načte dokumenty, případně všechny adresáře v podsložkách a na výstupu je zobrazí jako seznam. Tohle je podstatné, protože nestrukturované dokumenty nemohou být reprezentovány stejným způsobem jako data v databázi (řádky a sloupci). Výstupem pro každý dokument nebo složku je pole s jedním záznamem, které může být vybráno jako vstup pro následné uzly Text Extraction a Text Link Analysis. Použití tohoto uzlu je ukázáno na obrázku 4, kdy po vykonání uzlu Text Extraction vznikne model tohoto uzlu. Pro zobrazení výsledků se k tomuto nově vygenerovanému modelu musí připojit tabulka.



Obrázek 4: Použití uzlu File List. [Zdroj: vlastní]

Vlastnosti uzlu File List

Po poklepání na uzel File List se zobrazí záložka *Settings*, kde se nastavují parametry tohoto uzlu, což je znázorněno na následujícím obrázku (obrázek 5).



Obrázek 5: Nastavení uzlu File List – záložka Settings. [Zdroj: vlastní]

Dále je uveden seznam parametrů [23]:

Directory. Specifikace kořenové složky s dokumenty, která bude použita.

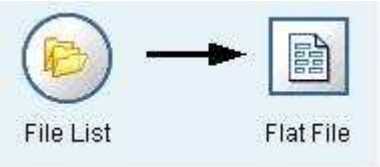
Seznam přípon. Zde jsou vybrány typy souborů a přípony, které budou ve složce použity ke zpracování. Přípony souborů, které nebudou vybrány, extraktor během zpracování přeskočí. Lze vyfiltrovat následující přípony:

Tabulka 2: Seznam přípon dokumentů. Zdroj: [23]

Přípona	Dokument
doc, rtf	Textový dokument
txt, text	Prostý textový soubor
htm, html, shtml	Webovská stránka
xml	Strukturovaný dokument xml
xls	Microsoft Excel
pdf	Dokument formátu PDF
ppt	Prezentace Microsoft Powerpoint

Výsledné pole uzlu File List lze vybrat jako vstup pro další uzly Text Extraction a Text Link Analysis. U obou z těchto uzlů se zaškrtnutím rámečku *Text field represents pathnames to documents* označí, že vybrané pole obsahuje cestu k dokumentům, kde je umístěn text.

Uzel File List patří mezi tzv. uzly CEMI, které nemohou být použity ve streamech, jenž jsou rozmístěny Solution Publisherem nebo jinou predikční aplikací. Solution Publisher slouží pro vyhodnocování nestrukturovaných dat v reálném čase V těchto případech musí být při použití uzlu File List připojen uzel Flat File, který vytvoří vstupní soubor např. pro analýzy uzlu Text Extraction. Rozmístění uzlu Flat File je znázorněno na obrázku 6.



Obrázek 6: Rozmístění uzlu File List. [Zdroj: vlastní]

V tomto případě se postupuje následovně: Vytvoří se stream, který k načtení seznamu dokumentů, ze kterých bude „dolováno“, používá zdrojový uzel File List. K tomuto uzlu se připojí uzel Flat File. V dialogu uzlu Flat File na záložce *Export* se zaškrtnutím *Generate an import node for this data* vytvoří zdrojový uzel, kterým se importují data. Poté se nově vygenerovaný zdrojový uzel připojí k uzlům Text Extraction nebo Text Link Analysis.

2.2 Uzel Text Extraction

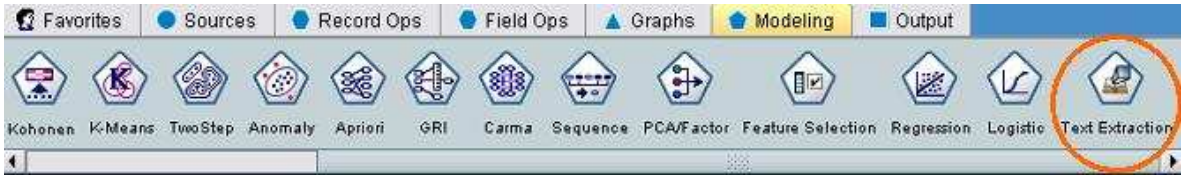
Použitím lingvistických zdrojů jsou extrahované termíny a slova podobného významu seskupena do jednoho termínu nazvaného konceptem. Modelovacím uzlem Text Extraction se vytváří model extrakce textu, který odhaluje a extrahuje významné koncepty ze strukturovaných i nestrukturovaných textů. Clementine tímto uzlem, pomocí vnitřního extraktoru a použitím metod NLP, extrahuje a organizuje koncepty do databáze konceptů. [23]

Tyto koncepty, v souladu s jejich frekvencemi a typy klasifikace, jsou kombinovány s existujícími strukturovanými daty a následně použity pro modelování. Modelování pomocí data miningových nástrojů systému Clementine slouží k získání lepších rozhodnutí. Použitím výsledků extrakce textu může být zlepšena přesnost modelů predikce.

Tento uzel se typicky používá v procesu, ve kterém jsou koncepty opakovaně extrahovány, zkoumány a očištěny. Lze použít zdrojové lingvistické soubory k očištění pravidel a slovníků, které jsou použity během extrakce, a které ovlivňují obsah a strukturu celkové koncepce. Pokud se provedou se zdroji jakékoliv změny, stream pro zobrazení aktuálních výsledků musí být opětovně spuštěn.

Je zde také možnost automatického překladu z jazyků jiných než angličtina: arabština, čínština nebo perština. Tato funkce umožňuje „dolování“ z textů, jejichž jazyku nerozumíme. Tento překlad je však možný pouze do angličtiny, protože Clementine obsahuje anglický slovník. K funkčnosti tohoto překladu potřebuje být v systému nainstalován Language Weaver Translation Server.

Modelovací uzel Text Extraction přijímá data z jakéhokoliv standardního zdrojového uzlu systému Clementine (uzel Database, uzel Flat File apod.). Cesta k externím dokumentům se vytvoří pomocí uzlu File List. Uzel Text Extraction je nainstalován se systémem Clementine a je k dispozici na paletě Modeling, jak je zobrazeno na obrázku 7.

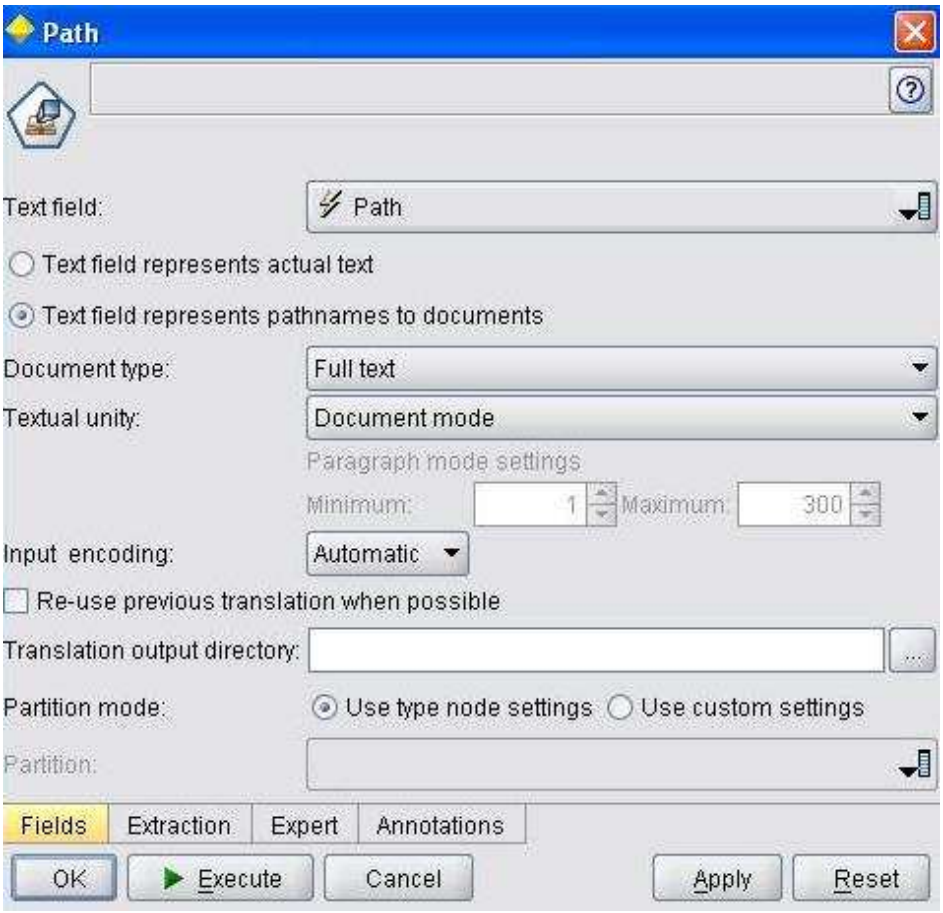


Obrázek 7: Modelovací paleta obsahující uzel Text Extraction. [Zdroj: vlastní]

Extrahované koncepty mohou být kombinovány s existujícími strukturovanými daty a použity k modelování. Uzel Text Extraction je plně integrován se systémem Clementine, takže můžeme text miningové streamy rozmístit přes Solution Publisher, který slouží pro vyhodnocování nestrukturovaných dat v reálném čase. Schopnost rozmístění těchto streamů zajišťuje uzavřená smyčka text miningových implementací. Organizace například nyní mohou analyzovat příchozí a odchozí hovory použitím predikčních modelů ke zvýšení přesnosti marketingových zpráv v reálném čase. Ze všech text miningových uzlů je tento uzel extrakce konceptů nejčastější. V případě extrakce vzorů je použit uzel Text Link Analysis.

2.2.1 Nastavení parametrů uzlu Text Extraction

K nastavení parametrů se uživatel dostane kliknutím na uzel Text Extraction. Je zde na výběr ze tří záložek. Základní parametry se nastavují na záložce *Fields* (obrázek 8).



Obrázek 8: Dialogové okno uzlu Text Extraction -Záložka Fields. [Zdroj: vlastní]

Záložka *Fields* se používá k nastavení položek pro data, ze kterých se extrahují koncepty. Na této záložce lze nastavit tyto hlavní parametry:

Text field: Výběr cesty k dokumentu obsahující text, který bude extrahován. Tato položka je závislá na zdrojovém uzlu. Jestliže je zadáno *Direction=None* nebo *Type=Typeless*, může se specifikovat jakýkoliv řetězec. Zde zmíněný název položky je výchozí název tohoto atributu. Do doby, než je uzel připojen ke streamu, je název tohoto streamu *No input*. Na výběr je zde z těchto dvou možností [23]:

1. **text field represents actual text**, kde takto vybraná volba obsahuje text, ze kterého by měly být koncepty extrahovány, nebo

2. **text field represents pathnames to documents**, kde vybraná volba obsahuje jednu nebo více cest k dokumentům, které musí být během extrakce načteny.

Document type. Tento parametr je dostupný pouze tehdy, pokud textové pole představuje cestu k dokumentu. Ke specifikaci struktury textu je možnost výběru jedné z nabízejících možností: Full text, Structured text nebo XML text.

2.2.2 Použití modelovacího uzlu Text Extraction

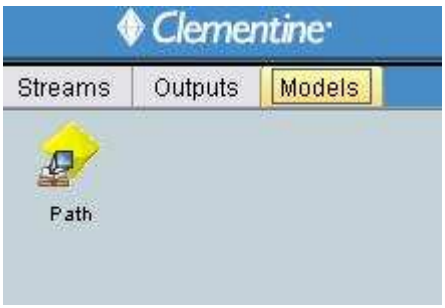
Modelovací uzel Text Extraction je používán k extrakci konceptů. Ke zpřístupnění dat lze použít jakýkoliv z těchto zdrojových uzlů: Database, Variable File nebo uzel Fixed File. Pro text nestrukturovaného formátu, který je umístěn v externích dokumentech se musí použít uzel File List. K tomuto vstupnímu uzlu se připojí modelovací uzel Text Extraction (obrázek 9).

Obrázek 9: Uzel File List s modelovacím uzlem Text Extraction. [Zdroj: vlastní]

2.2.3 Vygenerovaný model Text Extraction

Po úspěšném vykonání uzlu Text Extraction je vygenerován model Text Extraction. Model obsahuje seznam konceptů, které jsou použity pro odhalení klíčových konceptů ostatních textů.

Vygenerované modely jsou umístěny v paletě modelů (obrázek 10). Informace o vygenerovaných modelech se získají pravým kliknutím na uzel v paletě modelů a následným výběrem Browse z kontextové nabídky.



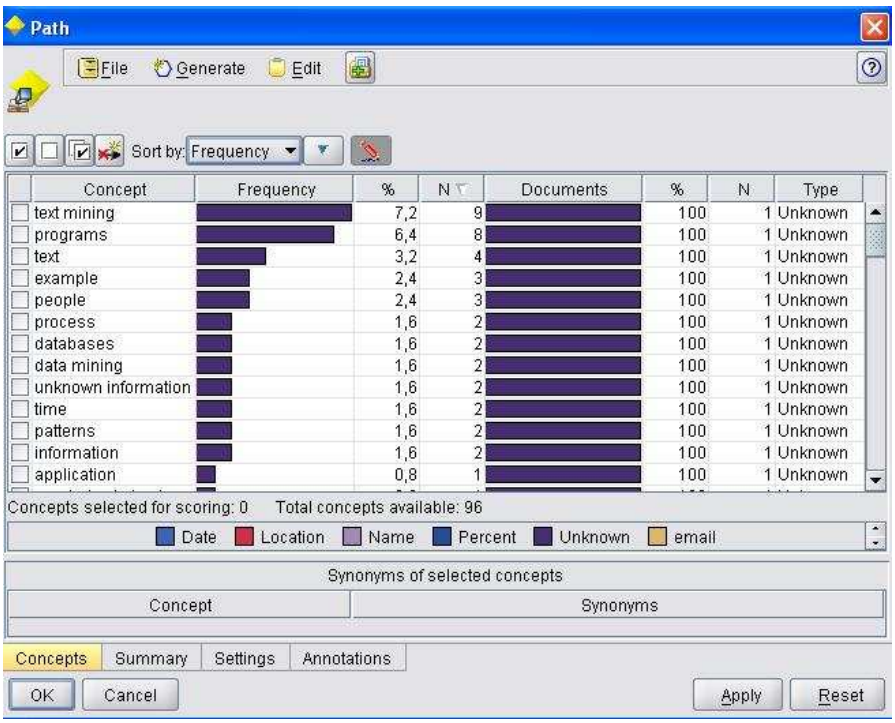
Obrázek 10: Paleta modelů. [Zdroj: vlastní]

Tento model je přidán ke streamu kliknutím na ikonu v paletě modelů a následným přetažením a připojením ke streamu. Po připojení modelu ke streamu jsou data připravena

k vytváření predikcí. Při vykonání streamu obsahující modelovací uzel Text Extraction jsou k datům přidána nová pole. Počet a struktura polí záleží na režimu vyhodnocování, které je nastaveno na záložce *Settings* uzlu File List.

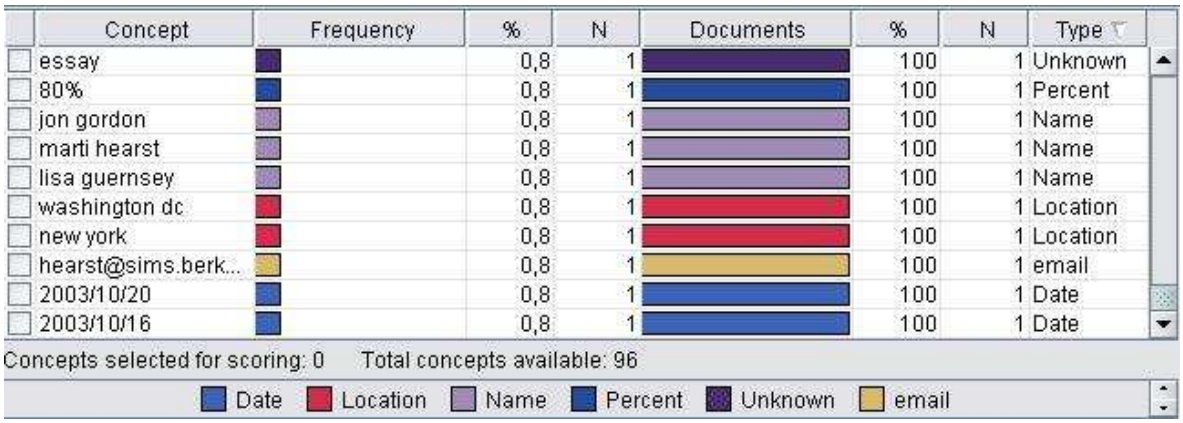
Data ve vytvořeném modelu musí obsahovat stejná vstupní pole a typy polí jako trénovací data, která jsou použita při vytvoření tohoto modelu. Když některé z polí chybí nebo typy těchto polí nesouhlasí, při vykonání streamu se objeví chybové hlášení.

Po úspěšném vykonání streamu se zobrazí tabulka konceptů. Záložka *Concepts* zobrazuje sadu konceptů, které byly vyextrahovány (obrázek 11). Koncepty jsou zobrazeny ve formě tabulky, ve které každý řádek odpovídá jednotlivým konceptům. Cílem tabulky je vybrat koncepty, které budou použity pro skórování. Skórováním se rozumí realizace procedur, které periodicky na základě zjištěných modelů generují skóre typů konceptů. Zaškrtnuté pole znamená, že koncept bude vybrán pro vyhodnocení.



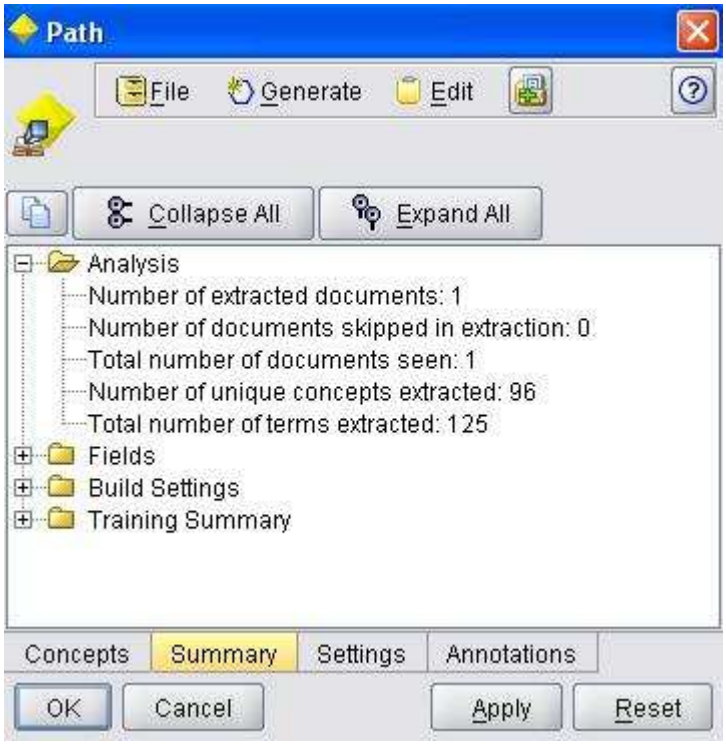
Obrázek 11: Vygenerovaný model uzlu text Extraction - záložka Concepts. [Zdroj: vlastní]

Výslednou tabulku lze seřadit podle konceptů, frekvence nebo typů. Z dokumentu byly vyextrahovány koncepty typů Date, Location, Name, Percent nebo email. Pro nezařazené typy je zvoleno Unknown. Model Text Extraction vyextrahoval celkem 96 konceptů. Nejčastějším konceptem je termín *text mining* s devíti výskyty. Jednotlivé typy konceptů jsou vidět na obrázku 12.



Obrázek 12: Extrahované typy konceptů. [Zdroj: vlastní]

Záložka Summary (obrázek 13) prezentuje informace o modelu (složka Analysis), polích použitých v modelu (složka Fields), nastavení použité při tvorbě modelu (složka Build Settings) a trénování modelu (složka Training Summary).



Obrázek 13: Vygenerovaný model uzlu text Extraction - záložka Summary. [Zdroj: vlastní]

Při prvním prohlédnutí modelovacího uzlu jsou složky na záložce *Summary* sbalené. K rozbalení výsledků, která nás zajímají, musíme použít ovládací prvek se znaménkem plus na levé straně složky. Složka Analysis obsahuje statistiky extrahovaných dokumentů a termínů.

Výsledný soubor procesu extrakce informací pomozí uzlu Text Extraction je k dispozici na přiloženém CD pod názvem **Text_extraction.str**.

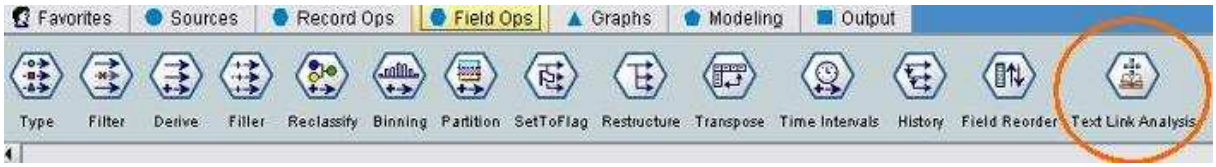
2.3 Uzel Text Link Analysis

Uzel Text Link Analysis extrahuje koncepty podobně jako uzel Text Extraction. Tento uzel ale navíc dokáže identifikovat vztahy mezi těmito koncepty, které jsou založené na známých vzorech. Extrakcí vzorů tedy mohou být odhaleny vztahy mezi těmito koncepty. [23]

Například extrahovaný název produktu nemusí být dost zajímavý. Pokud ale v těchto datech existují určité náznaky, tak použitím tohoto uzlu může být zjištěno, co si lidé o produktu myslí. Vztahy a asociace jsou identifikovány a extrahovány spojením známých vzorů z textu.

Uzel Text Link Analysis na vstupu přijímá, stejně jako uzel Text Extraction, standardní zdrojové uzly systému Clementine, mezi které patří uzly Database, Flat File. Uzel File List slouží pro načtení externích dokumentů.

