

Slovenská technická univerzita v Bratislave
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLOGIÍ

FIIT-5212-56312

Jakub Ševcech
Poznámkovanie webových stránok
Bakalárska práca

Študijný program: Informatika

Študijný odbor: 9.2.1 Informatika

Miesto vypracovania: Ústav informatiky a softvérového inžinierstva, FIIT STU Bratislava

Vedúci práce: prof. Ing. Mária Bieliková, PhD.

máj, 2011

ZADANIE BAKALÁRSKEHO PROJEKTU

Meno študenta: **Jakub Ševcech**
Študijný odbor: INFORMATIKA
Študijný program: Informatika
Názov projektu: **Poznámkovanie webových stránok**

Zadanie:

Množstvo informácií a snaha po efektívnom prístupe k nim vedie k novým metódam sprístupňovania informácií. Pri zobrazovaní informácií môže významne pomôcť, keď prezentovaný dokument obsahuje poznámky, ktoré napr. spresnia význam určitého slova alebo ho vysvetlia, alebo jednoducho odkážu na dodatočné informácie, či poukážu na chybu v dokumente. Poznámky predstavujú vlastne ďalší obsah, ktorý treba získať, analyzovať a aj efektívne zobraziť (každému podľa aktuálnych potrieb). Skúmajte možnosti ako pri zobrazovaní poznámok zohľadniť kontext dokumentu a aj kontext používateľa, či sociálnych väzieb používateľa. Vyberte si konkrétnu doménu, v rámci ktorej bude možné automaticky získať poznámky vo forme vysvetlení a navrhnite metódu, ktorá bude zohľadňovať kontext pri poznámkovaní. Vytvorte prototyp, ktorý realizuje navrhnutú metódu a experimentálne ju overte.

Práca musí obsahovať:

- Anotáciu v slovenskom a anglickom jazyku
- Analýzu problému
- Opis riešenia
- Zhodnotenie
- Technickú dokumentáciu
- Zoznam použitej literatúry
- Elektronické médium obsahujúce vytvorený produkt spolu s dokumentáciou

Miesto vypracovania: Ústav informatiky a softvérového inžinierstva, FIIT STU, Bratislava
Vedúci projektu: prof. Ing. Mária Bieliková PhD.

Termín odovzdania práce v zimnom semestri: dňa 13. decembra 2010
Termín odovzdania práce v letnom semestri: dňa 11. mája 2011

Bratislava, dňa 20. septembra 2010



prof. Ing. Pavol Návrat, PhD.
riaditeľ ÚISI

Anotácia

Fakulta informatiky a informačných technológií

Slovenská technická univerzita v Bratislave

Autor: Jakub Ševcech
Vedúci bakalárskej práce: prof. Ing. Mária Bieliková, PhD.
Bakalárska práca: Poznámkovanie webových stránok
Štúdijný program: Informatika
máj, 2011

Pri organizovaní a sprístupňovaní informácií ich adresátovi sa v súčasnosti často používajú rôzne druhy poznámok. Tieto poznámky sú vytvorené či už manuálne čitateľmi dokumentu alebo automaticky. V tejto práci sa venujeme automatickému vytváraniu poznámok do webových stránok. O poznámkach hovoríme ako o hocakých informáciách rozširujúcich dokument, či už tým, že opisujú informácie, ktoré obsahuje alebo k nemu pridávajú nejaké ďalšie, rozširujúce informácie. Analyzujeme známe prístupy k poznámkovaniu z rôznych pohľadov, napríklad na základe toho či je poznámka vytvorená manuálne používateľmi alebo automaticky. Navrhli sme metódu na automatické pridávanie poznámok ku kľúčovým slovám v texte v slovenskom jazyku. Vytvorené poznámky majú za úlohu poskytnúť definíciu neznámych slov, alebo rozširujúce informácie k potencionálne zaujímavým slovám. Vytvorené poznámky prispôbujeme na základe implicitnej spätnej väzby od používateľov. Navrhnutú metódu implementujeme a overujeme v prostredí výučbového rámca ALEF vyvíjaného na Fakulte informatiky a informačných technológií. Vytvorené poznámky sú v ALEFe uchovávané spolu s inými druhmi poznámok ako napríklad komentármi alebo tagmi, vytvorenými študentami.

Annotation

Faculty of Informatics and Information Technology
Slovak University of Technology

Name: Jakub Ševcech
Supervisor: prof. Ing. Mária Bieliková, PhD.
Bachelor thesis: Web-page annotation
Course: Informatics
2011, May

Nowadays different kinds of annotations are used for organising and providing information to their recipient. These annotations are created either manually or automatically by document readers. In this work we focus on automatic creation of annotations to Web pages. We talk about annotations as arbitrary information extending document whether by describing information contained in the document or by adding some more enriching information. We analyze the known approaches for document annotation from different perspectives, for example, on the basis of whether the annotation is created manually by document users or automatically. We proposed a method for automatic creation of annotations to the keywords in the text in Slovak language. Created annotations are intended to provide a definition of unfamiliar words or additional information to potentially interesting words. Created annotations are adapted using implicit feedback from the user. We implement and verify the proposed method in an environment of learning framework ALEF developed in the Faculty of Informatics and Information Technology. Created annotations are stored in ALEF along with other types of annotations such as comments or tags created by students.

Pod'akovanie

Chcem sa pod'akovať vedúcej mojej bakalárskej práce, pani profesorky Márii Bielikovej za jej rady, skúsenosti a vynaložený čas, ktorými mi výrazne pomohla pri písaní tejto práce.

Ďalej sa chcem pod'akovať ALEF tímu za možnosť spolupracovať na projekte a celej PeWe skupine za konštruktívne diskusie, ktoré mi často ukázali nový smer a upozornili ma na možné problémy.

V neposlednom rade sa chcem pod'akovať mojej rodine a priateľom, ktorý ma podporovali počas celého môjho štúdia.

Jakub Ševcech

Obsah

1 ÚVOD	1
2 PRÍSTUPY K POZNÁMKOVANIU DOKUMENTOV	3
2.1 Poznámky ako metadáta.....	3
2.2 Poznámky rozširujúce informácie v dokumente	4
2.3 Manuálne poznámkovanie.....	4
2.4 Automatické poznámkovanie.....	5
2.4.1 Vyhľadanie miesta pre pridanie poznámky.....	7
2.4.2 Získanie údajov na naplnenie poznámky.....	7
2.4.3 Zobrazenie poznámok.....	9
3 PRISPÔSOBENIE DOKUMENTOV	11
3.1 Na základe čoho prispôbovať dokumenty.....	11
3.1.1 Získanie spätnej väzby.....	11
3.1.2 Modelovanie používateľa.....	12
3.2 Čo prispôbovať.....	13
4 EXISTUJÚCE NÁSTROJE NA POZNÁMKOVANIE DOKUMENTOV	17
4.1 Nástroje podporujúce manuálne poznámkovanie.....	17
4.1.1 Annotea.....	17
4.1.2 Diigo.....	18
4.1.3 ALEF.....	18
4.2 Nástroje na automatické poznámkovanie.....	19
4.2.1 Wikify!.....	19
4.2.2 zLinks.....	20
4.2.3 Pannda.....	20
4.2.4 Dictionary Tooltip.....	21
4.2.5 Gnosis.....	22
4.3 Zhodnotenie existujúcich riešení.....	23
5 METÓDA NA POZNÁMKOVANIE WEBOVÝCH STRÁNOK	25
5.1 Vytvorenie poznámok k dokumentu.....	25
5.1.1 Mapovanie slov medzi textom a jeho prekladom.....	26
5.1.2 Získavanie informácií	27
5.2 Prispôsobenie poznámok.....	28
5.2.1 Interpretovanie implicitnej spätnej väzby.....	29
5.2.2 Získanie poradia odkazov.....	30
6 REALIZÁCIA NAVRHNUTEJ METÓDY	33
6.1 Adaptive Learning Framework.....	33
6.2 Vytvorenie poznámok.....	34
6.2.1 Mapovanie slov medzi textom a jeho prekladom.....	35
6.2.2 Získanie informácií na naplnenie poznámok.....	36
6.3 Zobrazenie a prispôsobenie poznámok.....	37
6.3.1 Zobrazenie poznámok.....	37
6.3.2 Prispôbovanie poznámok.....	38

7 OVERENIE A EXPERIMENTY	41
7.1 Mapovanie slov medzi textom a jeho prekladom.....	41
7.2 Vyhodnotenie kvality informácií na naplnenie poznámok.....	42
7.3 Overenie metódy v systéme ALEF.....	43
7.3.1 Vyhodnotenie porovnaním implicitnej a explicitnej spätnej väzby.....	43
7.3.2 Vyhodnotenie odpovedí študentov.....	45
8 ZHODNOTENIE	49
LITERATÚRA	51
PRÍLOHY	53
A TECHNICKÁ DOKUMENTÁCIA	55
A.1 Ukážka kódu na výpočet Levenshteinovej vzdialenosti.....	55
A.2 Ukážka výstupného XML súboru služby na vytvorenie poznámok.....	56
A.3 Ukážka kódu na vytvorenie tooltipu s poznámkou.....	57
A.4 Zjednodušený dátový model systému ALEF.....	58
A.5 Doménový model systému ALEF.....	59
B POUŽÍVATEĽSKÁ PRÍRUČKA	61
C INŠTALAČNÁ PRÍRUČKA	79
D PRÍSPEVOK PUBLIKOVANÝ NA KONFERENCII IIT.SRC 2011	81
E PRÍSPEVOK ZASLANÝ NA KONFERENCIU WEB INTELLIGENCE 2011	89
F DIPLOM ZÍSKANÝ V MEDZINÁRODNEJ SÚŤAŽI ELEARNING V PRAXI V RÁMCI KONFERENCIE ICETA 2010	95
G OBSAH ELEKTRONICKÉHO MÉDIA	99

1 Úvod

V súčasnom svete preplnenom informáciami nie je problém získať informácie. Problém je v týchto informáciách nájsť tú, ktorú práve potrebujeme. Pri vyhľadávaní informácií v prostredí internetu sme si zvykli používať rôzne služby na vyhľadávanie. Tieto služby nám však pre takmer každý dopyt vrátia milióny relevantných výsledkov. Na nás stále zostáva si z tohto obrovského množstva informácií vybrať tú, ktorú potrebujeme, pretože v podstate z týchto miliónov výsledkov potrebujeme len jeden jediný, ten ktorý nám dá odpoveď, ktorú hľadáme.

Na lepšiu orientáciu v takomto obrovskom priestore informácií sa úspešne používajú poznámky (anotácie) pripojené k dokumentom. Poznámky vlastne tvoria dodatočné informácie, ktoré nejakým spôsobom obohacujú pôvodný dokument. Tento dokument nemusí byť len text, ale napríklad aj obrázky, hudba alebo iný multimedialný obsah. Svoje uplatnenie našli anotácie napríklad v digitálnych knižniciach [3], kde sú s ich pomocou organizované a sprístupňované obrovské množstvá odborných textov. Podobné využitie našli aj v oblasti výučbových systémov, kde sú s ich pomocou organizované výučbové materiály a študenti tak dokážu nájsť potrebné informácie.

Proces poznámkovania si môžeme predstaviť ako činnosť podobnú tej, ktorú vykonáva čitateľ knihy alebo napríklad aj článku v novinách, keď si v texte vyznačuje zaujímavé pasáže, podčiarkuje dôležité slová, prípadne na okraj textu vkladá rôzne poznámky. V súčasnosti sa veľké množstvo nástrojov snaží tento proces napodobniť v prostredí webu a obohatiť ho s použitím moderných technických prostriedkov. Jeden zo spôsobov ako tento proces vylepšiť je pridaním nových metód na podporu tvorby poznámok. Moderné technické prostriedky však neposkytujú nové druhy poznámok, tie existujú už veľmi dávno. Pomocou nových prostriedkov sa vylepšujú len metódy na podporu tvorby, uchovávaní a prezentácie poznámok.

Jedným zo zaujímavých aspektov pri tvorbe poznámok, je podpora kolaborácie. To znamená, že viacerí používatelia toho istého dokumentu, spoločne vytvárajú poznámky, ktoré sú viditeľné aj ostatným používateľom. Takto sa pomocou poznámok od rôznych čitateľov, vytvára akási vrstva nad dokumentom, ktorá môže výrazne zvýšiť informačnú hodnotu dokumentu.

Doteraz sme hovorili len o poznámkach, ktoré vytváral človek pri študovaní dokumentu. Avšak kvôli už spomínanému preťaženiu informáciami je nemožné pridať poznámky ku všetkým dostupným dokumentom. Preto vznikol druhý smer ako vylepšiť proces poznámkovania a to tak, že tento proces sa automatizuje a poznámky už nebudú musieť vytárať len používatelia, ale aj stroje.

V tejto práci sa zaoberáme automatickým pridávaním poznámok do textu webovej stránky a ich prispôbením na základe spätnej väzby od používateľov poznámok. Vytvorené poznámky majú za cieľ definovať dôležité slovo v texte, alebo k tomuto slovu poskytnúť dodatočné informácie. Poznámky sú preto vo forme definície pojmu a vo forme odkazov na internetové stránky súvisiace s dôležitým termínom v texte internetovej stránky. Vytvorené poznámky následne upravujeme pre potreby používateľov na základe spätnej väzby získanej

sledovaním ich interakcie s poznámkami. Pri vytváraní poznámok sa primárne orientujeme na oblasť vzdelávania, ale našou ambíciou je, aby navrhnuté riešenie bolo širšie použiteľné.

V kapitole 2 analyzujeme rôzne druhy poznámok a prístupov k poznámkovaniu. V kapitole 3 opisujeme rôzne metódy na prispôsobovanie dokumentov. Niektoré existujúce riešenia na podporu manuálneho vytvárania poznámok do dokumentov a riešenia na automatické poznámkovanie dokumentov sú opísané v kapitole 4. Po opise existujúcich riešení pre vytváranie poznámok sa sústreďujeme na hlavnú časť dokumentu (kapitoly 5 a 6), kde navrhujeme a realizujeme metódu na vytváranie poznámok ku kľúčovým slovám v texte v slovenskom jazyku. Navrhovanú metódu overujeme (kapitola 7) v prostredí výučbového rámca ALEF [24].

2 Prístupy k poznámkovaniu dokumentov

Poznámky sa používajú na organizovanie a obohacovanie dokumentov napríklad v digitálnych knižniciach [3], výučbových systémoch [24] alebo na širokom webe [1]. V tomto dokumente sa zaoberáme automatickým poznámkovaním webových stránok, teda dokumenty, do ktorých sa pridávajú poznámky, tvorí text webovej stránky. Ďalej sa bude slovo dokument voľne zamieňať so slovným spojením textový dokument alebo v krátkosti len text. Ak sa budú informácie týkať aj iných dokumentov ako len textových, teda ak pôjde napríklad o rôzne multimediálne dokumenty, túto informáciu explicitne uvedieme.

Najskôr je potrebné definovať, čo sa v skutočnosti myslí pod pojmom poznámka (anotácia). Rôzni autori majú rôzny pohľad na podstatu poznámky. Základná, spoločná myšlienka je, že je to akási informácia pripojená k dokumentu. Na túto základnú myšlienku sa dá ďalej pozerieť dvoma hlavnými smermi [2]. Smer, kde sa na poznámku pozerá ako na metadáta pripojené k dokumentu, ktoré ho opisujú, alebo ako na informácie, ktoré daný dokument rozširujú.

Poznámky môžeme rozdeliť aj podľa toho, ako sú pripojené k dokumentu. Môžu byť súčasťou dokumentu, ako napríklad informácie o súbore, v ktorom je dokument uložený. Ale rovnako môžu byť poznámky samostatné dokumenty, pripojené k dokumentu, ktorého sa týkajú.

Pri automatickom poznámkovaní dokumentov môžeme rozlišovať poznámky aj podľa toho, kedy prebieha samotné vytváranie poznámok. Teda či sa vytvárajú poznámky počas sprístupňovania poznámok (on-line), alebo ešte pred samotným prijatím požiadavky na zobrazenie poznámky (off-line). Metódy na vytváranie poznámok pri ich sprístupňovaní musia riešiť prísne časové limity, zatiaľ čo metódy na vytváranie poznámok pred ich vyžiadáním nemajú takéto obmedzenie. Pri vytváraní poznámok „v predstihu“ je však potrebné nejakým spôsobom ohraničiť doménu, v ktorej sa budú poznámky vytvárať. Takéto ohraničenie môže byť napríklad obmedzenie sady opoznámkovaných dokumentov, obmedzenie témy, ktorej sa budú poznámky týkať a podobne. Možné sú aj kombinácie týchto dvoch prístupov, keď sa časť procesu vytvárania poznámok odohráva pred prijatím požiadavky na zobrazenie poznámky a časť sa vykonáva až po prijatí tejto požiadavky.

2.1 Poznámky ako metadáta

Metadáta pripojené k dokumentu sú vlastne informácie o existujúcom obsahu dokumentu. Účelom metadát nie je pridávanie nových informácií do dokumentu, ale opísanie už existujúcich informácií a vlastností dokumentu. Takouto dodatočnou informáciou je napríklad veľkosť dokumentu, čas vytvorenia alebo poslednej úpravy, ale aj téma, ktorej sa dokument týka. Pri obrázkoch to môže byť ich rozlíšenie, formát a podobne.

Poznámky slúžiace na pripájanie metadát k dokumentom musia mať presne stanovenú štruktúru, aby ich dokázali čítať nielen ľudia, ale najmä aby sa dali spracovať automaticky. Ak je štruktúra takýchto metadát dostatočne známa a podporovaná a tieto informácie sú v strojovo spracovateľnej forme, dokážu ich spracovať rôzne nástroje na automatizované

spracovanie informácií. Následne sa dajú použiť na vylepšenie už existujúcich nástrojov, ako napríklad už spomínané vyhľadávanie informácií na Internete.

Príkladom formátu na uchovávanie poznámok ako metadát k dokumentu je formát RDF¹, ktorý sa využíva vo webe so sémantikou. RDF je rozšírením jazyka XML². Vďaka tomuto formátu sa dajú k človeku zrozumiteľnému dokumentu, pripojiť údaje, ktoré dokáže spracovať stroj. V tomto prípade sú poznámky samostatné dokumenty a k dokumentu, na ktorý sa viažu, sú pripojené pomocou URI³.

2.2 Poznámky rozširujúce informácie v dokumente

Druhý spôsob ako pozeráť na poznámku, je vnímať ju ako informáciu pripojenú k dokumentu, ktorá tak tvorí ďalší obsah. Ak hovoríme o poznámkach, ktoré vytvára používateľ alebo čitateľ dokumentu, tak to môže byť napríklad informácia, ktorou používateľ vkladá myšlienku, ktorá ho napadla pri štúdiu dokumentu, vysvetlenie danej časti dokumentu, odkaz na iný dokument a podobne. Tento typ poznámok zahŕňa aj rôzne vyznačovanie dôležitých častí dokumentu, či už manuálne používateľmi alebo automaticky. Takéto poznámky sa podobajú na marginále, čiže „tematické okrajové ukazovatele na okraji textu v knihách (hlavne odborných)“, ako ich definuje Slovník slovenského jazyka⁴. V tomto prípade sa poznámky nemusia nachádzať len na okraji textu, ale môžu byť rôzne graficky zakomponované do štruktúry dokumentu. V súčasnosti existujú rôzne spôsoby ako zobrazovať takéto poznámky v elektronickom dokumente. Často sa používajú zvýraznenia častí textu pomocou zmeny štýlu písma, zmeny farby pozadia alebo pridaním rôznych značiek do textu. Dodatočné informácie sú zas zobrazované v texte pomocou rôznych tooltipov a pop-up okien.

Inými informáciami, ktorými sa dá obohatiť dokument sú napríklad definície odborných, cudzích, neznámych alebo inak na pochopenie náročných slov. Podobne sa dajú do textu webovej stránky pridať odkazy na iné stránky, ktoré by svojím obsahom dopĺňali pôvodnú stránku. Adresámi takýchto poznámok sú ľudia, keďže tento typ poznámky už nemusí mať presne stanovenú formu.

V tejto práci sa zaoberáme hlavne poznámkami, ktoré rozširujú informácie v dokumente a nie poznámkami, ktoré opisujú samotný dokument.

2.3 Manuálne poznámkovanie

Manuálne poznámkovanie je určitou analógiou k už spomínanému písaniu poznámok na okraj dokumentu. Poznámky v tomto prípade môžu byť rôzne grafické značky, ktoré zvýrazňujú časti textu (krúžkovanie, podčiarknutie textu, rôzne šípky, ...), prípadne prvky, ktoré obohacujú dokument o dodatočné informácie [18]. Takéto poznámky môžu byť napríklad výpisky z textu, myšlienky, ktoré napadli používateľa pri čítaní textu alebo vysvetlenia dôležitých termínov.

1 RDF, „Resource Description Framework“, <http://www.w3.org/RDF/>

2 XML, „eXtensible Markup Language“, <http://www.w3.org/XML/>

3 URI, „Uniform Resource Identifier“, <http://www.w3.org/TR/uri-clarification/>

4 Slovník slovenského jazyka, Hlavný redaktor dr. Štefan Peciar, CSc. Bratislava: Vydavateľstvo Slovenskej akadémie vied 1959 – 1968, <http://slovniky.juls.savba.sk>

Ak množstvo používateľov pridáva poznámky do dokumentov a ak sú tieto poznámky viditeľné aj pre ostatných čitateľov, tak ďalší používatelia môžu využívať takto vytvorené informácie a teda môžu získať väčší úžitok z čítania dokumentu. Tento efekt sa však dosiahne len v tom prípade, ak používatelia nie sú zahltený množstvom poznámok priradených k dokumentu. Vtedy by poznámky neriešili zahltenie informáciami, ale naopak ešte by tento problém prehlbovali. Je preto potrebné zaviesť nástroje, ktorými je možné skrývať nevhodné poznámky alebo filtrovať spomedzi všetkých len tie najdôležitejšie.

Podľa toho ako jednotlivé systémy na poznámkovanie podporujú spoluprácu používateľov pri tvorbe poznámok, dajú sa rozdeliť na dve skupiny:

- Systémy, ktoré poskytujú rozhranie umožňujúce vytváranie poznámok používateľmi.
- Systémy zabezpečujúce navyiac aj kolaboráciu medzi používateľmi navzájom.

Kolaborácia dáva možnosť používateľom vytvárať poznámky k dokumentom spoločne. To spôsobuje, že nad dokumentom sa vytvára vrstva, v ktorej sú pridané vedomosti celej skupiny používateľov, a tak sa výrazne zvyšuje informačná hodnota dokumentu. Kolaborácia pri poznámkovaní sa technicky zabezpečuje pridaním možnosti zdieľať túto poznámku s ostatnými používateľmi. Možnosť zdieľania je často doplnená o možnosť určenia skupiny používateľov, pre ktorú je daná poznámka viditeľná, podobne ako to robí napríklad nástroj Diigo⁵. Týmto spôsobom sa môžu vytvárať záujmové skupiny, ktoré zoskupujú dokumenty z určitej domény a vytvárajú k nim poznámky.