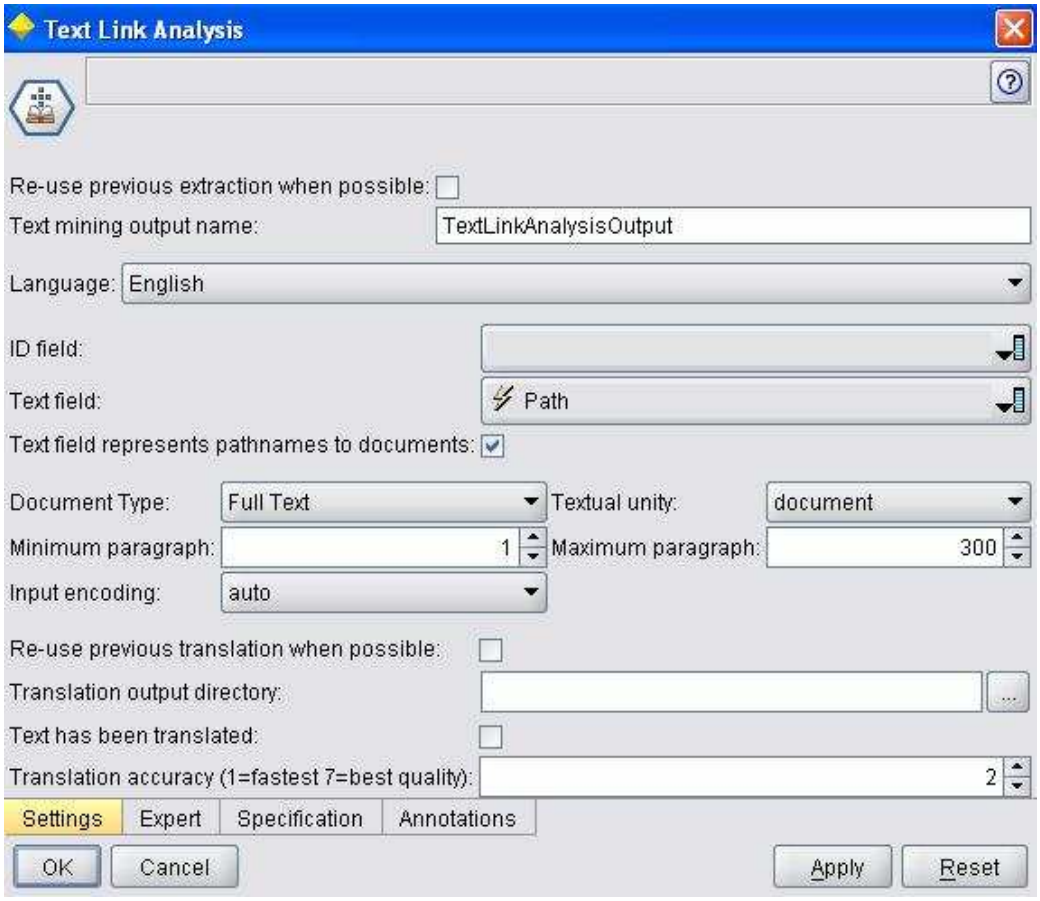
Uzel Text Link Analysis neposkytuje pouhou extrakci konceptů, ale navíc oproti uzlu Text Extraction poskytuje informace o vztazích mezi těmito koncepty. Tento uzel je tedy silnější alternativou k uzlu Text Extraction. Tento uzel se vyskytuje na paletě Fields Options (obrázek 14).



Obrázek 14: Paleta Fields Options obsahující uzel Text Link Analysis. [Zdroj: vlastní]

2.3.1 Nastavení uzlu Text Link Analysis

Základní nastavení uzlu Text Link Analysis se provede na záložce *Settings* tohoto uzlu (obrázek 15). Na této záložce lze nastavit vlastnosti dat načtených do systému Clementine i proces výstupu extrakce.



Obrázek 15: Dialogové okno uzle Text Linked Analysis - záložka Settings. [Zdroj: vlastní]

Na této záložce lze nastavit tyto parametry [23]:

Text mining output name. Tímto parametrem se určí název výsledku extrakce. Název by neměl obsahovat mezery a musí respektovat konvence názvů souborů daného operačního systému. Název by neměl začínat výrazy #,%,&,* nebo ^.

Language. Tímto parametrem se určí jazyk textu, který bude využíván při „dolování“. Ve výchozím nastavení je stanovena angličtina. Ze seznamu lze dále vybrat z těchto možností:

- **ALL**. Touto volbou se určí jazyk každého dokumentu, který bude použit při automatickém rozpoznání tohoto jazyka. Tato možnost skenuje data a během procesu extrakce automaticky rozpozná nejvhodnější vnitřní slovník pro dokumenty. Volba ALL je řízena parametry v souboru *LangIdentifier.ini*.
- Mezi další možnosti v nabídce patří arabština, čínština a perština.

Pro oddělení překladu od procesu extrakce lze použít uzel Translate. Při použití tohoto uzlu by se jako jazyk měla nastavit angličtina a určit text, který bude překládán.

ID field. Tento parametr určí oblast, která obsahuje identifikátor pro textové záznamy. Identifikátor musí být typu Integer. Parametr ID field slouží jako index pro jednotlivé textové záznamy. Tento parametr je použit, pokud textové pole představuje text, který bude „dolován“. Tento parametr ale nelze použít v případě, pokud textové pole představuje cestu k dokumentům.

Document type. Tento parametr je dostupný pouze tehdy, pokud se určí cesta k dokumentu. Strukturu textu specifikuje jedna z následujících možností: Full text, Structured text, XML text nebo XML Database text.

Záložka *Expert* obsahuje parametry, které ovlivňují extrakci a zpracování textu. Parametry v tomto dialogovém okně upravují základní i několik pokročilých režimů procesu extrakce informací. Nastavitelné parametry existují jako zdrojové soubory, které jsou přímo přístupné extraktorem vestavěným v tomto uzlu, a které řídí zpracování textu.

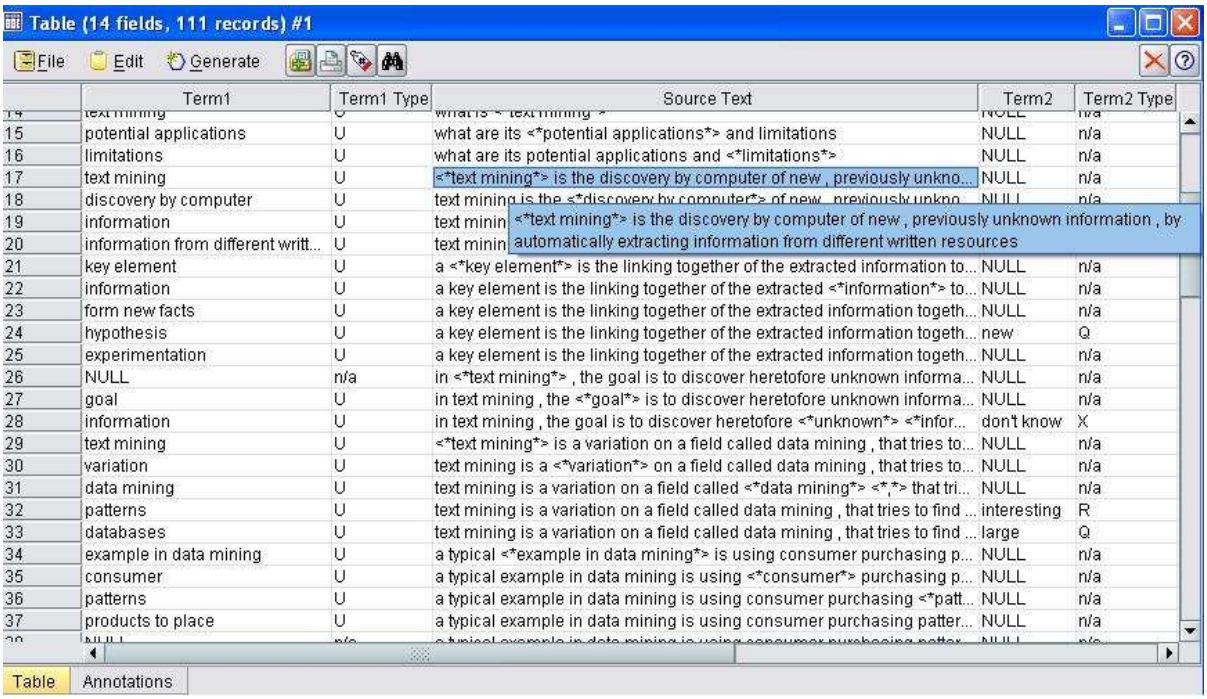
2.3.2 Použití uzlu Text Link Analysis

Uzel Text Link Analysis je používán k přístupu k datům a extrakci konceptů. K přístupu k datům lze použít jakýkoliv standardní zdrojový uzel. Typicky uzel Database, Variable File, Fixed File nebo uzel File List.

Následující obrázek 16 ukazuje použití zdrojového uzlu File List s modelovacím uzlem Text Link Analysis.

Obrázek 16: Uzel File List s modelovacím uzlem Text Link Analysis. [Zdroj: vlastní]

Uživatel nejprve přidá uzel File List. Tím určí, kde jsou požadované dokumenty umístěny. Poté uživatel k extrakci konceptů pro modelování streamu přidá uzel Text Link Analysis. Na záložce *Settings* uzlu Text Link Analysis uživatel zaškrtne možnost *text field represents pathnames to documents.* Tím se určí, že zadaná cesta představuje cestu k dokumentu a zaktivuje se část parametrů potřebných k nastavení tohoto dokumentu. Nakonec se k prohlížení konceptů, které byly extrahovány z textového dokumentu, připojí uzel Table. Výsledná tabulka je vidět na obrázku 17.



Obrázek 17: Výsledná tabulka. [Zdroj: vlastní]

V tabulce se vyextrahovalo 111 záznamů. Každý záznam má následující pole: zdrojový text, ze kterého byl koncept extrahován, jednotlivé termíny, typy těchto termínů. Pro nalezené vzory označené Term1 se nalezl vztah k dalšímu termínu, který je označen Term2. Pokud by jsme to uvedli na konkrétním příkladě, tak záznamu 32 *patterns* byl určen termín Term 2 *interesting*, tedy dohromady *pattern interesting* (zajímavé vzory). K tomuto novému termínu byl přidělen typ vztahu. Termíny, kterým nebyl přiřazen žádný další vztah jsou označeny nulovou hodnotou NULL, a tedy typ vztahů je též nulový n/a. Zkratky typů vztahů pro proces uzlu Text Link Analysis jsou X, Q a R. Zkratky tedy nyní neoznačují typy termínů, ale typy vztahů. Tyto typy vztahů byly nalezeny ve slovnících, jejichž zkratky jsou: Q pro slovník kontextového značení (*ContextualQualifier_dictionary*), R pro slovník přesných vztahů (*Positive_dictionary*) a X pro slovník neurčených vztahů (*Uncertain_dictionary*).

Výsledný soubor extrakce informací pomocí uzlu Text Link Analysis je k dispozici na přiloženém CD pod názvem **Text_link_analysis.str**.

2.4 Shrnutí kapitoly

Clementine je komerční software určený pro data mining. Uzly pro účely text miningu jsou v systému integrované, jen je potřeba je přes rozhraní CEMI doinstalovat. Po doinstalování tohoto doplňku v systému přibudou uzly Text Extraction a Text Link Analysis. Uzel Text Extraction se vyskytuje na paletě uzlů Modeling. Jde o modelovací uzel, který z textového dokumentu extrahuje koncepty. Konceptem se obvykle rozumí termín. K extrahovaným

konceptům se určí i jejich typ – názvy, produkty, datumy apod. Typy konceptů jako jména, místa apod. tento nástroj pozná podle vnitřních slovníků. Typ e-mail pozná podle znaku zavináče @, typ procenta podle znaku pro procento % a datum podle zástupného znaku rrrr/mm/dd. Druhým uzlem je Text Link Analysis. Jde o uzel na paletě Fields Options. Jde o podobný uzel uzlu Text Extraction, ale kromě extrakce konceptů navíc extrahuje i významné vztahy mezi těmito koncepty.

Jako zdrojový uzel se určí databáze nebo uzel File List. Uzel File List umožňuje dolovat koncepty ze sedmi textových formátů. Výsledná analýza je ale pro každý formát totožný. Všechny formáty dokumentů jsou totiž při analýze převedeny do jednotného modelu. Analyzuje se totiž samotný text, a nikoliv struktura dokumentu jako je to u některých jiných text miningových nástrojů.

Tento nástroj extrahuje termíny celkem z 10 jazyků. Jednotlivé typy termínů jsou pro každý z podporovaných jazyků uloženy ve formě slovníků. Slovníky typů termínů pro české texty chybí.

Nejvýznamnější charakteristika systému Clementine je shrnuta v následující tabulce 3.

Tabulka 3: Charakteristika systému Clementine. [Zdroj: vlastní]

Charakteristika	Popis
Možnosti text miningu	• Extrakce informací
Funkce extrakce informací	• Text Extraction • Text Link Analysis
Výhody	• Snadná konfigurace a příjemné uživatelské rozhraní • Více jazyků a formátů k extrakci
Nevýhody	• Drahý • Málo funkcí extrakce informací • Nepodporuje české texty

3 RapidMiner

RapidMiner je data miningový nástroj poskytovaný zdarma pod licencí GNU[3]. Tento nástroj byl získán ze stránek produktu RapidMiner [14]. Jsou dvě verze instalace tohoto produktu. Prvním je instalace software pro operační systém Windows s grafickým prostředím, kdy je potřeba z domovské stránky produktu stáhnout a následně nainstalovat instalátor rapidminer-XXX-install.exe, přičemž XXX značí verzi produktu. Druhou možností je instalace knihoven Java, které jsou dostupné přes příkazový řádek. RapidMiner je kompletně napsán v jazyce Java, což dává možnost spuštění knihoven Java téměř na jakémkoli operačním systému. Tato instalace ovšem vyžaduje mít v systému nainstalován Java Runtime Environment (JRE) verze 5 nebo vyšší. [14]

RapidMiner je prostředí pro procesy data miningu a strojového učení. Modulární koncept operátoru umožňuje návrh zřetězených operátorů pro obrovské množství problémů. Manipulace s daty je transparentní vzhledem k operátorům. Nemusí se tedy vyrovnávat s danými datovými formáty nebo odlišnými datovými pohledy – jádro systému RapidMiner dbá na potřebné transformace. V dnešní době je RapidMiner celosvětově nejrozšířenějším produktem open source pro data miningová řešení a je vědeckými pracovníky hojně využíván. [15]

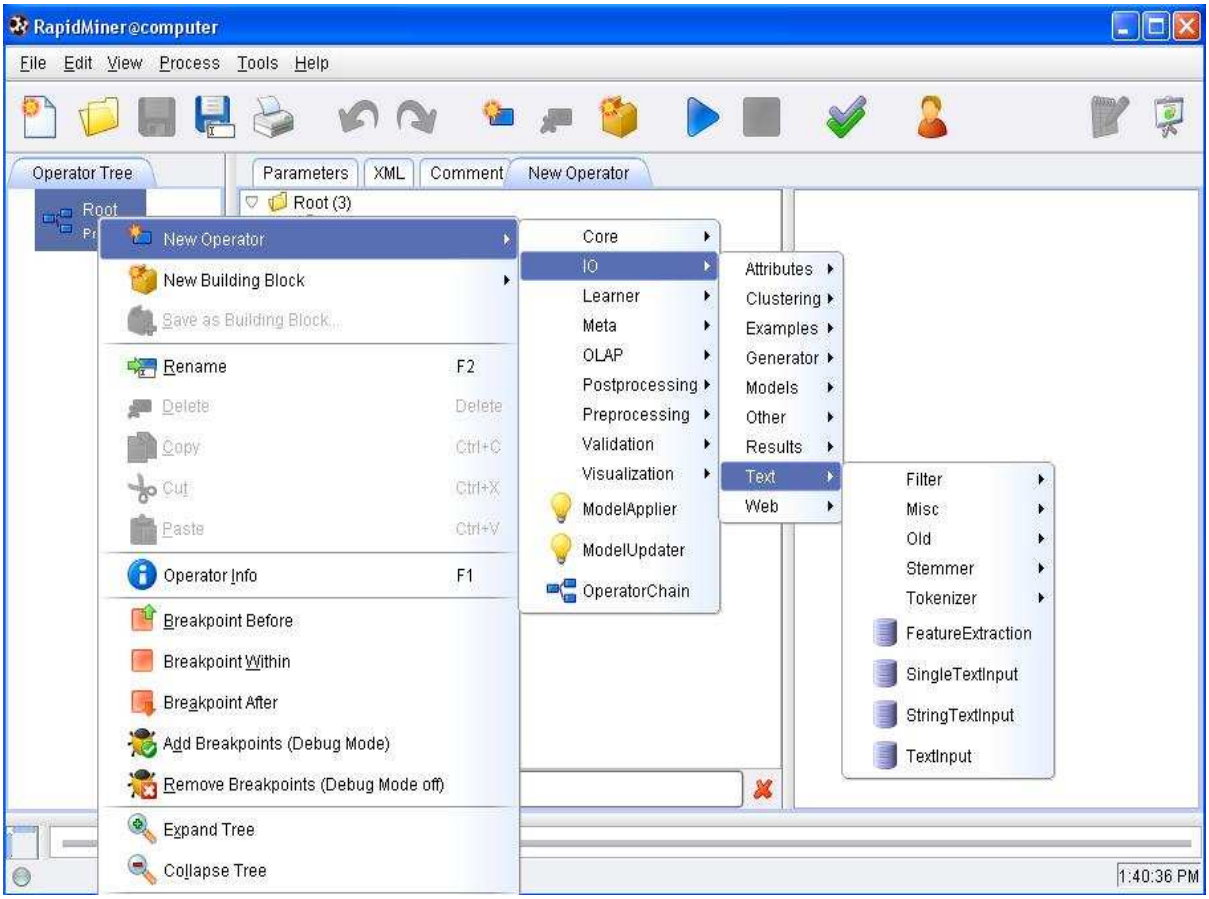
Reálné procesy objevování znalostí se typicky skládají z komplexu kroků předzpracování dat, strojového učení, evaluace a vizualizace. Proto by data miningová platforma měla umožnit zřetězení operátorů, poskytovat transparentní zpracování dat, snadnou manipulaci s parametry atributů, optimalizaci, flexibilitu a rozšiřitelnost. [15]

RapidMiner představuje novou koncepci transparentního zpracování dat a procesu modelování, které uživatelům usnadňují konfiguraci. Dostatečně příjemné grafické uživatelské prostředí (GUI) a skupina skriptovacích jazyků založených na XML vnáší RapidMiner do integrovaného vývojářského prostředí pro data mining. Kromě toho XML konfigurační soubory definují standardizovaný formát pro data miningové procesy. [15]

Na obrázku 18 je grafické uživatelské prostředí programu RapidMiner. Ve výchozím nastavení je v panelu stromu operátorů pouze jeho kořen. Kliknutím pravého tlačítka myši na tento kořen se rozvine kontextové menu s nabídkami operací. V nabídce **New Operator** se k tomuto kořeni stromu operátorů mohou připojit nové operátory. RapidMiner nabízí se

[3] GNU (General Public License) je licence pro svobodný software

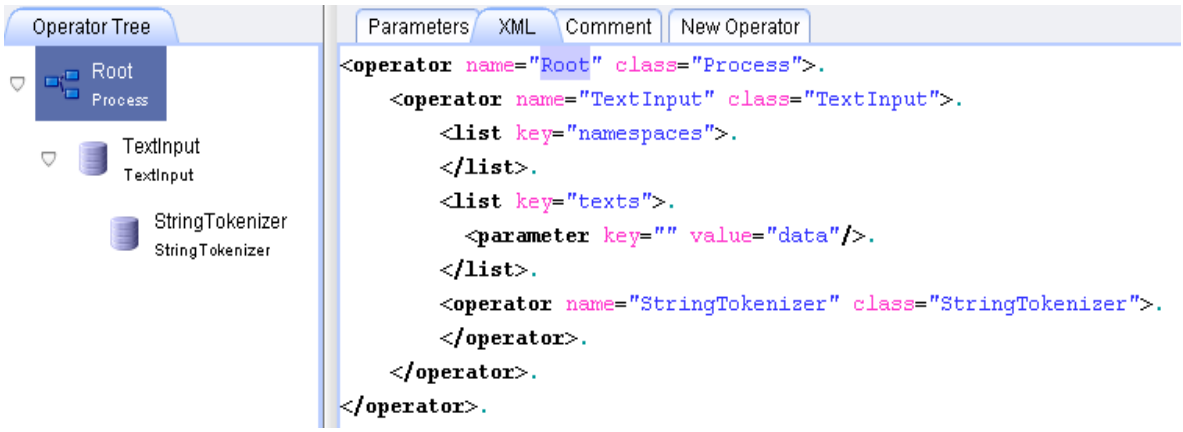
všemi doplňky něco přes 400 operátorů ze všech oblastí data miningu. Po doinstalování doplňku pro text mining vznikne nabídka **Text**, kde je na výběr z několika text miningových operátorů.



Obrázek 18: Grafické uživatelské rozhraní programu RapidMiner. [Zdroj: vlastní]

GUI RapidMineru je použito k návrhu operátorů, k interaktivní kontrole běžících procesů a k nepřetržitému monitorování vizualizace výsledků. Ke kontrole průběžných výsledků a datového toku mezi operátory je zde možnost použití tzv. bodů zlomu. Grafické rozhraní usnadňuje způsob použití jednotlivých operátorů, stromů těchto operátorů nebo komplexních stromů operátorů.

RapidMiner, jak bylo dříve uvedeno, používá XML (eXtensible Markup Language). Je to široce používaný jazyk vhodný pro popis strukturovaných objektů. Zde slouží k popisu modelovacích procesů pomocí stromu operátorů (obrázek 19). XML se stal standardním formátem pro výměnu dat. Navíc je tento formát snadno vnímán jak počítači tak i lidmi. Všechny procesy RapidMineru jsou popsány v XML formátu. V tomto případě XML rozumíme skriptovací jazyk pro data miningové procesy.



Obrázek 19: Zápis XML zanořených operátorů. [Zdroj: vlastní]

Na obrázku 19 je vidět zanoření tří operátorů a jejich zápis v XML. XML je platformně nezávislý značkovací jazyk. Tento jazyk je především určen pro výměnu dat mezi aplikacemi. Jazyk XML nemá žádné předdefinované tagy, je třeba definovat vlastní značky, které budou používány. XML bývá tzv. well formed, čili dobře strukturovaný, aby byl efektivně zpracován počítači i dobře vnímán lidmi. Formát XML musí tedy splňovat následujících 6 pravidel pro dobře strukturovaný dokument [6]:

1. záleží na velikosti písmen (case sensitive),
2. dokument musí mít právě jeden kořenový element,
3. všechny tagy musejí být uzavřené,
4. elementy se nesmějí křížit,
5. hodnoty atributů musejí být uzavřeny v uvozovkách nebo apostrofech,
6. pokud není v prologu uvedeno jinak, použije se jako kódování UTF-8.

V ukázce automaticky vygenerované systémem RapidMiner jsou tyto požadavky splněny. Tři elementy *operator*, které jsou dobře zanořeny a ukončeny. Každý element má atribut *name*, což je unikátní název operátoru a atribut class, který označuje třídu operátoru. Parametry mají dvojici atributů *key* - *value*, které označují klíčový atribut a jeho hodnotu. Atributy elementů jsou uzavřeny v uvozovkách a vše je psáno malými písmeny. Kódování se nastavuje ve vlastnostech nástroje a tedy úvodní prolog s verzí xml a kódování není v zápise XML zobrazeno.

Vícevrstvý datový koncept

Nejvýznamnější charakteristikou systému RapidMiner je schopnost zanoření operátorů a tvorba komplexních stromů operátorů. Pro podporu této charakteristiky působí datové jádro

systému RapidMiner jako datová základna systému řízení a poskytuje vícevrstvé pojetí datových pohledů na centrální tabulku dat. Prvním pohledem může být například vybrána podmnožina příkladů a druhým pohledem vybereme podmnožinu znaků.Výsledkem je jeden pohled, který odráží oba pohledy. Jinými pohledy jsou vytvářeny nové atributy nebo filtrovány data. Počet vrstev pohledů není omezen. [15]

Tohle vícevrstvé pojetí je také efektivní způsob při ukládání odlišných pohledů ve stejné tabulce dat. Je to zvláště důležité pro automatické úlohy předzpracování dat, jakými jsou tvorba nebo selekce znaků. [15]

Bez ohledu na to, zda datová sada je uložena v paměti, v souboru nebo v databázi, RapidMiner k jejich vyjádření používá speciální typ datové tabulky. [15]

Transparentní zpracování dat

RapidMiner podporuje flexibilní uspořádání procesů, které umožňuje vyhledání nejlepšího schématu učení a manuální předzpracování datových a učících se úloh. Jednoduchá adaptace a evalvace různých návrhů procesu umožňují porovnání jednotlivých řešení. [15]

RapidMiner dosahuje transparentního zpracování dat podporou několika typů datových zdrojů, skrytím interních transformací dat a jejich oddělením od uživatele. Díky koncepci modulárního operátoru lze k optimalizaci výkonu nahradit pouze jeden operátor, zatímco zbytek návrhu procesu zůstává stejný. Tohle je důležitý znak při optimalizaci reálných aplikací. [15]

Vstupní objekty operátorů jsou zpracovány následnými operátory. Jestliže vstupní objekty nejsou těmito operátory vyžadovány, jsou jednoduše přeskočeny a mohou být použity novějšími nebo vnějšími operátory. Tímto umožněním předání objektů z jednoho operátoru přes několik dalších k cílovému operátoru se zvyšuje flexibilita systému RapidMiner. [15]

Typické objekty zpracovávané operátory jsou predikční modely, evaluační vektory apod. Operátory mohou ke vstupním objektům přidat informace, např. popisky k dříve neoznačeným příkladům nebo nové znaky při použití operátorů tvorby znaků. [15]

Rozšíření systému RapidMiner

RapidMiner podporuje implementaci uživatelem definovaných operátorů. Pro implementaci těchto operátorů uživatel definuje očekávané vstupy, výstupy, povinné a volitelné parametry a funkce operátoru. Vše ostatní již zařídí RapidMiner. Zápis operátorů v XML automaticky vytváří odpovídající elementy v grafickém rozhraní. [15]

Pro účely text miningu je potřeba instalace doplňku Word Vector Tool (WVTool). Je to flexibilní knihovna programovacího jazyka Java, která slouží k modelování textů. WVTool je konkrétně postaven na modelu vektorového prostoru. Tento model slouží k vytváření vektoru slov představující textové dokumenty. V modelu vektorového prostoru je dokument představen vektorem, který označuje významnost daných termínů tohoto dokumentu. Termíny jsou obvykle slova přirozeného jazyka, ale mohou to být obecně entity, které jsou redukovány na lingvistický základ. [21]

Dříve byl tento model vektorového prostoru při automatickém zpracování textu a získávání informací významný. Nyní je tento model pro mnoho úloh automatického zpracování textu, jakými jsou klasifikace textu, shlukování, sumarizace a získávání informací, pouze počátečním bodem. [21]

Cílem WVTool je poskytnout snadnou použitelnost a rozšiřitelnost Java knihovny při tvorbě vektorů slov. Nástroj je těsně integrován s prostředím strojového učení, který umožňuje vykonání experimentů přímo použitím textových dat. I proto plugin WVTool vyplňuje mezeru mezi propracovanými lingvistickými balíky jako systém GATE na jedné straně a mnoha dílčími řešeními, které jsou částmi různých textových a IR aplikací na straně druhé. Nejpodobnější k WVTool je Bow package, což je nástroj napsaný v jazyce C, který slouží pro tvorbu vektorů slov, shlukování a klasifikaci textu. [21]

3.1 Počáteční konfigurace

Při počáteční konfiguraci je potřeba nastavit operátor *TextInput* a případně vytvořit seznam slov. Operátor *TextInput* generuje z kolekce textu vektory slov. Tento operátor převede vstupní text do tzv. sady příkladů (example set).

Seznam slov obsahuje statistiku termínů, které se objevují v textech. Tyto seznamy slov se ukládají do souborů a jsou důležité při pozdějším použití.

V některých případech je potřeba tento seznam slov načíst z vytvořeného souboru. Operátor pak využívá tyto informace k nalezení, které termíny v textu vzít do úvahy a jak ováhovat atributy (zvláště pro váhu tf/idf)

3.1.1 Operátor TextInput

Operátor *TextInput* vytváří z kolekce textů sadu příkladů *ExampleSet*. Výstup *ExampleSet* obsahuje jeden řádek pro každý textový dokument a jeden sloupec pro každý termín.

Textová kolekce musí být specifikována jedním ze dvou způsobů [21]:

1. pokud je určen parametr *texts*, každá dvojice klíč-hodnota musí obsahovat popis třídy a adresář, ve kterém jsou texty. Ve výchozím nastavení jsou parametry *default_encoding*, *default_language* a *default_type* použity pro všechny vstupní dokumenty.
2. jinak operátor očekává *ExampleSet* na vstupu. Sady příkladů obsahující speciální názvy a popisky jsou zasazeny do těchto pěti obvyklých atributů:
    a) *document_source* – soubor, slovník nebo URL určující text
    b) *type* – typ dokumentu
    c) *encoding* – kódování obsahu dokumentu
    d) *language* – jazyk obsahu dokumentu
    e) *the label attribute* – třída nápisů textu

K operátoru *TextInput* se musí přidat vnitřní operátory, jenž představují potomky tohoto operátoru. Tento operátor slouží jen pro načtení dokumentu a bez těchto vnitřních operátorů extraktor nepozná co se má vykonat. V každém kroku je zde pomocí bodů přerušení možnost kontroly funkce na vstupních datech.

Pro nastavení parametrů operátoru *TextInput* je zde možnost přepnutí mezi dvěma režimy. V režimu pro začátečníky (*Beginners mode*) je pouze jeden parametr *texts*, kde se načtou texty určené ke zpracování. V režimu pro pokročilejší uživatele (*Expert mode*) lze nastavit spoustu volitelných parametrů.

3.1.2 Vytvoření a údržba seznamu slov

Pro mnoho aplikací je užitečné vytvořit a udržovat seznamy slov (a tedy dimenze vektorového prostoru) manuálně. Tuto funkčnost poskytuje operátor *InteractiveAttributeWeighting* v kombinaci s operátory *TextInput* a *CorpusBased Weighting*. [21]

Vždy je vhodnější definovat výsledné seznamy slov jako parametr operátoru *TextInput*, které lze importovat do souboru a později v procesu aplikace opětovně načíst. [21]

Počáteční seznam slov lze vytvořit použitím následujících zřetězených operátorů: *TextInput*, *CorpusBaseWeighting* a *InteractiveAttributeWeighting*. *TextInput* vytvoří počáteční seznam slov. Operátor *CorpusBasedWeighting* ohodnotí každý termín v seznamu s ohledem na jeho významnost. Váha daného termínu je vypočítána sečtením vah tohoto termínu ze všech dokumentů ve třídě. Cílem této metody je dát vysokou váhu termínům, které jsou pro danou

třídu významné. Součty ostatních tříd slouží jako základní znalost o tom, jak významné termíny jsou v celém korpusu (ačkoli operátor může být použit pouze s jednou třídou). Při použití operátoru *InteractiveAttributeWeighting* se objeví okno, které uživateli nabídne seznam slov. Kliknutím na ikonu nad tabulkou lze termíny seřadit abecedně nebo podle jejich váhy. Tlačítko vedle každé položky slouží k výběru klíčových slov (nastavením jejich vah na 1 nebo 0). Pro uložení seznamu slov slouží tlačítko save. Výsledný soubor obsahuje řádky následujícího formátu [21]:

<termín>: <váha>

Seznam slov lze použít dvěma způsoby. K použití aktuálních vah nejdříve uživatel operátorem *TextInput* vytvoří vektory slov a poté pro dosažení výsledného *ExampleSet* použije operátory *AttributeWeightsLoader* a *AttributesWeightsApplier*. K selekci významných termínů ze seznamu slov nejdříve použije *AttributeWeightsLoader*. *TextInput* vytvoří vektory, které ve formě dimenze obsahují pouze termíny ze seznamu slov, které mají váhu větší než nula. [21]

Po přidání nových dokumentů do korpusu jsou nově přidané termíny obvykle zařazeny mezi významné a měli by tedy být přidány do seznamu slov. Operátorem *InteractiveAttributeWeighting* se použije funkce k načtení původního seznamu slov. Při nastavení parametru přepsání „overwrite“ budou hodnoty načtené ze souboru a vygenerované operátorem *TextInput* přepsány. Všechny termíny, pro které je již rozhodnuto, zda by měli nebo neměli být v seznamu slov, jsou chráněny. Všechny nové termíny budou v seznamu setříděny podle jejich vah. [21]

3.2 Extrakce informací

WVTool není zamýšlen jako propracovaný systém procesu extrakce informací. Tento doplněk poskytuje jednoduché, ale výkonné dotazy k získání strukturovaných informací z polostrukturovaných dat.

Nástroj podporuje 2 základní způsoby extrakce informací [21]:

1. dotazy XPath
2. regulární výrazy

3.2.1 Extrakce informací pomocí XPath

XPath je jazyk, který umožňuje vybírat jednotlivé části dokumentu. Pro potřeby tohoto jazyka je dokument chápán jako stromová struktura, kde jsou jednotlivé uzly tvořeny elementy, atributy a obsahem elementů. Výsledkem výrazu v XPath je pak skupina uzlů. V tabulce 4 následuje výčet základních výrazů XPath.

Tabulka 4: Základní konstrukce jazyka XPath. Zdroj: [15]

Výrazy XPath	Popis
/	Kořen dokumentu, oddělovač cesty
.	Současný uzel
..	Nadřazený uzel (rodič)
*	Jakýkoliv potomek
//	Prohledání celého dokumentu (traversal)
name	Element
@name	Atribut
[..]	Podmínka
f(x,y)	Funkce s argumenty

Pomocí XPath se mohou extrahovat informace pouze ze strukturovaných dokumentů. V případě HTML je kód nejprve převeden do XHTML. Potom se mohou stanovit obvyklé dotazy XPath na HTML. Pro tento účel je v systému RapidMiner vytvořen nový experiment a ke kořenovému uzlu Root připojen operátor *Extractor* (obrázek 20). Ve vlastnostech k tomuto operátoru musí být vybrán adresář, kde se nacházejí požadované soubory. Tohle je provedeno pomocí parametru *texts*.

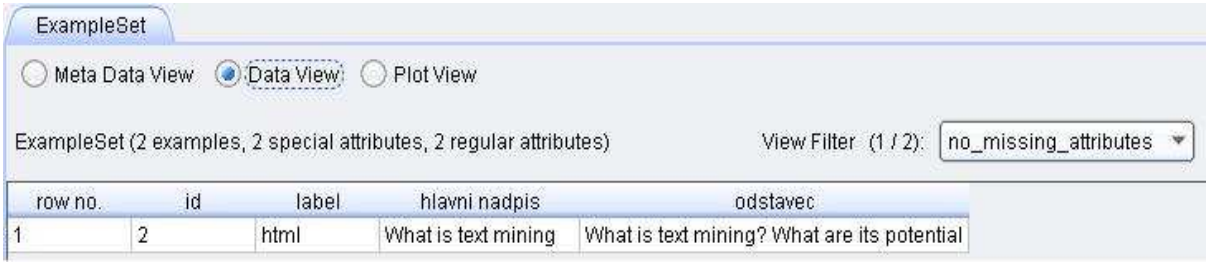
Obrázek 20: Strom operátoru při dotazu XPath. [Zdroj: vlastní]

Nyní může být zadán dotaz kliknutím na parametr *attributes*. Stisknutím tlačítka **Add** je přidán nový dotaz. Na levé straně bude zadán název atributu. Pole na pravé straně bude obsahovat dotaz XPath, který extrahuje požadované informace. Může být přidáno libovolné množství řádků, tedy atributů. Na následujícím obrázku je ukázka dotazu XPath s dvěma atributy (obrázek 21).

Obrázek 21: Parametr attributes. [Zdroj: vlastní]

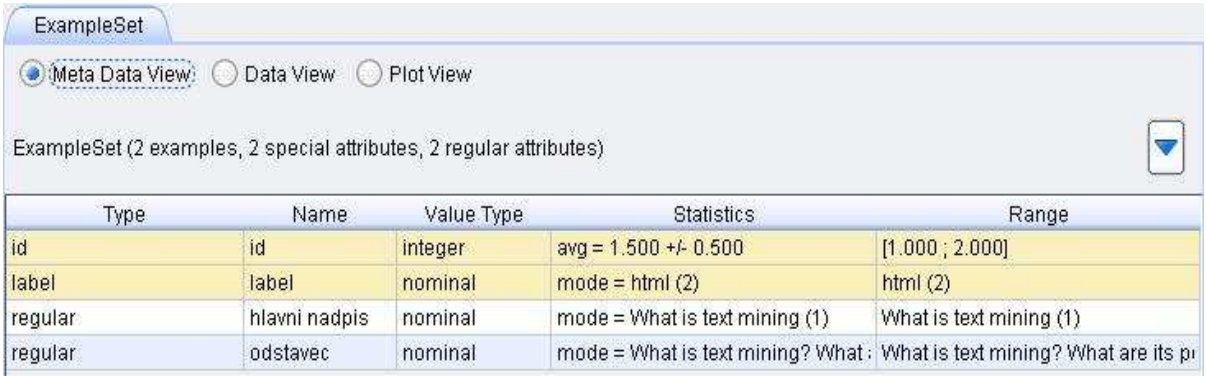
Že jde o dotaz XPath, nikoliv regulární výraz pozná parser podle počátečního znaku "/". Jmenný prostor XHTML je automaticky svázán s identifikátorem h. Například výrazem /h:html/h:body/text() by bylo vyselektováno kompletní tělo souboru HTML. Právě jmenné prostory jsou u XPath častým zdrojem problémů. Pokud zdrojový soubor používá určité jmenné prostory, tyto jmenné prostory musí být nadeklarovány v atributu *namespaces*. Jde totiž o názvy elementů, atributů a potřebných nastavení strukturovaného dokumentu. Číselné atributy musí jako první znak obsahovat znak mřížky #. [21]

Pro extrakci byly zvoleny tyto atributy: hlavní nadpis stránky a odstavec s atributem *class* nastaveným na *text*. Před spuštěním dotazu se tlačítkem **Preview** uživatel může přesvědčit, zda je XPath dotaz sestaven správně. Jestliže je vše v pořádku, může se spustit experiment. Po stisknutí tlačítka **Run** pro spuštění procesu se zobrazí výsledná sada příkladů *ExampleSet* (obrázek 22).



Obrázek 22: Výsledek XPath dotazu – Data View. [Zdroj: vlastní]

Uživatel má na výběr vizualizaci výsledků ze tří možností. Plot View zobrazuje výsledek ve formě grafů, což se v tomto případě nehodí. Pro přehledné zobrazení výsledků byla v tomto případě vybrána možnost **Data View**. Výsledek se zobrazil na jednom řádku, což znamená, že se extrahovalo pouze z jednoho dokumentu. Sloupce odpovídají jednotlivým atributům včetně atributů identifikačních. Hodnoty jednotlivých polí odpovídají extrahovaným XPath dotazům. Možnost **Meta Data View** (obrázek 23) nabízí statistiku extrahovaných výrazů. Typ hodnoty je v tomto případě jmenný (nominal), v případě extrakce čísel by tato hodnota byla numeric.


ExampleSet

Meta Data View | Data View | Plot View

ExampleSet (2 examples, 2 special attributes, 2 regular attributes)

Type	Name	Value Type	Statistics	Range
id	id	integer	avg = 1.500 +/- 0.500	[1.000 ; 2.000]
label	label	nominal	mode = html (2)	html (2)
regular	hlavni nadpis	nominal	mode = What is text mining (1)	What is text mining (1)
regular	odstavec	nominal	mode = What is text mining? What :	What is text mining? What are its p

Obrázek 23: Výsledek XPath dotazu – Meta Data View. [Zdroj: vlastní]

Výsledný soubor celého procesu extrakce informací pomocí dotazů XPath je k dispozici na přiloženém CD pod názvem **XPath.xml**.

3.2.2 Extrakce informací regulárními výrazy

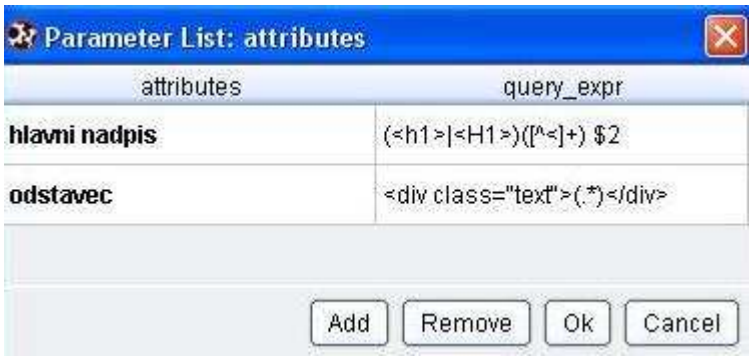
Regulární výraz je speciální řetězec znaků, který představuje určitý vzor pro textové řetězce. Regulární výrazy lze v současné době najít téměř ve všech programovacích jazycích. Slouží pro validaci e-mailové adresy, ve formulářích na webu před odesláním ke kontrole zadávaných dat apod. Existuje více druhů regulárních výrazů. RapidMiner je napsán v jazyce Java, a proto tento jazyk využívá regulární výrazy tohoto programovacího jazyka

Regulární výrazy jsou aplikovány na vstupní text, který chceme zpracovat. Výsledkem je sada shod. Pokud v sadě není shoda žádná, výraz nesouhlasí se vstupním textem. Při jedné shodě regulární výraz souhlasí na jednom místě vstupního textu a při několikanásobné shodě i na více místech. Vzory jsou základní prvky pro vytváření regulárních výrazů. Vzorem se popisuje, jak má přípustný řetězec vypadat. Následuje přehled základních pravidel pro vytváření vzorů (tabulka 5).

Tabulka 5: Základní pravidla pro vytváření vzorů. Zdroj: [21]

Výraz	Popis	Příklad výrazu	Odpovídající řetězce
^	Začátek textu	^abc	abc, abcd, abc123…
$	Konec textu	x$	abcx, x, aaax…
(…)	Logická skupina		
[…]	Výčet možných znaků	[0-9]	1, 2, 4, 5, 6…
[^…]	Obrácený výčet znaků	[^0-9]	a, b, z, x…
{…}	Pevný počet výskytů	ab{3}c	abbbc
{…,…}	Rozsah počtu výskytů	ab{1,2}c	abc, abbc
{…,}	Minimální počet výskytů		
*	Žádný nebo více znaků	A*hoj	hoj, Ahoj, AAhoj, AAAhoj…
+	Jeden nebo více znaků	A?hoj	Ahoj, AAhoj, AAAhoj…
?	Žádný nebo jeden znak	A?hoj	Ahoj, hoj
…|…	Alterace - více možností	A(hoj|uto|aa)	Ahoj, Auto, Aaa
.	Libovolný znak (kromě \n)	A.oj	Aaoj, Aboj, Acoj, Adoj…
\t	Zastupuje znak tabulátoru		
\r	Zastupuje návratu hlavy		
\n	Zastupuje nový řádek	\r\n	{konec řádku}
\w	Ekvivalent [a-zA-Z_0-9]	\w+	sba34, 45, A1, fgBc…
\W	Ekvivalent [^a-zA-Z_0-9]	\W+	\, –, +, *…
\d	Ekvivalent [0-9]	\d+	753, 4, 678, 3…
\D	Ekvivalent [^0-9]	\D+	ahoj, abc, df…
\s	Zahrnuje neviditelné znaky		
\S	Zahrnuje viditelné znaky	\S+	Abc123, asd-dgg, fb…

Postup extrakce je stejný jako u XPath. Jediný rozdíl je v tom, že namísto dotazu XPath bude použit regulární výraz. Postup je vidět na následujícím obrázku (obrázek 24). V poli *attributes* uživatel zadá libovolný název atributu a v poli *query_expr* bude regulární výraz. V případě, že před název atributu bude vložen znak #, potom tento atribut bude považován za numerický. Nový dotaz se do seznamu parametrů přidá stisknutím tlačítka **Add**.