2.4 Automatické poznámkovanie

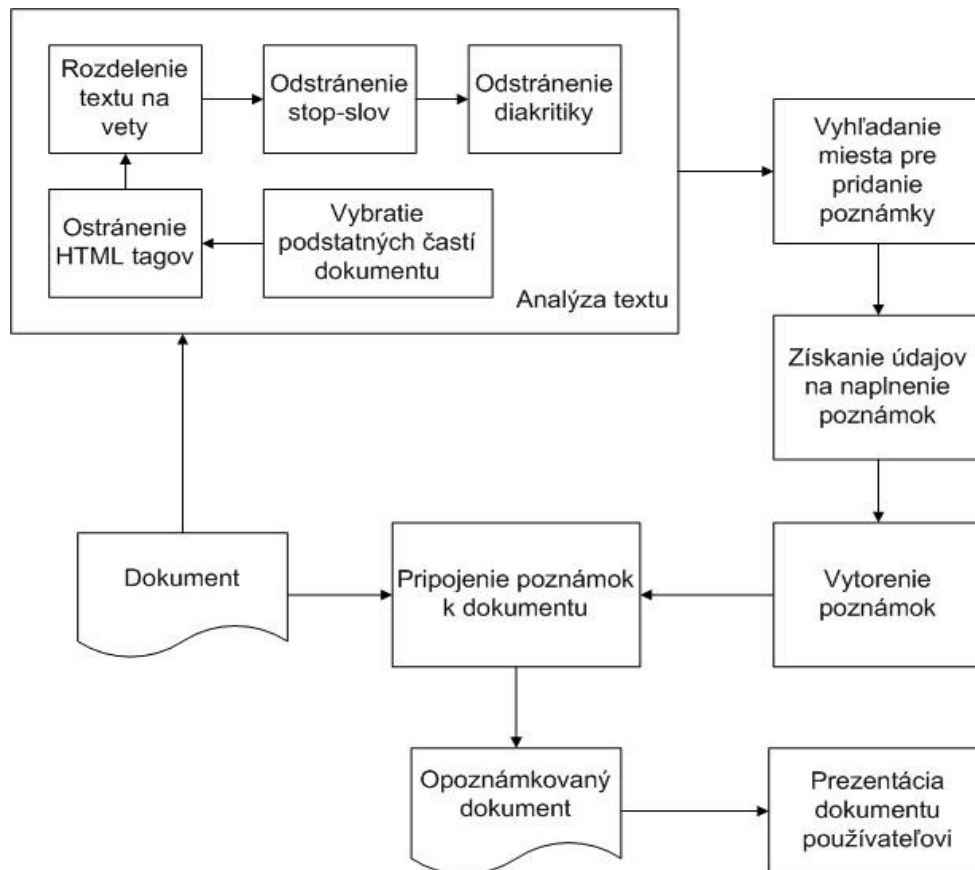
Automatické poznámkovanie spočíva v tom, že poznámky nevytvára čitateľ dokumentu, ale stroj na základe analýzy dokumentu. Pri manuálnom poznámkovaní nebolo potrebné starať sa o získavanie informácií do poznámok a o umiestnenie poznámky na správne miesto do textu, tieto problémy zostali na pleciach samotného používateľa. Pri automatickom poznámkovaní je potrebné vyriešiť oba tieto problémy a navyiac treba riešiť aj veľkú časť problémov, ktoré sa vyskytovali aj pri poznámkovaní používateľmi.

Proces poznámkovania pozostáva z viacerých krokov (pozri Obr. 1). Pri vytváraní poznámok k dokumentu je potrebné najskôr vykonať analýzu dokumentu, teda previesť dokument do takej podoby, aby s ním vedel poznámkovač ďalej pracovať. Úpravy, ktoré sa pri tejto analýze vykonávajú môžu byť napríklad:

- Vybratie podstatných častí dokumentu. Napríklad pri webových stránkach to znamená odstránenie reklamy, rôznych ovládacích prvkov a menu. Jednou zo základných metód, ktoré sa na túto úlohu používajú je analýza používania interpunkcie v texte. Predpokladá sa totiž, že rôzne reklamy a ovládacie prvky používajú interpunkciu len zriedka na rozdiel od hlavného textu stránky.
- Rozdelenie textu na vety.
- Odstránenie diakritiky a interpunkcie.

5 Diigo, <http://www.diigo.com/>

- Odstránenie tzv. stop-slov, teda slov, ktoré sú pre ďalšie spracovanie nežiadúce. Z textu sa odstraňujú napríklad spojky, častice, zámená a podobné slová, ktoré nemajú význam pre ďalšie spracovanie textu.
- Odstránenie HTML tagov.



Obr. 1. Proces automatického pridávania poznámok do dokumentu

Po analýze dokumentu poznámkovač určí, ku ktorým častiam dokumentu je potrebné pridať poznámky, teda v texte nájde kandidátov na pridanie poznámky. Tento kandidát môže byť napríklad kľúčové slovo v texte, pomenované entity, ktoré označujú napríklad osobu, miesto, krajinu, organizáciu alebo inak významné slovo. Ďalej je možné pridávať poznámky k väčším častiam textu, ako len jedno alebo niekoľko málo slov. To môže byť napríklad celý odsek, ktorý je niečím zaujímavý v porovnaní s ostatným textom.

Keď už máme určené časti textu, ku ktorým pridáme poznámky, je potrebné získať informácie, ktorými sa tieto poznámky naplnia. Až po získaní týchto informácií je možné vytvoriť poznámky a zobrazíť ju používateľovi.

Používateľ s takouto poznámkou interaguje a na základe tejto interakcie je možné neskôr upravovať poznámky. Tento posledný krok už nie je základnou súčasťou poznámkovania. Poznámky teda môžu existovať a plniť svoj účel aj bez toho aby sa prispôbovali za základe spätnej väzby.

2.4.1 Vyhľadanie miesta pre pridanie poznámky

Ako sme už uviedli, miesta v dokumente, ku ktorým chceme pridať poznámky, môžu byť kľúčové slová, pomenované entity, dôležité úseky textu alebo inak významné časti dokumentu. Pre vyhľadávanie týchto miest sa používa viacero rôznych metód. Používajú sa napríklad metódy z oblasti spracovania prirodzeného jazyka (NLP, Natural Language Processing), ktoré sa používajú na vetný rozbor, strojový preklad, vyhľadávanie pomenovaných entít, rozlišovanie významu slova a veľa ďalších [8].

Na vyhľadanie kľúčových slov v texte sa používajú rôzne ATR algoritmy [23]. Napríklad algoritmus TF-IDF hľadá kľúčové slová tak, že pozerá aká je frekvencia výskytu slova v analyzovanom dokumente a súčasne sleduje aká je frekvencia výskytu tohto slova v ostatných dokumentoch. Predpokladá, že ak je vysoká frekvencia výskytu slova v dokumente, tak toto slovo má veľký význam pre dokument. Existujú však slová ako napríklad spojky a častice, ktoré sa nachádzajú často v takmer každom dokumente, a teda nemajú rozlišovaciu schopnosť medzi dokumentami, nezávisle od toho aké je toto slovo významné pre konkrétny dokument. Algoritmus TF-IDF odstraňuje problémy spôsobené týmito slovami tak, že sleduje frekvenciu výskytu slova aj v ostatných dokumentoch a teda vie určiť, či sa dané slovo objavuje pravidelne a je bežne používané. Takéto slová sú následne penalizované pri výpočte ich dôležitosti v dokumente.

Výsledkom týchto algoritmov je zoznam slov s priradenými hodnotami významnosti vzhľadom k dokumentu. Podobných algoritmov je viac a využívajú rôzne vlastnosti slov a dokumentov ako napríklad algoritmus Weirdness, ktorý predpokladá, že významné slová v spracovávanom dokumente majú výrazne odlišné rozloženie výskytu ako tie isté slová vo všeobecnej sade dokumentov [4].

Pre vyhľadanie pomenovaných entít sa používajú prístupy založené na gramatikách, strojovom učení a hybridné prístupy [17]. Na prvý pohľad by sa mohlo zdať, že nájsť pomenované entity v texte je pomerne ľahké, pretože sa dá vytvoriť zoznam napríklad všetkých riek alebo miest a stačí porovnať každé slovo z textu s týmto zoznamom. Tento prístup však nefunguje hneď z niekoľkých dôvodov:

- Zoznam pomenovaných entít sa stále mení, pribúdajú napríklad nové spoločnosti alebo nové významné osobnosti.
- Význam slova v texte nie je tvorený len samotným slovom, ale aj kontextom, v ktorom sa dané slovo v texte nachádza. Príkladom slova, pri ktorom by takýto prístup robil problémy, je slovo Brazília. Brazília je krajina ale aj hlavné mesto tejto krajiny a jeho skutočný význam je možné zistiť len na základe kontextu, v ktorom bolo toto slovo použité.

2.4.2 Získanie údajov na naplnenie poznámky

Na získanie informácií na naplnenie poznámky môžeme využiť pomerne veľa možností. Môžeme použiť napríklad predpripravenú bázu údajov [19]. Tento postup je však vhodný len vtedy ak máme presne ohraničenú doménu, ktorej sa budú poznámky týkať. Ak by sme chceli

zmeniť doménu kde by sa poznámky používali, museli by sme rozšíriť pôvodnú alebo vyrobiť novú bázu dát a to by si vyžadovalo nemalé úsilie.

Druhý spôsob je vyhľadávať voľne dostupné informácie a vkladať ich do anotácie, podobne ako to robia nástroje ako Gnosis⁶ alebo Dictionary Tooltip⁷. Nástroj Gnosis sa snaží identifikovať pomenované entity v texte a k týmto entitám následne vyhľadáva dodatočné informácie s pomocou verejne dostupných služieb ako Google alebo Wikipédia. V súčasnej dobe je pre tento účel možné použiť veľké množstvo vyhľadávacích nástrojov a podobných služieb, ktoré poskytujú informácie na základe nejakého dopytu, najčastejšie vo forme zoznamu kľúčových slov.

Dopyty vo forme zoznamu kľúčových slov však nemajú žiadnu špecifikovanú syntax a teda často nejednoznačne špecifikujú informáciu, ktorú chceme získať. Existujú spôsoby ako sa dopytovať údajov, na základe presne stanovených syntaktických pravidiel. Na základe týchto pravidiel špecifikujeme podmienku, ktorú musí dokument spĺňať a ako výsledok získame množinu výsledkov, ktoré spĺňajú špecifikáciu. Podobne ako to robí napríklad nástroj Sig.ma⁸, ktorý poskytuje rozhranie pre vyhľadávanie informácií na internetových stránkach, ktoré podporujú štandardy pre sémantický web.

Dopytmi s presne stanovenou syntaxou sa dajú oveľa presnejšie špecifikovať požiadavky na výsledok a teda je možné sa k nemu dopracovať rýchlejšie a pritom tieto informácie presne spĺňajú naše požiadavky vyjadrené dopytom. Príkladom takéhoto dopytovacieho jazyka je jazyk SPARQL⁹. SPARQL je jazyk na dopytovanie sa údajov, ktoré sú uložené vo formáte RDF. Tieto informácie sú uložené po trojiciach alebo tiež tripletoch, subjekt – predikát – objekt.

Dopyt v jazyku SPARQL môže vyzeráť napríklad nasledovne:

```
SELECT ?abstract
WHERE {
  { <http://dbpedia.org/resource/Computer_software>
    <http://dbpedia.org/ontology/abstract> ?abstract .
  FILTER langMatches( lang(?abstract), 'en') }
}
```

Ako je vidieť z príkladu, syntax jazyka SPARQL je veľmi podobná jazyku SQL. Výsledkom tohto dopytu je abstrakt v anglickom jazyku k zdroju Computer_software. V tomto dopyte je subjekt URI adresa zdroja na Dbpedii¹⁰:

http://dbpedia.org/resource/Computer_software, nasleduje znova URI, ale tentokrát vyjadruje predikát: <http://dbpedia.org/ontology/abstract>, ktorý

6 Gnosis, <http://www.opencalais.com/Gnosis>

7 Dictionary Tooltip, <http://www.dictionarytip.com/>

8 Sig.ma, <http://sig.ma>

9 SPARQL, <http://www.w3.org/TR/rdf-sparql-query/>

10 DBpedia, <http://wiki.dbpedia.org/About>

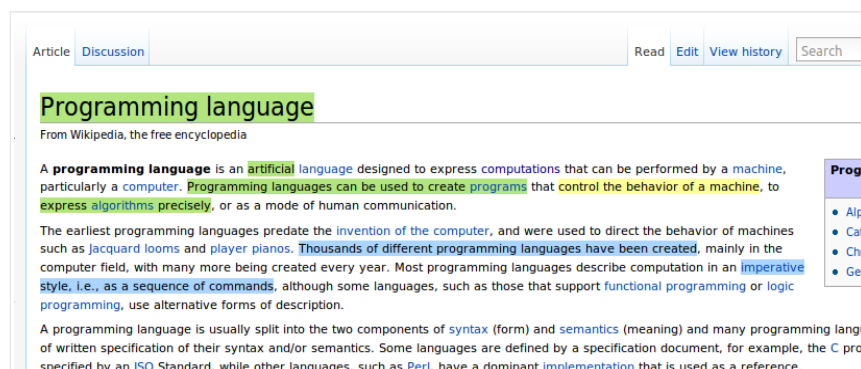
hovorí o tom, že hľadáme abstrakt. Objekt je v tomto príklade premenná, čiže ten je zatiaľ neznámy. Nakoniec je tu filter, ktorý hovorí, že chceme len abstrakty v anglickom jazyku.

Výsledkom tohto dopytu je množina tripletov, ktoré spĺňajú zadanú podmienku. Premenná `abstract` sa nahradí za konkrétny abstrakt z množiny výsledkov.

Problém pri používaní dopytov s presne stanovenou syntaxou je ten, že táto syntax je často náročná a používateľ ju nepozná, alebo jednoducho nevie presne špecifikovať informáciu, ktorú hľadá. Vznikajú teda nástroje, ktoré sa snažia prekonať tieto problémy a poskytujú rozhranie na jednoduchšie vytvorenie takýchto dopytov. Jedným takýmto nástrojom je Zitgist Query Builder¹¹.

2.4.3 Zobrazenie poznámok

Spôsobov ako vizualizovať poznámky existuje veľa a pri konkrétnej realizácii sa zväčša používajú kombinácie viacerých z nich.



Obr. 2: Príklad poznámok vo forme zvýraznení častí textu pomocou nástroje Diigo

Spôsoby zobrazenia poznámok môžeme rozdeliť do dvoch skupín, podľa toho aký druh poznámky sa pomocou nich prezentuje. Môžeme zobrazovať poznámky, ktoré majú za úlohu zvýrazniť časti textu alebo také, ktoré do textu pridávajú nejaký dodatočný textový, obrazový alebo iný multimediálny obsah.

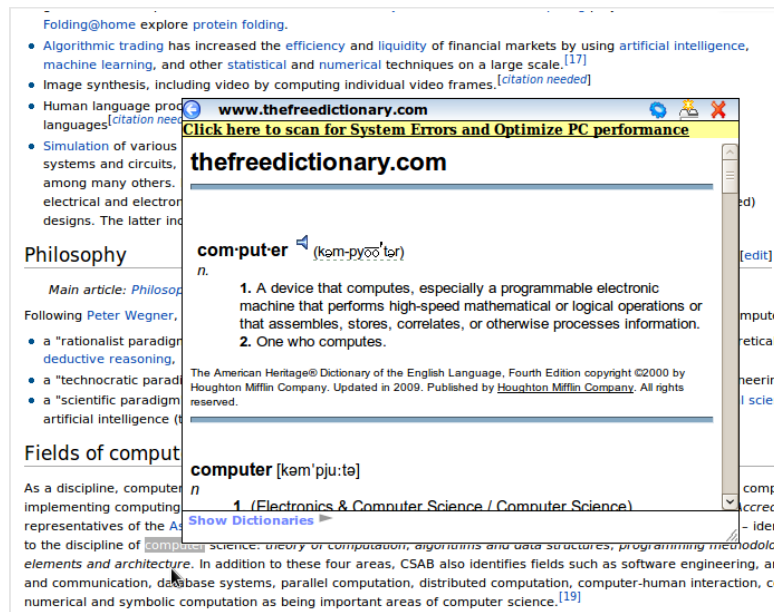
Poznámky, ktoré zvýrazňujú časti textu môžu mať formu napríklad podčiarknutia, zmeny farby pozadia textu, zakrúžkovania textu, zmeny štýlu, veľkosti alebo farby písma. Môžu sa zobrazovať aj tak, že do textu pridávajú nejaký grafický prvok, napríklad šípku alebo iný symbol.

Poznámky, ktoré do dokumentu pridávajú nejaký dodatočný obsah sa často zobrazujú pomocou pop-up okien alebo tooltipov. Takéto pop-up okno sa môže aktivovať buď prechodom myši nad oblasťou, ku ktorej bola poznámka pripojená, alebo kliknutím na túto oblasť. Pri oboch spôsoboch aktivácie si však treba dať pozor na to, ako sú použité. Ak sa okno s poznámkou aktivuje prechodom myši nad opoznámkovanou oblasťou, je dôležité zabezpečiť, aby otvárajúce sa okná nepôsobili rušivo. Preto by sa pred zobrazením okna s poznámkou mal nastaviť čas, ktorý treba počkať, aby sa predišlo náhodnému otváraníu okien.

¹¹ Zitgist Query Builder, http://zitgist.com/products/query_builder.html

Takéto neželané správanie môže zneprijemniť prácu s dokumentom a pri väčšom množstve poznámok môže dokonca znemožniť plnohodnotnú prácu.

Podobne ak sa okno zobrazí až po kliknutí na opoznámkovanú oblasť, treba aby bola táto opoznámkovaná oblasť textu nejako graficky oddelená od ostatných častí dokumentu. Často sa pri tomto spôsobe aktivácie okna použije niektoré zo zvýraznení, ktoré sme opisovali vyššie. Tu znova vzniká nebezpečenstvo, že pri väčšom množstve poznámok pridaných k dokumentu, môže množstvo zvýraznení v texte pôsobiť rušivo.



Obr. 3: Príklad poznámky vo forme tooltipu vytvoreného nástrojom DictionaryTooltip

3 Prispôsobenie dokumentov

Podľa definície Brusilovského [9], adaptívny hypermediálny systém je každý hypertextový alebo hypermediálny systém, ktorý zobrazuje niektoré vlastnosti používateľa do modelu používateľa a tento model používa na prispôsobenie dokumentu. Adaptácia poznámok znamená upravovanie poznámok, ich obsahu, výzoru, umiestnenia podľa používateľa a kontextu jeho práce, ktorý je vyjadrený modelom používateľa.

Takéto adaptívne systémy môžu podporovať používateľa v navigácii medzi dokumentami, alebo mu môžu ponúkať poznámky k dokumentu, ktorý práve číta.

Pri prispôsobovaní dokumentov sa treba zaoberať dvoma základnými otázkami:

1. Na základe čoho prispôsobovať dokument.
2. Čo a ako prispôsobovať, teda ktorú časť dokumentu adaptovať a ako sa bude toto prispôsobenie prejavovať.

3.1 Na základe čoho prispôsobovať dokumenty

Pre prispôsobovanie dokumentov je kľúčové získanie spätnej väzby, na základe ktorej bude toto prispôsobovanie prebiehať. Bez správne získanej a interpretovanej spätnej väzby je následná úprava dokumentov neefektívna a môže dokonca znížiť kvalitu upravených dokumentov. Pre správne interpretovanie tejto spätnej väzby pre konkrétnych používateľov si rôzne systémy často vytvárajú model používateľa, na základe ktorého následne personalizujú dokumenty.

3.1.1 Získanie spätnej väzby

Pri získavaní spätnej väzby od používateľov sa môžeme sústreďovať na dva základné druhy spätnej väzby: explicitnú a implicitnú. Pri explicitnej spätnej väzbe žiadame používateľa aby nám vedome poskytol svoje názory, postrehy alebo iné potrebné informácie. Pri implicitnej spätnej väzbe naopak sledujeme jeho akcie a na základe nich odvodzujeme potrebné informácie, pričom používateľ nevie či a ako jeho interakciu systém sleduje a interpretuje.

Explicitná spätná väzba

Pri použití tohto spôsobu získavania spätnej väzby si priamo pýtame od používateľa konkrétne informácie. Nenastáva tu problém s nesprávnym interpretovaním získaných informácií, pretože používateľ si je vedomý toho, že nám túto spätnú väzbu poskytuje a teda mu môžeme do istej miery dôverovať. To že používateľ musí vedome poskytovať spätnú väzbu však môže spôsobovať problémy. Táto aktivita používateľovi priamo nepomôže k vyriešeniu cieľa, ktorý sleduje pri používaní systému. Často sa teda stáva, že používatelia nie sú ochotní poskytovať spätnú väzbu a získavame jej len veľmi málo. Je teda potrebné nejakým spôsobom motivovať používateľa k poskytovaniu tejto väzby. Takouto motiváciou môže byť napríklad lepšie fungovanie systému, personalizovaného pre potreby používateľa, alebo nejaký iný druh „odmeny“.

Implicitná spätná väzba

Ak bol pri explicitnej spätnej väzbe používateľ nútený vykonať dodatočnú akciu na poskytnutie spätnej väzby, pri získavaní implicitnej spätnej väzby nemusí vykonávať žiadne dodatočné kroky, nemusí dokonca ani vedieť že systém nejakú spätnú väzbu potrebuje a získava. Táto spätná väzba sa totiž získava len na základe interpretovania interakcie používateľa so systémom. Tým že túto spätnú väzbu získavame nepriamo, odvodením z akcií používateľa, sa však do nej pridávajú rôzne šумы a nežiaduce vplyvy. Treba teda správne interpretovať interakciu, tak aby sa minimalizoval vplyv týchto šumov a aby boli získané informácie čo najpresnejšie.

Možné problémy s interpretovaním spätnej väzby si ukážeme na príklade interpretovania akcie kliknutia na odkaz v ponúkanom zozname. Na základe tohto kliknutia by sme chceli určiť kvalitu odkazov poskytnutých napríklad nástrojom na vyhľadávanie internetových stránok.

Naivný spôsob ako interpretovať kliknutie na odkaz ako implicitnú spätnú väzbu by bolo spočítať kliknutia a na základe ich počtu určiť, ktorý odkaz z ponúkaného zoznamu je ten najlepší alebo najrelevantnejší. Pri tomto postupe sa ale zanedbávajú niektoré faktory, ktoré vplyvajú na používateľa [14]. Odkazy sú totiž zoradené v zozname a používatelia prirodzene preferujú klikanie na prvé odkazy v zozname. Používateľ sa na odkazy nachádzajúce sa ďalej v zozname častokrát nedostane, pretože na vyšších pozíciách našiel požadovaný odkaz, alebo radšej preformuloval dopyt použitý na vyhľadanie odkazov. Ďalším vplyvom, ktorý bol zanedbaný je fakt, že odkazy sú častokrát už v tomto zozname zoradené na základe nejakého kľúča. Získaná spätná väzba je teda ovplyvnená týmto usporiadaním. V neposlednom rade používatelia do istej miery dôverujú nástroju, ktorý používajú a predpokladajú, že najlepšie výsledky nájdu na najvyšších priečkach v zozname.

Druhým možným postupom ako interpretovať kliknutie na odkaz je priradiť cenu kliku v závislosti od toho, na ktorej pozícii v zozname sa nachádzal zvolený odkaz. Pre použitie tejto metódy je však potrebné poznať funkciu, podľa ktorej sú rozdelené početnosti kliknutí na jednotlivé pozície v zozname. Táto funkcia je pre každý nástroj odlišná a bolo by teda potrebné pre každý systém získať množstvo tréningových dát, na základe ktorých by sa táto funkcia určila.