Obrázek 24: Parametr attributes - regulární výrazy. [Zdroj: vlastní]

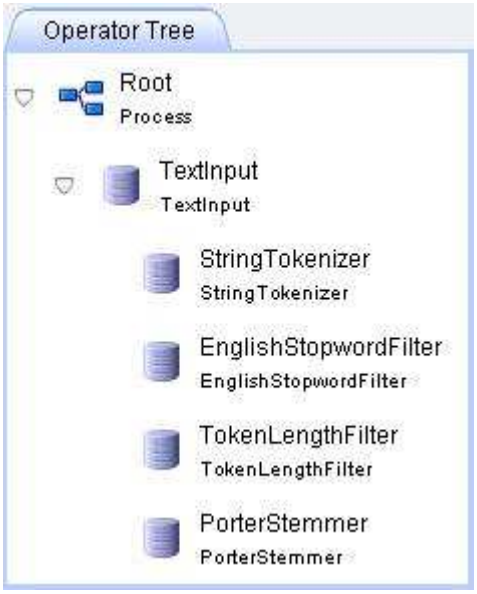
RapidMiner rozlišuje mezi dotazy XPath a regulárními výrazy pomocí prvního znaku dotazu. Pokud dotaz začíná znakem /, tak je tento výraz interpretován jako dotaz XPath. Jestliže regulární výraz jako první znak obsahuje znak lomítka /, použije se úniková sekvence.

Po stisknutí tlačítka **Run** pro spuštění procesu lze vidět výsledný *ExampleSet*. V případě XPath i regulárních výrazů byly extrahovány stejné hodnoty. Výsledek je tedy shodný.

Výsledný soubor celého procesu extrakce regulárními výrazy je k dispozici na přiloženém CD pod názvem **Regularni_vyrazy.xml**.

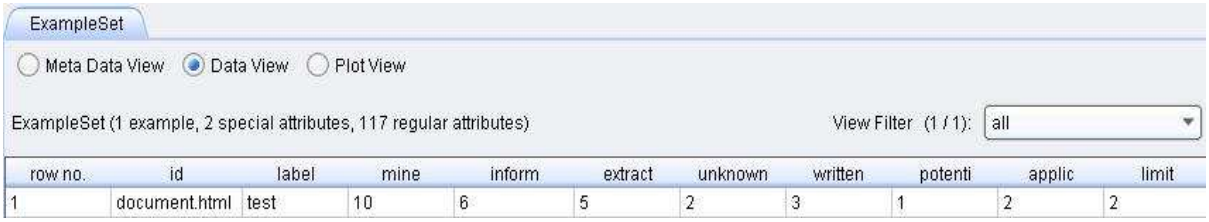
Extrakce obsahu HTML

Na následujícím obrázku je vidět použití operátoru *TextInput* spolu s vnitřními operátory *StringTokenizer*, *EnglishStopwordFilter*, *TokenLengthFilter a PorterStemmer* při extrakci obsahu webové stránky s anglickým textem (obrázek 25). Funkce operátorů je následující: *TextInput* vygeneruje vektory slov z textové kolekce, *StringTokenizer* rozdělí text na sadu tokenů, *EnglishStopwordFilter* je standardní seznam stopword pro anglické texty. Seznam stopwords obsahuje tokeny, které by neměly být uvažovány při vektorizaci a jsou tedy vyfiltrovány. *TokenLengthFilter* filtruje termíny založené na minimálním počtu znaků, které musí obsahovat, a *PorterStemmer* je porter stemmer pro anglické texty, který vykonává algoritmus tohoto porter stemmeru. Stemming je technika, která redukuje slova na kořen tohoto slova. [21]



Obrázek 25: Strom operátorů při extrakci obsahu webové stránky. [Zdroj: vlastní]

Po stisknutí tlačítka **Run** pro spuštění procesu je vidět výsledný *ExampleSet* (obrázek 26). Zde lze zvolit vizualizaci výsledků opět ze tří možností. Byla vybrána možnost **Data View**. Jednomu řádku odpovídá jeden dokument. U operátoru *TextInput* byl nastaven parametr *text_query* na <p>([^<]+) $1, což znamená extrakci termínů ze všech odstavců značících tagem <p>.



Obrázek 26: Výsledný ExampleSet – regulární výrazy. [Zdroj: vlastní]

V tomto případě došlo k extrakci 117 regulárních atributů. Nejčastějším termínem ve vzorovém textu je slovo mine čili dolovat. Tato extrakce umožňuje pouhý výčet a sumu jednotlivých termínů.

Výsledný soubor celého procesu extrakce obsahu webové stránky je k dispozici na přiloženém CD pod názvem **Extrakce_obsahu_HTML.xml**.

3.3 Shrnutí kapitoly

RapidMiner je data miningový nástroj vyvíjený pod licencí GNU komunitou open source. Pro účely text miningu se musí nainstalovat doplněk WVTool, který do nástroje přidá několik nových operátorů text miningu.

Výhodou systému RapidMiner je, že je napsán v Javě a je tedy multiplatformní. RapidMiner jde spustit z příkazového řádku jako knihovna programovacího jazyka Java nebo nejintuitivnější cestou je tvorba experimentů v grafickém rozhraní programu. Výhodou je též implementace vlastních operátorů.

Extrakce informací probíhá dvěma způsoby. Prvním způsobem je extrakce pomocí dotazů XPath. Zde je použití pouze pro strukturované dokumenty XML a HTML. Druhým způsobem je extrakce pomocí regulárních výrazů. Jsou to výrazy, které podobně jako dotazy XPath extrahují informace z dokumentu. Výhodou regulárních výrazů oproti XPath je to, že se nemusí extrahovat vyloženě ze strukturovaných dokumentů. Při extrakci se do parametru attributes operátoru *FeatureExtraction* zadá dotaz XPath případně regulární výraz. Extrakce je omezena jen na dotaz v parametru attributes.

Kromě extrakce jednotlivých částí strukturovaného lze extrahovat i obsah textového dokumentu. V tomto případě se ke kořeni stromu musí připojit operátor *TextInput* pro zadání textu, ze kterého proběhne extrakce. Jako vnitřní operátory se musí připojit operátory *StringTokenizer*, *EnglishStopwordFilter*, *TokenLengthFilter*, *PorterStemmer*. Tyto vnitřní operátory slouží pro tokenizaci a následné oddělení jednotlivých slov. Extrakce obsahu celého dokumentu lze aplikovat jen na anglické a německé texty. Jako vnitřní operátor se musí zadat

operátor seznamu termínů. Tento operátor je k dispozici jen pro 2 zmíněné varianty (*EnglishStopwordFilter* a *GermanStopwordFilter*). Při absenci tohoto operátoru dochází k nekorektní extrakci termínů.

Nejvýznamnější charakteristika systému RapidMiner je shrnuta v následující tabulce 6.

Tabulka 6: Charakteristika systému RapidMiner. [Zdroj: vlastní]

Charakteristika	Popis
Možnosti text miningu	• Extrakce informací • Klasifikace • Shlukování textů
Funkce extrakce informací	• Dotazy XPath • Regulární výrazy
Výhody	• Zdarma • Implementace vlastních operátorů
Nevýhody	• Neextrahuje typy termínů • Málo podporovaných formátů • Málo jazyků k extrakci

4 GATE

GATE (*General Architecture for Text Engineering*) je software pro text mining. Byl vyvinut na Univerzitě v Sheffieldu v roce 1995 skupinou studentů a učitelů. GATE je software šířený pod licencí GNU. Protože je založen na knihovnách programovacího jazyka Java, což je multiplatformní programovací jazyk, může být vyvíjen a užíván téměř na všech operačních systémech. [4]

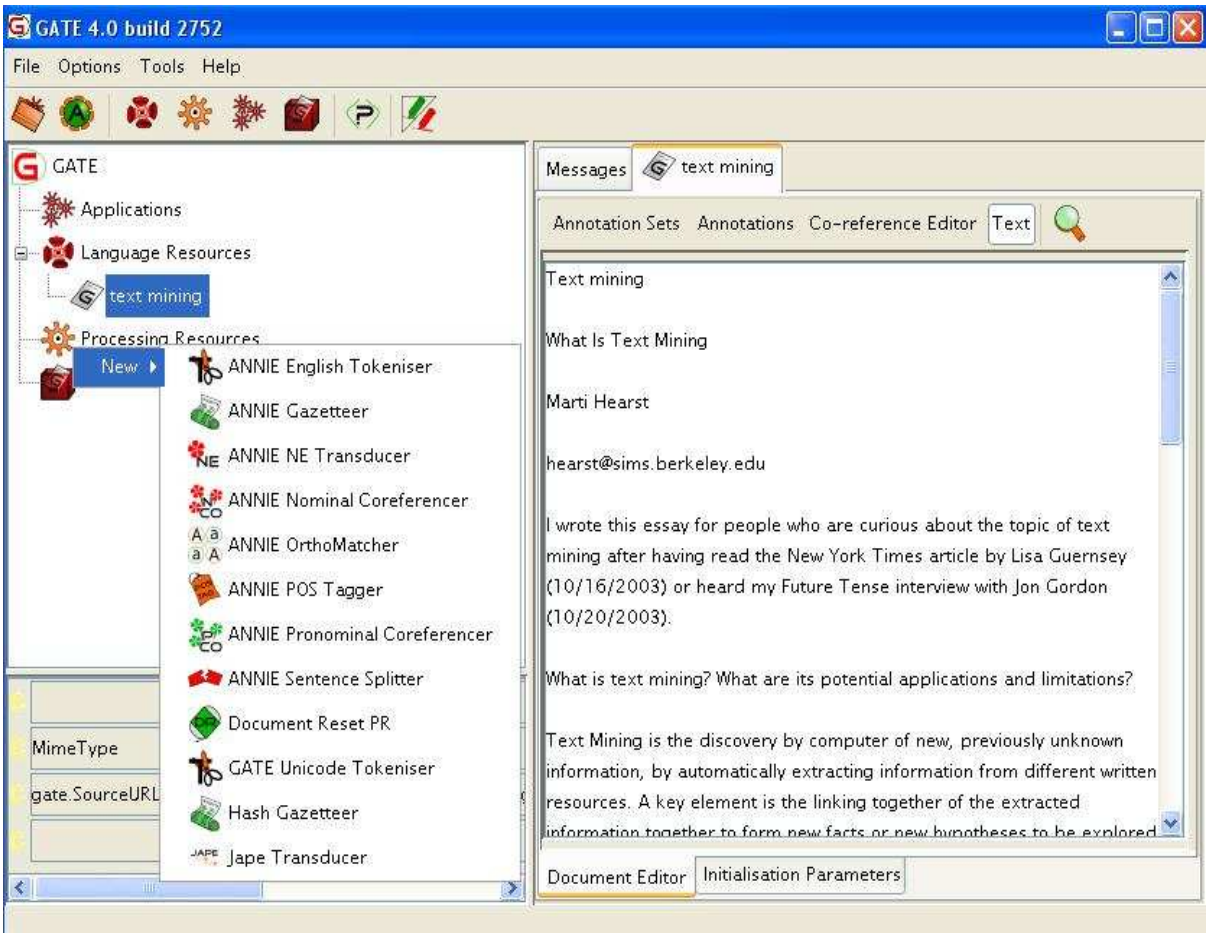
GATE poskytuje podporu pro úlohy text miningu jako extrakci informací, hodnocení aplikací, tvorbu a anotaci korpusu. Korpusem se rozumí *rozsáhlý vnitřně strukturovaný a ucelený soubor textů daného jazyka elektronicky uložený a zpracovávaný.* [25] Mezi současné využití patří tvorba a anotace korpusů v mnoha jazycích, např. Americký národní corpus (American National Corpus), Britský národní korpus (British National Corpus) a Český národní korpus.

GATE zahrnuje mnoho nástrojů pro úlohy zpracování textů jako tokenizace, identifikace vět, značení částí textu, větný rozbor, vyhledávání ve slovníku klíčových slov, vyhledávání informací a rozpoznávání entit. K těmto procesům systém potřebuje mít přístup skrze jazykové zdroje. GATE má tři typy jazykových zdrojů: dokumenty, korpusy a ontologie. Ontologie je slovníkem, který slouží k uchovávání a předávání znalosti týkající se určité problematiky. GATE pro usnadnění práce s aplikacemi, které nepoužívají angličtinu, využívá standard kódování znaků Unicode. Unicode je používán k vývoji aplikací a korpusů v různých slovanských, germánských, románských a indických jazycích. [20]

GATE je distribuován se systémem extrakce informací, nazvaný ANNIE, který detekuje např. osoby, názvy organizací, klíčová slova, datumy, čas a jednotky měn. Používá slovník se seznamem klíčových slov, a to měst, zemí, organizací nebo třeba dnů týdne. Další částí ANNIE je sémantické značení, které používá pravidla jazyka JAPE, ve kterém jsou popsány vzory. Vzory jsou určeny textovými řetězci nebo anotacemi, které byli dříve vytvořeny pomocí modulů, jakými jsou tokenizer, slovník klíčových slov nebo analýza struktury dokumentu. Začleněné moduly také rozpoznají relace mezi entitami a detekovanými odkazy. GATE též obsahuje nástroje pro tvorbu zdrojů nových jazyků a pro hodnocení výkonu text miningových systémů. [20]

Obrázek 27 ukazuje hlavní okno aplikace s načteným jedním dokumentem. V tomto prostředí se vyskytují čtyři hlavní oblasti:

1. panel nabídek s nabídkami **File**, **Options**, **Tools** a **Help**,
2. v levé horní oblasti se nachází strom zdrojů zahrnující položky **Applications**, **Language Resources**, **Processing Resources** a **Data Stores**,
3. malý panel na levé straně v dolní oblasti je prohlížeč zdrojů,
4. největší oblast zahrnuje hlavní prohlížeč zdrojů, který zobrazuje načtené dokumenty a spuštěné aplikace. Pomocí záložek se přepínají události. Na obrázku 27 je aktivní záložka načteného textu. Na záložce *Messages* se zobrazují chybové zprávy nebo doplňující informace.



Obrázek 27: Grafické uživatelské prostředí nástroje GATE. [Zdroj: vlastní]

Strom zdrojů a oblast prohlížeče zdrojů se navzájem ovlivňují a poskytují zobrazení zdrojů různými způsoby. Vizualizační zdroje integrované v GATE mohou mít malý i velký náhled, zatímco datové sklady mají jen malý náhled a dokumenty jen velký náhled.

Všechny zdroje, aplikace a datové sklady, které jsou současně načteny v systému, se objevují ve stromu zdrojů. Kliknutím na zdroj se objeví prohlížeč zdrojů v jedné z oblasti zdrojů.

GATE podporuje mnoho formátů jako XML, RTF, HTML, SGML, e-mail a čistý text, které systém pro podporu anotací upraví do jednotného modelu. Tři mechanismy uložení jsou: relační databáze, Java objekty a vnitřní formát založený na XML. Dokumenty mohou být opětovně exportovány do jejich původních formátů s anotacemi nebo bez nich. Kódování textu je z důvodu podpory vícejazyčného datového zpracování založeno na Unicode, takže systémy vyvinuté pomocí GATE mohou být převedeny do nově vytvořených jazyků bez doplňkových zdrojů pro daný jazyk. [17]

GATE je rozhraní pro vývoj technologií zpracování jazyka ve více vrstvách. Poskytuje tři typy zdrojů: jazykové zdroje, které souhrnně pojednávají o datech, procesní zdroje, které se vztahují k algoritmům a vizualizační zdroje, které představují komponenty vizualizace a editace. [5]

Po načtení a otevření dokumentu je zavolán analyzátor struktury dokumentu, ve kterém jsou v závislosti na typu dokumentu vytvořeny jazykové zdroje. Tyto jazykové zdroje obsahují text originálního dokumentu, a jedna nebo více sad anotací, které obsahují značky dokumentů (např. html tagy). [5]

Anotace jsou během analýzy textu aktualizovány zdroji pro zpracování, ale mohou být vytvořeny i během editace anotace v uživatelském rozhraní systému GATE. Každá anotace patří k sadě anotací, která má typ, dvojici offsetů[4], sadu atributů a hodnot. Atributy jsou řetězce a hodnotami mohou být Java objekty. Atributy a hodnoty jsou určeny ve schématu anotace, která při manuální anotaci usnadňuje validaci a vstup. [5]

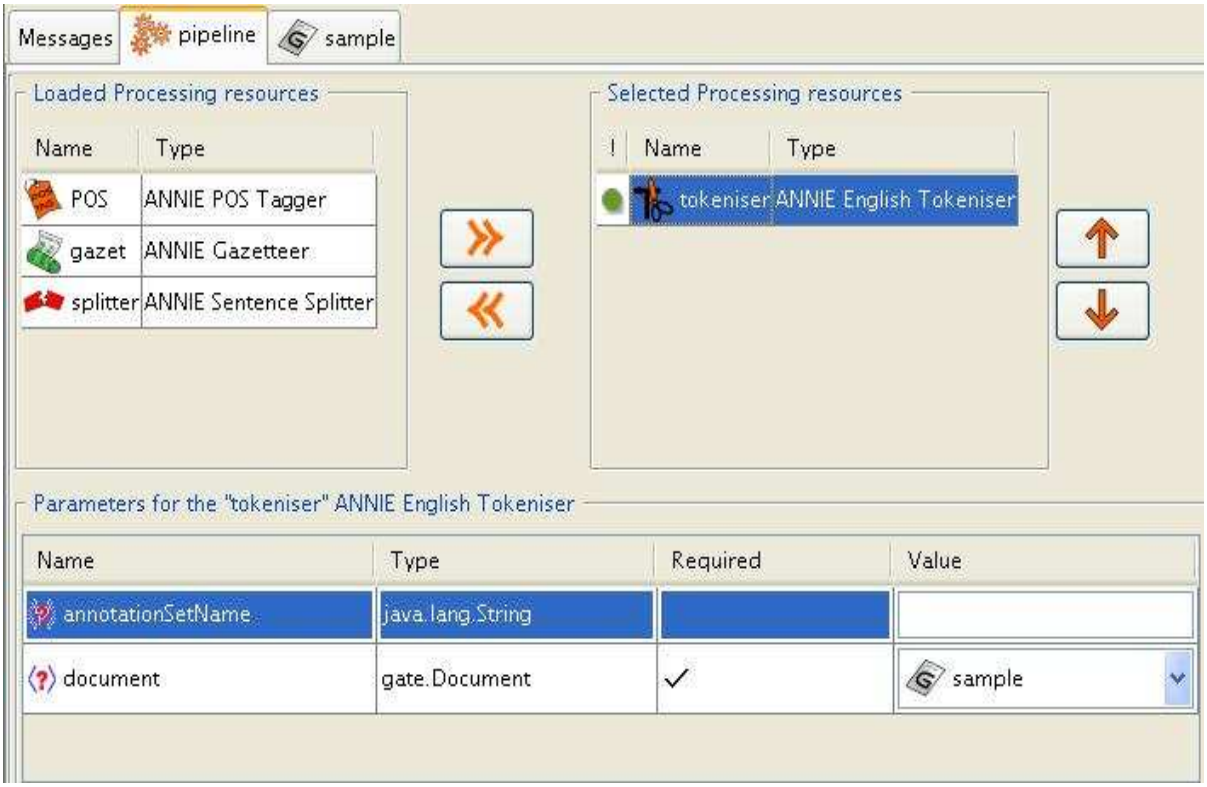
4.1 Vytvoření a spuštění aplikace

Po načtení všech zdrojů lze vytvořit a spustit aplikaci. Tohle je provedeno pravým kliknutím na položku **Applications**, následným výběrem **New**, a poté buď **Corpus Pipeline** nebo **Pipeline**. Aplikace *Pipeline* je spuštěna pouze při načtení jednoho dokumentu, zatímco *Corpus Pipeline* je použita při načtení celého korpusu.

Spojení je zahájeno dvojitým kliknutím uživatele na popisek vytvořeného spojení a následným výběrem procesních zdrojů potřebných ke spuštění aplikace. Tento proces je vidět na obrázku 28. Procesní zdroje se nejprve objeví v oblasti načtených zdrojů. Teprve poté se tyto procesní zdroje přemístí do oblasti vybraných zdrojů na pravé straně okna nastavení. Vybrané komponenty musí být pro korektní zpracování umístěny ve správném pořadí

[4] Rozpětí textu anotace

(začínající odshora). Pokud tomu tak není, správné pořadí se provede výběrem komponenty a následným použitím šipek nahoru/dolů.



Obrázek 28: Procesní zdroje. [Zdroj: vlastní]

Označením jednotlivých vybraných procesních zdrojů se pro tuto komponentu v panelu níže objeví seznam parametrů. Je potřebné zajistit, aby všechny požadované parametry, které jsou v poli *Required* označené fajfkou, byly nastaveny. U každé komponenty je nutné parametrem *document* nastavit dokument ke zpracování. Při použití C*orpus Pipeline* se cesta korpusu pro vybrané komponenty nastaví pouze jednou. Při použití *Pipeline* musí být dokument vybrán pro každý použitý zdroj zpracování. Kliknutím na **Run** se spustí aplikace dokumentu nebo celého korpusu.

Uložení a obnovení jazykových zdrojů

K uložení textu do datovém skladu musí být, pokud ještě neexistuje, vytvořen nový datový sklad. Nový datový sklad se vytvoří pravým kliknutím na **Data Store** na levé straně panelu a následně vyběrem **Create Data Store**. Poté se vybere vhodný typ datového skladu. Tím se vytvoří slovník, který slouží jako datový sklad.

Do datového skladu může být uložen celý korpus (v tomto případě struktura korpusu bude zachována) nebo lze uložit jednotlivé dokumenty. Uložení korpusu se provede kliknutím pravé myši na název korpusu a výběrem možnosti **Save to…**. Uložení individuálních

dokumentů do datového skladu se provede pravým kliknutím na název jednotlivých dokumentů. Poté následuje stejná procedura.

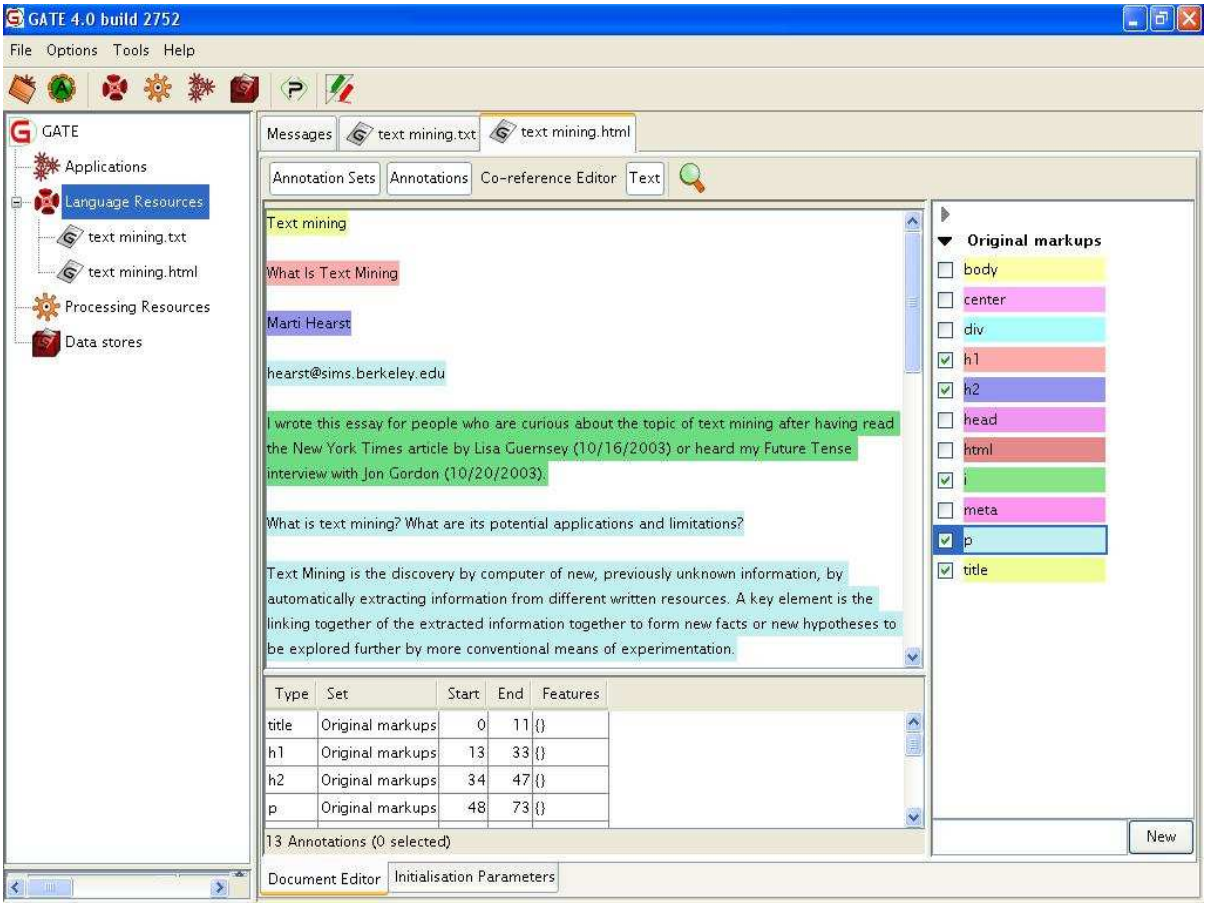
Při načtení dokumentu z datového skladu není vhodné jej načítat jako jazykový zdroj. Na místo toho je vhodné jej otevřít pravým kliknutím na **Data Store** a výběrem možnosti **Open Data Store**. Objeví se seznam datových skladů, které jsou dostupné. V hlavním okně se objeví strom datového skladu. Dvojitým kliknutím na korpus nebo dokument ve stromu zdrojů se tyto dokumenty otevřou. K uložení korpusu a dokumentu do stejného skladu se jednoduše zvolí možnost **Save**.

4.2 Anotace

Anotací dokumentu je myšlen proces obohacování dokumentu dodatečnými informacemi vztahující se k obsahu textu. Proces anotace je zahájen dvojklikem na název souboru na levé straně panelu, který je načten v jazykových zdrojích. Po kliknutí na **Annotations Sets** na pravé straně panelu se začnou prohledávat sady anotací. To způsobí zobrazení prohlížeče anotací, který ukazuje dostupné sady anotací a jejich odpovídající typy. Jestliže není spuštěna žádná aplikace extrakce informací, zobrazené anotace budou pouze ty, které odpovídají analýze formátu dokumentu automaticky provedené systémem GATE ihned při načtení dokumentu (např. HTML nebo XML tagy). Jestliže je aplikace extrakce informací spuštěna, mohou být zobrazeny také jiné typy a sady anotací. Fonty a barvy anotací jsou editovány dvojklikem na název anotace. [4]

K prohlédnutí anotací a jejich atributů uživatel klikne na položku **Annotations** v horní části hlavního okna. Prohlížeč anotací se objeví pod hlavním oknem spolu se zobrazeným textem. Tento prohlížeč obsahuje pouze anotace vybrané ze seznamu anotací. Seznamy mohou být setříděny vzestupně nebo sestupně podle sloupců kliknutím na odpovídající nadpis sloupce.

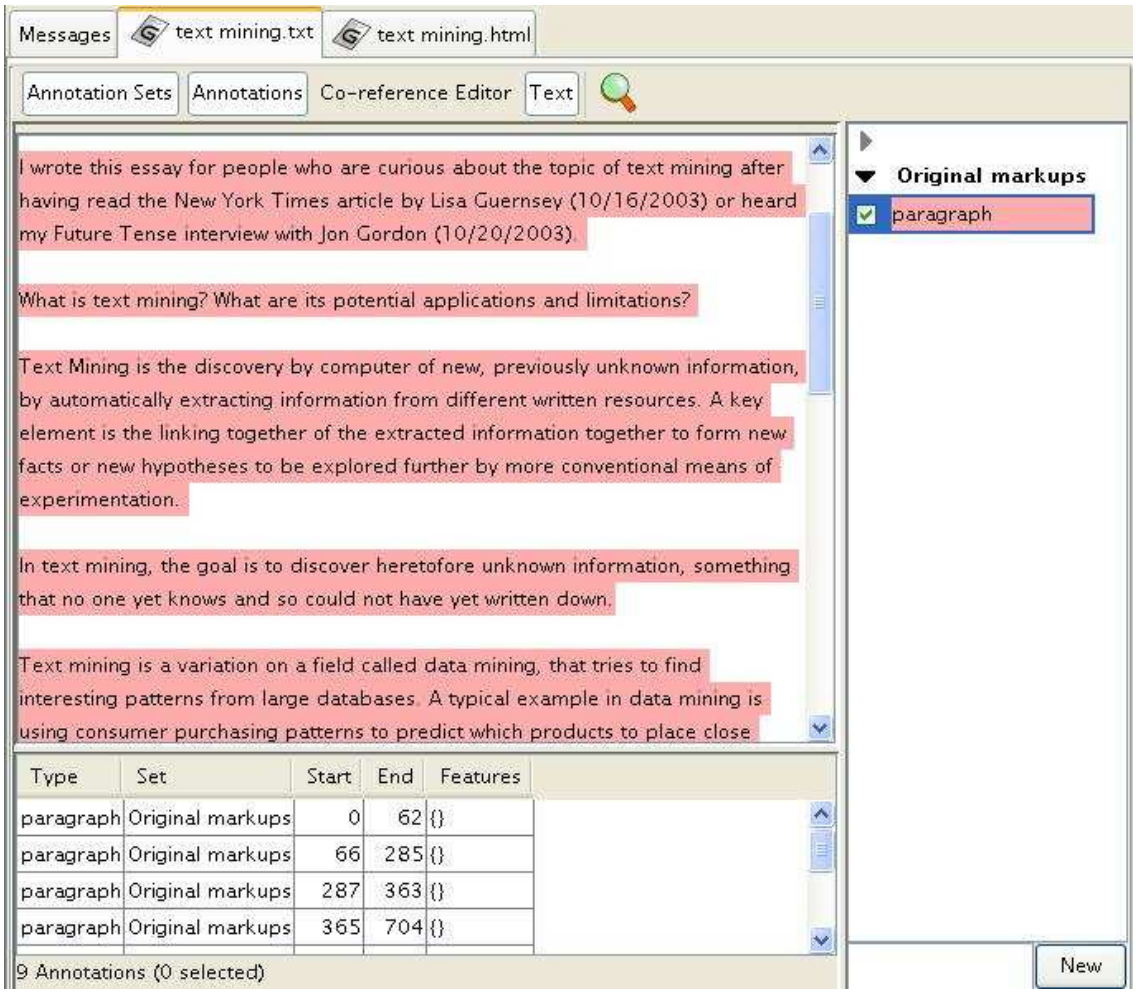
Typ anotace je zvolen kliknutím na příslušné zaškrtávací políčko. Textové segmenty odpovídající těmto anotacím budou zvýrazněny v hlavním textovém okně, jak je zobrazeno na obrázku 29.



Obrázek 29: Anotace webové stránky. [Zdroj: vlastní]

Webová stránka je anotována podle elementů struktury zdrojového kódu. Typy anotací jsou právě tyto elementy. Jednotlivé elementy jsou barevně zvýrazněny. Po zaškrtnutí typu anotace na paletě sady anotací se zaškrtnutý element zvýrazní shodnou barvou i v textu. Na paletě *Annotations* jsou vypsány atributy anotace, včetně typu anotace, počáteční a koncové pozice anotace.

Jako druhý dokument bude zpracováván prostý textový soubor txt. Anotace tohoto dokumentu je znázorněna na následujícím obrázku (obrázek 30).



Obrázek 30: Anotace textového souboru txt. [Zdroj: vlastní]

U textového souboru je pouze jediný typ anotace, a to *paragraph* čili odstavec. Zvýraznění celého dokumentu je tedy stejnou barvou. Paleta *Annotations* znázorňuje stejně jako u webové stránky typ, sadu, počáteční a koncovou pozici anotace.

Hlavní prohlížeč slouží i k zobrazení dodatečných informací. Stane se tak kliknutím na záložku *Messages*. Pokud se při procesu zpracování objeví chyba, záložka zpráv se zvýrazní červeně a objeví se okno chybové zprávy.

Text načteného dokumentu lze editovat v prohlížeči *Document Editor*. Existují zde obvyklá platformová specifika vyjmout (cut), kopírovat (copy), vložit (paste) a příslušné klávesové zkratky v závislosti na nainstalovaném operačním systému.

4.3 Extrakce informací

Nástroj GATE je distribuován se systémem extrakce informací nazvaným ANNIE. ANNIE obsahuje tyto hlavní komponenty: tokenizer, gazetteer, identifikace vět a gramatické značení. Pro načtení a spuštění ANNIE z vývojového prostředí vybere uživatel z menu **File** nabídku

Load ANNIE system. Ke spuštění ve výchozím stavu zvolí *With Defaults*. Automaticky se načtou všechny zdroje ANNIE a vytvoří spojení korpusu. Tohle spojení je načteno s vhodnými zdroji vybranými ve správném pořadí a výchozí vstupní a výstupní sadou anotací. [4]

Pokud uživatel zvolí možnost bez výchozího stavu *Without defaults*, načtou se stejné procesní zdroje, ale pro každý zdroj se objeví okno, které uživateli umožňuje specifikovat název a umístění zdroje. Je to přesně stejná procedura jako pro načítání zdrojů při zpracování individuálních dokumentů. Rozdíl je pouze v tom, že v případě *Load ANNIE system with defaults* systém automaticky vybere všechny procesní zdroje obsažené v ANNIE. Po načtení zdrojů bude vytvořeno spojení korpusu nazvané ANNIE. [4]

Spuštění aplikace je zahájeno kliknutím na **Run** (z editoru *Serial Application* nebo pravým kliknutím na název aplikace a volbou **Run**). Výsledky jsou zobrazeny dvojklikem na název dokumentu v levém panelu. Až do výběru anotací z anotační sady nebude zobrazen panel *AnnotationsSet* ani *Annotations*.

GATE byl původně vyvinut v kontextu extrakce informací (IE). S GATE jsou distribuovány komponenty IE v mnoha jazycích, tvarech a velikostech. GATE je distribuován se systémem IE nazvaným ANNIE. ANNIE je založen na určitých algoritmech a jazyku JAPE. Tento jazyk pomocí regulárních výrazů vyhledává anotace v dokumentech. Komponenty ANNIE jsou včleněny do systému GATE přes jazykové zdroje. [4]

4.3.1 Tokenizer

Tokenizací korpusu je tento korpus rozložen na základní prvky, se kterými bude dále pracováno. Tokenizace korpusu ovlivňuje další práci s daty a jsou na ní závislé výsledky následných analýz. [18]

Po načtení a otevření dokumentu systémem GATE je prvním krokem při zpracování textu tokenizace tohoto dokumentu, což je proces segmentace textů do jednotek představujících slova, čísla, interpunkce a jiné elementy. Jsou vytvořeny dva typy anotace: *Token* pro slova, čísla, symboly, interpunkci a *SpaceToken* pro mezery a řídící znaky. Charakteristiky zpracované během tohoto procesu jsou typy tokenů (slovo, interpunkce, číslo, mezera, řídící znak apod.), jejich délky a ortografické charakteristiky (kapitálky, malá písmena, počáteční písmena apod.). Proces tokenizace může být modifikován změnou tokenizačních pravidel systému.

Tokenizační pravidla [4]

Pravidlo má levou (LSP) a pravou stranu pravidla (PSP). Levá strana je regulárním výrazem, který musí být propojen se vstupem. Pravá strana popisuje anotace, které jsou přidány k sadě anotací. Levá strana je oddělena od pravé znakem '>'. Následující operátory jsou použity na levé straně:

- | (nebo)
- * (0 nebo více výskytů)
- ? (0 nebo 1 výskyt)
- + (1 nebo více výskytů)

Pravá strana používá jako oddělovač ';' a má následující formát:

{LSP} > {Typ anotace}; {atribut 1} = {hodnota1};…; {atribut n} = {hodnota n}

Detaily o dostupných konstruktorech jsou dány v souboru tokenizeru (DefaultTokeniser.Rules).

Následující tokenizační pravidlo je pro slovo začínající velkým písmenem:

"UPPERCASE_LETTER" "LOWERCASE_LETTER"* >
Token;orth=upperInitial;kind=word;

Tento kód oznamuje, že sekvence musí začít velkým písmenem následovaným žádným nebo více malými písmeny. Tato sekvence bude anotována jako typ *Token*. Atribut *orth* (ortography) má hodnotu *upperInitial* a atribut *kind* má hodnotu *word*.

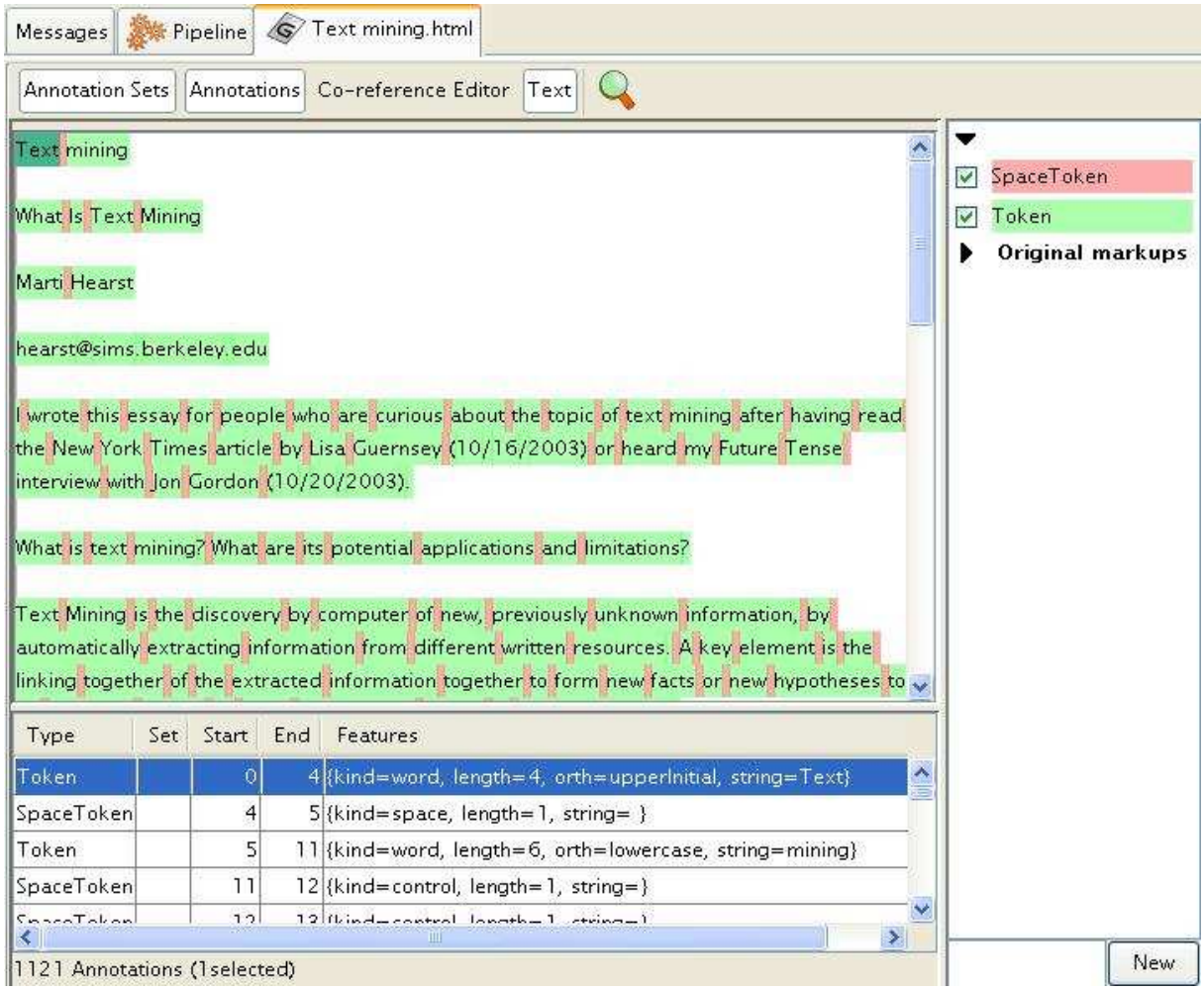
Typy tokenů

Ve výchozím nastavení pravidel jsou možné následující typy *Tokenu* a *SpaceTokenu* [4]:

1. **Word.** Word je definován jako sada velkých a malých písmen včetně spojovníku (ale ne jiné formy interpunkce). Word má atribut *orth*, pro který jsou definovány čtyři hodnoty:
    a. upperInitial – počáteční písmeno je velké, zbytek malým
    b. allCaps – všechna písmena jsou velká
    c. lowerCase – všechna písmena malá
    d. mixedCaps – kombinace velkých a malých písmen mimo výše zmíněných kategorií
2. **Number.** Number je definován jako kombinace číslic. Není zde žádné další členění tohoto typu.

3. **Symbol.** Zde jsou definovány dva typy symbolů: symbol měny (např. ‘$’) a symbol (např. ‘&’, ’^’). Jsou prezentovány číslem měny respektive jinými symboly.
4. **Punctuation.** Zde jsou definovány tři typy interpunkce: start_punctuation, end_punctuation a ostatní interpunkční znaménka („:“). Každé interpunkční znaménko je samostatným tokenem.
5. **SpaceToken.** Neviditelné znaky jsou rozděleny do dvou typů SpaceTokenu podle toho, zda se jedná o znaky pro mezery nebo řídící znaky. Jakákoli souvislá (a homogenní) sada mezer nebo řídících znaků je definována jako SpaceToken.

Výše popsané typy používá výchozí tokenizer. Pokud je potřeba, mohou se vytvořit alternativní tokenizery. Vhodný tokenizer se v tomto případě vybere při zpracování textu. Proces tokenizace je znázorněn na následujícím obrázku (obrázek 31).



Obrázek 31: Tokenizace. [Zdroj: vlastní]

Na obrázku 31 je vidět, že vzniklo 1121 anotací, přičemž jedna je vybrána. V poli sady anotací jsou vytvořeny dva typy anotace: Token a SpaceToken. Dále je zde uvedena počáteční

(Start) a koncová pozice (End). V charakteristice sady anotací jsou názorně vidět atributy a jejich hodnoty pro každý typ anotace. Anotace, která se vybere ze sady anotací, je též zvýrazněna přímo v textu. Na vybrané anotaci **Text** je vidět, že atribut délky *length* má hodnotu 4, atribut *kind* má hodnotu word, jedná se o počáteční velké písmeno (*upperInitial*) a v atributu *string* je hodnotou řetězec „Text“. Výsledný soubor tokenizace dokumentu je k dispozici na přiloženém CD pod názvem **Tokenizer.xml**.