Ďalším spôsobom ako interpretovať tieto kliknutia je nepriradovať ceny klikom, ale považovať tieto kliknutia za tvrdenia, že zvolený odkaz je lepší ako niektoré iné v zozname [14]. Je možné použiť hneď niekoľko stratégií ako tieto tvrdenia vyhľadávať. Príkladom takejto stratégie je povedať, že všetky nekliknuté odkazy v zozname, ktoré sú na vyššej pozícii ako kliknutý odkaz, sú od neho horšie. Z týchto tvrdení je následne možné odvodiť kvalitu odkazov v porovnaní s ostatnými odkazmi v zozname, avšak nie je možné odvodiť absolútne ohodnotenie kvality odkazov.

3.1.2 Modelovanie používateľa

Pre lepšie prispôsobovanie dokumentov pre potreby konkrétneho používateľa si systémy často vytvárajú model používateľa. Do tohto modelu sa premietajú rôzne vlastnosti používateľa. Jednou z takýchto vlastností sú jeho vedomosti. Vedomosti používateľa sú špeciálne dôležité

vo výučbových systémoch, kde je to základná vlastnosť, na základe ktorej sa systém prispôsobuje. Vedomosti medzi rôznymi používateľmi systému o nejakom koncepte sa môžu výrazne líšiť a zároveň sa môžu líšiť aj vedomosti toho istého používateľa v priebehu času. Adaptívny systém musí byť schopný tieto rozdiely zaznamenať a zohľadniť pri prispôbovaní.

Inou vlastnosťou používateľa, na základe ktorej možno prispôbovať obsah sú používateľove ciele, teda zámer, s ktorým používateľ pracuje so systémom. Tento zámer sa môže počas práce so systémom meniť a to aj počas jedného sedenia. Adaptívny systém by mal dokázať takúto zmenu cieľa odhaliť a podľa toho zmeniť spôsob akým vykonáva personalizáciu. Ako príklad takéhoto sedenia si môžeme zobrať vyhľadávanie webových stránok pomocou internetového vyhľadávača. Počas vyhľadávania jednej informácie môže používateľ zrazu zmeniť cieľ hľadania, prípadne môže zadať náhodný dopyt, ktorý s pôvodnou témou vyhľadávania vôbec nesúvisí. Používateľ môže mať tiež otvorených viacero okien s vyhľadávačom a v každom okne môže hľadať informácie na rôznu tému. Preto pri personalizácii na základe cieľov používateľa je veľmi dôležité správne určiť trvanie sedenia.

Model používateľa sa najčastejšie reprezentuje pomocou stereotypov alebo pomocou prekryvného modelu [10]. Stereotypy sú vymedzené kategórie, do ktorých je používateľ zaradený. Príkladom takýchto kategórií môže byť „Začiatočník“, „Pokročilý“ alebo „Expert“. Inicializácia stereotypu, do ktorého používateľ patrí, sa najčastejšie vykonáva pomocou dotazníka na začiatku práce so systémom. Pri ďalšej práci sa už stereotyp určuje na základe interakcie so systémom.

V prekryvnom modeli sú znalosti používateľa reprezentované množinou konceptov a vzťahmi medzi nimi. Tieto tvoria akúsi sieť, v ktorej je obsiahnutý model používateľa. Pre každého používateľa sa dá určiť vedomosť o koncepte a zistením vedomostí o každom koncepte sa dá získať aj informácia o používateľovej znalosti domény ako celku.

3.2 Čo prispôbovať

Druhou dôležitou otázkou, ktorou je potrebné sa zaoberať je to, ktorú časť obsahu je treba používateľovi prispôbovať. Tieto časti sa dajú rozdeliť na dve časti [9]:

- adaptívna prezentácia a
- adaptívna navigácia.

Adaptívna prezentácia znamená prispôbovanie prezentovaného obsahu na základe modelu používateľa, ktorý si systém vytvoril. Z adaptívnej prezentácie sa občas oddeľuje prispôbovanie obsahu ako samostatná časť [7]. Vtedy sa pod adaptáciou prezentácie myslí zmena formy prezentovaného obsahu, napríklad zmenou farby alebo veľkosti písma a pod prispôbením obsahu sa myslí modifikácia samotného obsahu dokumentu.

Na prispôbovanie obsahu sa používajú napríklad techniky:

- *Vkladanie/odstraňovanie fragmentov*, kedy sa do dokumentu pridávajú vhodné fragmenty, alebo sa naopak odstraňujú tie nevhodné v závislosti od modelu používateľa.

- *Alternatívne fragmenty* je možné použiť ak existujú viaceré varianty fragmentov, napríklad s rôznou náročnosťou.
- *Roztiahovací text (strečtext)*, teda text ktorý môže byť zobrazovaný v skrátenej alebo v rozťahnutej podobe.
- *Usporiadúvanie fragmentov* znamená usporiadanie fragmentov na základe ich vhodnosti. Môže to byť napríklad zoznam odkazov, kde na prvých priečkach sú tie najvhodnejšie z nich.

Adaptívna prezentácia v sebe zahŕňa napríklad:

- *Alternatívne štýly*, ak existuje viacero variant štýlov zobrazovaných fragmentov.
- *Anotáciu fragmentov*, kde niektoré fragmenty majú zmenený štýl zobrazenia, alebo sú k nim pridané rôzne grafické prvky ako značky alebo ikony, čím sa dosiahne ich zvýraznenie alebo naopak skrytie.

Prispôbovanie navigácie znamená poskytovať používateľovi pomoc tak, aby sa dostal k dokumentom, o ktoré bude mať pravdepodobne záujem. Takáto pomoc sa snaží o skrátenie cesty v hypertexte k potenciálne zaujímavým dokumentom, prípadne ak prezerané dokumenty nie sú priamo previazané s ostatnými zaujímavými dokumentami, tak k nim vytvára cestu. Na dosiahnutie týchto cieľov sa používajú techniky ako [9]:

- *Prispôsobenie doménovej mapy*, teda prispôbovanie štruktúry hypertextu, tak aby sa v ňom používateľ dokázal rýchlejšie dostať k zaujímavým dokumentom.
- *Odporúčanie dokumentov*, teda priame vedenie k potenciálne zaujímavým dokumentom poskytovaním zoznamu odporúčaných dokumentov.
- *Usporiadanie zoznamu odkazov*, kde sa predpokladá, že čím je odkaz zaujímavejší pre používateľa, tým je v zozname odkazov na vyššej pozícii. Pri tejto forme úpravy navigácie je však nevyhnutné dbať na to, aby časté zmeny v poradí odkazov nemiatli používateľov, ktorí si už na nejaké usporiadanie zvykli.
- *Skrývanie odkazov* k nerelevantným dokumentom. Pri použití tejto formy prispôsobenia navigácie treba rovnako zohľadniť riziko zmätenia používateľov.
- *Pridávanie poznámok k odkazom*. Tieto poznámky nemusia byť len vo forme vysvetľujúceho textu, ale napríklad aj zmeny v štýle písma, zmeny pozadia textu, pripojenia ikony alebo iného grafického prvku k časti textu.

Pre vyhľadanie miest, kde sa dajú použiť tieto techniky je možné použiť napríklad kolaboratívne odporúčanie [13], alebo odporúčanie založené na analýze obsahu. Systémy kolaboratívneho odporúčania sledujú správanie používateľov a pre konkrétneho používateľa upravujú navigáciu na základe správania sa jemu podobných používateľov. Systémy odporúčania založené na analýze obsahu sledujú obsah stránky a porovnávajú ho s modelom používateľa. Potenciálne zaujímavé dokumenty sa následne odporúčajú používateľovi.

V súvislosti s pridávaním poznámok do dokumentov a ich následným prispôbovaním používateľom je možné použiť spomínané techniky na prispôbovanie dokumentov, ako aj

ich rôzne kombinácie. Ako príklad môžeme uviesť zvýrazňovanie slov ku ktorým sú priradené poznámky pomocou anotácie fragmentov, skrývanie nevhodných poznámok alebo naopak zvýrazňovanie dôležitých poznámok. Rovnako ak používame poznámky vo forme rôznych tooltipov, v ktorých je zobrazený dodatočný obsah, prispôsobovanie dokumentov sa dá použiť aj na samotný obsah týchto poznámok.

4 Existujúce nástroje na poznámkovanie dokumentov

Nástroje na poznámkovanie dokumentov sa dajú rozdeliť do dvoch skupín: tie, ktoré nechávajú používateľa vytvárať poznámky k dokumentu, prípadne mu poskytujú pomocné funkcie pri vytváraní poznámok a tie, ktoré vytvárajú poznámky do dokumentu automaticky.

4.1 Nástroje podporujúce manuálne poznámkovanie

Analyzovali sme niektoré nástroje, ktoré poskytujú používateľovi funkcie na manuálne vytváranie poznámok. Tieto nástroje sú charakteristické tým, že pri poznámkovaní elektronických dokumentov sa často snažia napodobniť rôzne metódy vytvárania poznámok používané pri písaní poznámok do papierových dokumentov. Poskytujú teda rôzne funkcie na zvýrazňovanie slov, písanie doplnujúcich textov, umožňujú používať rôzne grafické značky a podobne.

4.1.1 Annotea

Annotea [15] je projekt, ktorý sa pod záštitou W3C, snaží využiť manuálne poznámkovanie webových stránok na vytváranie sémantickej vrstvy nad dokumentami. Poznámky sú reprezentované vo formáte RDF oddelene od poznámkovaného dokumentu. Annotea špecifikuje spôsob, akým sú poznámky pripájané k dokumentom a spôsob akým, sú poznámky uchovávané a získavané z RDF databáz a nie spôsob akým sú tieto poznámky vizualizované. Vznikli teda viaceré klientské aplikácie, ktoré poskytujú používateľské prostredie na zobrazovanie a vytváranie poznámok. Takýmito nástrojmi sú napríklad webový prehliadač Amaya, rozšírenie pre prehliadač Mozilla Firefox s názvom Annotea Ubimarks¹² alebo prehliadač Annozilla, postavený na prehliadači Mozilla Firefox. Klientské aplikácie komunikujú s anotačným serverom, kde uchovávajú vytvorené poznámky, alebo získavajú poznámky, ktoré pripájajú k zobrazovaným dokumentom.

Annotea umožňuje uchovávať metadáta k dokumentu, teda poznámky tvoria informácie, ktoré opisujú existujúci dokument. Používateľ pridáva do dokumentu poznámky (annotation), záložky (bookmark), témy (topic) alebo odpovede (reply). Každý z pridaných objektov je identifikovaný pomocou URI a je uložený vo formáte RDF. Poznámky sú teda oddelené od samotného dokumentu. Súbor s poznámkami môžu byť uložené na serveri, lokálne alebo môžu byť zverejnené priamo vo webových dokumentoch. Annotea umožňuje okrem vytvárania poznámok aj ich získavanie z rôznych zdrojov a následné pripájanie k prezeraným dokumentom. Používateľ si môže zvoliť nielen to z akého zdroja chce získavať anotácie k dokumentom, ktoré prezerá, ale môže si tiež zvoliť kam chce ukladať poznámky, ktoré sám vytvoril.

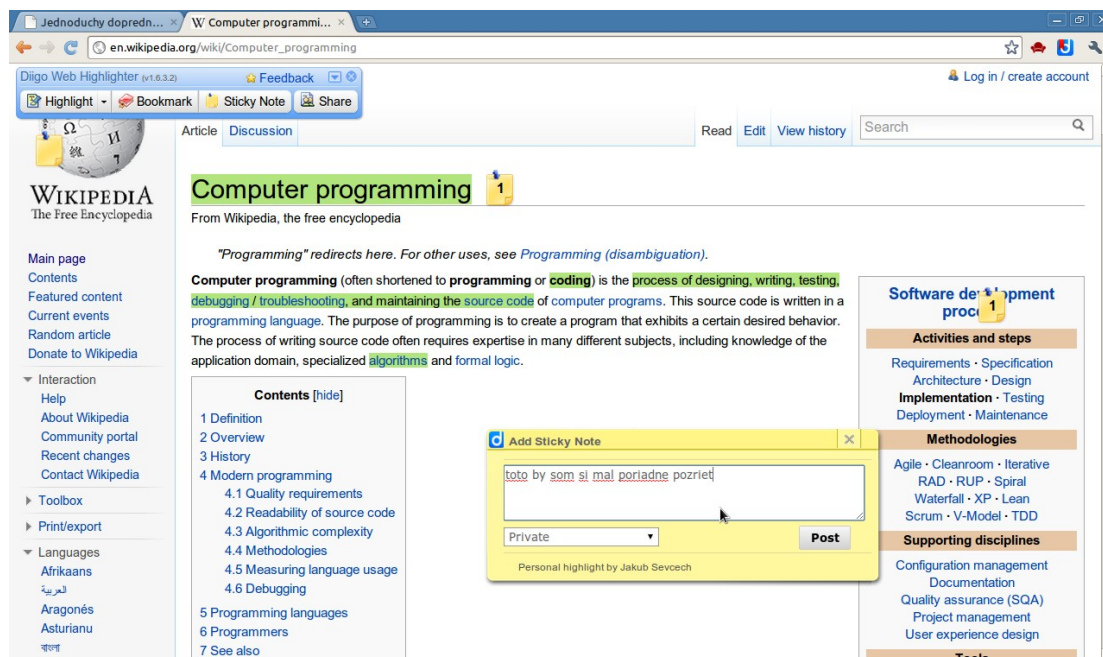
Annotea samotná je len architektúra na pripájanie poznámok k dokumentom, ich uchovávanie a získavanie. Používateľské rozhranie teda určuje klientská aplikácia, ktorá sa použije na prácu s týmito poznámkami. Annotea umožňuje značnú voľnosť ako pri vizualizácii poznámok tak aj pri ich získavaní z anotačných serverov a uchovávaní.

¹² Annotea Ubimarks, <http://www.annotea.org/mozilla/ubi.html>

4.1.2 Diigo

Diigo (Obr. 4) je služba dostupná vo forme lišty do všetkých najpoužívanejších webových prehliadačov, ako aj vo forme aplikácie do mobilných zariadení ako sú iPhone a iPad alebo telefónov s operačným systémom Android. Používateľ si môže pomocou tejto služby upravovať webové stránky a tieto úpravy ukladať na server. Pri ďalšej návšteve opoznámkovanej stránky sa už používateľovi nezobrazuje pôvodná stránka, ale tá, ktorú si upravil. Služba dovoľuje zvýraznenie textu, pridávanie poznámok do webových stránok a vytváranie záložiek s použitím tagov na ich organizáciu. Dovoľuje tiež zaznamenávanie webových stránok do formy obrázku, s možnosťou pridania orámovania k zaujímavým častiam, prípadne s možnosťou pridania šípok a iných značiek spolu s textom na popísanie vytvoreného obrázku.

Diigo umožňuje tiež vytvárať skupiny používateľov. V rámci skupiny je možné zdieľať vytvorené poznámky. Zdieľaním poznámok v rámci skupiny je možné kolaboratívne vyhľadávať zaujímavé stránky a pridávať ďalší obsah do stránok.



Obr. 4: Príklad webovej stránky opoznámkovanej pomocou nástroja Diigo

4.1.3 ALEF

ALEF [24] je adaptívny výučbový rámec vyvíjaný na Fakulte informatiky a informačných technológií. Tento nástroj umožňuje používateľom rozširovať informácie vo výučbových materiáloch tým, že im umožňuje pridávať rôzne druhy poznámok k výučbovým textom, ktoré obsahuje. Podporované je:

- pridávanie komentárov k častiam textu,
- hlásenie chýb v texte,

- pridávanie externých odkazov k častiam textu,
- zvýrazňovanie dôležitých častí textu a
- vytváranie tagov.

Poznámky sú zobrazované zvýraznením opoznámkovaného textu, pomocou bočného pásiku, ktorý zobrazuje pozíciu poznámky alebo pomocou rôznych zásuvných modulov (Obr. 5). Zásuvné moduly slúžia na vytváranie rôznych druhov poznámok, ich zobrazovanie a prácu s nimi. ALEF poskytuje tiež možnosť filtrovať poznámky podľa ich typu, čo zabezpečuje ich prehľadné použitie aj keď ich počet postupne rastie.

The screenshot shows the ALEF web application interface. At the top, there is a navigation bar with the ALEF logo and menu items: Administrácia, Debug, and SI. The user is identified as Jakub Sevcich (administrátor) with an 'Odhlásiť' button. The main content area displays a course page for '1.2 Softvérové inžinierstvo - dôležité pojmy'. The page includes a sidebar with a list of topics, a main content area with a diagram and text, and a right sidebar with a score of 4.7, a list of reported errors, and external sources. The diagram shows 'Čas' and 'Cena' pointing to 'Splnenie požiadaviek na výrobok', which is labeled 'Kvalita'. Below the diagram, there is a list of bullet points related to quality and standards.

Obr. 5: Príklad poznámok vo výučbovom rámci ALEF

4.2 Nástroje na automatické poznámkovanie

Automatickému poznámkovaniu dokumentov sa v súčasnosti venuje pomerne veľa úsilia, či už v oblasti výskumu alebo nasadzovania aplikácií. V tejto časti prezentujeme viac nástrojov ako v časti týkajúcej sa manuálneho poznámkovania, keďže aj naša práca sa zameriava na automatické poznámkovanie webových stránok.

4.2.1 Wikify!

Wikify! [20] je nástroj, ktorý spája metódu na automatické vyhľadávanie kľúčových slov v texte a metódu na rozpoznávanie významu slov, založenú na analýze článkov vo Wikipédii. S pomocou týchto dvoch metód pridáva k slovám vo webových stránkach odkazy na články vo Wikipédii. Nástroj v texte automaticky identifikuje koncepty a spojí ich s článkami na Wikipédii. Poznámky majú teda v tomto nástroji podobu odkazov na článok vo Wikipédii, súvisiaci s anotovaným slovom.

Nástroj vytvára pre každú webovú stránku podobnú štruktúru odkazov ako môžeme vidieť v článkoch na Wikipédii. Teda k dôležitým slovám v texte sa pridáva odkaz na súvisiacu stránku. Pri pridávaní odkazov do dokumentu nástroj napodobňuje postup, ktorým takéto

odkazy pridáva do dokumentu samotný autor článku vo Wikipédii. Identifikuje pravdepodobne najdôležitejšie slová a vety v texte a následne rozpoznáva presný význam týchto slov a hľadá k nim najvhodnejšie existujúce články.

Previazanie textov s článkami z Wikipédie umožňuje čitateľovi dokumentu rýchlo nájsť informácie, ak potrebuje vysvetlenie slova alebo dodatočné informácie. Nástroje na vytváranie týchto odkazov sú nezávislé od použitého jazyka. Je teda možné použiť túto metódu na spracovanie textov z rôznych jazykov a na ich previazanie s rôznymi jazykovými edíciami Wikipédie.

4.2.2 zLinks

zLinks¹³ je služba, ktorá na strane servera pridáva k odkazom na stránkach dodatočné informácie a odkazy na iné súvisiace stránky. Funguje ako plugin do redakčného systému WordPress¹⁴. Doména, pre ktorú služba poskytuje poznámky je teda obmedzená stránkami, ktoré obsahuje daný redakčný systém. Služba pridáva ikonu ku každému odkazu, ku ktorému dokáže poskytnúť dodatočné informácie. Po presnutí myši nad túto ikonu sa zobrazí tooltip s dodatočnými informáciami k odkazu a s ďalšími odkazmi na súvisiace stránky. Informácie, ktoré sa pridávajú k odkazom sa získavajú z prostredia LinkedData, zo zdrojov ako FOAF alebo DBpedia.

Služba tiež ponúka možnosť pridávať poznámky k odkazom priamo používateľmi. Poznámky je možné označiť ako verejné, čo spôsobí, že všetci návštevníci stránky budú môcť pri tomto odkaze vidieť aj vytvorenú poznámku. Táto funkcia je však len vedľajšou funkcionalitou, hlavným účelom nástroja je automatické pridávanie informácií k odkazom.

Tento nástroj teda pridáva dodatočné informácie k odkazom umiestnených v stránkach spravovaných pomocou redakčného systému s nainštalovaným pluginom zLinks. Nevýhodou tohto riešenia je, že ak používateľ číta text na stránke a hľadá nejaké dodatočné informácie, tak priamo nasleduje odkaz v texte a nesnaží sa najskôr si o tomto odkaze zobraziť dodatočné informácie. Túto funkcionalitu teda návštevník využije len vtedy ak sa vrátil naspäť na stránku, pretože nenašiel informácie, ktoré hľadal. Zaujímavejšie použitie získaných informácií by bolo, ak by sa nepriradňovali k existujúcim odkazom ale by sa pomocou nich vytvárali odkazy na tých miestach, kde sú v stránke potrebné alebo by sa tieto informácie priradňovali priamo k dôležitým slovám v texte.

4.2.3 Pannda

Nástroj Pannda [1] vyhľadáva dôležité slová v dokumente a dopĺňa k nim dodatočné informácie. Poznámky sa do webovej stránky pridávajú počas sprístupňovania stránky. Informácie na naplnenie poznámok sa získavajú z predpripravanej bázy údajov uchovávané vo forme ontológií. Pri vytváraní poznámok je teda obmedzená doména informácií, ktoré dokáže táto služba poskytnúť.

Poznámky sú naviazané ku konkrétnym slovám v texte stránky. Slová na priradenie poznámky sa vyhľadávajú porovnávaním slov v texte stránky so slovami uchovanými v báze

13 zLinks, <http://zitgist.com/products/zlinks/zlinks.html>

14 WordPress, <http://wordpress.org/>

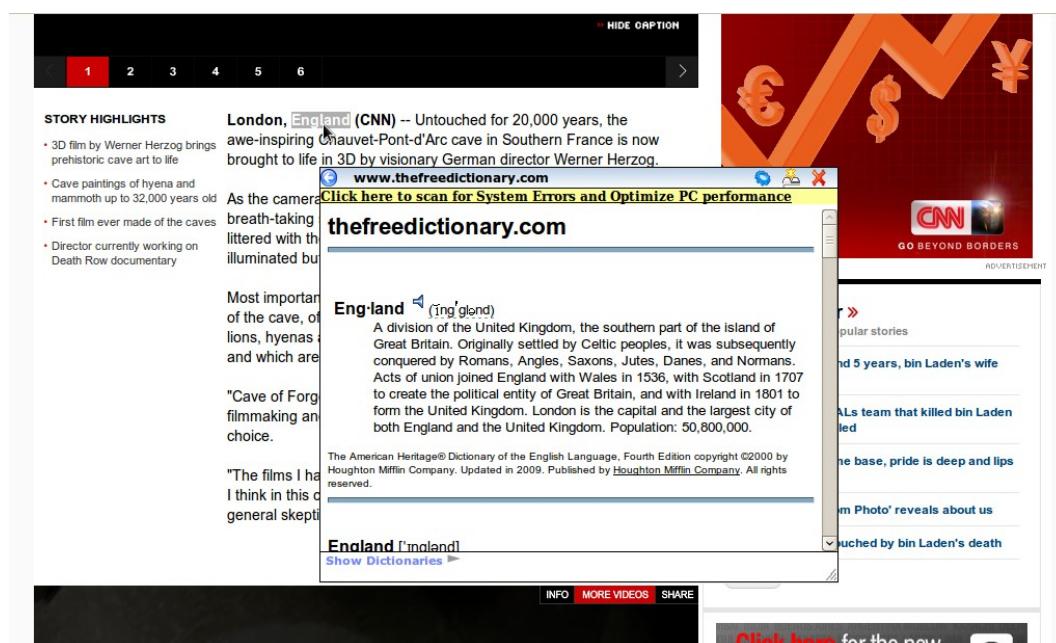
dát, pomocou regulárnych výrazov alebo na základe vzorov závislých na jazyku. Následne sa z bázy dát vyberajú dodatočné informácie, ktoré sa použijú na naplnenie poznámok.