4.3.2 Gazetteer

Seznamy gazetteer jsou textové soubory s jedním záznamem na řádku. Každý seznam obsahuje slova nebo sekvence slov představující klíčová slova nebo termíny, které označují určité znalosti. Indexní soubor je použit k definici různých seznamů v jednotlivých aplikacích. Při klasifikaci termínů do kategorií je pro každý seznam definován hlavní typ *major type* a volitelně vedlejší typ *minor type* v následujícím formátu [5]:

Terms.lst:Class:SubClass,

kde *Terms.lst* je název seznamu a *Class* i *SubClass* jsou řetězce. Indexní soubor je použit k vytvoření automatu, který rozpozná a klasifikuje klíčová slova. Když konečný stav automatu spojí jakýkoliv řetězec s termínem patřící do *Terms.lst*, je vyprodukován typ anotace *LookUp Annotation*, který rozdělí spojené sekvence. Znaky přidané k anotaci jsou *majorType* s hodnotou Class a *minorType* s hodnotou SubClass. Gramatika používá informace o procesu vyhledání v jejich pravidlech. Tyto informace používá k vyprodukování významných anotací. [5]

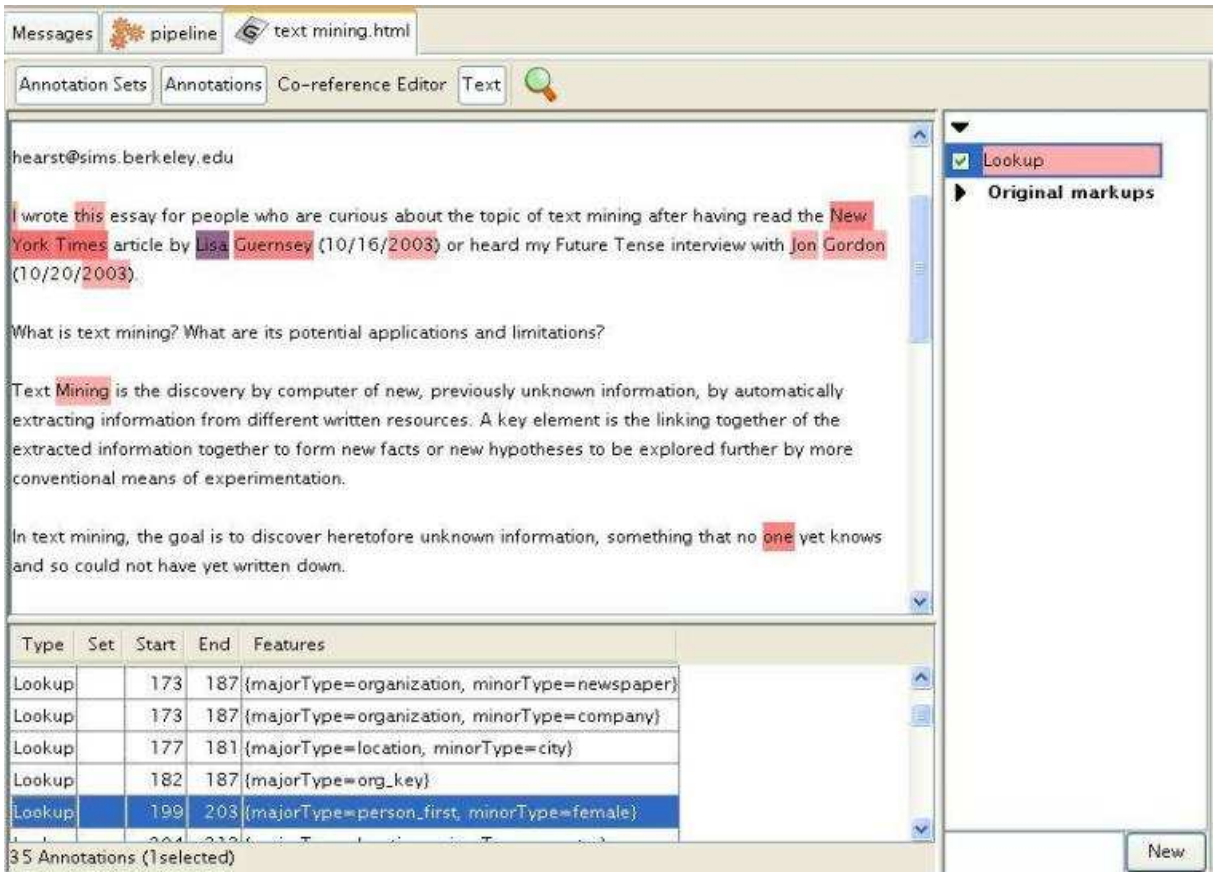
Indexní soubor (lists.def) je používán k přístupu k těmto seznamům. Pro každý seznam je určen hlavní a volitelně vedlejší typ. Tyto seznamy jsou sestaveny do konečných stavů. Textové tokeny, které jsou spojeny těmito stavy budou anotovány se znaky určenými hlavními a vedlejšími typy. Pravidla gramatiky poté určí typy, které budou identifikovány v jednotlivých událostech. Každý seznam gazetteer by měl být umístěn ve stejném adresáři jako indexní soubor. [4]

Když například potřebujeme identifikovat určitý den, potom bychom měli v gramatice určit vedlejší typ *day*, aby se porovnávali pouze informace o určitých dnech. Jestliže chceme identifikovat datum, musíme určit *date* jako hlavní typ. Poté lze pomocí tokenů anotovat informaci o datumu.

Zde je plný seznam parametrů výchozího Default Gazetteeru [5]:

- **listsUrl** URL souboru, který obsahuje seznam vzorů (obvykle lists.def)
- **encoding** Kódování znaků použitých při čtení seznamu vzorů.
- **gazetteerFeatureSeparator.** Znak použitý k přidání libovolného znaku k záznamům gazetteeru.
- **caseSensitive** Gazetteer by měl být během porovnání citlivý na písmena.
- **document** Dokument ke zpracování
- **annotationSetName** Název anotace, kde budou vytvořeny výsledné anotace Lookup.
- **WholeWordsOnly** Pokud je tato vlastnost nastavena na true, řetězec ve vstupním dokumentu bude porovnáván, jestliže je ohraničen jinými znaky než písmena, znaky pro mezery nebo kombinace znaků pro mezery (podle standardu Unicode).
- **LongestMatchOnly** Tento parametr je významný pouze tehdy, pokud seznam pro vyhledávání obsahuje vhodné předpony nebo jiné záznamy v seznamu. Ve výchozím nastavení (když je tento parametr nastaven na true) se pouze porovnává nejdelší záznam. Nastavení tohoto parametru na false zapříčiní, že gazetteer bere do úvahy všechny předpony.

Proces gazeteeru je vidět na následujícím obrázku (obrázek 32).



Obrázek 32: Gazeteer. [Zdroj: vlastní]

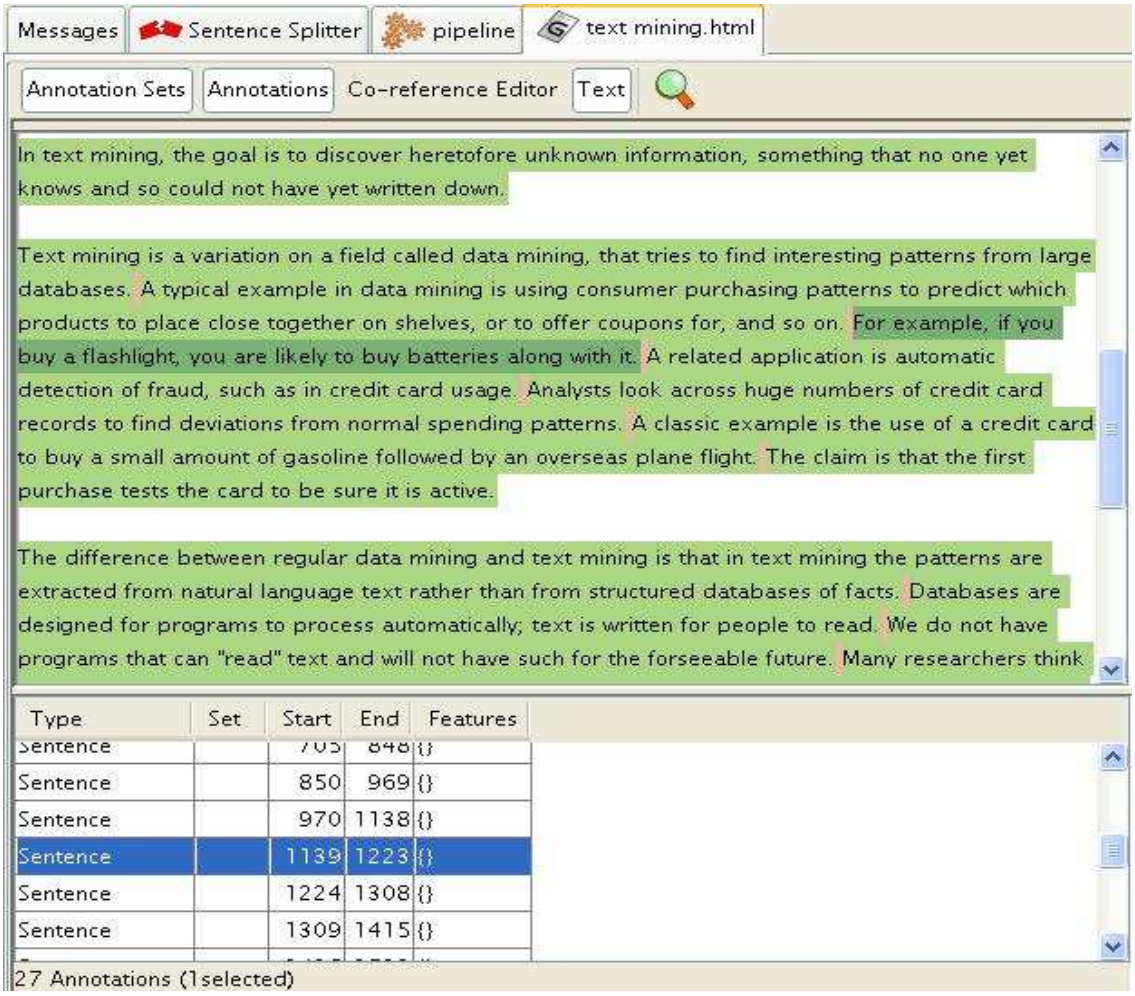
Ze vzorového textového souboru (příloha 1) vzniklo 35 anotací. Extrahovala se klíčová slova jako jména, názvy, města, čísla, datumy apod. V textu byla pro názorný případ vybrána jedna anotace **Lisa**. Gazetteer u vybrané anotace určil, že hlavním typem je osoba (person) a jako vedlejší typ určil, že jde o ženu (female). Podobně také u anotace **New York Times** určil hlavní typ organizaci (organization) a vedlejší typ noviny (newspaper). V případě gazeteeru extrakce u webové stránky a textového souboru byla totožná. Výsledný soubor procesu gazetteer je k dispozici na přiloženém CD pod názvem **Gazetteer.xml**.

4.3.3 Identifikace vět

Důležitým krokem po tokenizaci je proces identifikace vět, což je proces segmentace textu do vět. Tohle je v GATE implementováno přes množinu konečněstavových převodníků, které využívají soubor gramatiky. Proces využívá anotace vyprodukované dle postupů tokenizace (např. prezence interpunkčních znamének, řídících znaků a zkratek obsažených v dokumentu). Tento proces vytváří typ anotace *Sentence* a podle uvozovek indikuje, zda věta je nebo není citovaný výraz (obrázek 33). [5]

Identifikace vět je souhrn převodníků konečných stavů, který rozděluje text do vět nalezením hranic těchto vět. Tento proces používá seznam gazetteeru k rozlišení teček na konci vět od jiných znaků. [4]

Každá věta je anotována typem *Sentence*. Každý konec věty je anotován jako typ *Split*. Konec věty je rozlišen pomocí několika možných typů: „.“, „interpunkce“, „CR“ (zalomení řádku), nebo „multi“ (série interpunkčních znaků jako např. „?!?!“. Identifikace vět je proces aplikačně nezávislý. [4]

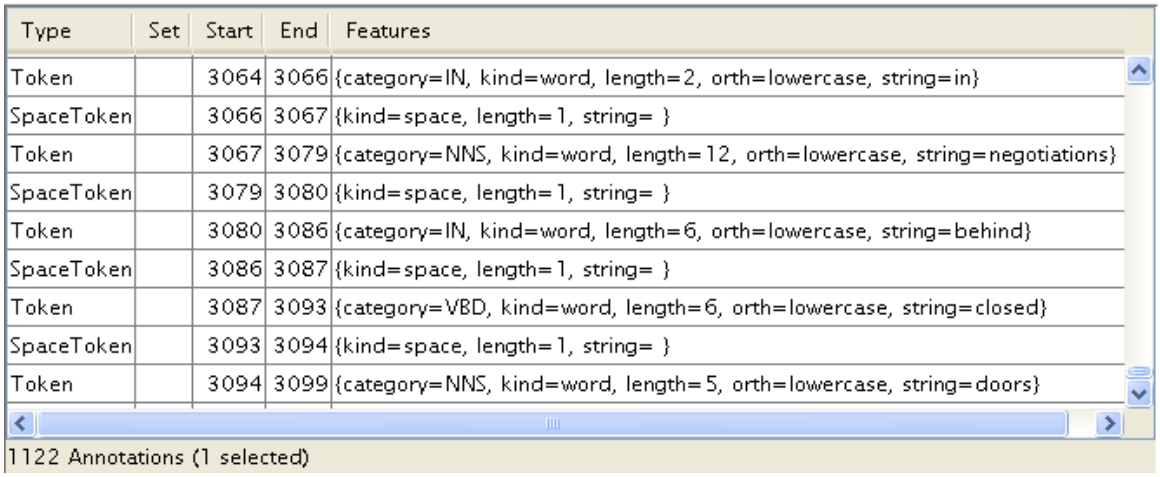


Obrázek 33: Identifikace vět. [Zdroj: vlastní]

Proces identifikace vět tedy rozdělí věty na typ *Sentence*. Proces identifikace vět následuje po procesu tokenizace. Identifikace vět totiž anotuje jen typ *Sentence* a typ *Split* pozná z procesu tokenizace. Tato aplikace je stejná jak pro webovou stránku, tak textový soubor. Výsledný soubor procesu identifikace vět je k dispozici na přiloženém CD pod názvem **Identifikace_vet.xml**.

4.3.4 Gramatické značení

Při zpracování textu je obvykle požadován proces gramatického značení. Je to proces asociace formy každého slova v textu. Tohle značení je v GATE implementováno modifikovanou verzí Brillova taggeru. Proces je závislý na jazyku a spoléhá (stejně jako Brill tagger) na dva zdroje – lexikon a transformační pravidla, která jsou již natrénována na korpus. Výchozí gramatické značení je v GATE již natrénováno na anglický jazyk. Tento proces nevytváří žádný nový typ anotace, ale obohacuje anotaci Token znakem *category*, který značí části vět (obrázek 34). [5]



Type	Set	Start	End	Features
Token		3064	3066	{category=IN, kind=word, length=2, orth=lowercase, string=in}
SpaceToken		3066	3067	{kind=space, length=1, string= }
Token		3067	3079	{category=NNS, kind=word, length=12, orth=lowercase, string=negotiations}
SpaceToken		3079	3080	{kind=space, length=1, string= }
Token		3080	3086	{category=IN, kind=word, length=6, orth=lowercase, string=behind}
SpaceToken		3086	3087	{kind=space, length=1, string= }
Token		3087	3093	{category=VBD, kind=word, length=6, orth=lowercase, string=closed}
SpaceToken		3093	3094	{kind=space, length=1, string= }
Token		3094	3099	{category=NNS, kind=word, length=5, orth=lowercase, string=doors}

1122 Annotations (1 selected)

Obrázek 34: Gramatické značení. [Zdroj: vlastní]

Pro spuštění procesu gramatického značení se musí do procesních zdrojů ještě přidat procesy tokenizeru a identifikace vět. Anotovány budou 3 typy: *Sentence*, *Token* a *SpaceToken*. Tyto typy anotací vyprodukují již zmíněný tokenizer a proces identifikace vět. Proces gramatického značení ovšem k těmto typům anotací doplní atribut *category*, který značí kategorie anotovaných slov.

Na příkladě věty v angličtině bude ukázáno použití procesu gramatického značení. Ve větě „The company acquired the building for manufacturing“ by slovo „building“ mělo značit podstatné jméno a „acquired“ sloveso. Tento proces má vyřešit základní nejasnosti anglického jazyka, kdy slovo „building“ značí jak podstatné jméno tak sloveso. [5]

Jedním ze způsobů získávání kanonické formy z každého slova dokumentu je aplikace lemmatiseru, která analyzuje každé slovo identifikací kořene slova a jeho přípony. Tento proces obohacuje anotaci tokenu o dva znaky – *root* pro kořen slova a *affix* pro příponu slova.

Pod pojmem lemmatizace je nazývána činnost, kterou převádíme slova na základní gramatický tvar (nazývaný též lemma). Způsob, kterým je tohoto stavu dosahováno je závislý především na použitém jazyku a jeho gramatice. Čeština patří mezi jazyky gramaticky velmi

ohebné, což ztěžuje práci lemmatizátoru, nástroje pro lemmatizaci. K lemmatizaci je možné přistupovat dvěma základními metodami, a to lemmatizací na základě výčtu všech tvarů daného slova (slovníku všech tvarů slov) nebo lemmatizaci na základě gramatiky jazyka (způsobů tvorby jednotlivých tvarů slov). [18]

Takto popsané zpracování je však pouze na úrovni slov, neřeší otázku mnohoznačnosti slov, či různých tvarů stejně psaných, avšak s rozdílným významem. Při používání jazyka člověk automaticky rozpozná správný význam z kontextu, pro strojové zpracování je však takovéto rozpoznání velmi složité, jak co do metod realizace, tak i výpočetně vzhledem k nárůstu možných kombinací. Přesný výběr adekvátního významu slova není dosud při automatickém zpracování uspokojivě vyřešen. [18]

Význam jednotlivých hodnot atributu *category* je v příloze 2 nazvané **Morfologické a syntaktické kategorie**. Výsledný soubor procesu gramatického značení je k dispozici na přiloženém CD pod názvem **Gramaticke_znaceni.xml**.

4.4 Shrnutí kapitoly

GATE je text miningový nástroj, který je poskytován zdarma ve formě open source pod licencí GNU. GATE pracuje se zdroji. Tento nástroj využívá tří typy zdrojů: jazykové, procesní a vizualizační. Jazykové zdroje jsou dokumenty, korpusy a ontologie. Procesní zdroje jsou jednotlivé komponenty extrakce informací tohoto systému. Mezi vizuálními zdroji jsou určité grafické prvky.

Systém extrakce informací je nazván ANNIE. ANNIE se skládá z těchto hlavních komponent: tokenizer, gazetteer, identifikace vět, gramatické značení. Dále jsou k dispozici procesy sémantického značení a ortomatcher. Tokenizací korpusu je tento korpus rozložen na základní prvky, se kterými se pracuje v dalších komponentách. Gazetteer je seznam s klíčovými slovy, která tato komponenta nalezne v textu. U ní se určuje kategorie a volitelně podkategorie. Například u jmen osob jako kategorii určí samotné jméno osoby a v podkategorii určí, zda jde o mužské či ženské jméno. Proces identifikace vět nalézá v textu znak tečky pro ukončení věty a podle toho pozná, kdy věta končí. Pokud se vyskytne tečka u řadové číslovky, tuto tečku jako konec věty nepovažuje. Gramatické značení produkuje značky slovního druhu jako anotace každého slova nebo symbolu. Tohle značení nejprve potřebuje procesy tokenizace a identifikace vět. Sémantickým značením se pomocí vytvořených pravidel v jazyku JAPE hledají vzory. Ortomatcher slouží k rozpoznání relací mezi entitami a klasifikuje neklasifikované entity pomocí již nalezených.

Všechny procesy extrakce informací lze aplikovat pro 10 jazyků. Pro ostatní jazyky včetně češtiny funguje jen proces tokenizace. Tohle oddělení slov pozná podle znaku pro mezeru mezi slovy, což je pro všechny jazyky stejné. Pro další procesy extrakce informací systém při zpracování textu nepodporovaného jazyka zahlásí chybu.

Nejvýznamnější charakteristika systému GATE je shrnuta v následující tabulce 7.

Tabulka 7: Charakteristika systému GATE. [Zdroj: vlastní]

Charakteristika	Popis
Možnosti text miningu	• Extrakce informací • Vyhledávání informací • Klasifikace textu • Zpracování přirozeného jazyka
Funkce extrakce informací	• Tokenizer • Gazetteer • Identifikace vět • Gramatické značení
Výhody	• Zdarma • Široká škála procesů extrakce informací • Intuitivní uživatelské prostředí
Nevýhody	• Obtížnější konfigurace • Nepodporuje české texty

5 Srovnávací studie

V této kapitole budou srovnány text miningové nástroje. Pro srovnávací studii byl vybrán komerční produkt Clementine a dva open source projekty RapidMiner a GATE. Srovnávací studie představuje vícekriteriální rozhodovací problém, který bude řešen pomocí analytického hierachického procesu (AHP). Jádrem této metody je Saatyho metoda párového porovnání. Pro řešení problému je použita výpočetní technika s příslušným softwarem, a to MS Office 2000, Matlab 6.5 a CDP 3.04.

Cílem je nalezení optimální alternativy na základě stanovených kritérií. Za alternativy byly zvoleny porovnávané text miningové nástroje: Clementine, RapidMiner a GATE. Nejprve bude v této studii přiblížena výchozí situace a definován rozhodovací problém. Dále je uveden postup řešení rozhodovací situace, a to včetně stanovení kritérií, stanovení vah, tvorby rozhodovací tabulky a její normalizace. Následuje popis práce v prostředí CDP.

V tabulce 8 je uvedena charakteristika porovnávaných nástrojů.

Tabulka 8: Charakteristika nástrojů. [Zdroj: vlastní]

Charakteristika	Clementine	RapidMiner	GATE
Verze	10.1	4.0	4.0
Výrobce	SPSS	Rapid-I	-
Licence	Komerční software	GNU General Public License	GNU General Public License
Programovací jazyk	CEMI	Java	Java
České prostředí programu	ne	ne	ne
Extrakce českých textů	částečně	částečně	částečně
Operační systém	Windows	Windows, UNIX	Windows, UNIX
Možnost tvorby nových pluginů	ne	ano	omezeně
Výhody	Velké množství jazyků k extrakci a velké množství podporovaných formátů	Zdarma, podpora uživatelsky definovaných operátorů	Zdarma, široká škála procesů extrakce informací
Nevýhody	Drahý, pouze 2 způsoby extrakce informací	Neextrahuje typy termínů, méně formátů, méně jazyků k extrakci, obtížnější konfigurace	Obtížnější konfigurace

5.1 Postup řešení

Při postupu řešení je potřeba nejdříve stanovit kritéria a stanovit váhu těmto kritériím. Poté se vytvoří rozhodovací tabulka, která bude normalizována.