Poznámky sú zobrazované ako malá ikona vedľa anotovaného slova. Toto riešenie však nie je veľmi vhodne zvolené, pretože mení usporiadanie textu na stránke. Pri väčšom množstve poznámok v texte sú už zmeny v zarovnaní textu príliš veľké a spolu s veľkým množstvom samotných ikon zhoršujú orientáciu v texte.

Po kliknutí na ikonu pri anotovanom slove sa otvorí nové okno s obsahom anotácie. Systém podporuje niektoré akcie používateľa nad poznámkami a to: pridanie komentáru a označenie poznámky za správnu, prípadne chybnú. Systém je postavený ako klient-server aplikácia, kde klient vo forme rozšírenia pre internetový prehliadač, zobrazuje poznámky používateľovi a poskytuje rozhranie pre interakciu s poznámkami. Serverová časť vykonáva analýzu dokumentu a pridávanie poznámok do dokumentu. Poznámky sú posielané zo serveru oddelene od samostatného dokumentu a to vo formáte vytvorenom v projekte Annotea.

4.2.4 Dictionary Tooltip

Dictionary Tooltip (Obr. 6) je nástroj vo forme rozšírenia pre webový prehliadač Firefox alebo Google Chrome. Poskytuje definície ľubovoľného slova na internetovej stránke. Po dvojkliku na slovo na stránke sa zobrazí tooltip s definíciou slova. Nástroj získava definície z množstva rôznych on-line slovníkov a služieb. Medzi použitými slovníkmi je napríklad dictionary.com, alebo urbandictionary.com, ale použité je aj vyhľadávanie článkov z Wikipédie alebo odkazov pomocou Google Search. Informácie sú vyhľadávané na základe dopytov na použité služby, kde dopyt tvorí samotné spracovávané slovo. Kvalita výsledných informácií teda závisí výhradne od kvality služieb použitých na ich vyhľadanie.



Obr. 6: Definícia slova pomocou nástroje DictionaryTooltip

Pri ponechaní pôvodných nastavení, nástroj zobrazuje zoznam definícií z rôznych slovníkov v jednom okne. Pri zvolení konkrétneho slovníka alebo inej stránky, aplikácia vyhľadá definíciu slova pomocou zvoleného nástroja a v tooltipe zobrazí celú stránku nástroja, v ktorom hľadal informácie. To že sa v tooltipe zobrazuje pôvodná stránka služby, bez úprav, uberať na použiteľnosti nástroja. Ak totiž chce používateľ použiť viacero služieb na vyhľadanie informácií k danému slovu, musí sa pre každú službu najskôr naučiť aká je jej grafická úprava a ako sa ovláda a až potom môže efektívne využívať poskytnuté informácie. Ak by mal nástroj vytvorený mechanizmus, pomocou ktorého by dokázal získať informácie z každej z použitých služieb a dokázal by ich prezentovať v jednotnej forme, používateľ by si zvykol len na jednu formu prezentácie a dokázal by pracovať s hociktorou z ponúkaných služieb oveľa rýchlejšie a pohodlnejšie.

Pri opakovanom získavaní informácií o tom istom slove nástroj nezískava odznova všetky informácie z webových služieb, ktoré používa, ale pri prvom zobrazení definície k slovu si túto definíciu uloží na strane používateľa. Vďaka tejto báze definícií nemusí používateľ čakať na vytvorenie poznámky k slovu vždy, ale stačí raz pri prvom zobrazení definície. Keďže sa informácie o slove ukladajú po prvom zobrazení na strane klienta, nástroj umožňuje pridať k týmto informáciám aj vlastnú textovú poznámku.

4.2.5 Gnosis

Gnosis (Obr. 7) je jedna z mnohých aplikácií, ktoré využívajú službu Open Calais¹⁵. Tento nástroj analyzuje webové stránky pri ich pristupovaní a získava z nich rôzne pomenované entity ako osoby, miesta, organizácie a podobne. Tieto pomenované entity sú zvýraznené v texte stránky a po prejení myšou nad takto zvýrazneným slovom sa zobrazí tooltip, v ktorom sú zobrazené zdroje, v ktorých je možné vyhľadať dodatočné informácie o tomto slove. Podporované zdroje sú napríklad Google a Wikipédia, ale aj služby ako Technorati (<http://technorati.com>) a Reuters (<http://www.reuters.com>).

Nástroj funguje ako rozšírenie do internetového prehliadača Firefox alebo Internet Explorer. Po inštalácii má nástroj nastavený predvolený zoznam webových sídiel, na ktorých vytvára poznámky do stránok. Používateľ si ale môže pridať ďalšie sídla, pre ktoré si praje zvýrazňovať pomenované entity. Medzi stránkami v základnej ponuke nástroja sú stránky ako: BBC (<http://www.bbc.com/>), Wikipedia (<http://www.wikipedia.org/>) alebo New York Times (<http://www.nytimes.com/>).

Nástroj poskytuje tri základné funkcie:

1. Zhrnutie v bočnom paneli, teda zoznam všetkých slov alebo fráz, ku ktorým vie na stránke vygenerovať anotácie. Tento zoznam je roztriedený podľa kategórií, do ktorých patria vyhľadané pomenované entity. Kategórie, ktoré nástroj rozlišuje sú napríklad osoby, spoločnosti alebo krajiny.
2. Automatické spracovanie stránky pri jej načítavaní. Pri načítavaní stránky sa asynchrónne vyhľadávajú slová, ku ktorým je možné vyhľadať dodatočné informácie.

15 Open Calais, <http://www.opencalais.com/>

- Poznámky sa vytvárajú len v stránkach, ktoré používateľa zaujímajú, teda tie, ktoré si pridal do zoznamu stránok na spracovanie.



Obr. 7: Poznámky vytvorené pomocou služby Gnosis

4.3 Zhodnotenie existujúcich riešení

V súčasnosti existuje veľké množstvo riešení, ktoré sa snažia rôznymi formami pridávať poznámky do textu. Nástroje na manuálne poznámkovanie používajú množstvo rôznych metód na podporu používateľov vo vytváraní poznámok do textu. Mnohé z nich podporujú kolaboráciu čitateľov dokumentov pri tvorbe poznámok. Túto funkcionality najčastejšie podporujú možnosťou zdieľať poznámky medzi všetkými používateľmi ale častokrát aj medzi jednotlivými skupinami používateľov.

Nástroje pre automatické poznámkovanie zas používajú rôzne spôsoby na vyhľadanie častí dokumentu, ktoré sú vhodné na priradenie poznámok. Často sa používajú metódy na vyhľadanie pomenovaných entít v texte ale aj metódy, ktoré vyhľadávajú slová z predpripravenej bázy dát pomocou porovnávania slov a pomocou regulárnych výrazov. Na vyhľadanie informácií na naplnenie poznámok sa tiež používajú rôzne metódy, od metód založených na vyberaní informácií z predpripravenej bázy dát, po vyhľadávanie informácií pomocou verejne dostupných služieb na vyhľadávanie informácií.

Všetky analyzované nástroje na automatické vytváranie poznámok sa zameriavali na dokumenty v anglickom jazyku. Jedinú výnimku tvoril nástroj Wikifi!, ktorého niektoré súčasti sa dajú použiť nezávisle od jazyka. Zaujímavým smerom ďalšieho rozvoja by teda bolo vytváranie poznámok do textov v inom ako anglickom jazyku

Iným zaujímavým smerom je vytváranie a zobrazovanie poznámok prostredníctvom rôznych mobilných zariadení. Spomedzi analyzovaných riešení len nástroj Diigo poskytoval možnosť vytvárať poznámky prostredníctvom mobilných telefónov.

5 Metóda na poznámkovanie webových stránok

V predchádzajúcej kapitole sme analyzovali niekoľko existujúcich riešení na automatické vytváranie poznámok do webových stránok. Väčšina z opísaných riešení sa však zameriavala na vytváranie poznámok do dokumentov v angličtine. My navrhujeme metódu na vytváranie poznámok do webových stránok v slovenskom jazyku s potenciálom použiť ju aj pre iné jazyky s podobnou štruktúrou. Vytvorené poznámky sú pripojené ku kľúčovým slovám v texte stránky. Účelom poznámok je definovať kľúčové slovo, ku ktorému sú priradené, alebo poskytnúť k nemu dodatočné informácie, ktoré rozšíria informácie obsiahnuté v texte webovej stránky. Metódu navrhujeme pre doménu výučbového systému, ale je použiteľná aj širšie.

5.1 Vytvorenie poznámok k dokumentu

Vstupom do procesu vytvárania poznámok je text dokumentu v slovenskom jazyku, ku ktorému chceme poznámky pridať. Výstupom sú poznámky oddelené od dokumentu. Poznámky sú s dokumentom previazané prostredníctvom kľúčových slov, ku ktorým boli vytvorené. Tieto kľúčové slová určujú, ku ktorej časti dokumentu je poznámka priradená. Pre vytvorenie kvalitných poznámok je teda potrebná spoľahlivá metóda na extrakciu kľúčových slov, ku ktorým následne môžeme vytvoriť poznámky.

V súčasnosti sú uspokojivé výsledky strojového spracovania textu dosahované len pri spracovaní anglického textu. Tomuto faktoru nasvedčovali aj naše pokusy, kde sme porovnávali kvalitu kľúčových slov extrahovaných zo slovenského textu, textu preloženého po slovách do angličtiny a textu preloženého do angličtiny. Prvé dva prístupy poskytovali neporovnateľne horšie výsledky ako prístup, kde sme extrahovali kľúčové slová z textu preloženého do angličtiny. Z tohto dôvodu poznámkovaný text najskôr preložíme do angličtiny. Bez prekladu textu do angličtiny by výrazne klesla kvalita nájdených kandidátov na priradenie poznámky a teda aj kvalita výsledných poznámok.

Ako sme už v predchádzajúcich častiach spomínali kandidáti na priradenie poznámok môžu byť rôzne kľúčové slová, pomenované entity ale aj väčšie časti dokumentu. Keďže extrahovaním kľúčových slov z textu preloženého do angličtiny získame slová v angličtine a my chceme poznámky priradiť k slovám v slovenčine, potrebujeme týchto kandidátov na priradenie poznámok spojiť s ich ekvivalentami v pôvodnom texte. Na tento krok sme navrhli metódu na mapovanie slov medzi textom a jeho prekladom. Po získaní kandidátov na vytvorenie poznámky, ktorých vieme priradiť k slovám v pôvodnom texte, môžeme pokračovať vo vytváraní poznámok vyhľadaním informácií na ich naplnenie.

Posledným krokom pri poznámkovaní dokumentu je spojiť nájdené informácie do poznámok a zobraziť ich používateľovi. Metóda pre poznámkovanie dokumentu, ktorú sme navrhli, sa skladá z nasledujúcich šiestich krokov:

1. Preklad textu do angličtiny.
2. Vyhľadanie kandidátov na pridanie poznámky.
3. Spojenie nájdených kandidátov so slovami v pôvodnom texte

4. Vyhľadanie informácií na naplnenie poznámok
5. Vytvorenie poznámok
6. Umiestnenie poznámok do textu

5.1.1 Mapovanie slov medzi textom a jeho prekladom

Pri automatickom preklade textu do cudzieho jazyka je takmer isté, že spätný preklad bude mať oproti pôvodnému textu veľa odlišností, dokonca v niektorých príkladoch úplne iný význam. Zároveň však chceme vyhľadávať informácie v rôznych jazykoch. Objavuje sa teda problém ako spojiť informácie, ktoré sme našli k preloženému textu, späť k slovám v pôvodnom texte [5].

Väčšina metód, ktoré sa používajú na mapovanie slov v paralelných textoch, využíva rôzne štatistické metódy [11], ale používajú sa tiež metódy založené na pozícii slov v texte [6] a metódy založené na prístupoch využívajúcich bilingválny slovník.

My sme navrhli metódu, ktorá na spájanie prekladov slov používa bilingválny slovník. Pri spájaní prekladov sme sa sústredili na fakt, že v slovenčine existuje viac tvarov toho istého slova, keďže slovenčina je flektívny jazyk. Na mapovanie všetkých tvarov slov by sme potrebovali slovník, v ktorom by boli všetky tvary slov a k nim priradené preklady. Takýto slovník by bol však obrovský, ťažko by sa s ním pracovalo a ešte ťažšie by sa vytváral. Rozhodli sme sa preto použiť oveľa menší slovník a metódu na mapovanie sme navrhli tak aby si dokázala poradiť s rôznymi tvarmi slova. Predpokladáme, že na zohľadňovanie rôznych tvarov slov nám stačí, ak budeme so slovami pracovať na úrovni reťazcov znakov. Dôležitejšie ako presná metóda na vytváranie základných tvarov slov je pre nás efektívnosť navrhovanej metódy. Rozhodli sme sa teda použiť podobnú metódu ako Garabík [12] vo svojom morfológickom analyzátoze, kde sú slová porovnávané s ich základnou formou (lemou) pomocou Levenshteinovej vzdialenosti [16]. Levenshteinova vzdialenosť dvoch slov je vlastne minimálny počet Levenshteinových operácií potrebných na to, aby sme previedli jeden reťazec písmen na iný. Pri tomto prevode sú povolené tri operácie: vloženie písmena, odstránenie písmena a nahradenie písmena za iné.

My sme ďalej využili vlastnosť slovenčiny, že pri skloňovaní slova sa mení len koncovka, teda prípona slova. Zmena prípony slova mení jeho význam len minimálne, zatiaľ čo ak sa zmení jeho predpona alebo koreň slova, význam slova sa zmení výrazne. Preto sme menili cenu Levenshteinovej operácie podľa toho, na ktorej pozícii v slove sa táto operácia vykonala. Cenu Levenshteinovej operácie sme použili konštantnú na začiatku, teda v koreni slov a na posledných znakoch slova sme cenu Levenshteinovej operácie lineárne znižovali.

Pri mapovaní slov medzi textom a jeho prekladom predpokladáme, že tieto dva texty sú po vetách správne namapované, teda ekvivalentné vety sú v oboch textoch v rovnakom poradí.

Pri tomto predpoklade prechádzame postupne textom po vetách a spájame slovo s jeho prekladom v rámci viet. Toto spájanie prebieha tak, že si zvolíme jednu vetu z preloženého textu a prechádzame všetkými slovami v tejto vete. K slovu nájdeme pomocou slovníka všetky jeho preklady a vypočítame Levenshteinovu vzdialenosť medzi každým prekladom slova a slovami v slovenskej vete. Pre všetky dvojice slov v spracovávaných vetách,

stanovíme vzdialenosť ako najmenšiu Levenshteinovu vzdialenosť spomedzi vzdialeností medzi prekladmi anglického slova získanými zo slovníka a slovom zo slovenskej vety. Tie dvojice slov, ktoré majú menšiu Levenshteinovu vzdialenosť ako nejaká stanovená hranica, prehlásime za namapované ekvivalenty.

Algoritmus na mapovanie ekvivalentných slov sa dá opísať aj nasledujúcim pseudokódom:

```
original_sentences = split_to_sentences(original_text)
translated_sentences = split_to_sentences(translated_text)
for i = 0 to |original_sentences| do
    for each word ow in original_sentences[i] do
        for each word tw in translated_sentences[i] do
            translations = find_translations_for(tw)
            for each translation t in translations do
                if levenshtein_distance(t,ow) < threshold
                    declare_mappings(ow,tw)
                end
            end
        end
    end
end
```

Tento algoritmus je jednoducho dátovo paralelizovateľný, keďže algoritmus sekvenčne prechádza spracovávaný text po vetách a spracovanie jednotlivých viet na sebe nijako nezávisí.

Keďže pri mapovaní slov sa považujú za ekvivalenty páry slov, pre ktoré platí, že preklad jedného slova a druhé slovo majú Levenshteinovu vzdialenosť menšiu ako nejaká hranica, tak sa môže stať, že k jednému slovu vo vete sa pripojí viacero slov z druhej vety. Takéto slová môžu byť napríklad synonymá, ktoré majú rovnaký preklad do druhého jazyka alebo rovnaké slová, ktoré sú v texte použité opakovane a môžu byť len inak vyskloňované.

5.1.2 Získavanie informácií

Na získanie informácií na naplnenie poznámok používame verejne dostupné služby na vyhľadávanie informácií. Iný prístup ako sa dajú získavať potrebné informácie je napríklad použitie predpripravenej bázy údajov.

Použitie verejne dostupných služieb nám poskytuje hneď niekoľko výhod oproti použitiu predpripravenej bázy dát ako zdroju informácií na naplnenie poznámok:

- Metóda na vytváranie poznámok je nezávislá na doméne. Pri použití predpripravenej bázy dát by sme museli pre každú novú doménu rozširovať existujúcu alebo vytvárať novú bázu dát, čo by si vyžadovalo nemalé úsilie.
- Poznámky môžu obsahovať rôzne druhy informácií (text, odkazy, obrázky ...) v závislosti od služieb použitých na ich získanie.

Za nevýhodu však možno považovať fakt, že kvalita získaných informácií, a teda aj výsledných poznámok, výrazne závisí od kvality služieb použitých na získanie informácií na naplnenie poznámok.

Získavanie informácií prebieha dvoma spôsobmi. Dva spôsoby sú zvolené preto, aby bolo množstvo informácií dostatočne veľké a rôznorodé.

Prvý spôsob získavania informácií je pomocou odkazu na zdroj v prostredí LinkedData. Webové služby ako AlchemyAPI¹⁶ alebo OpenCalais dokážu v texte vyhľadať koncepty alebo pomenované entity, o ktorých sa v texte hovorí a k týmto konceptom dokážu vrátiť aj odkaz na príslušný zdroj v prostredí LinkedData. Na vyhľadanie týchto informácií je možné použiť napríklad dopytovací jazyk SPARQL. Pre zväčšenie množstva informácií, ktoré sa takto dajú získať je možné okrem samotného zdroja využiť aj služby rôznych sémantických vyhľadávačov, ako napríklad Sindice¹⁷.

Druhý spôsob je vyhľadávanie informácií pomocou rôznych služieb, kde dopyt na získanie informácií je vo forme zoznamu kľúčových slov. Príkladom takejto služby je Google Search alebo Delicious¹⁸.

Predpokladáme, že tento druhý spôsob bude v porovnaní s prvým poskytovať oveľa viac informácií zo širšej oblasti, avšak súčasne predpokladáme, že tieto informácie budú obsahovať aj viac nerelevantných informácií. Preto získavanie informácií pomocou služieb, ktoré prijímajú dopyt vo forme zoznamu kľúčových slov, používame ako doplnkový spôsob získavania informácií, pre zväčšenie ich množstva.

Pri získavaní informácií sa snažíme nájsť dva druhy informácií: informácie vo forme odkazov na súvisiace stránky a vo forme textu, obrázkov alebo iného multimediálneho obsahu súvisiaceho so slovom, ku ktorému hľadáme poznámku.

5.2 Prispôbenie poznámok

Pre zvýšenie kvality a užitočnosti dokumentov sa tieto často prispôbujú pre potreby používateľov. Podobným spôsobom je možné prispôbovať aj poznámky. V kapitole 3 sme opisovali spôsoby, ktorými je možné interpretovať implicitnú spätnú väzbu od používateľov. Opísali sme tiež spôsoby ako možno prispôbovať dokumenty konkrétnemu používateľovi na základe modelu používateľa. V tejto časti navrhujeme metódu na prispôbovanie

¹⁶ AlchemyAPI, <http://www.alchemyapi.com/>

¹⁷ Sindice, <http://www.sindice.com/>

¹⁸ Delicious, <http://www.delicious.com/>

poznámok pre používateľov na základe implicitnej spätnej väzby získanej pozorovaním interakcie s poznámkami.

Metóda na vytváranie poznámok, pripája poznámky ku konkrétnym slovám v texte poznámkovaného dokumentu. Tieto poznámky sú vo forme definície slova a odkazov na stránky súvisiace so slovom, ku ktorému je poznámka pripojená. Prispôsobovanie poznámok sa teda dá robiť dvoma spôsobmi:

1. Prispôsobovaním formy, akou sa v texte označuje slovo, ku ktorému sa viaže poznámka.
2. Prispôsobovaním obsahu poznámky.

Pri návrhu metódy na prispôsobovanie poznámok sa sústreďujeme na upravovanie obsahu poznámky. Z metód na prispôsobovanie obsahu dokumentu opísaných v časti 3.2 používame usporiadanie zoznamu odkazov. Toto usporiadanie je na základe relevancie odkazu k poznámkovanému textu a slovu, ku ktorému je priradená poznámka. Pri tomto spôsobe upravovania obsahu poznámky sa predpokladá, že odkazy, ktoré sú na vyšších priečkach v zozname, používateľ zobrazí častejšie alebo skôr ako odkazy, ktoré sú v zozname umiestnené na nižších pozíciách. Pri preusporiadavaní obsahu dokumentov je potrebné dbať na to aby zmena dokumentu nepôsobila na používateľa rušivo. Pri prispôsobovaní poznámok nie je tento problém tak výrazný, ako pri prispôsobovaní celých dokumentov, pretože sa predpokladá, že používateľ bude opakovane používať tú istú poznámku len zriedka, zatiaľ čo dokument ako celok používateľa zobrazujú oveľa častejšie, a teda si rýchlejšie zvyknú na jeho presné usporiadanie.

5.2.1 Interpretovanie implicitnej spätnej väzby

Hodnotenie relevancie odkazov odvodzujeme z implicitnej spätnej väzby získanej pozorovaním toho, ktoré odkazy používateľa nasledujú. Kliknutie na odkaz interpretujeme ako tvrdenie, že zobrazený odkaz je lepší ako niektorý iný odkaz v zozname, podobne ako sme to opisovali v časti 3.1.1. Na získavanie týchto tvrdení navrhujeme použiť dve stratégie založené na pozícií zobrazených odkazov v zozname [14].

1. Zobrazený odkaz je lepší ako odkazy na vyšších pozíciách v zozname, ktoré neboli zobrazené.
2. Zobrazený odkaz je lepší ako bezprostredne nasledujúci odkaz, ak tento nebol zobrazený.

Takto získané tvrdenia sú tvorené trojicou $T = (u_z, u_p, v)$, kde:

- u_z – identifikácia zobrazeného odkazu, teda uzol, ktorý používateľ nasledoval
- u_p – identifikácia porovnávaného odkazu, teda odkaz v porovnaní s ktorým je zobrazený odkaz lepší
- v – váha tvrdenia

Váhu tvrdenia odvodzujeme dvomi spôsobmi v závislosti od toho, či je k dispozícii informácia o trvaní zobrazenia pre daný druh odkazu.

- Ak nie je k dispozícii informácia o trvaní zobrazenia, váha tvrdenia je vždy konštantná.
- Ak je k dispozícii čas trvania zobrazenia odkazu, váha tvrdenia je rovná spomínanej konštante len ak je toto trvanie dlhšie ako stanovený prah, inak je váha nastavená na nulu.

Interakciou používateľov s odkazmi, vloženými do poznámky získavame sadu tvrdení, pomocou ktorej môžeme určiť aké relevantné sú zobrazované odkazy.

5.2.2 Získanie poradia odkazov

Spôsobom opísaným v predchádzajúcej časti dokážeme získať z implicitnej spätnej väzby od používateľov sadu tvrdení o tom, že jeden odkaz je lepší ako niektorý iný. Pre získanie poradia odkazov, ktoré vyjadruje ich relevanciu, navrhujeme vytvoriť z týchto tvrdení a samotných odkazov orientovaný graf. Ohodnotením uzlov v tomto grafe získame ohodnotenie relevancie odkazov. Následným usporiadaním odkazov podľa ohodnotenia relevancie získame nové poradie odkazov.

V navrhovanom grafe sú uzli reprezentované odkazmi, ktoré chceme usporiadať. Hrany grafu tvoria tvrdenia o kvalite odkazov, kde hrana je orientovaná v smere od porovnávaného uzla k zobrazenému uzlu, teda k uzlu, ktorý je podľa tvrdenia lepší. Keďže pri vytváraní tvrdení o kvalite môže vzniknúť viacero tvrdení pre rovnaké dvojice odkazov, v grafe by nám vznikalo veľké množstvo násobných hrán. Takéto tvrdenia teda spojíme do jedného, tak aby výsledné tvrdenie malo váhu rovnú súčtu váh tvrdení z ktorých vzniklo. Vo vytvorenej sade tvrdení sa rovnako môžu nachádzať protirečivé tvrdenia, teda také, ktoré majú vymenený zobrazený a porovnávaný odkaz. Takéto tvrdenia sa vo výslednom grafe premietnu do dvoch, opačne orientovaných hrán.

Na takto vytvorený graf odkazov aplikujeme PageRank [21] algoritmus upravený tak, aby zohľadňoval váhy hrán. Pôvodný PageRank algoritmus sa dá vyjadriť rovnicou:

$$PR(p_i) = (1-d) + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)}$$

kde:

- p_i – i -ty uzol grafu, pre ktorý počítame jeho ohodnotenie
- $PR(p_i)$ – ohodnotenie vypočítané pre i -ty uzol grafu
- d – tlmiaci faktor (damping factor)
- $M(p_i)$ – množina uzlov, z ktorých smeruje hrana do uzlu p_i
- $L(p_j)$ – počet odchádzajúcich hrán z uzla p_j

Hodnotenie uzlu sa teda odvíja od počtu hrán, ktoré do neho smerujú a od hodnotenia uzlov, ktoré sú na opačných koncoch týchto hrán. V pôvodnom návrhu algoritmu PageRank

použitom na ohodnocovanie kvality internetových stránok je hodnota $1-d$ interpretovaná ako pravdepodobnosť, že používateľ pri prechádzaní stránok prejde na nejakú náhodnú stránku. Pri použití v nami upravenom algoritme sa dá táto hodnota interpretovať ako pravdepodobnosť, že používateľ klikol na nejaký odkaz náhodou alebo preto, lebo nevedel či je pre neho relevantný.

Upravený algoritmus okrem počtu hrán smerujúcich do uzlu a ohodnotení susedných uzlov, zohľadňuje aj váhy hrán smerujúcich do uzlu, pre ktorý počítame ohodnotenie. Takto upravený algoritmus je možné zapísať rovnicou:

$$PR(p_i) = (1-d) + d \sum_{p_j \in M(p_i)} \frac{PR(p_j) W(p_j, p_i)}{S(p_j)}$$

kde:

$W(p_j, p_i)$ – váha hrany smerujúcej z uzla p_j do uzla p_i

$S(p_j)$ – súčet váh odchádzajúcich hrán z j -teho uzla

Aplikovaním navrhovaného algoritmu na graf odkazov a tvrdení o ich kvalite získame ohodnotené uzli, teda odkazy. Zoradením odkazov podľa klesajúceho ohodnotenia získame nové poradie, ktoré vyjadruje to aké, dobré sú odkazy. Poznámku s takto preusporiadanými odkazmi môžeme následne zobrazit' používateľom.

Zoradovaním odkazov na základe tvrdení o ich kvalite sme dosiahli, že výsledné poradie odkazov nie je ovplyvnené poradím odkazov, v ktorom boli odkazy zobrazované pri získavaní spätnej väzby. Je teda možné používateľom zobrazovať poradie odkazov priebežne, počas práce s poznámkou a súčasne získavať spätnú väzbu na jeho ďalšie vylepšovanie. Odpadá teda fáza tréovania, keďže poradie sa môže vytvárať počas používania poznámok. Odkazy môžeme zobrazovať používateľom bez toho, aby sme mali na začiatku nejakú spätnú väzbu o ich kvalite. Správne poradie odkazov sa postupne vyvinie z interakcie so zobrazovaným zoznamom.

6 Realizácia navrhutej metódy

Navrhovaná metóda je realizovaná v dvoch častiach:

- webová služba na vytvorenie poznámok k dokumentom a
- modul do výučbového systému ALEF [24] na vizualizáciu poznámok a ich prispôsobenie na základe implicitnej spätnej väzby od používateľov.

Tým že sme oddelili vytváranie poznámok od zvyšnej časti systému a realizujeme ju ako webovú službu sme dosiahli to, že sme schopný vytvárať poznámky nezávisle na systéme, v ktorom sa zobrazujú. Toto nám dovoľuje vytvorené poznámky používať aj v iných systémoch. Možné použitie je napríklad v proxy-serveri, kde by sa poznámky pridávali do každej stránky, na ktorú používateľ cez takýto proxy-server pristupuje. Inou možnosťou je použiť webovú službu podobne ako my, kde sme poznámky pridávali na strane servera už pri samotnom vytváraní stránky. Táto služba sa dá použiť aj v rôznych redakčných systémoch ako je napríklad WordPress¹⁹, ale aj na strane klienta.

Doména výučbového systému, v ktorom sme realizovali zobrazenie samotných poznámok, nám poskytuje výborné podmienky na overenie navrhovaných metód. Samotný systém ALEF nám poskytuje funkcionality a údaje potrebné pri realizácii aplikácie, ako aj fungujúce prostredie, používané pri výučbe viacerých predmetov na Fakulte informatiky a informačných technológií.

6.1 Adaptive Learning Framework

Adaptive Learning Framework (ALEF) [24] je výučbový systém vyvíjaný na Fakulte informatiky a informačných technológií. Ako už z názvu systému vyplýva, vyznačuje sa tým, že je adaptívny a teda sa dokáže prispôbovať používateľom na základe modelu používateľa.

Údaje výučbového kurzu sú rozdelené do dvoch častí:

- Výučbové objekty vo forme vysvetľujúcich textov, otázok a cvičení
- Metadáta, teda koncepty, vzťahy medzi konceptami a vzťahy medzi výučbovými objektami a konceptami.

Na podporu prispôsobenia systému študentom je v vytvorený prekryvný model používateľa, kde sa znalosti používateľa zaznamenávajú ako znalosti jednotlivých konceptov z doménového modelu výučbového kurzu. Znalosti konceptov sú odvodené od toho, ako študent vyriešil cvičenia a otázky, ktoré mu systém ponúka. Takto získaný model používateľa sa využíva napríklad na odporúčanie textov a cvičení, ktoré by si mal študent preštudovať. ALEF poskytuje možnosť aby študenti sami tvorili a rozširovali obsah výučbových kurzov. Táto funkcionality je zabezpečená pomocou rôznych druhov poznámok, ktoré môžu študenti pridávať k jednotlivým výučbovým textom obsiahnutých v systéme. Študenti môžu napríklad vytvárať otázky a komentáre k textu, môžu hlásiť chyby, ktoré objavili v texte a pridávať rôzne externé zdroje, ktoré súvisia s výučbovým objektom. Systém poskytuje tiež možnosť zvýrazňovať zaujímavé časti textu a pridávať k nemu tagy.

¹⁹ WordPress, <http://wordpress.org/>

ALEF je rámec, ktorý umožňuje rozširovanie a tvorbu vzdelávacích aplikácií. Je navrhnutý tak aby bolo možné do neho jednoducho zapracovávať nové moduly. Túto vlastnosť využívame aj my, keďže navrhovaná metóda na zobrazovanie a prispôsobovanie poznámok je zapracovaná do systému ALEF ako jeden z jeho modulov.

ALEF je implementovaný v jazyku *Ruby*²⁰, s použitím rámca *Ruby on Rails*²¹. Údaje potrebné pre chod systému sú uchovávané v *MySQL* databáze. Používateľ komunikuje so systémom prostredníctvom webového rozhrania, kde sa vo veľkej miere využíva jazyk *JavaScript* a technológia *AJAX*. Tieto technológie používame aj na realizácii oboch častí navrhovanej aplikácie.

Bližší opis systému ALEF je uvedený v technickej dokumentácii v prílohe A a v používateľskej príručke v prílohe B.

6.2 Vytvorenie poznámok

Vytvorenie poznámky je realizované pomocou *REST*²² webovej služby, implementovanej v jazyku *Ruby*, s použitím rámca *Ruby on Rails*. Táto služba prijíma text alebo adresu webovej stránky v slovenčine. Ak je na vstupe adresa webovej stránky, použijeme službu *Readability*²³ na to aby sme zo stránky na tejto adrese získali text, ktorý posunieme na ďalšie spracovanie. Proces vytvárania poznámok je nasledujúci:

1. Preloženie textu do angličtiny pomocou služby *Google Translate*²⁴.
2. Vyhľadanie kľúčových slov v preloženom texte.
3. Nájdenie ekvivalentných slov k extrahovaným kľúčovým slovám pomocou metódy na mapovanie ekvivalentných slov medzi textom a jeho prekladom.
4. Vyhľadanie informácií na naplnenie poznámok.

Výstupom implementovanej webovej služby je *XML* alebo *JSON* dokument so slovami, ku ktorým boli vytvorené poznámky a samotné poznámky. Poznámky sú vo forme definície kľúčového slova, odkazov na súvisiace stránky a vo forme prezentácií. Implementovaná služba je dostupná na adrese: <http://leela.fiit.stuba.sk/word-definition>

Poznámky k jednotlivým kľúčovým slovám sú uchovávané v *MySQL* databáze, pre prípad opakovaného vytvárania poznámok pre rovnaké slová. Pri opätovnom vyžiadaní poznámky k tomu istému slovu služba nezískava informácie nanovo, ale použije informácie uložené v databáze. Takto sme dosiahli zvýšenie priemernej rýchlosti poskytovania poznámok. Dátový model použitý na reprezentáciu týchto údajov je zobrazený na obrázku číslo 8.

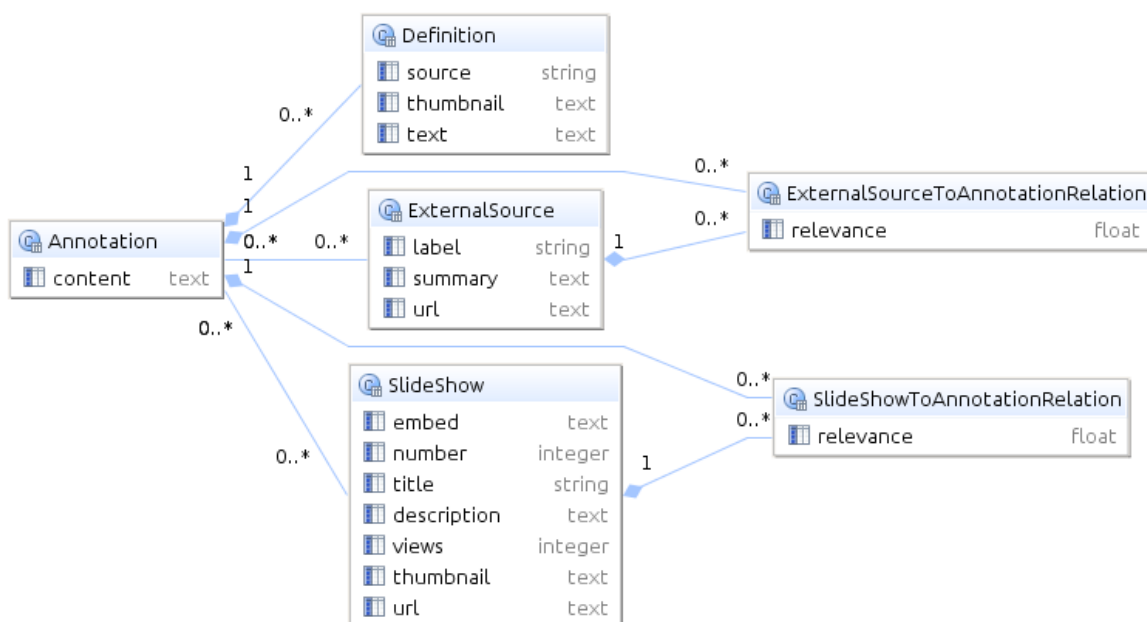
20 Ruby, <http://www.ruby-lang.org/en/>

21 Ruby on Rails, <http://rubyonrails.org/>

22 REST, http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

23 Readability, <https://www.readability.com/>

24 Google Translate, <http://translate.google.com>



Obr. 8: Dátový model webovej služby na vytvorenie poznámok

6.2.1 Mapovanie slov medzi textom a jeho prekladom

Realizácia metódy na mapovanie ekvivalentných slov je implementovaná v jazyku *Ruby* ako časť webovej služby na vytváranie poznámok. Na hľadanie mapovania ekvivalentných slov medzi slovenským textom a jeho prekladom do angličtiny je použitý bilingválny slovník. Ako slovník je pri hľadaní mapovania použitý Malý Anglicko-Slovenský slovník²⁵ vytvorený pre podporu Open Source programov. Rôzne tvary slov sú porovnávané na základe ich Leveshteinovej vzdialenosti. Pri hľadaní mapovania slov postupujeme podľa algoritmu opísaného v časti 5.1.1. Jednotlivé korky mapovania ekvivalentných slov sú nasledovné:

1. Preloženie textu do angličtiny.
2. Rozdelenie textu na vety.
3. Odstránenie stop-slov, teda slov, ktoré v texte nenesú žiadny význam. Takéto slová sú napríklad spojky a častice.
4. Prechod textu po vetách a vyhľadanie mapovania s pomocou Levenshteinovej vzdialenosti.

Pri výpočte Levenshteinovej vzdialenosti meníme cenu jednotlivých Levenshteinových operácií v závislosti od pozície v slove, na ktorej sa vykonávajú. Využívame pri tom fakt, že zmena v koreni slova mení v slovenčine význam slova výrazne, zatiaľ čo zmena v koncovke slova zmení len tvar slova, zatiaľ čo jeho význam ovplyvní len minimálne. Cena Levenshteinovej operácie je konštantná na začiatku slova, teda v koreni slova a lineárne klesá ku na posledných znakoch slova.

25 Malý Anglicko-Slovenský slovník, <http://www.sk-spell.sk.cx/mass-msas>

Nasledujúci úsek kódu zobrazuje funkciu použitú na výpočet ceny jednotlivých Levenshteinových operácií:

```
def self.cost(position, wordLength)
  return 2.0 if position <= 3
  return 2.0 if (position <= wordLength-6)
  r = 3 > wordLength-6 ? 3 : wordLength-6
  return 1-((position-r)/(wordLength-r+1.0))
end
```

Vytvorili sme 4 verzie funkcie na mapovanie slov.

1. Základná funkcia, je implementáciou navrhovanej metódy, ktorá na mapovanie slov využíva Levenshteinovu vzdialenosť.
2. Druhá používa na nájdenie ekvivalentov k slovám, ktoré sa nepodarilo namapovať základnou metódou, dodatočný predpoklad. Predpokladáme, že ak sú dve slová v jednom texte vedľa seba, tak budú vedľa seba aj v preklade tohto textu. Doplnili sme teda do základnej funkcie časť, ktorá po namapovaní slov pomocou Levenshteinovej vzdialenosti prechádzala namapované dvojice slov a ak má dvojica v oboch textoch nenamapovaného suseda, tak týchto susedov spojí a označí ich za ekvivalentné preklady.
3. Ďalšia funkcia je opäť vylepšením základnej funkcie. Tentokrát sme upravili slovník, s ktorým základná funkcia pracuje. V slovníku, ktorý sme používali neboli všetky tvary anglických slov. Previedli sme teda anglické slová v tomto slovníku na základný tvar pomocou Porterovho algoritmu [22]. Následne sme v slovníku nevyhľadávali anglické slová ale základné tvary anglických slov. Takto sme dosiahli, že všetky tvary vyhľadávaných anglických slov sa zlúčili do jedného základného tvaru.
4. Posledná, štvrtá funkcia je kombináciou oboch predchádzajúcich vylepšení, teda pracuje s upraveným slovníkom a zároveň zohľadňuje pozíciu slov vo vete.

Na základe overenia kvality a množstva namapovaných slov pomocou všetkých štyroch funkcií sme sa rozhodli pri realizácii webovej služby na vytváranie poznámok použiť funkciu používajúcu druhé vylepšenie, teda pracujúcou s predspracovaným slovníkom. Postup a výsledky realizovaných experimentov sú opísané v časti 7.1.

6.2.2 Získanie informácií na naplnenie poznámok

Získavanie informácií na naplnenie poznámok prebieha dvomi spôsobmi:

- vyhľadanie informácií na základe dopytu vo forme kľúčového slova a
- vyhľadanie informácií v prostredí LinkedData.

Pre získavanie kľúčových slov z textu, je použitá webová služba AlchemyAPI. Pre extrahované kľúčové slová následne vyhladáme ich ekvivalenty v pôvodnom texte. Na tento krok je použitá metóda na mapovanie ekvivalentných slov navrhnutá v časti 5.1.1. Tieto mapovania sú potrebné na to, aby sme dokázali určiť, ku ktorým slovám v texte webovej stránky je potrebné priradiť poznámky. Okrem toho tieto slová využívame aj na vyhládanie informácií na naplnenie samotných poznámok. Na vyhládanie týchto informácií sú použité nasledujúce služby:

- Google Search, kde výsledkom je zoznam odkazov na stránky vyhovujúce dopytu.
- DictService²⁶, kde výsledkom je definícia slova zadaného ako dopyt.
- SlideShare²⁷, kde výsledok je prezentácia vyhládaná na základe kľúčových slov.

Služba AlchemyAPI použitá na vyhládanie kľúčových slov, dokáže okrem kľúčových slov vyhládať aj koncepty, o ktorých text hovorí a pomenované entity, ktoré sa v texte nachádzajú. Spolu s nájdenými konceptami a pomenovanými entitami služba AlchemyAPI vracia aj URI, ktorou sú identifikované medzi zdrojmi napríklad DBpedia alebo FOAFu. Odkazy na zdroje v DBpedii používame na získavanie abstraktov a odkazov na stránky súvisiace s týmito zdrojmi. Na dopytovanie sa DBpedia používame knižnicu *ActiveRDF*²⁸ pre jazyk Ruby, ktorá poskytuje objektový prístup k zdrojom vo formáte *RDF*, podobne ako nástroje ako *ActiveRecord*²⁹ poskytujú objektový prístup k relačným databázam.

6.3 Zobrazenie a prispôsobenie poznámok

Pre zobrazenie poznámok sme implementovali modul do výučbového systému ALEF. Keďže systém ALEF pracuje s uzavretou množinou stránok, nevytvárame poznámky pri načítavaní stránky, teda on-line, ale off-line pre všetky stránky v systéme naraz. Tým že na vytváranie poznámok používame webovú službu by sme síce mali možnosť pridávať poznámky do webovej stránky pri jej načítavaní, ale takto dosiahneme vyššiu rýchlosť, keďže nie je potrebné pri každom načítavaní stránky analyzovať text stránky, získavať informácie na naplnenie poznámok a vytvárať samotné poznámky. Súčasne je jednoduchšie prispôbovať poznámky pre potreby používateľov, ak sa pri každom zobrazení poznámky pracuje s tým istým základom poznámky a ten sa prispôbuje používateľovi.

Vytvorené poznámky uchováваме ako časť dátového modelu v systéme ALEF, oddelene od samotných textov. Vkladanie poznámok do webovej stránky sa vykonáva už pri skladaní stránky na strane servera.

6.3.1 Zobrazenie poznámok

Poznámky sú pripojené ku kľúčovým slovám vo výučbových materiáloch v systéme ALEF. Slová, ku ktorým je pripojená poznámka, sú v texte zvýraznené podčiarknutím a zmenou farby pri prechode kurzorom myši nad týmto slovom (Obr. 9). Po kliknutí na zvýraznené slovo sa zobrazí tooltip (Obr. 11), v ktorom je zoznam poznámok priradených k tomuto slovu,

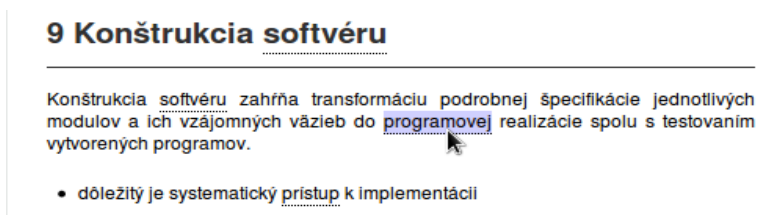
26 DictService, <http://services.aonaware.com/DictService/DictService.aspx>

27 SlideShare, <http://www.slideshare.net/>

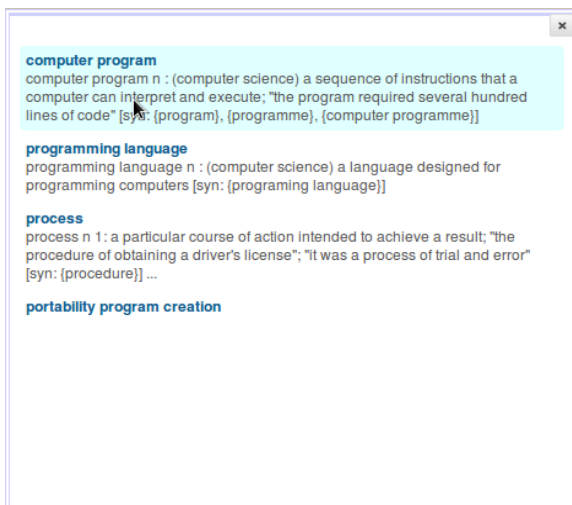
28 ActiveRecord, <http://activerdf.org/>

29 ActiveRecord, <http://ar.rubyonrails.org/>

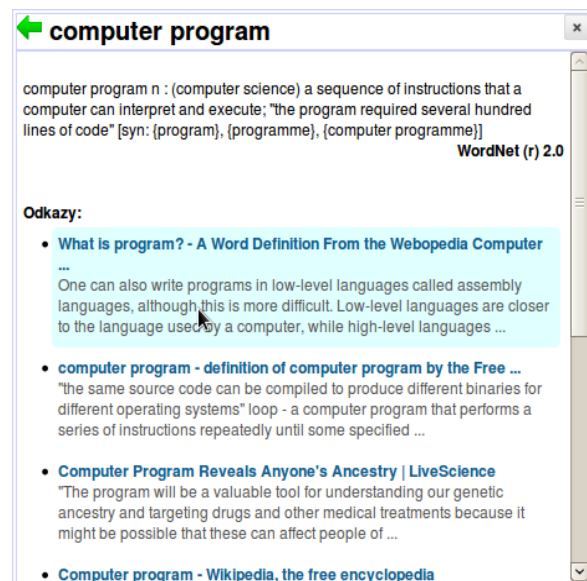
spolu so skrátanou definíciou z každej poznámky. Po kliknutí na jednu zo zobrazených poznámok sa obsah tooltipu zmení a bude zobrazovať samotný obsah poznámky (Obr. 10). Poznámka je tvorená definíciou slova, ku ktorému je priradená, zoznamom odkazov na súvisiace stránky a zoznamom prezentácií. K odkazom na súvisiace stránky je pripojená krátka sumarizácia obsahu stránky na ktorú odkaz smeruje. Po kliknutí na súvisiaci odkaz sa na novej karte webového prehliadača otvorí stránka, na ktorú tento odkaz smeruje. Po kliknutí na odkaz na prezentáciu priradenú k poznámke sa opäť zmení obsah tooltipu a nahradí ho zvolená prezentácia.