5.1.1 Stanovení kritérií

Pro porovnání těchto tří nástrojů bylo vybráno 6 nejvýznamnějších kritérií:

1. **Cena**. Clementine je komerční systém, který nemá jednotnou cenu. Cena se pohybuje v závislosti na objednaném balíčku nebo multilicenci. Ostatní dva nástroje jsou šířeny pod licencí open source a jsou tudíž nabízeny zdarma. Kritérium cena je značeno od nejlevnějšího produktu (1) po nejdražší produkt (5).
2. **Technická náročnost**. Tímto kritériem se myslí obtížnost konfigurace nástroje a hardwareová náročnost. Clementine je z těchto tří porovnávaných variant nejnáročnější na hardware, ale zase dosahuje nejsnadnější konfigurace. RapidMiner není vůbec náročný na hardware, ale zato je nejobtížnější při konfiguraci. GATE je téměř nenáročný jak na hardware, tak na konfiguraci. Do úvahy se bere i počet nastavitelných parametrů.
3. **Počet možností při extrakci informací.** Tímto kritériem se rozumí počet možností nástroje při extrakci informací z dokumentů. Clementine v rámci extrakce informací a celého text miningu nabízí 2 uzly: Text Extraction a Text Link Analysis. Tedy extrakci konceptů a extrakci vzorů těchto konceptů. RapidMiner nabízí extrakci informací pomocí dotazů XPath a regulárních výrazů. GATE nabízí nejširší paletu komponent při extrakci informací: tokenizer, gazeteer, identifikaci vět, gramatické značení.
4. **Počet podporovaných formátů**. Clementine podporuje sedm formátů dokumentů: doc, xml, xls, txt, ppt, pdf, html. RapidMiner pouze tyto čtyři formáty: pdf, html, xml, txt. GATE má největší podporu formátů: xml, rtf, html, sgml, txt, email, doc, pdf. Všechny nástroje podporují tyto formáty html a txt. Pomocí těchto společných formátů byly zvoleny dokumenty pro porovnání.
5. **Počet jazyků k extrakci**. Všechny nástroje používají k extrakci defaultně angličtinu. Clementine dále podporuje extrakci z těchto jazyků: němčina, francouzština, italština,

španělština, portugalština a holandština. Po doinstalování doplňku Language Weaver lze navíc extrahovat z arabštiny, čínštiny a perštiny. Language Weaver tyto tři jazyky překládá do angličtiny a extraktor v tomto případě pracuje s angličtinou. RapidMiner nabízí pouze angličtinu a němčinu. GATE nabízí opět širokou paletu jazyků: němčina, francouzština, čínština, arabština, rumunština, ruština, hinduinština a jazyk cebuano, což je jazyk používaný na Filipínách.

6. **Hodnocení rozhraní**. Kritérium, kde je posuzována kvalita a intuitivnost grafického rozhraní. Posuzuje se zde i snadnost použití. Clementine je intuitivní nástroj ovládaný pomocí uzlů. RapidMiner i GATE jsou naprogramovány programovacím jazykem Java. RapidMiner byl původně ovládán jen přes příkazový řádek, ale přesto má vlídné grafické prostředí, i když pro běžného uživatele jsou zde mírné komplikace při nastavení.

Cena, technická náročnost a snadnost používání jsou „oznámkována“ 1 až 5 a jsou to minimalizační kritéria. Ostatní kritéria jsou maximalizační.

5.1.2 Stanovení vah

Pro stanovení vah kritérií byla použita metoda Saatyho matic. Tato metoda vychází ze dvou kroků. Nejprve se zjistí preferenční vazby dvojic kritérií uspořádaných v tabulce (čtvercové), v jejíž řádcích a sloupcích jsou zapsána kritéria ve stejném pořadí. Velikost preference je vyjádřena podle následující tabulky:

Tabulka 9: Základní stupnice preferencí. Zdroj [13]

Hodnotící stupeň	Porovnání prvků x a y	Vysvětlení
1	x je stejně důležité jako y	Oba prvky přispívají stejnou měrou k výsledku.
2	x je slabě důležitější než y	První prvek je slabě důležitější než druhý.
3	x je mírně důležitější než y	Zkušenosti a úsudek mírně preferují první prvek před druhým.
4	x je více důležitý než y	O něco silnější preference než předchozí.
5	x je důležitější než y	Silná preference prvního prvku před druhým.
6	x je mnohem více důležitější než y	O něco silnější preference než předchozí.
7	x je silně důležitější než y	Velmi silná preference prvního prvku před druhým.
8	x je velmi silně důležitější než y	O něco silnější preference než předchozí.
9	x je extrémně důležitější než y	Skutečnosti upřednostňující první prvek před druhým mají nejvyšší stupeň průkaznosti.

Z tabulky vyplývá, že na hlavní diagonále této matice preferencí kritérií budou jedničky.

Vždy se porovnává kritérium v řádku s kritériem ve sloupci. Pokud tuto matici kritérií označíme S, pak prvky pod hlavní diagonálou budou dány vztahem [13]:

$$s_{ij} = \frac{1}{s_{ji}} \qquad (3)$$

pro všechna i a j. Prvky s_{ij} vyjadřují odhad podílů vah kritérií v_i a v_j, takže $s_{ij} \cong v_i/v_j$.

Se znalostí Saatyho matic můžeme nyní stanovit váhy kritérií exaktními nebo aproximativními postupy. Byla zvolena exaktní metoda založená na výpočtu vlastního vektoru matice relativních důležitostí, neboť je považována za nejpřesnější. Je ovšem složitější a při větším rozsahu vyžaduje použití počítače. Pro výpočet normovaných vah (v intervalu <0;1>) byl využit programový prostředek Matlab 6.5. Zdrojový kód je uveden jako Příloha 3. Další výhodou této metody je existence jisté kontroly, že byly matice sestaveny správně. Tu zajišťuje tzv. index konzistence KI. Pokud platí, že KI ≤0,1, pak byla matice sestavena správně. Jednotlivé Saatyho matice preferencí s výslednými váhami a KI jsou znázorněny v následujících tabulkách:

Tabulka 10: Saatyho matice preferencí 1. [Zdroj: vlastní]

	Uživatelské	Ostatní	Váhy	KI
Uživatelské	1,00	3,00	0,75	0
Ostatní	0,33	1,00	0,25	

Tabulka 11: Saatyho matice preferencí 2. [Zdroj: vlastní]

	Počet funkcí	Počet podporovaných formátů	Počet jazyků k extrakci	Hodnocení rozhraní	Váhy	KI
Počet funkcí	1,00	3,00	5,00	7,00	1,00	0,039
Počet podporovaných formátů	0,33	1,00	3,00	5,00	0,33	
Počet jazyků k extrakci	0,20	0,33	1,00	3,00	0,20	
Hodnocení rozhraní	0,14	0,20	0,33	1,00	0,14	

Tabulka 12: Saatyho matice preferencí 3. [Zdroj: vlastní]

	Cena	Technická náročnost	Váhy	KI
Cena	1,00	0,20	0,1667	0
Technická náročnost	5,00	1,00	0,8333	

Z výše uvedených tabulek je vidět, že žádný z koeficientů konzistence nepřekročil hodnotu 0,1. Matice a normované váhy jsou tedy určeny správně.

5.1.3 Rozhodovací tabulka

Výsledná rozhodovací tabulka má pak tuto podobu:

Tabulka 13: Rozhodovací tabulka. [Zdroj: vlastní]

		Uživatelské				Ostatní	
	Váhy:	0,75				0,25	
		Počet funkcí extrakce informací	**Počet podporovaných formátů**	**Počet jazyků k textové analýze**	**Hodnocení rozhraní**	**Cena**	**Technická náročnost**
Nástroj	Váhy:	0,565	0,2622	0,1175	0,0553	0,8333	0,1667
Clementine		2	7	10	2	5	2
RapidMiner		2	4	2	5	1	3
GATE		4	8	10	3	1	1

5.1.4 Normalizace dat

Normalizace dat se provádí pro odstranění vlivu jednotek a velikostí dat. Rozhodovací problém bude řešen v programovém prostředku CDP, kde určujeme váhy kritérií v intervalu <0;100>. 0 je nejhorší a 100 je nejlepší skóre. Normalizace dat je tedy dána vztahem [13]:

$$x_{ij}^{*} = \frac{x_{ij}}{\sum_{i} x_{ij}} \cdot 100 \qquad (4)$$

pro všechna i a j, kde i je počet řádků, j je počet sloupců, x_{ij} jsou původní hodnoty a x^{*}_{ij} jsou normalizované hodnoty. Váhy již byly znormovány do intervalu <0;1>, stačilo proto je pouze vynásobit číslem 100. Výpočet byl proveden v programovém prostředku MS Excel 2003. Výsledná normovaná rozhodovací tabulka má pak tento tvar:

Tabulka 14: Normovaná rozhodovací tabulka. [Zdroj: vlastní]

		Uživatelské				Ostatní	
	Váhy:	0,75				0,25	
		Počet funkcí extrakce informací	**Počet podporovaných formátů**	**Počet jazyků k textové analýze**	**Hodnocení rozhraní**	**Cena**	**Technická náročnost**
Nástroj	Váhy:	0,565	0,2622	0,1175	0,0553	0,8333	0,1667
Clementine		25	37	45	50	9	33
RapidMiner		25	21	9	13	45	25
GATE		50	42	45	38	45	42

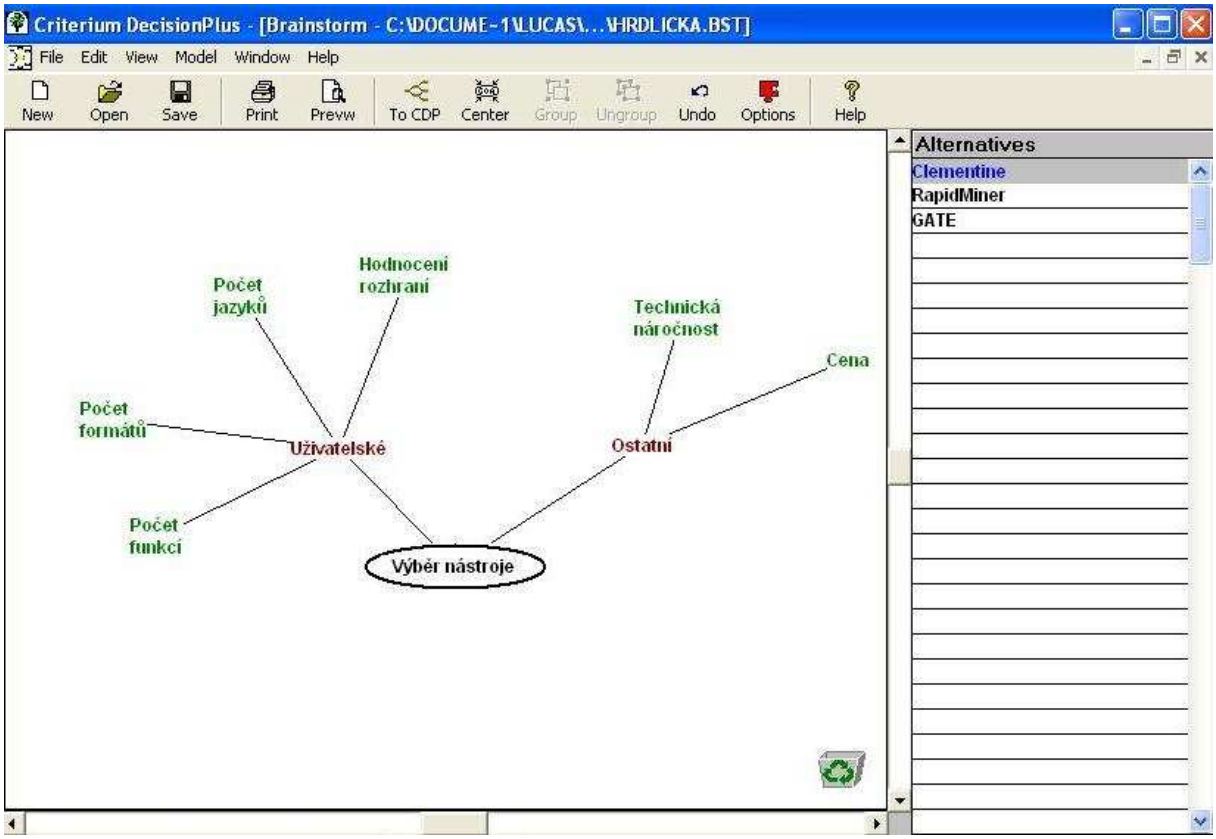
5.2 Práce v prostředí CDP

Pro řešení rozhodovacího problému byl použit programový prostředek CDP (Criterium Decision Plus) verze 3.04. Tento program na podporu rozhodování vyvinula a nabízí firma InfoHarvest, Inc. Program je založen na rozdělení do více hierarchických úrovní, který nabízí i přímé bodování s vlastní volitelnou stupnicí. Výsledky jsou rovněž zobrazovány jak v číselné, tak i v grafické podobě.

Criterium Decision Plus je k dispozici v anglickém jazyce a lze si vyzkoušet jeho Student verzi, která je ke stažení na stránkách firmy InfoHarvest [9]. Student verze není nijak časově omezena, ale lze použít pouze 20 bloků (kritérií a variant).

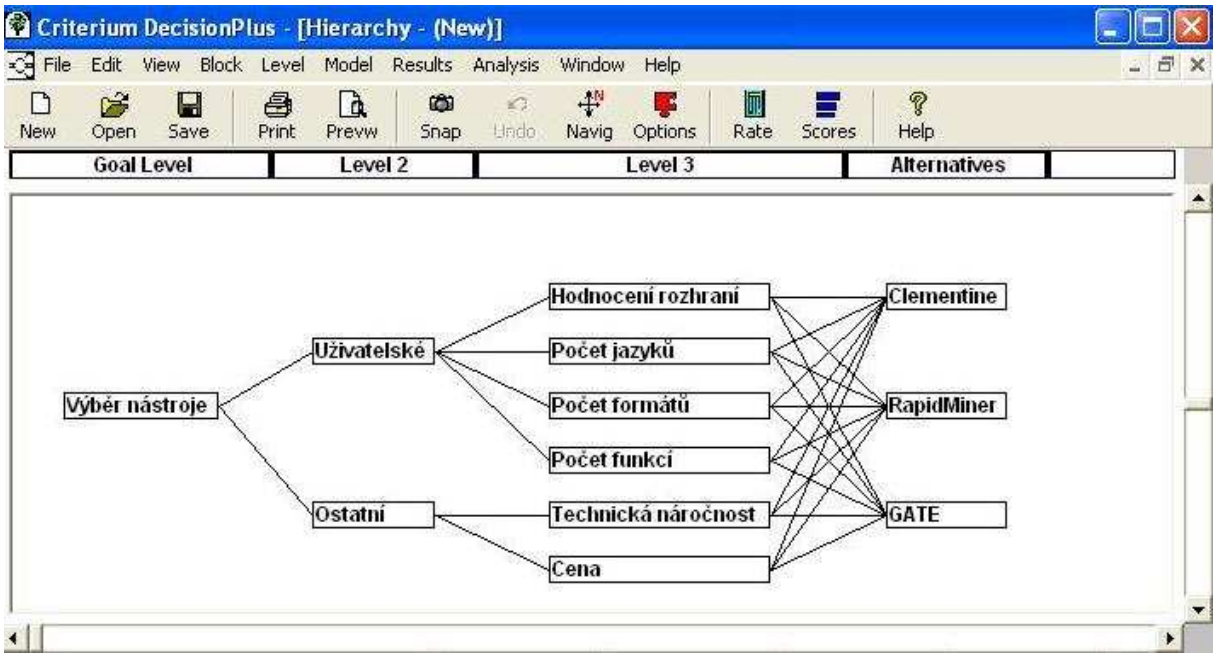
5.2.1 Definice rozhodovacího problému

Nejprve byl proveden brainstorming, což je takové rychlé zaznamenání všech nápadů. Cílem bylo nalézt rozhodovací kritéria a jejich optimální počet. Z nabídky File byl vybrán příkaz New pro vytvoření prázdného souboru. Poté byly zaznamenány jednotlivá kritéria vybraných alternativ. Výsledek brainstormingu je uveden na následujícím obrázku 35.



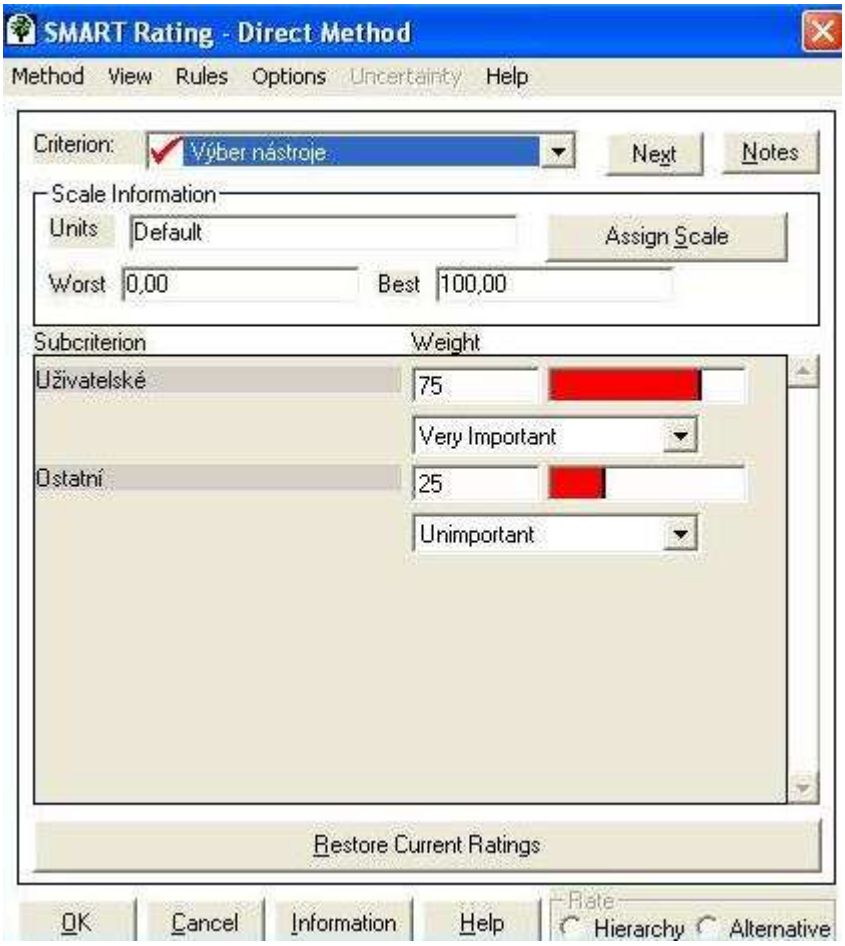
Obrázek 35: Brainstorming. [Zdroj: vlastní]

Z nabídky Model byl vybrán příkaz Generate hierarchy pro převedení schématu brainstormingu na hierarchický model. Výstup je znázorněn na dalším obrázku (obrázek 36).



Obrázek 36: Hierarchický model rozhodovacího procesu. [Zdroj: vlastní]

Dále byly nastaveny váhy jednotlivých kritérií pomocí příkazu Rate Subcriteria z nabídky Block. Příklad nastavení skóre kritérií je uveden na obrázku 37.

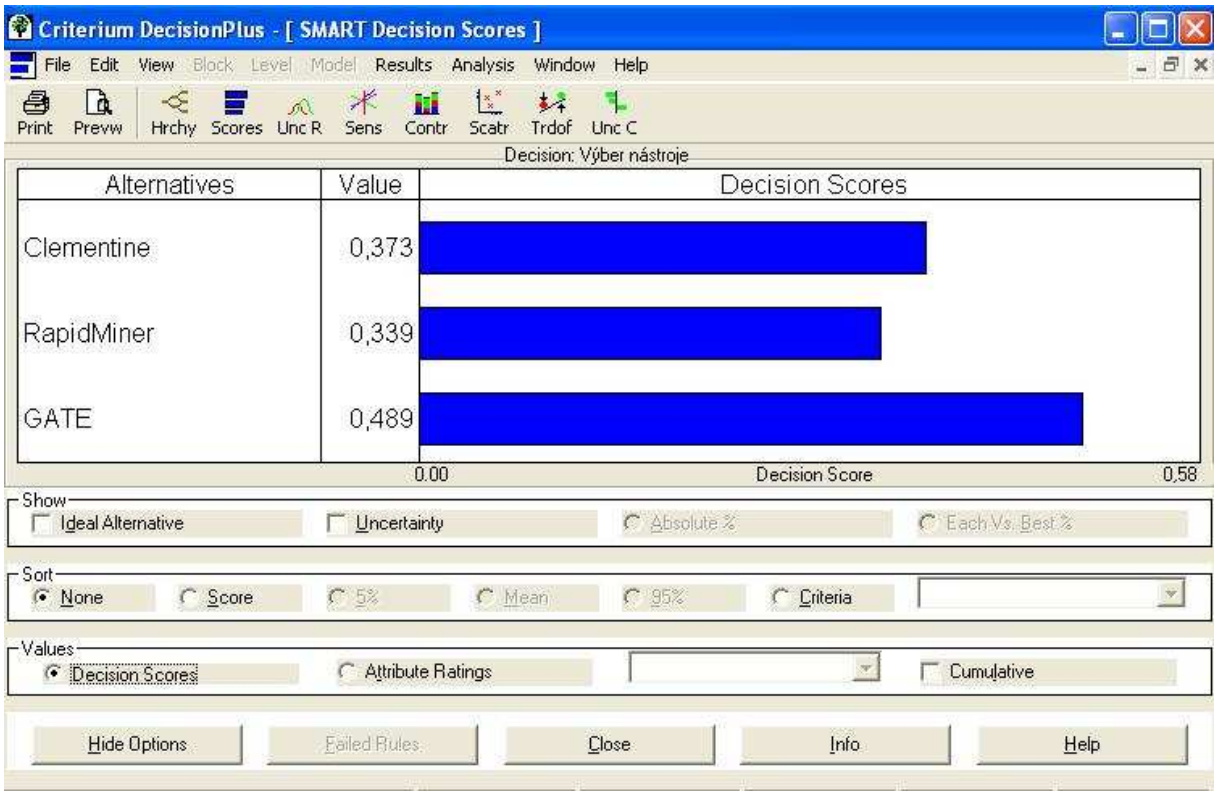


Obrázek 37: Příklad nastavení skóre kritérií. [Zdroj: vlastní]

5.2.2 Zobrazení výsledků

Program CDP umožňuje vytvořit celou škálu grafického zobrazení, jakožto výstupní odezvu na zadané vstupní hodnoty. Tyto grafické výstupy slouží k zobrazení optimálního výsledku rozhodnutí podle zadaných a předem nadefinovaných vstupních dat.