Obr. 9: Zvýraznenie opoznámkovaného slova



Obr. 11: Tooltip na výber z poznámok priradených ku kľúčovému slovu



Obr. 10: Tooltip zobrazujúci telo poznámky

6.3.2 Prispôbovanie poznámok

Prispôbovanie poznámok je realizované formou usporiadania odkazov v tele poznámky. Poradie odkazov je vypočítané na základe implicitnej spätnej väzby získanej sledovaním odkazov, na ktoré používatelia klikli. Na základe metódy opísanej v časti 5.2.1 vytvárame tvrdenia o kvalite odkazov v zozname. Pri vytváraní týchto tvrdení pre odkazy na súvisiace stránky má váha tvrdenia vždy hodnotu jedna. Pri vytváraní tvrdení pre prezentácie priradené k poznámkam je váha závislá od trvania zobrazenia. Ak mal študent poznámku zobrazenú

kratšie ako 15 sekúnd, hodnota váhy tvrdenia je nula. Ak zobrazenie trvalo dlhšie ako 15 sekúnd, táto hodnota je jedna. Trvanie zobrazenia meriame na strane používateľa pomocou JavaScriptu, s využitím volaní `window.onfocus` a `window.onblur`. Tieto volania sa spustia vtedy ak okno alebo karta prehliadača získa alebo stratí fokus. Meraním času pri spustení týchto akcií sa vytvárajú zoznamy časov príchodov a odchodov zo stránky. Pri zatvorení tooltipu s prezentáciou alebo prechode tooltipu do iného stavu sa z týchto časov vypočíta celkový čas zobrazenia prezentácie.

Na odvodenie nového poradia je použitá iteratívna verzia upraveného PageRank algoritmu, opísaného v kapitole 5.2.2. Iteratívna verzia znamená, že sa opakovane vypočítava ohodnotenie pre všetky uzly grafu, pokiaľ zmena v ohodnotení uzlov nie je menšia ako stanovená prahová hodnota. Ako prahovú hodnotu sme experimentálne zvolili hodnotu 0,01. Usporiadáním ohodnotených odkazov získame nové poradie odkazov.

7 Overenie a experimenty

Overenie navrhovaných metód sme vykonali formou otvoreného experimentu v prostredí výučbového systému ALEF, ako aj formou čiastkových uzavretých experimentov. Metódu na mapovanie slov medzi textom a jeho prekladom, ako aj kvalitu informácií získaných pomocou verejne dostupných služieb sme overili v uzavretých experimentoch s pomocou expertov. Experimentom vo výučbovom systéme ALEF sme overili použiteľnosť implementovanej aplikácie a metódu na prispôsobovanie poznámok na základe implicitnej spätnej väzby získanej od používateľov.

7.1 Mapovanie slov medzi textom a jeho prekladom

Overenie výsledkov metódy na mapovanie slov sme realizovali porovnaním úspešnosti štyroch funkcií, ktoré sme implementovali. Test sme realizovali na časti textu skrípt k predmetu Princípy softvérového inžinierstva. Testovacia vzorka mala po odstránení stop-slov 1928 slov.

Výsledky experimentu sú zhrnuté v tabuľke 1. Sledovali sme počet slov, ktoré sa namapovali správne, nesprávne a počet takých slov, kde sa síce slovo namapovalo na správny preklad, ale okrem neho sa k tomuto prekladu namapovali aj ďalšie, nesprávne slová. V poslednom stĺpci tabuľky je pomer počtu všetkých namapovaných slov a počtu slov v testovacej vzorke.

Funkcia	Správne	Nesprávne	Viac	Namapované/ Všetky
Základná	92.75 %	5.82 %	1.60 %	45.38 %
Zohľadňovanie pozícií slov	55.14 %	32.92 %	11.93 %	84.85 %
Predspracovaný slovník	92.45 %	5.58 %	1.95 %	63.59 %*
Obe vylepšenia	64.07 %	24.95 %	10.96 %	96.08 %*

Tab. 1: Výsledky experimentu na overenie metódy na mapovanie slov

* Pri použití metód, ktoré pracovali s upraveným slovníkom sa niektoré slová vo vete zlúčili do jedného základu slova, preto počet všetkých po vetách unikátnych základov slov je len 1689 oproti celkovému počtu slov 1928

Z tabuľky 1 vidíme, že pomer správne namapovaných slov ku všetkým namapovaným slovám je pri základnej metóde viac ako 90 %. Nanešťastie táto metóda dokázala najšť mapovanie ani nie pre polovicu všetkých slov v testovacej vzorke. Metóda zohľadňujúca pozíciu nenamapovaných slov bola navrhnutá tak aby dokázala najšť mapovanie práve pre tie slová, ku ktorým základná metóda nedokázala priradiť ich ekvivalent. Aplikovaním tejto metódy sme síce dosiahli počet všetkých namapovaných slov viac ako 80 %, avšak do nájdených mapovaní toto vylepšenie zaviedlo veľké množstvo chýb. Druhé vylepšenie, používajúce

predspracovaný slovník sa tiež snaží o zvýšenie podielu namapovaných slov ku všetkým slovám v spracovávanej vzorke. Táto funkcia mierne zvýšila tento podiel oproti základnej metóde a pritom dokázala zachovať počet správne namapovaných slov na viac ako 90 % zo všetkých namapovaných slov. Posledná overovaná metóda, kombinujúca obe vylepšenia, dokázala nájsť mapovanie pre najväčší počet slov spomedzi všetkých overovaných metód. Zároveň dokázala mierne zlepšiť kvalitu mapovania v porovnaní s metódou používajúcou len zohľadňovanie pozícií slov.

Pri výbere metódy, spomedzi štyroch navrhovaných je potrebné si zvoliť, či má metóda, ktorú chceme použiť, poskytovať kvalitné mapovania a uspokojíme sa aj s menším počtom namapovaných slov, alebo vyžadujeme mapovanie pre veľký počet slov aj za cenu kvality mapovania.

7.2 Vyhodnotenie kvality informácií na naplnenie poznámok

Na naplnenie poznámok sme použili rôzne verejne dostupné služby na vyhľadávanie informácií. Overovali sme kvalitu informácií, ktoré dokážeme pomocou týchto služieb získať. Postup overovania bol nasledovný:

1. Vytvorili sme sadu 26 textov z oblasti softvérového inžinierstva.
2. Tieto texty sme spracovali pomocou služby AlchemyAPI. Výsledkom spracovania bola sada kľúčových slov a sada konceptov, ktorých sa tento text týkal. K získaným konceptom nám služba AlchemyAPI vrátila aj odkazy reprezentujúce tieto koncepty v prostredí LinkedData.
3. S použitím vyhladaných kľúčových slov ako dopytu sme vyhľadali dodatočné informácie pomocou služieb GoogleSearch, SlideShare a DictService.
4. S použitím knižnice ActiveRDF a odkazov na zdroje v prostredí LinkedData sme vyhľadali dodatočné informácie v DBpedii.
5. Požiadali sme expertov, aby určili, či sú vyhľadané definície slov, odkazy na súvisiace stránky a prezentácie relevantné ku kľúčovým slovám a k textu, v ktorom boli tieto slová vyhľadané.

Pomocou služieb Google Search a DBpedia sme získali informácie vo forme odkazov na súvisiace stránky. Služba SlideShare nám vrátila prezentácie a služba DictService definície slov.

Expertmi mali za úlohu určiť, či sú tieto informácie relevantné ku kľúčovým slovám, ku ktorým boli vyhľadané a súčasne či sú relevantné k analyzovanému textu. Teda určiť či sa nájdené definície týkajú slova, ku ktorému sú priradené a súčasne či opisujú toto slovo v rovnakom význame, v akom bolo opísané v analyzovanom texte. V prípade odkazov na internetové stránky a odkazov na prezentácie mali za úlohu si túto stránku (prezentáciu) prezrieť a určiť, či sa týka slova, ku ktorému je priradená a či opisuje toto slovo v rovnakom význame ako je opísané v texte.

Podarilo sa nám ohodnotiť relevanciu informácií pre 29 súborov z toho 16 rôznych. Výsledky overenia sú zhrnuté v tabuľke číslo 2. Sledovali sme počet relevantných informácií a počet

nerelevantných informácií. Pre každú z overovaných služieb bolo malé percento informácií takých, že sa nám nepodarilo vyhodnotiť ich relevanciu, toto bolo spôsobené napríklad poškodeným odkazom. V poslednom stĺpci tabuľky je celkový počet informácií (odkazov, prezentácií alebo definícií), na ktorých sme vykonali experiment.

Služba	Relevantné	Nerelevantné	Spolu
Google	70.01 %	26.98 %	897
DBpedia	63.29 %	31.90 %	395
DictService	59.64 %	40.36 %	280
SlideShare	26.32 %	72.79 %	1246

Tab. 2: Výsledky experimentu na overenie metódy na mapovanie slov

Pre jednotlivé služby sú veľké rozdiely v kvalite informácií, ktoré pomocou nich vieme získať. Je preto dôležité si pri výbere služieb použitých na naplnenie poznámok vybrať také, ktoré poskytnú najviac relevantných informácií, keďže kvalita výsledných poznámok výrazne závisí od kvality informácií, ktorými sú naplnené. Žiadna z použitých služieb sa pri poskytovaní informácií nepriblížila k stopercentnej úspešnosti, je tu teda priestor na vylepšovanie spôsobu na získavanie informácií, napríklad lepším formulovaním dopytu.

7.3 Overenie metódy v systéme ALEF

V systéme ALEF sme vyhodnocovali užitočnosť a použiteľnosť poznámok priradených ku kľúčovým slovám v texte. Okrem toho sme overovali metódu na usporiadanie odkazov v poznámke na základe implicitnej spätnej väzby získanej sledovaním klikania študentov na odkazy zobrazené v tele poznámky. Overenie metódy na zoradovanie odkazov prebiehalo porovnaním implicitnej a explicitnej spätnej väzby získanej od používateľov. Overenie metódy na pridávanie poznámok do textu prebiehalo pomocou otázok, ktoré sa systém priebežne pýtal študentov počas ich práce. Experiment prebiehal na kurze k predmetu Princípy softvérového inžinierstva od 16.4.2011 do 2.5.2011. ALEF vtedy používalo 195 študentov, z nich 65 študentov zobrazilo aspoň jedenkrát poznámku. Spolu bolo zobrazených 2074 poznámok. Študenti v nich 1878-krát nasledovali zobrazený odkaz a 415-krát zobrazili vloženú prezentáciu.

7.3.1 Vyhodnotenie porovnaním implicitnej a explicitnej spätnej väzby

Metóda na zoradovanie odkazov v poznámke používa na usporiadanie odkazov tvrdenia o kvalite odkazov získané sledovaním klikania študentov na odkazy (kapitola 5.2). Overenie tejto metódy prebiehalo porovnaním získaného poradí s poradím, ktoré sme získali explicitným hodnotením kvality odkazov vybranými študentami. Vybrali sme 34 študentov, ktorí okrem toho, že si mohli prezerat' poznámky, mali aj možnosť vybrať z ponúknutých odkazov tie najlepšie. Požiadali sme ich aby z ponúknutých odkazov vybrali tri najlepšie. Odkazy sme následne zoradili podľa počtu priradených explicitných hodnotení a porovnali

s poradím získaným na základe implicitnej spätnej väzby. Na zoradenie odkazov sme použili len také explicitné hodnotenia, na ktorých sa aspoň dvaja študenti zhodli, teda také, ktoré boli vybrané aspoň dvakrát.

Sledovali sme presnosť s akou vieme určiť, ktoré odkazy majú byť na prvých pozíciách v zozname, s tým že sme nesledovali poradie odkazov na týchto prvých pozíciách. Presnosť výberu je vlastne pomer počtu vybraných relevantných dokumentov a počtu všetkých vybraných dokumentov. Všeobecný vzorec pre výpočet presnosti (precision) je nasledovný:

$$presnosť = \frac{||\{relevantné\} \cap \{vybrané\}||}{||\{vybrané\}||}$$

V našom prípade je počet vybraných dokumentov rovný počtu prvých pozícií v zozname, ktoré sledujeme a počet relevantných vybraných dokumentov je počet zhodných odkazov na prvých pozíciách v oboch porovnávaných zoznamoch. Potom sa výraz pre výpočet presnosti výberu dá zapísať ako:

$$presnosť = \frac{\text{počet spoločných odkazov}}{\text{počet sledovaných pozícií}}$$

Vzťah pre výpočet úplnosti (recall) je pomer počtu vybraných relevantných dokumentov k celkovému počtu relevantných dokumentov. V našom prípade je počet relevantných a vybraných dokumentov rovnaký, pretože relevantné sú všetky odkazy na zvolenom počte prvých pozícií v zozname získanom explicitným ohodnotením kvality odkazov a vyberáme práve tento počet dokumentov. To znamená, že výsledný vzťah pre presnosť a úplnosť je v našom prípade rovnaký. Preto sledujeme len presnosť výberu.

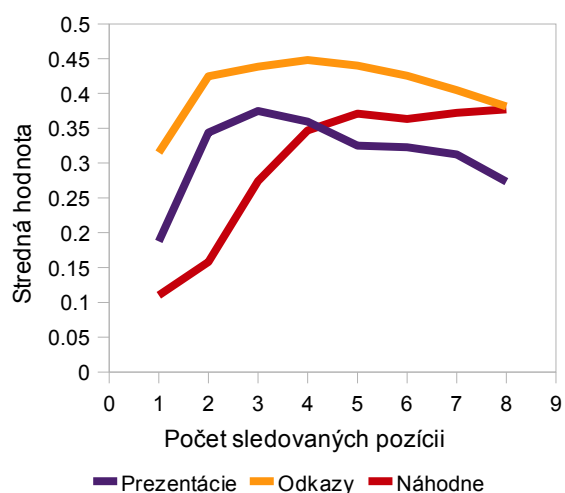
Počas zbierania implicitnej spätnej väzby sa študentom zobrazovali odkazy v náhodnom poradí. V posledných dňoch experimentu sme nasadili funkciu na zoradovanie odkazov, aby ju mohli študenti používať a aby sme získali ich vyjadrenia o kvalite usporiadania. Predtým ako sme nasadili túto funkciu, sme vytvorili obraz databázy, na ktorej sme následne robili vyhodnotenie metódy na usporiadanie odkazov. Ak by sme vyhodnocovali toto poradie s údajmi získanými počas celého trvania experimentu, tieto posledné by boli ovplyvnené tým, že študenti vedeli o tom, že v systéme nastala nejaká zmena a snažili sa vyhodnotiť, či je toto nové poradie správne. Obraz databázy vytvorený pri nasadení funkcie na usporiadanie odkazov ako aj finálny obraz databázy je umiestnený na priloženom médiu.

Od študentov sme získali 1340 explicitných hodnotení kvality odkazov a 213 explicitných hodnotení kvality prezentácií. Prešli sme cez všetky poznámky a zoznamy k nim priradených odkazov. Ak sme mali k týmto odkazom dostupné aj explicitné hodnotenia ich kvality porovnali sme zoznam usporiadaný na základe explicitnej a implicitnej spätnej väzby.

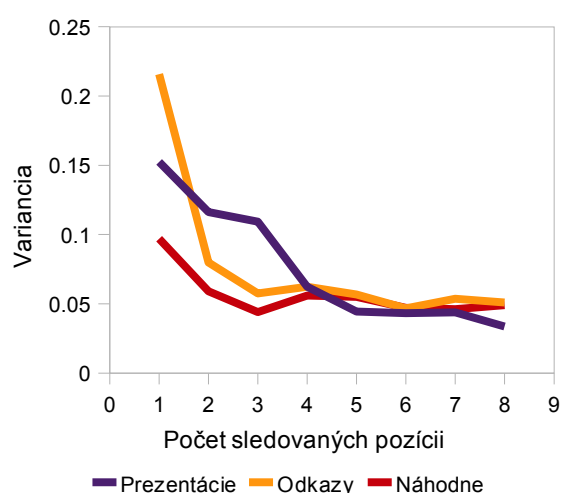
Postupne sme vypočítali presnosť pre výber odkazov na jednu, dve, tri až postupne na všetky prvé pozície v zozname. Zo získaných presností sme vypočítali strednú hodnotu, ako aritmetický priemer presností z rôznych poznámok, ktoré sledujú rovnaký počet prvých pozícií. Následne sme vypočítali rozptyl k týmto stredným hodnotám. Výsledky porovnania sú zhrnuté v grafoch na obrázku číslo 12 a číslo 13. V týchto grafoch je zobrazená stredná hodnota a rozptyl presností výberu pre zoznamy odkazov a zoznamy prezentácií zoradených

pomocou overovanej metódy. Tieto výsledky sú porovnané s hodnotami získanými pre náhodne usporiadaný zoznam odkazov.

Z grafu vývoja strednej hodnoty presnosti (Obr. 12) vidíme, že implementovaná metóda dosahuje vždy lepšiu presnosť výberu ako náhodne usporiadaný zoznam odkazov. Ak by sme mali k dispozícii úplný zoznam odkazov usporiadaných podľa ich relevancie, graf strednej hodnoty by musel stúpať postupne k hodnote 1. My sme však mali k dispozícii len usporiadaný zoznam prvých najlepších odkazov. Preto keď sme nemali s akým zoznamom porovnávať zoznam získaný pomocou navrhovanej metódy, nemohol rásť počet zhodných odkazov na prvých pozíciách a teda stredná hodnota výberu začala klesať.



Obr. 12: Stredná hodnota presnosti výberu prvkov na prvé pozície v zozname



Obr. 13: Rozptyl presnosti výberu prvkov na prvé pozície v zozname

Stredná hodnota presnosti výberu vypočítaná pre zoznamy prezentácií má nižšie hodnoty, pretože študenti s prezentáciami interagovali výrazne menej (329 zobrazení) ako s odkazmi (1602 zobrazení). Metóda na usporiadanie odkazov teda používala oveľa menej údajov na vytvorenie poradia. Súčasne bolo aj menej explicitných hodnotení kvality prezentácií (213 hodnotení) ako hodnotení odkazov (1340 hodnotení), takže zoznam najlepších prezentácií, s ktorým sme výsledky navrhovanej metódy porovnávali bol taktiež oveľa kratší. Rozptyl nameraných stredných hodnôt sa podľa očakávania znižuje s rastúcim počtom sledovaných pozícií pri všetkých troch meraniach.

7.3.2 Vyhodnotenie odpovedí študentov

Počas práce študentov s výučbovým systémom ALEF, sa im náhodne zobrazovali tri druhy otázok, ktorými sme vyhodnocovali kvalitu vytvorených poznámok a to či sú poznámky pripojené k správnym slovám v texte. V posledných dňoch experimentu sme do systému vložili funkciu na usporiadanie odkazov na základe implicitnej spätnej väzby. Na overenie tejto novej funkcionality sme vymenili sadu otázok, ktoré sme sa študentov pýtali a pýtali sme sa už len dve otázky na overenie kvality vytvoreného usporiadania odkazov. Študenti odpovedali spolu na 582 otázok. Ich odpovede sú zhrnuté v tabuľke 3. Sada otázok, ktoré sa zobrazovali používateľom pri práci so systémom ALEF bola nasledovná:

1. Poznámky s dodatočnými informáciami pripojenými k slovám v texte
 - a. používam ich na získanie dodatočných informácií.
 - b. občas v nich hľadám nejaké informácie.
 - c. nepoužívam.
2. Sú pre vás definície pripojené k slovám v texte užitočné?
 - a. Áno, našiel som tam zaujímavé informácie.
 - b. Niektoré informácie boli pre mňa užitočné.
 - c. Len veľmi málo informácií v nich bolo pre mňa užitočných.
 - d. Viac-menej tam neboli užitočné informácie.
 - e. Nepoužíval som tieto poznámky.
3. Sú poznámky s dodatočnými informáciami priradené k vhodným slovám v texte?
 - a. Áno.
 - b. Väčšinou áno ale občas sú priradené k nesprávnym slovám.
 - c. Väčšinou nie.
 - d. Nie.
 - e. Neviem povedať.

Predchádzajúce otázky sa zobrazovali len do nasadenia funkcionality na zoraďovanie odkazov. Následne sa sada zobrazovaných otázok zmenila a zobrazovali sa len nasledujúce dve otázky.

4. Vyjadruje podľa vás usporiadanie informácií v poznámkach s dodatočnými informáciami to ako sú tieto informácie pre vás užitočné?
 - a. Áno.
 - b. Väčšinou áno.
 - c. Nie, poradie sa mi zdá náhodné.
 - d. Neviem povedať.
5. Sú podľa vás v poznámkach s dodatočnými informáciami na prvých priečkach v zozname uvedené užitočné informácie?
 - a. Áno.
 - b. Väčšinou áno.
 - c. Nie, sú náhodne rozmiestnené.
 - d. Nie sú tu žiadne relevantné informácie.
 - e. Neviem určiť.

Otázka	Počet odpovedí v percentách					Celkový počet odpovedí
	a.	b.	c.	d.	e.	
1.	30.43 %	66.46 %	3.11 %	–	–	161
2.	19.30 %	60.82 %	16.37 %	1.75 %	1.75 %	171
3.	23.70 %	60.12 %	8.09 %	1.16 %	6.94 %	173
4.	2.63 %	36.84 %	57.89 %	2.63 %	–	38
5.	2.56 %	48.72 %	41.03 %	5.13 %	2.56 %	39

Tab. 3: Počet odpovedí používateľov na otázky

Na otázky 1 a 2, o používaní poznámok a o užitočnosti definícií pripojených k slovám v texte študenti väčšinou odpovedali pozitívne. Z toho usudzujeme, že dodatočné informácie pripojené ku kľúčovým slovám poskytnuté vo forme poznámky zobrazenej ako tooltip sú použiteľný a užitočný prostriedok na poskytovanie doplňujúcich informácií o dôležitých slovách v texte.

Pri overovaní toho, či sú poznámky priradené k správnym slovám vo výučbovom texte sa študenti vyjadrovali zväčša pozitívne, ale nájdu sa aj také poznámky, ktoré sú priradené k nesprávnym slovám. Metóda na mapovanie slov teda funguje dobre, ale stále je tu priestor na jej vylepšovanie.

Odpovedí na otázky, ktoré sa začali zobrazovať po nasadení funkcionality na usporiadanie odkazov je v porovnaní s odpoveďami na ostatné otázky veľmi málo. Toto prisudzujeme faktu, že na odpovedanie na ne mali študenti menej času ako aj faktu, že na predmete Princípy softvérového inžinierstva ich už nečakali žiadne cvičenia ani kontrolné práce, takže neboli tak motivovaný pracovať so systémom. Metódu na zoraďovanie odkazov sme však overovali hlavne pomocou porovnania poradia získaného pomocou explicitnej a implicitnej spätnej väzby, takže tieto otázky možno považovať len za doplnok k overeniu poradia odkazov. V týchto otázkach sa študenti vyjadrovali rozporuplne z čoho usudzujeme, že usporiadanie poznámok bolo v mnohých prípadoch nedostatočné. Veríme však, že pri väčšom množstve údajov použitých na usporiadanie odkazov by boli výsledky oveľa lepšie.

8 Zhodnotenie

V rámci projektu sme navrhli a implementovali metódu na vytvorenie poznámok do textu webovej stránky v slovenskom jazyku. Navrhovaná metóda má za úlohu poskytovať čitateľovi dokumentu definíciu neznámych slov, alebo dodatočné informácie k potencionálne zaujímavým slovám. Pre vyhľadanie kľúčových slov, ku ktorým priradíme poznámky sme preložili text spracovávanej stránky do angličtiny. Z tohto prekladu vyplynula potreba vytvoriť mapovanie medzi extrahovanými anglickými slovami a slovenskými slovami v pôvodnom texte webovej stránky. Navrhli a implementovali sme teda metódu na hľadanie ekvivalentných slov medzi slovenským textom a jeho prekladom do angličtiny. Na vytvorenie tohto mapovania sme použili bilingválny slovník a porovnávanie slov pomocou Levenshteinovej vzdialenosti. Použitie Levenshteinovej vzdialenosti nám umožnilo spracovávať rôzne tvary slovenských slov na úrovni reťazcov. Okrem tejto metódy sme implementovali aj jej dve vylepšenia, ktorými sme sa snažili zvýšiť počet namapovaných slov. Jedno vylepšenie zohľadňuje pozíciu slov vo vete a to druhé pracuje s predspracovaným slovníkom, za účelom zohľadnenia rôznych tvarov slov v anglickom jazyku.

Na získanie informácií na naplnenie poznámok sme použili rôzne verejne dostupné služby na získavanie informácií. Overili sme kvalitu informácií, ktoré dokážeme takýmto spôsobom získať. Zistili sme, že kvalita získaných informácií výrazne závisí od použitej služby. Ďalej by teda bolo potrebné rozvíjať spôsoby ako sa dopytovať týchto informácií, aby sme zvýšili ich kvalitu. Takéto vylepšenie by mohlo byť napríklad vylepšenie spôsobu tvorby dopytu zohľadňovaním kontextu dokumentu alebo modelu používateľa.

Časť metódy na vytváranie poznámok do textu webových stránok, ktorá spracováva text stránky a vytvára samotné poznámky sme implementovali ako webovú službu. Časť zodpovednú za vizualizáciu a prispôsobenie poznámok sme implementovali ako modul do výučbového rámca ALEF vyvíjaného na Fakulte informatiky a informačných technológií.

Prispôsobenie poznámok je vo forme usporiadania odkazov zobrazených ako obsah poznámky podľa ich relevancie. Relevanciu odkazov sme odvodzovali na základe implicitnej spätnej väzby získanej sledovaním klikania používateľov.

Navrhovanú metódu sme overili po častiach pomocou niekoľkých uzavretých experimentov a ako celok v systéme ALEF využívaného na predmete Princípy softvérového inžinierstva, kde študenti používali poznámky pripojené ku kľúčovým slovám vo výučbových materiáloch.

Pri implementovaní navrhovanej metódy sme prišli na veľa zaujímavých aspektov, ktoré by sme chceli v ďalšej práci rozpracovať. Jedným takýmto aspektom je použitie rôznych metód na prispôbovanie poznámok. V súčasnej práci sme sa zaoberali prispôbovaním poradia prezentovaných odkazov, ale existuje množstvo ďalších techník pre rôzne formy informácií v obsahu poznámky, ktorými by sme sa chceli v budúcej práci zaoberať. Ďalším zaujímavým smerom, ktorý by sme chceli rozvíjať je použitie rôznych zdrojov informácií, na základe ktorých je možné prispôbovať poznámky a následne upravovať poznámky pre každého používateľa na základe modelu používateľa.

Literatúra

- [1] ADAM M.: An approach to automated on-line annotation, Proc. of research project workshop Tools for Acquisition, tent annotation is useful approach also to assist students Organization and Presenting of Informatio (2007)
- [2] AGOSTI M., FERRO N.: A formal model of annotations of digital content, Transactions on Information Systems (TOIS), ACM (November 2007)
- [3] AGOSTI M., FERRO N.: Annotations: Enriching a Digital Library, Research and Advanced Technology for Digital Libraries, Springer Berlin / Heidelberg (2003)
- [4] AHMAD K., GILLAM L., TOSTEVIN L.: University of Surrey Participation in TREC 8: Weirdness Indexing for Logical Document Extrapolation and Retrieval (WILDER), The Eighth Text REtrieval Conference (TREC-8), National Institute of Standards and Technology (1999)
- [5] BALLESTEROS L., CROFT B.: Dictionary methods for cross-lingual information retrieval, Database and Expert Systems Applications, ser. Lecture Notes in Computer Science, (1996)
- [6] BARBU A.-M.: A Positional Linguistics-Based System for Word Alignment., Lecture Notes in Computer Science, (2004)
- [7] BIELIKOVÁ M.: Adaptívna prezentáciaí hypermédií na webe, Proc. of DATAKON 2003, (2003)
- [8] BOGURAEV B., KENNEDY C.: Applications of term identification technology: domain description and content characterisation, Natural Language Engineering, Cambridge University Press (1999)
- [9] BRUSILOVSKY P.: Adaptive hypermedia, User Modeling and User-Adapted Interaction, (2001)
- [10] BRUSILOVSKY P., MILLÁN E.: User Models for Adaptive Hypermedia and Adaptive Educational Systems, Brusilovsky, P., Kobsa, A., Nejd, W. (eds.) Adaptive Web 2007., LNCS (2007)
- [11] GALE W., CHURCH K.: Identifying word correspondences in parallel texts., In Fourth DARPA Workshop on Speech and Natural Language, (1991)
- [12] GARABÍK R.: Slovak morphology analyzer based on Levenshtein edit operations, Workshop on intelligent and knowledge oriented technologies (WIKT), (2006)
- [13] HOLUB M., BIELIKOVÁ M.: Behavior Based Adaptive Navigation Support, Proceedings of the Workshop on the Practical Use of Recommender Systems, Algorithms and Technologies, (2010)
- [14] JOACHIMS T., GRANKA L., PAN B., HEMBROOKE H., and GAY G.: Accurately interpreting clickthrough data as implicit feedback, Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, ACM (2005)

- [15] KAHAN J., KOIVUNEN M.-R: Annotea: An open RDF infrastructure for shared Web annotations, Proceedings of the 10th International Conference on World Wide Web, (2001)
- [16] LEVENSHTEIN V. I.: Binary Codes Capable of Correcting Deletions, Insertions and Reversals., Soviet Physics Doklady (1966)
- [17] MANSOURI A., AFFENDEY L. S., MAMAR A.: Name Entity Recognition Approaches, International Journal of Computer Science and Network Security, (2008)
- [18] MARSHALL C. C.: Toward an ecology of hypertext annotation, UK Conference on Hypertext, (1998)
- [19] MIHAL V., BIELIKOVA M.: An Approach to Annotation of Learning Texts on Programming within a Web-Based Educational System, SMAP, IEEE Computer Society (2009)
- [20] MIHALCEA R., CSOMAI A.: Wikify!: linking documents to encyclopedic knowledge, Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, ACM (2007)
- [21] PAGE L., BRIN S., MOTWANI R. and WINOGRAD T.: The PageRank citation ranking: Bringing order to the web, Stanford Digital Library Technologies Project, (1998)
- [22] PORTER M. F.: An algorithm for suffix stripping, Program (1980)
- [23] ZHANG, Z., IRIA, J., BREWSTER, C., CIRAVEGNA, F.: A Comparative Evaluation of Term Recognition Algorithms., The sixth international conference on Language Resources and Evaluation, LREC (2008)
- [24] ŠIMKO M., BARLA M. BIELIKOVÁ M.: ALEF: A Framework for Adaptive Web-based Learning 2.0., Key Competencies in the Knowledge Society, World Computer Congress 2010 (2010)

Prílohy

A Technická dokumentácia

B Používateľská príručka

C Inštalačná príručka

D Príspevok publikovaný na konferencii IIT.SRC 2011

E Príspevok zaslaný na konferenciu Web Intelligence 2011

F Diplom získaný v medzinárodnej súťaži eLearning v praxi v rámci konferencie
ICETA 2010

G Obsah elektronického média

A Technická dokumentácia

A.1 Ukážka kódu na výpočet Levenshteinovej vzdialenosti

```
class LevenshteinDistance
  def self.distance(word1,word2)
    word1 = Unicode.downcase(word1)
    word2 = Unicode.downcase(word2)
    m = word1.length
    n = word2.length
    matrix = []
    (m+1).times { matrix << [] }
    matrix[0][0]=0.0

    m.times do |i|
      matrix[i+1][0]=matrix[i][0]+cost(i+1,m)
    end
    n.times do |i|
      matrix[0][i+1]=matrix[0][i]+cost(i+1,n)
    end
    # koniec inicializacie matice
    m.times do |i|
      n.times do |j|
        if(word1[i..i]==word2[j..j]) # netreba vkonat ziadnu operaciu
          matrix[i+1][j+1]=matrix[i][j]
        else # cena je cena opercie na danom nieste plus minimalna cena susednych prvkov matice
          cost = [cost(i+1,m),cost(j+1,n)].min
          matrix[i+1][j+1]=[matrix[i][j+1]+cost, # pridane znaku do prveho slova
                           matrix[i+1][j]+cost, # pridane znaku do druheho slova tj. zmazanie znaku v
                           # prvom slove
                           matrix[i][j]+cost].min #vymena znaku

          end
        end
      end
    end
    matrix[m][n] # vysledok je v dolnom pravom rohu matice
  end

  def self.cost(pos,wordLength)
    return 2.0 if pos <= 3
    return 2.0 if (pos <= wordLength-6)

    r = 3 > wordLength-6 ? 3 : wordLength-6
    1-((pos-r)/(wordLength-r+1.0));
  end
end
```

A.2 Ukážka výstupného XML súboru služby na vytvorenie poznámok

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
  <original_text>Postupnosť definícií a krokov potrebných na dosiahnutie riešenia problému
  realizovaných pomocou programovacieho jazyka sa nazýva program.</original_text>
  <translated_text> Sequence of definitions and the steps needed to achieve solutions to the
  problem made by a programming language called program.</translated_text>
  <concepts>
    <concept>
      <references>
        <reference>
          <url>http://en.wikipedia.org/wiki/Programming_language</url>
          <title>Programming language - Wikipedia, the free encyclopedia</title>
          <summary> By purpose, programming languages might be considered general purpose,
  system programming languages ...</summary>
        </reference>
        ...
      </references>
      <slides>
        <slide>
          <title>Programming language Ruby and the Rails framework </title>
          <url>http://www.slideshare.net/radekmika/programming-language-ruby-and-the-rails-
  framework-v-09-beta</url>
          <description/>
          <thumbnail>http://cdn.slidesharecdn.com/rubyextendedv2-100202112251-phpapp02-
  thumbnail</thumbnail>
          <embed>&lt;div style="width:425px" id="__ss_3055352"&gt;&lt;strong
  style="display:block; ...</embed>
          <number>3055352</number>
          <views>2416</views>
        </slide>
      </slides>
      <definitions>
        <definition>
          <source>dbpedia</source>
          <text> A programming language is an artificial language designed to express
  computations that can be performed by a machine ...</text>
        </definition>
        ...
      </definitions>
      <text>programming language</text>
      <mappings>
        <mapping>programovacieho jazyka</mapping>
      </mappings>
      <relevance>0.945151</relevance>
    </concept>
    ...
  </concepts>
</result>
```

A.3 Ukážka kódu na vytvorenie tooltipu s poznámkou

```
class WordDefinitionWidgetCell < Apotomo::StatefulNotCrashingWidget

  def transition_map
    {
      :init => [:init, :definition, :selection],
      :selection => [:init, :definition],
      :definition => [:init, :definition, :slide, :selection],
      :slide => [:init, :definition, :slide]
    }
  end

  # zakldany stav tooltipu (v tomto stave je tooltip schovany)
  def init
    # ...
  end

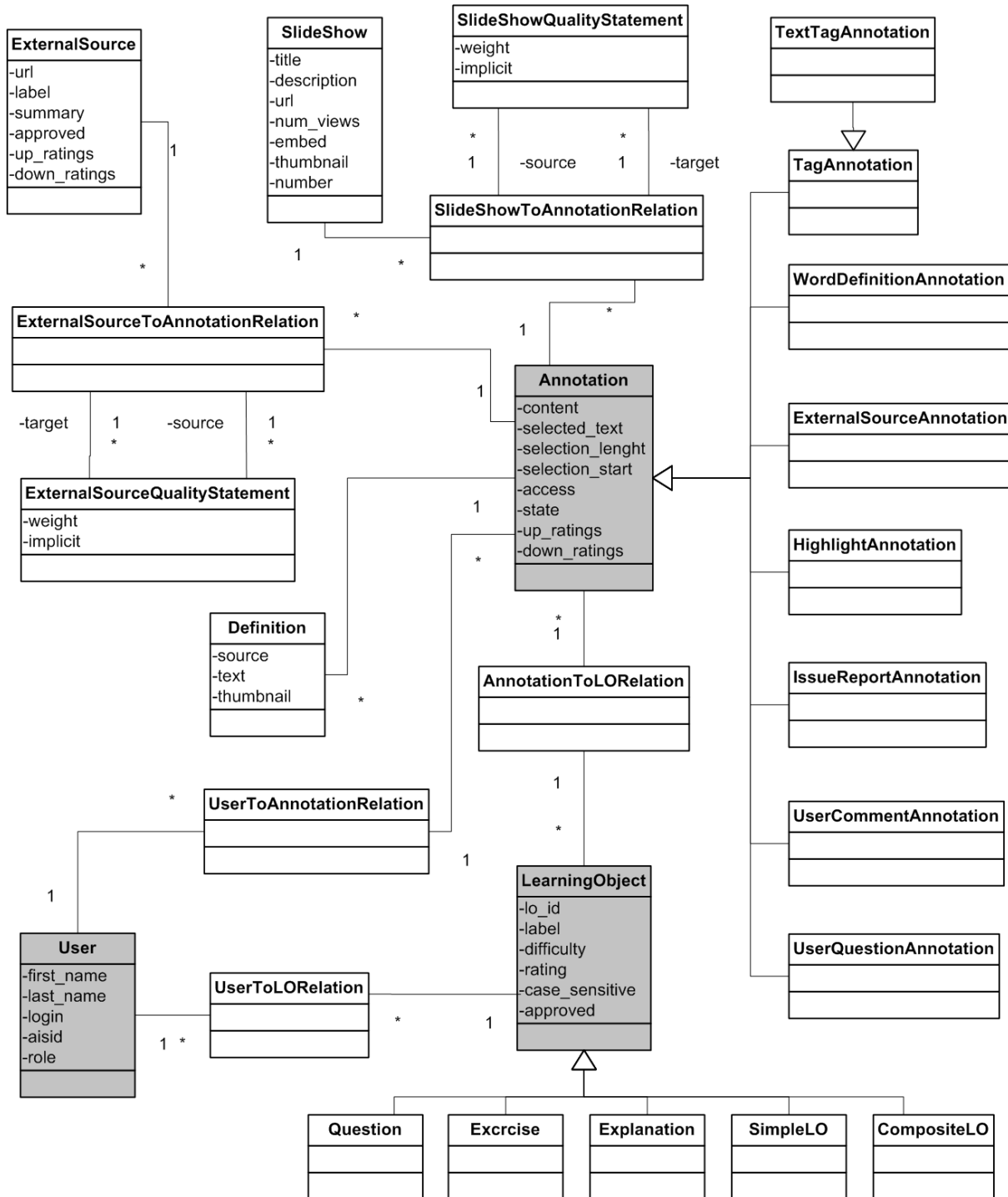
  # tooltip na vyber poznanky, ktora sa ma zobrazit
  def selection
    # ...
  end

  # tooltip zobrazujuci telo poznanky
  def definition
    # ...
  end

  private
  #funkcia na vytvorenie grafu odkazov reprezentovaneho pomocou matice susednosti
  def buildAdjacencyMatrix(objectsArray, outgoingEdgesArrays)
    # ...
  end