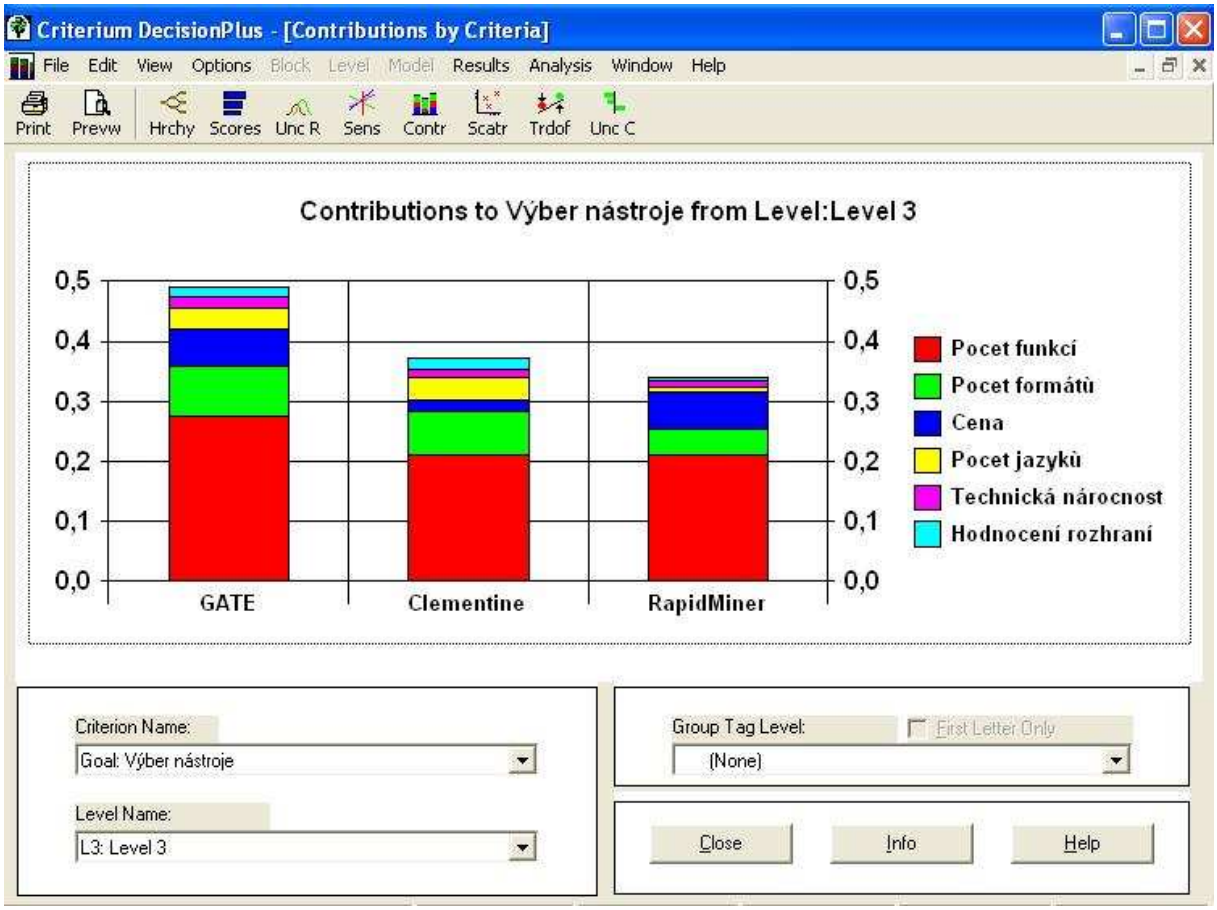
Pro zobrazení výsledků byl nejprve použit příkaz Decision Scores z nabídky Results. Následující obrázek 38 ukazuje výsledné ohodnocení alternativ.



Obrázek 38: Výsledné skóre alternativ. [Zdroj: vlastní]

Z obrázku je patrné vyhodnocení optimálního výběru text miningového nástroje podle zvolených a následně nastavených vah parametrů. V levé části jsou patrné jednotlivé text miningové systémy, které byly alternativní veličinou. Hodnota představuje poměr hodnoty, vytvořené ze zadaných vah jednotlivým parametrům. V tomto případě hodnota 0 nejméně odpovídá pravděpodobnosti výběru daného systému a hodnota 1 odpovídá nejvyšší pravděpodobnosti výběru daného systému. Čím více se bude tato hodnota blížit hodnotě 1, je pravděpodobnější, že systém splňuje požadovaná kritéria výběru podle nastavených vah jednotlivých parametrů. Nejlépe byla ohodnocena alternativa číslo tři GATE, se skórem 0,489. Druhý skončil Clementine se skórem 0,373 a poslední alternativou je RapidMiner se skórem 0,339.

Další obrázek znázorňuje, v jakém poměru přispívají jednotlivá kritéria na celkové skóre alternativ (obrázek 39).



Obrázek 39: Podíl kritérií na celkové skóre alternativ. [Zdroj: vlastní]

Na obrázku je vidět, že nejlepších výsledků dosahuje nástroj GATE. Nejdůležitější z uživatelských kritérií je kritérium počtu funkcí, a proto byla u všech nástrojů nastavena pro toto kritérium největší váha. V tomto kritérii vítězí GATE. GATE je v počtu podporovaných formátů a počtu extrahovaných jazyků téměř nastejno se systémem Clementine. RapidMiner v uživatelských kritériích dosahuje nejhorších výsledků. Jen hodnocení kritéria ceny je přijatelné, protože spolu se systémem GATE jsou nabízeny zdarma. Technická náročnost a hodnocení rozhraní mají přiděleny menší váhy, a proto se podílejí na celkovém hodnocení nejméně.

6 Závěr

Cílem diplomové práce bylo porovnání vybraných text miningových nástrojů. Pro porovnání byly zvoleny dva open source produkty s komerčním profesionálním produktem. Váha kritéria cena byla stanovena minimální, aby se největší důraz dával funkcím, které nástroje nabízejí v oblasti text miningu. Analýza těchto nástrojů byla soustředěna na extrakci informací. Text miningové nástroje byly analyzovány na stejných vzorových dokumentech, webovské stránce a textovém souboru. Jako jazyk vzorového dokumentu byla zvolena angličtina. Angličtina je u všech nástrojů plně podporována pro proces extrakce informací. Konečné porovnání bylo provedeno metodou AHP. Součástí práce bylo též vysvětlení základních pojmů této oblasti

Prvním analyzovaným nástrojem byl komerční nástroj Clementine. Jde především o data miningový nástroj, který pro potřeby text miningu umožňuje pomocí rozhraní CEMI doinstalování dvou uzlů: Text Extraction a Text Link Analysis. Jde o modelovací uzly, které z textu extrahují významné koncepty. Uzel Text Link Analysis extrahuje navíc významné vztahy mezi těmito koncepty.

Silným prvkem tohoto nástroje je extrakce konceptů včetně jejich frekvence a typů konceptů. Typy konceptů tento nástroj pozná především podle vnitřních slovníků. Výhodou tohoto nástroje je snadná konfigurace a příjemné uživatelské prostředí, podpora více typů formátů dokumentů a extrakce ze značného množství jazyků. Mezi největší nevýhodu lze zařadit cenu. Jde především o data miningový nástroj a jen pro potřeby text miningu je tato cena nepřijatelná.

Druhým analyzovaným nástrojem byl RapidMiner. Jde opět v první řadě o data miningový nástroj, kdy pro potřeby text miningu se musí doinstalovat doplněk. V tomto případě jde ale o nástroj šířený pod licencí open source čili zdarma. RapidMiner nabízí dva způsoby extrakce informací – pomocí dotazů XPath a regulárních výrazů. V případě dotazů XPath se jedná o extrakci ze strukturovaných dokumentů XML nebo HTML. V případě regulárních výrazů tato podmínka odpadá.

Mezi výhody tohoto nástroje lze zařadit, že tento nástroj je zdarma a v případě znalosti programovacího jazyka Java nabízí možnost tvorby vlastních operátorů. Extrakce pomocí dotazů XPath či regulárních výrazů ze strukturovaných dokumentů je možná i z českých textů. Extrahovat obsah celého dokumentu lze jen pro angličtinu a němčinu. Mezi nevýhody

lze uvést, že se neextrahují typy termínů, ale jen výskyt slov a k dispozici je jen omezená statistika extrahovaných termínů.

Třetím nástrojem byl analyzován nástroj GATE. V tomto případě jde o text miningový nástroj, který je nabízen zdarma pod licencí GNU. Systém extrakce informací je nazván ANNIE. ANNIE se skládá z těchto komponent: tokenizer, gazetteer, identifikace vět, gramatické značení, sémantické značení a ortomatcher. Tokenizací korpusu je tento korpus rozložen na základní prvky, se kterými se pracuje s dalšími komponentami. Gazetteer je seznam s klíčovými slovy, které tato komponenta nalezne v textu. Proces identifikace vět nalézá v textu znak tečky pro ukončení věty. Gramatické značení produkuje morfologické a syntaktické kategorie.

Mezi výhody tohoto systému patří, že je poskytován zdarma, podporuje mnoho jazyků k extrakci a poskytuje široké spektrum funkcí v oblasti text miningu. Nástroj je silný především v oblasti anotování korpusu a extrakce informací. Za výhodu lze také považovat intuitivní uživatelské prostředí a pro náročné uživatele vytvoření nebo alespoň pozměnění již zmíněných procesů. Mezi nevýhody lze považovat, že se nedají extrahovat typy tokenů z českých textů a podporuje menší počet formátů dokumentů.

V následující části diplomové práce byly tyto nástroje porovnány metodou analytického hierarchického procesu. Pro porovnání bylo zvoleno 6 kritérií: počet funkcí extrakce informací, počet podporovaných formátů, počet jazyků k extrakci, hodnocení rozhraní, cena a technická náročnost. Těmto kritériím byly následně přiděleny váhy pomocí metody Saatyho matic a využití softwaru Matlab 6.5. Poté proběhlo sestavení rozhodovací matice a následné použití v programu Criterium Decision Plus.

Stanovené cíle byly splněny. Po definici základních pojmů, analýze jednotlivých systémů a porovnání těchto nástrojů byla stanovena optimální alternativa. Touto alternativou byl zvolen systém GATE. Jde o text miningový systém, který nabízí nejvíce možností v oblasti text miningu.

Použitá literatura

[1] *Automatické referování* [online]. [cit. 2008-8-2]. Dostupné z: <https://is.muni.cz/auth/th/110477/ff_b/automaticke_referovani.doc>

[2] BERRY, Michael J. A., LINOFF, Gordon S. *Data Mining Techniques: for marketing, sales and customer support*. John Wiley & Sons, 1997. 454 s. ISBN 80-251-0952-6.

[3] *Členění data miningových úloh* [online]. [cit. 2008-8-2]. Dostupné z: < http://datamining.xf.cz/view.php?cisloclanku=2002102801>

[4] *Developing Language Processing Components with GATE version 4*.0 [online]. [cit. 2008-8-2]. Dostupné z: < http://gate.ac.uk/sale/tao/tao.pdf >

[5] DO PRADO, Hercules Antonio., EDILSON. *Emerging Technologies of Text Mining: Techniques and Applications,* Idea Group Reference, 2007. 358 s. ISBN 1599043734.

[6] *Extensible Markup Language (XML) 1.0 (Fifth Edition)* [online]. [cit. 2008-8-2]. Dostupné z: < http://www.w3.org/TR/REC-xml/>

[7] FAYYAD, Usama; PIATETSKY-SHAPIRO, Gregory; SMYTH, Padhraic. *From data mining to knowledge discovery in databases*. American Association of Artificial Intelligence, 1996

[8] FELDMAN, Ronen, SANGER, James. *Text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge University Press, 2007. 410 s. ISBN 978-0-521-83657-9

[9] *InfoHarvest Inc., maker of decision analysis tools including Criterium Decision Plus (CDP)* [online]. [cit. 2008-8-2]. Dostupné z: <http://www.infoharvest.com/ihroot/index.asp >

[10] JACKSON, Peter, MOULINIER, Isabelle. *Natural Language Processing for Online Applications: Text Retrieval, Extraction and Categorization*, John Benjamins Publishing, 2000. 218 s. ISBN 1588112500.

[11] *JISC: Text Mining* [online]. [cit. 2008-8-2]. Dostupné z: < http://www.jisc.ac.uk/publications/publications/bptextminingv2.aspx>

[12] MAŘÍK, Vladimír. *Umělá inteligence* 4. Praha: Academia, 2003. ISBN 80-200-1044-0.

[13] RAMÍK, Jaroslav. *Vícekriteriální rozhodování – Analytický hierarchický proces (AHP).* 1. vyd. Karviná: Slezská univerzita v Opavě, Obchodně-podnikatelská fakulta, 1999. 216 s. ISBN 80-7248-047-2.

[14] *Rapid - I - RapidMiner Community Edition* [online]. [cit. 2008-8-2]. Dostupné z: < http://rapid-i.com/content/blogcategory/38/69/>

[15] *RapidMiner 4.0: User Guide, Operator Reference, Developer Tutorial* [online]. [cit. 2008-8-2]. Dostupné z: < http://downloads.sourceforge.net/yale/rapidminer-4.0-tutorial.pdf>

[16] SIRMAKESSIS, Spiros. *Text Mining and its Application*, Springer: New York, 2000. 968 s. ISBN 3-540-20238-2.

[17] SINGH, M., Paul. *Practical Handbook of Internet Computing*, Chapman & Hall/CRC, 2004. 1144 s. ISBN 1584883812.

[18] ŠŤOVÍČEK, Karel. *Komparace nástrojů pro zpracování dokumentů*. Univerzita Pardubice, 2008. 63 s. ISBN

[19] *Text and web mining* [online]. [cit. 2008-8-2]. Dostupné z: < http://www.ailab.si/blaz/predavanja/ozp/gradivo/2003-mladenic-soleunet.pdf>

[20] *The Digital Library* [online]. [cit. 2008-8-2]. Dostupné z: < http://www.dcs.shef.ac.uk/~valyt/download/greenstone-gate.pdf>

[21] *The Word Vector Tool and the RapidMiner Text Plugin: User Guide, Operator Reference, Developer Tutorial* [online]. [cit. 2008-8-2]. Dostupné z: < http://downloads.sourceforge.net/yale/rapidminer-4.0-tutorial.pdf>

[22] *Text mining a jeho možnosti (aplikace)* [online]. [cit. 2008-8-2]. Dostupné z: < http://www.fi.muni.cz/usr/jkucera/pv109/2003p/xsedlac5.htm>

[23] *Text Mining for Clementine 4.0 User's Guide* [online]. [cit. 2008-8-2]. Dostupné z: < http://www.spss.com/se/support/pdf/TextMiningforClementine.pdf>

[24] *Text Mining with Information Extraction* [online]. [cit. 2008-8-2]. Dostupné z: < http://www.cs.utexas.edu/users/ml/papers/discotex-melm-03.pdf>

[25] *Úvod do korpusové lingvistiky* [online]. [cit. 2008-8-2]. Dostupné z: < http://nlp.fi.muni.cz/nlp/aisa/NlpCz/Uvod_do_korpusove_lingvistiky.html>

[26] WEISS, Sholom M. *Text Mining: Predictive methods for analyzing unstructured information*, Springer: New York, 2005. 237 s.ISBN 0-387-95433-3.

[27] *What is Text Mining?* [online]. [cit. 2008-8-2]. Dostupné z: < http://people.ischool.berkeley.edu/~hearst/text-mining.html>

[28] WITTEN, Ian H., FRANK, Eibe. *Data mining: Practical machine learning tools and techniques*, Morgan Kaufmann: San Francisco, 2005. 525 s. ISBN 0-12-088407-0.

Seznam obrázků

Seznam tabulek

Seznam příloh

Seznam použitých zkratek

AHP	Analytický hierarchický proces
DM	Data Mining
CDP	Criterium Decision Plus
GATE	General Architecture for Text Engineering
GUI	Graphical User Interface
IE	Extrakce informací
IR	Vyhledávání informací
JRE	Java Runtime Environment
KDD	Dobývání znalostí z databází
NLP	Zpracování přirozeného jazyka
TM	Text Mining
WVTool	Word Vector Tool
XML	eXtensible Markup Language

Přílohy

Příloha 1: Zdrojový text

What Is Text Mining

Marti Hearst

hearst@sims.berkeley.edu

I wrote this essay for people who are curious about the topic of text mining after having read the New York Times article by Lisa Guernsey (10/16/2003) or heard my Future Tense interview with Jon Gordon (10/20/2003).

What is text mining? What are its potential applications and limitations?

Text Mining is the discovery by computer of new, previously unknown information, by automatically extracting information from different written resources. A key element is the linking together of the extracted information together to form new facts or new hypotheses to be explored further by more conventional means of experimentation.

In text mining, the goal is to discover heretofore unknown information, something that no one yet knows and so could not have yet written down.

Text mining is a variation on a field called data mining, that tries to find interesting patterns from large databases. A typical example in data mining is using consumer purchasing patterns to predict which products to place close together on shelves, or to offer coupons for, and so on. For example, if you buy a flashlight, you are likely to buy batteries along with it. A related application is automatic detection of fraud, such as in credit card usage. Analysts look across huge numbers of credit card records to find deviations from normal spending patterns. A classic example is the use of a credit card to buy a small amount of gasoline followed by an overseas plane flight. The claim is that the first purchase tests the card to be sure it is active.

The difference between regular data mining and text mining is that in text mining the patterns are extracted from natural language text rather than from structured databases of facts. Databases are designed for programs to process automatically; text is written for people to read. We do not have programs that can "read" text and will not have such for the forseeable

future. Many researchers think it will require a full simulation of how the mind works before we can write programs that read the way people do.

There are programs that can, with reasonable accuracy, extract information from text with somewhat regularized structure. For example, programs that read in resumes and extract out people's names, addresses, job skills, and so on, can get accuracies in the high 80 percents.

The fundamental limitations of text mining are first, that we will not be able to write programs that fully interpret text for a very long time, and second, that the information one needs is often not recorded in textual form. If I tried to write a program that detected when a where a new word came into existence and how it spread by analyzing web pages, I would miss important clues relating to usage in spoken conversations, email, on the radio and TV, and so on. Similarly, If I tried to write a program that processes published documents in order to guess what will happen to a bill in Washington DC, I would fail because most of the action still happens in negotiations behind closed doors.

Příloha 2: Morfologické a syntaktické kategorie

CC	koordinační spojka
CD	základní číslovka
DT	determinant
EX	existenciální THERE
FW	cizí slovo
IN	předložka nebo souřadící spojka
JJ	přídavné jméno
JJR	přídavné jméno – komparativ
JJS	přídavné jméno – superlativ
LS	značka položek seznamu
MD	způsob
NN	podstatné jméno – jednotné číslo
NNS	podstatné jméno – množné číslo
NP	vlastní jméno – jednotné číslo
NPS	vlastní jméno – množné číslo
PDT	výraz před členem
POS	přivlastňovací tvar
PP	osobní zájmeno
RB	příslovce
RBR	příslovce –komparativ
RBS	příslovce – superlativ
RP	částice
SYM	symbol
TO	literál TO
UH	citoslovce
VBD	sloveso – minulý čas
VBG	sloveso – gerundium nebo příčestí přítomné
VBN	sloveso – minulé příčestí
VBP	sloveso – mimo 3. osobu jednotného čísla v přítomném čase
VB	sloveso – infinitiv
VBZ	sloveso – 3. osoba jednotného čísla v přítomném čase
WDT	wh- determinant
WP	wh- zájmeno (what, who, whom)
WRB	wh- příslovce (how, where, why)

Příloha 3: Zdrojový kód pro výpočet vah kritérií

```
% Popis programu: stanovení vah kritérií ve třech maticích.
% M je počet kriterií
M1=2
M2=3
M3=2

% definování saatyho matic párových porovnání kritérií
S1=[1 3;
    1/3 1 ]

S2=[1 1/7 1/3;
    7 1 5;
    3 1/5 1 ]
S3=[1 1/5;
    5 1]
 % vypočet maximální vlastní hodnoty lambda max matice S a jejího
% vlastního vektoru
[vektor_sigma1, lambda_S1]=eig(S1)
[vektor_sigma2, lambda_S2]=eig(S2)
[vektor_sigma3, lambda_S3]=eig(S3)
% max_lambda je největší vlastní číslo z lambda_S
max_lambda1=norm(lambda_S1)
max_lambda2=norm(lambda_S2)
max_lambda3=norm(lambda_S3)
% definování vlastního vektoru příslušného max_lambda jako prvku (1,1)
% matice lambda_S
vektor_max1=vektor_sigma1(1:M1,1)
vektor_max2=vektor_sigma2(1:M2,1)
vektor_max3=vektor_sigma3(1:M3,1)

% výpočet normovaného váhového vektoru z vektoru vektor_max, tj. hodnoty
% od 0 do 1
for i=1:1:M1
    a(i)=vektor_max1(i,1)/sum(vektor_max1)
end
vahvekt_W1=a(1:M1)'

for i=1:1:M2
    a(i)=vektor_max2(i,1)/sum(vektor_max2)
end
vahvekt_W2=a(1:M2)'

for i=1:1:M3
    a(i)=vektor_max3(i,1)/sum(vektor_max3)
end
vahvekt_W3=a(1:M3)'

%index konzistence by měl být <=0.1
%určuje jak byla sestavena matice S
ind_nekonz1=(max_lambda1-M1)/(M1-1)
ind_nekonz2=(max_lambda2-M2)/(M2-1)
ind_nekonz3=(max_lambda3-M3)/(M3-1)
```

Příloha 4: Obsah přiloženého CD

Popis jednotlivých adresářů a souborů:

- **/ (kořenový adresář)**
  - Text mining.html
  - Text mining.txt
- **/ Clementine**
  - Text_extraction.str
  - Text_link_analysis.str
- **/ RapidMiner**
  - XPath.xml
  - Regularni_vyrazy.xml
  - Extrakce_obsahu_HTML.xml
- **/ Gate**
  - Tokenizer.xml
  - Gazetteer.xml
  - Identifikace_vet.xml
  - Gramaticke_znaceni.xml
- / **Srovnávací studie**
  - Brainstorming.bst
  - Hierarchie.cdp
  - Vahy_kriterii.m