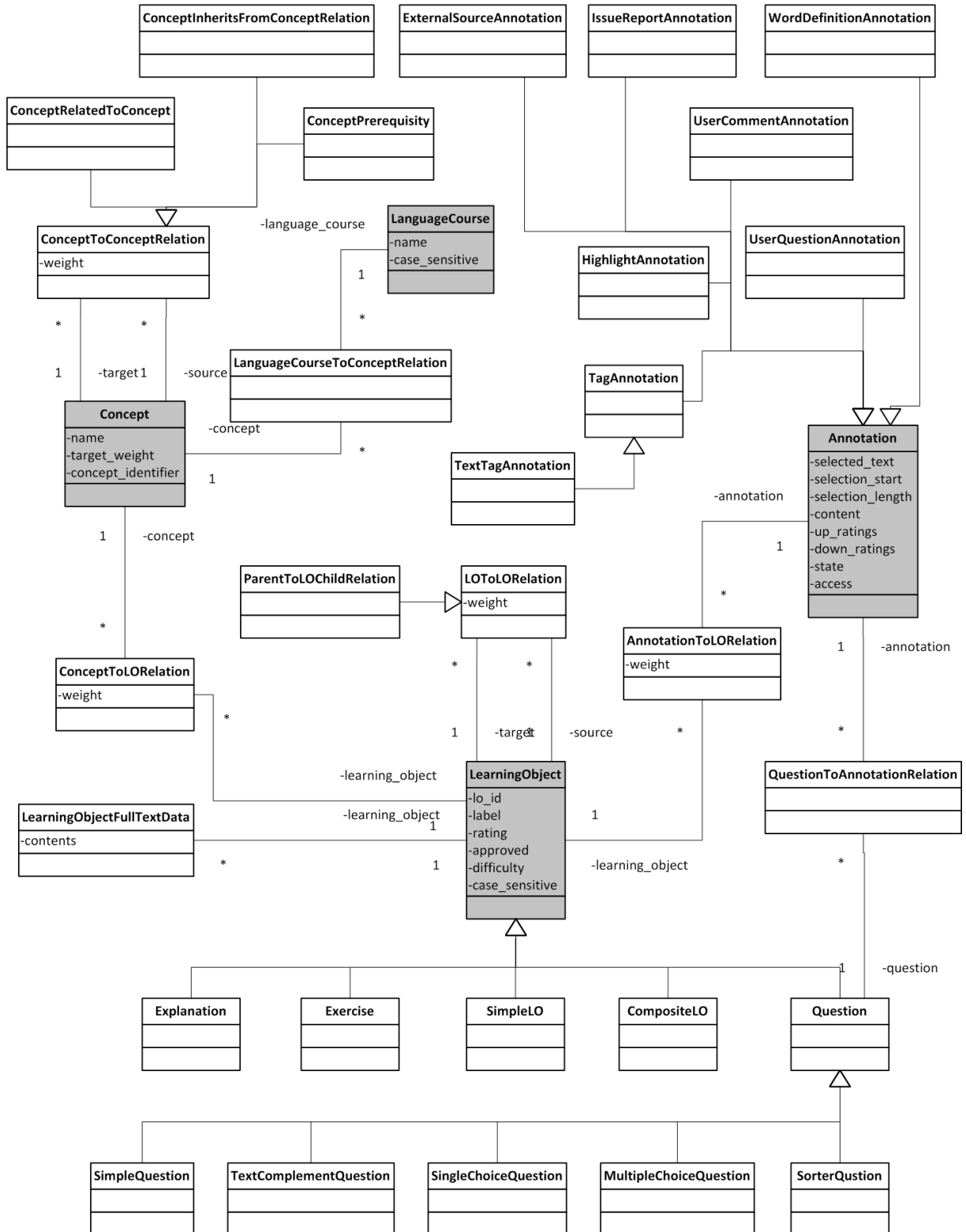
  #implementacia iterativnej verzie PageRank algoritmu
  #graf je zadany pomocou matice susednosti
  def pageRank(adjacency_matrix)
    # ...
  end
end
```

A.4 Zjednodušený datový model systému ALEF



A.5 Doménový model systému ALEF

Autori: Marián Šimko, Michal Barla, Pavel Michlík, Martin Labaj, Vladimír Mihál, Maroš Uncík, Róbert Móro, Jakub Ševcech



B Používateľská príručka

*Autori: Mária Bieliková, Marián Šimko, Michal Barla, Pavel Michlík, Martin Labaj,
Vladimír Mihál, Maroš Unčík, Jakub Ševcech, Máté Fejes*

1 Prístup do systému ALEF

Systém ALEF (Adaptive Learning Framework) je adaptívny výučbový systém na podporu výučby na predmetoch Funkcionálne a logické programovanie a Procedurálne programovanie na Fakulte informatiky a informačných technológií na Slovenskej Technickej Univerzite v Bratislave. Aktuálnu ukážkovú verzia tohto systému možno nájsť na webovej adrese

<http://alef.fiit.stuba.sk/>

1.1 Prihlásenie do systému

Po návšteve webovej adresy výučbového systému ALEF sa zobrazí úvodná obrazovka, kde sa vo vrchnej časti nachádza prihlasovací formulár (Obr. 1). V prípade, že máte vytvorené konto v systéme ALEF, zadajte svoje prihlasovacie meno a heslo.

ALEF

- › **Kto som?**
Ahojte, volám sa ALEF. Som *adaptívny* webový výučbový systém a som založený na konceptoch únes toľko spomínaného *webu 2.0*. Motivujem študentov k aktívnemu učeniu sa a vzájomnej spolupráci pomocou rôznych zaujímavých strojčekov, ktoré do mňa naskladali moji autori.
- › **Prečo som tu?**
Chcem, aby sa študenti učili *efektívnejšie*. Som tu, aby som posunul hranice súčasného vzdelávania v prostredí webu a urobil ho atraktívnejším pre každého tvoj, že sa mu prispôbim. Snažím sa, aby sa neprispôboval len výučbové texty, ale aj inými *študentmi vloženými obsah*, akým sú komentáre, tagy a pod. Časť rozhodnutí o tom, čo, ako a kedy sa učiť, prevezmem na seba (a na moje škóvné strojčeky), a študentovi tak ušetrim čas. Naučí sa toho viac, ako keby sa mal sám prehrabovať v hromadách výučbových materiálov, či hľadať vhodné príklady na precvičovanie.
- › **Pre koho som stvorený?**
Som stvorený pre všetkých *pozanáchtivých* ľudí. V mojej súčasnej inkarnácii však slúžim študentom na Fakulte informatiky a informačných technológií pri výučbe *funkcionálneho programovania* a *logického programovania* (na predmete *Funcionálne a logické programovanie*), *procedurálneho programovania* (na predmete *Procedurálne programovanie*) a *softvérového inžinierstva* (v predmetoch *Princípy softvérového inžinierstva* a *Tvorba softvérových systémov*).
- › **Čo mám rád?**
Mám rád študentov, ktorí mi roba dobre. Dobre mi roba vtedy, keď do mňa často klikajú a keď mi na otázky, ktoré sa ich pýtam, odpovedajú pravdivo. Teším sa, keď vložia múdru poznámku, či prídajú novú zaujímavú otázku, ktorá pomôže ďalším pri učení.
- › **Čo nemám rád?**
Nemám rád výpadky elektriny, sieťového spojenia a zlyhania diskov. Stále nemám svoj dedikovaný domov. Skúste preto nepretlažiť moje krahké spojenie s Internetom a ak budem náhodou pomalší, tak so mnou majte strpenie.
- › **Kto ma vytvoril?**
Vznikol som (a neustále ma vylepšujú) v rámci série projektov zameraných na podporu vzdeávania na Ústave informatiky a softvérového inžinierstva na FIIT STU. Som veľmi pyšný na to, že ma stvoril múdre havičky a škóvné ručičky študentov doktorandského, inžinierskeho a bakalárskeho štúdia pod vedením prof. Márie Bielikovej.

Používateľská príručka

Obr. 1. Úvodná obrazovka a prihlasovací formulár.

2 Práca so systémom

Po prihlásení do systému sa nezávisle od role používateľa zobrazí úvodná obrazovka systému (Obr. 2). Ak nie je pri komponente uvedené inak, zobrazuje sa používateľom oboch rolí. Úvodná obrazovka je rozdelená na štyri časti:

1. navigačná časť
 - zobrazuje odporúčanie pre študentov
 - zobrazuje menu pre navigáciu
2. obsahová časť
 - zobrazuje aktuálne zvolený výučbový materiál
3. anotačný pásik znázorňujúci poznámky a nahlásené chyby
4. časť so zásuvnými modulmi
 - zobrazuje zásuvné moduly v systéme

The screenshot shows the AleF system interface. At the top, there is a navigation bar with the AleF logo and menu items: Administrácia, Lisp, Prolog, and C. The user's name, Vladimír Mihal (študent), and a link to report a problem (Odhlásiť) are also visible. The main content area is divided into several sections:

- Podobné témy**: A blue sidebar on the left containing a search bar and a list of topics. The current topic is "znak nového riadku '\n' biely znak". Below this, there are sections for "Odporúčame" (Recommendations) and "Texty" (Texts). The "Texty" section is highlighted with a yellow circle labeled "1".
- Bílé znaky**: The main content area displays the title "Bílé znaky" and a paragraph of text explaining white space characters. A yellow circle labeled "2" is placed over the text.
- Filter**: A search filter bar is located above the main content area.
- Annotations and Errors**: A horizontal bar below the main content area displays a note from Pavel Herout about purchasing a book. A yellow circle labeled "3" is placed over this bar.
- Right Sidebar**: Contains several modules:
 - Tvoje skóre**: Shows a score of 51.9% and a message "Máš najlepšie skóre zo všetkých!".
 - Nahlásené chyby**: Shows a message "V tejto časti kurzu nie sú hlásené žiadne chyby.".
 - Externé zdroje**: Shows a message "K tomuto obsahu nie sú k dispozícii žiadne odkazy na externé zdroje" and a button "Pridaj externý zdroj".
 - Tagy**: Shows a message "K tomuto obsahu nie sú k dispozícii žiadne tagy" and a button "Pridaj tag". A yellow circle labeled "4" is placed over this section.

Obr. 2. Úvodná obrazovka.

2.1 Navigačná časť

Hlavný komponent navigačnej časti tvorí menu. Jeho obsahom je zoznam všetkých vzdelávacích objektov. Zoznam je usporiadaný podľa kapitol, t.j. v rámci daných

vzdelávacích objektov sa môžu nachádzať aj ďalšie. Kliknutím na kapitolu sa vybraná položka rozbalí a zobrazia sa k nej patriace podkapitoly.

Menu (Obr. 3) sa skladá z troch záložiek:

- **Texty** – záložka obsahuje zoznam výučbových textov,
- **Otázky** – záložka obsahuje zoznam otázok, na ktoré je možné odpovedať,
- **Cvičenia** – záložka obsahuje zoznam cvičení, ktoré je možné vyriešiť.

Obr. 3. Ukážka navigačnej časti.

V navigačnej časti sa okrem menu nachádza oblak kľúčových slov (nad menu, Obr. X), ktorý dopĺňa funkcionality menu. V oblaku sú zobrazené kľúčové pojmy, ktoré reprezentujú určité témy učiva.

Oblak slov pozostáva z dvoch častí:

- **Podobné témy** – obsahuje kľúčové slová, ktoré opisujú aktuálne otvorený vzdelávací objekt,
- **Odporúčame** – koncepty v tejto časti odporúčajú témy, ktoré sú určené na základe aktuálnych vedomostí študenta.

Konkrétne vzdelávacie objekty z danej témy sa získavajú kliknutím na vybrané kľúčové slovo v oblaku. Po kliknutí sa v menu vyznačia vzdelávacie objekty relevantné so zvoleným pojmom.

Po kliknutí na koncept sa pri vyznačených položkách v menu zobrazí jedna z anotácií uvedených nižšie:

- ★ – vzdelávací objekt je do veľkej miery relevantný s vybraným slovom,

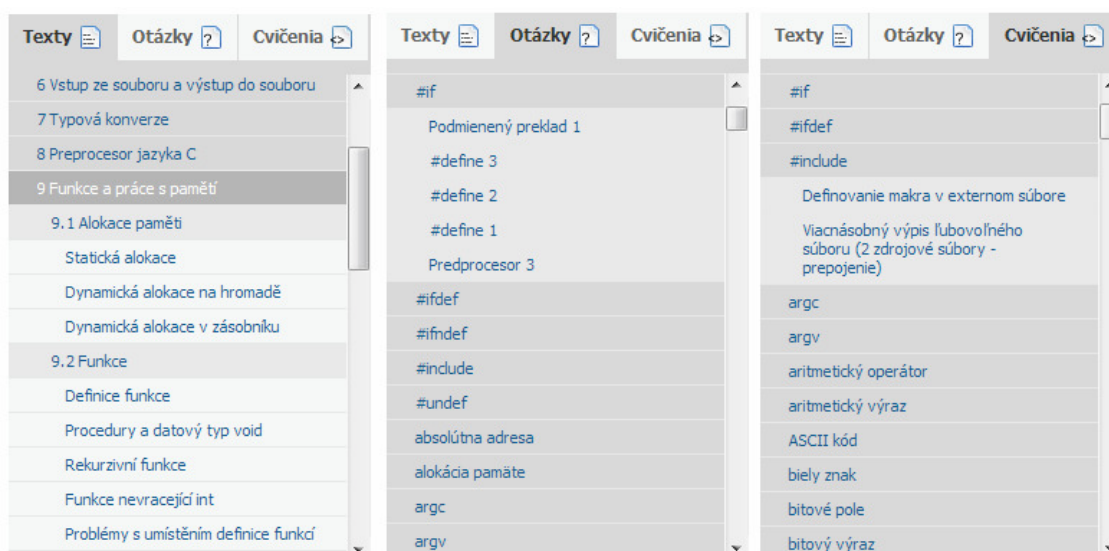
- 🌟 – vzdelávací objekt je mierne relevantný s vybraným slovom,
- 🌟 – relevancia vzdelávacieho objektu je minimálna,
- 📄 – vzdelávací objekt nie je relevantný s výberom, ale v rámci neho sa nachádzajú podkapitoly, pri ktorých je zobrazená jedna z prvých troch anotácií, t.j. treba ho rozbaľiť, aby sme sa dostali k požadovaným materiálom.

Vyznačenie objektov sa dá zrušiť kliknutím na ikonu 🚫 v pravom dolnom rohu oblaku.

2.2 Obsahová časť

V závislosti od vybraného výučbového materiálu sa môžu v obsahovej časti nachádzať tri typy zobrazených materiálov:

1. **otázka,**
2. **cvičenie,**
3. **výučbový text.**



Obr. 4. Ukážky menu. Zľava záložka výučbových textov, v strede záložka otázok a vpravo záložka príkladov.

2.2.1 Odpovedanie na otázku

Otázka je typ výučbového materiálu, na ktorý sa dá odpovedať a systém vyhodnotí odpoveď študenta. V závislosti od typu otázok sa zobrazí príslušný formulár na jej zodpovedanie (Obr. 5). V každom formulári študent vyplní správnu odpoveď a klikne na tlačidlo *Odpovedaj*. Systém následne vyhodnotí odpoveď študenta a zobrazí výsledok (Obr. 6).

Makro s parametrom 1 Vaše aktuálne hodnotenie: ?

Čo platí pre makro s parametrom?

niekedy sa takéto makrá nazývajú aj vkladané funkcie

pred použitím makra v programe, sa makro nahradí konkrétnym textom

sú veľmi vhodné na nahradenie dlhých funkcií, lebo sa výpočet v nich prebieha rýchlejšie

syntakticky ich môžeme zapísať ako: #define meno_makra(arg1, arg2 ..., argN) hodnota_makra

Odpovedaj

Neviem odpovedať

Obr. 5. Formulár pre zodpovedanie otázky.

Makro s parametrom 1 Vaše aktuálne hodnotenie: ?

Táto otázka bola vyriešená

Čo platí pre makro s parametrom?

niekedy sa takéto makrá nazývajú aj vkladané funkcie

pred použitím makra v programe, sa makro nahradí konkrétnym textom

sú veľmi vhodné na nahradenie dlhých funkcií, lebo sa výpočet v nich prebieha rýchlejšie

syntakticky ich môžeme zapísať ako: #define meno_makra(arg1, arg2 ..., argN) hodnota_makra

Odpovedali ste nesprávne!

Odhodnoťte užitočnosť:

 Neúžitočný

 Neutrálny

 Užitočný

Odporúčame:

Tieto materiály vám môžu pomôcť:

- » [8.1 Makra bez parametrov – príkaz define \(#define\)](#)
- » [8.2 Makra s parametrami \(makro\)](#)

Obr. 6. Zodpovedaná otázka.

Zelenou farbou sú vyznačené správne odpovede, červenou sú označené nesprávne odpovede. Čiernym rámčekom sú vyznačené odpovede, ktoré označil študent.

Pre rýchlu navigáciu možno použiť tlačidlo *Predchádzajúci príklad* resp. *Ďalší príklad*, ktoré zobrazia ďalšiu otázku na riešenie.

2.2.2 Odpovedanie na cvičenie

Cvičenie je typ výučbového materiálu, na ktorý sa dá vyriešiť a následne oznámiť systému postup v riešení (Obr. 7). Študent postupne kliká na jednotlivé možnosti, ktoré mu systém ponúka pri riešení. Možnosti sú nasledovné:

1. Poznám riešenie

Systém zobrazí správne riešenie. Následne si študent vyberie z možností:

- a. Moje riešenie je také isté, ako uvádza vzorové riešenie*
- b. Moje riešenie je iné, ale myslím, že správne*
- c. Moje riešenie je nesprávne, ale teraz už tomu rozumiem*
- d. Moje riešenie je nesprávne a stále tomu nerozumiem*

2. Nepoznám riešenie

Systém zobrazí pomôcku. Následne si študent vyberie z možností:

a. Už poznám riešenie

Systém zobrazí správne riešenie. Následne si študent vyberie z možností:

- i. Moje riešenie je také isté, ako vzorové riešenie*
- ii. Moje riešenie je iné, ale myslím, že správne*
- iii. Moje riešenie je nesprávne, ale už tomu rozumiem*
- iv. Moje riešenie je nesprávne a stále tomu nerozumiem*

b. Stále nepoznám riešenie

Systém zobrazí správne riešenie. Následne si študent vyberie z možností:

- i. Riešeniu rozumiem*
- ii. Riešeniu nerozumiem*

Pre rýchlu navigáciu je možné použiť tlačidlo *Predchádzajúci príklad* resp. *Ďalší príklad*, ktoré zobrazia ďalšie cvičenie na riešenie.

Definovanie makra na test veľkosti písmena

Vaše aktuálne hodnotenie:

Zadanie:

Definujte makro `je_veľké(x)`, ktoré vráti hodnotu `0` ak znak nie je veľké písmeno a hodnotu `1` ak je znak veľké písmeno.

Poznám riešenie

Nepoznám riešenie

Pavel Herout: Učebnice jazyka C - knihu si môžete zakúpiť v internetovom obchode www.eplanet.sk.

◀ Predchádzajúci príklad Nasledujúci príklad ▶

Obr. 7. Formulár pre zodpovedanie cvičenia.

2.2.3 Výučbový text

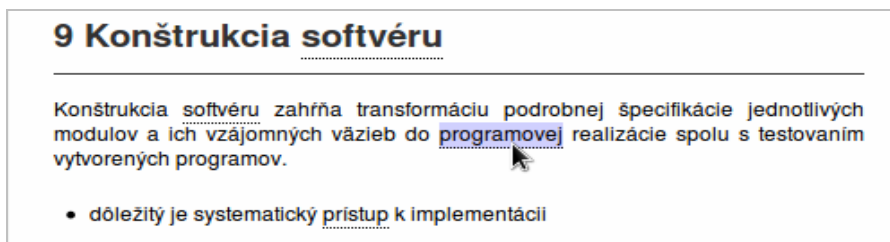
Výučbový text je statický text, ktorý môže obsahovať tabuľky, obrázky, grafy, hypertextové odkazy a ďalšie prvky. V niektorých výučbových textoch sú vložené vnorené cvičenia a otázky, ktoré sa viažu k danému výučbovému textu (Obr. 9). Tieto otázky a cvičenia je možné vyriešiť.

- Šípkami vľavo a vpravo sa možno pohybovať medzi vloženými otázkami a príkladmi,
- Samotné odpovedanie na otázku alebo príklad prebieha rovnako ako je to opísané v častiach Odpovedanie na otázka a Odpovedanie na cvičenie.

2.2.4 Poznámky priradené ku kľúčovým slovám v texte



K niektorým kľúčovým slovám vo výučbovom texte sú priradené automaticky generované poznámky. Tieto poznámky rozširujú informácie o danom kľúčovom slove tým že poskytujú jeho definíciu a odkazy na súvisiace stránky.

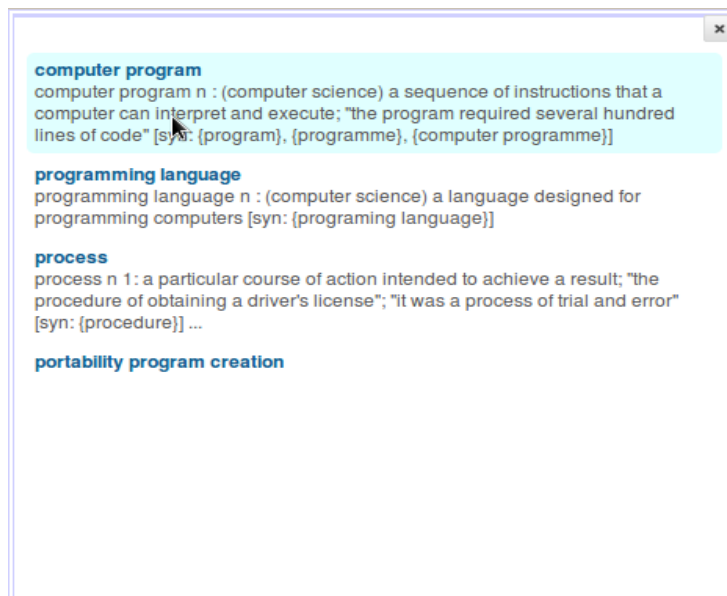
Slová, ku ktorým je takáto poznámka priradená sú v texte zvýraznené podčiarknutím a zmenou farby po prechode myšou nad takýmto slovom (Obr. 8).



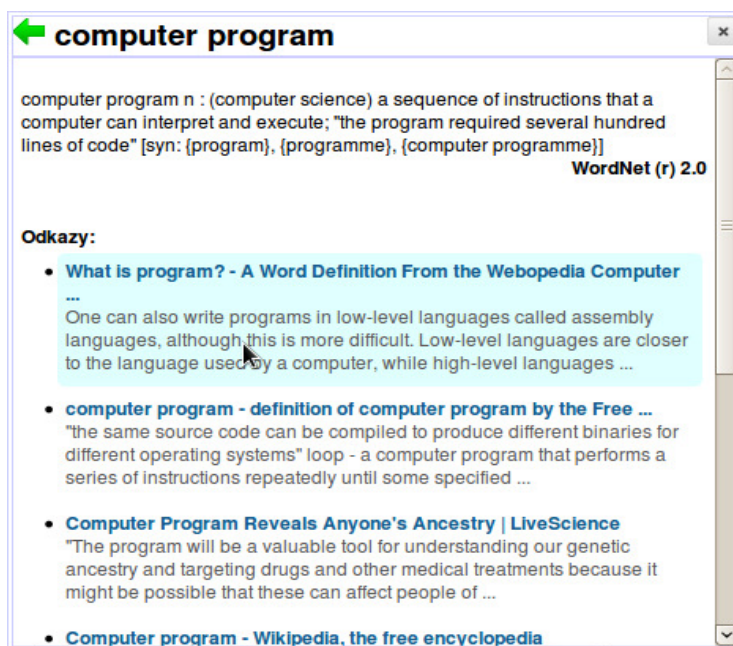
Obr. 8. Zvýraznenie slov ku ktorým je pripojená automaticky generovaná poznámka

Poznámka sa zobrazí vo forme tooltipu po kliknutí na zvýraznené slovo. Tooltip obsahujúci poznámku má viacero stavov medzi ktorými je možné prechádzať. Po kliknutí na zvýraznené slovo v texte sa zobrazí zoznam poznámok priradených k tomuto slovu (Obr. 9). Zvolením jednej z týchto poznámok sa tooltip presunie do stavu, kde zobrazuje obsah tejto poznámky (Obr. 10). Obsah poznámky tvorí definícia slova, zoznam odkazov na súvisiace stránky a zoznam pripojených prezentácií. Po kliknutí na odkaz na súvisiacu stránku sa na novej karte prehliadača otvorí táto stránka. Po kliknutí na odkaz zo zoznamu prezentácií sa tooltip presunie do ďalšieho stavu, kde bude v jeho tele zobrazená zvolená prezentácia.

V každom stave tooltipu je možné vrátiť sa o krok späť pomocou tlačidla . Zatvoriť tooltip je možné kliknutím mimo tooltipu do výučbového textu, kliknutím na tlačidlo  alebo stlačením klávesy Esc.



Obr. 9. Tooltip so zoznamom poznámok priradených k zvolenému slovu



Obr. 10. Tooltip zobrazujúci obsah poznámky

2.3 Anotačný pásik

Anotačný pásik (Obr. 11) znázorňuje komentáre vytvorené používateľmi a nahlásené chyby vo výučbových objektoch. Ich prítomnosť reprezentuje modrý obdĺžnik. Anotácia je zobrazená ukázaním kurzorom na modrú oblasť. Kliknutím alebo podržaním myši nad modrou oblasťou zostane anotácia zobrazená trvalo. Ďalším kliknutím ju opäť uvoľníme.

Pre filtrovanie medzi poznámkami a nahlásenými chybami slúži filter (Obr. 11 vpravo hore). Prvá ikona predstavuje zobrazenie tagov vložených z textu, druhá ikona predstavuje zvýraznené výseky textu, tretia predstavuje zobrazenie poznámok a posledná zobrazenie nahlásených chýb. Používateľ si tak môže zvoliť, ktoré anotácie chce v danom výučbovom materiáli vidieť.

8 Preprocesor jazyka C

Preprocesor je něco, co Pascal nezná. My jsme zatím využívali víceméně nevědomky nejčastěji používaný příkaz preprocesoru `#include`. Preprocesor má ovšem navíc řadu jiných možností, které dávají jazyku C další "sílu". Je vhodné se s těmito příkazy seznámit, protože, i když je nebudeme hned využívat úplně všechny, je dobré alespoň vědět, že něco takového existuje.

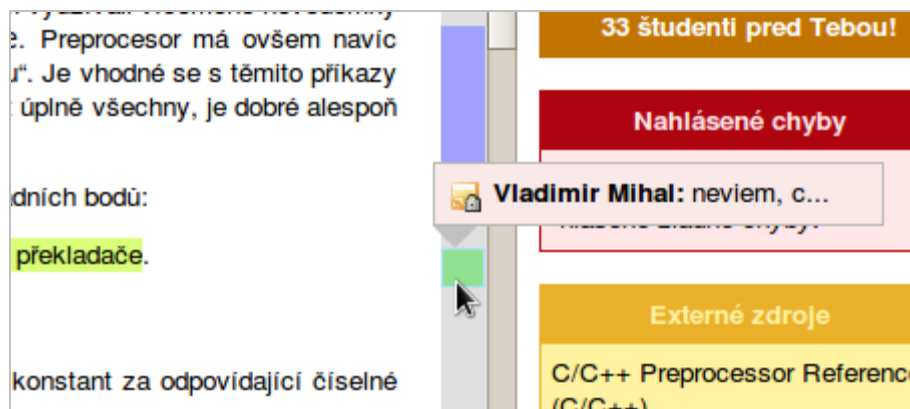
Činnost preprocesoru se dá shrnout do několika základních bodů:

- Zpracovává zdrojový text programu *přes* použitím překladače.
- Nekontroluje syntaktickou správnost programu.
- Provádí pouze *záměnu textů*, např. identifikátorů konstant za odpovídající číselné hodnoty^[1].
- Vypustí ze zdrojového textu všechny komentáře.
- Přípravuje podmíněný překlad.

Poznámka:

- Řádka, která je určena pro zpracování preprocesorem musí začínat znakem `##`

Obr. 11. Anotačný pásik. Nachádza sa napravo od výučbového materiálu. Modrou sú znázornené anotácie vložené používateľmi.







Obr. 12. Zobrazenie anotácie.

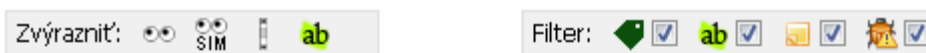
2.4 Zvýraznenie dôležitých fragmentov

Zvýrazňovač dôležitosti fragmentov umožňuje podfarbiť časti zobrazeného výučbového objektu podľa informácií o tom, ako používatelia v minulosti pracovali s týmto objektom. Predstavuje tak pomôcku pri zrýchlenom učení, revízii, opakovaní už naučeného textu a pod.

Ovládacie tlačidlá (Obr. 13) sa zobrazia nad výučbovým objektom vľavo hore v prípade, že sú informácie o činnosti používateľov pre daný objekt dostupné a je

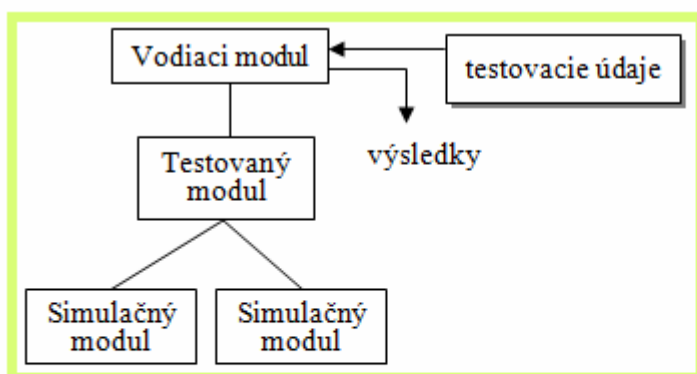
použitý podporovaný prehliadač (*Mozilla Firefox*). Jednotlivé tlačidlá umožňujú zobrazit' zvýraznenie podľa primárneho zdroja informácií:

1.  Pohľad,  Pohľad odhadnutý podľa myši – podľa času, ktorý sa používatelia pozerali na dané časti dokumentu,
2.  Opotrebovanie dokumentu – podľa virtuálneho opotrebovania dokumentu, čiže ako dlho boli jednotlivé časti zobrazené (vhodné pre dlhšie texty, ktorých obsah bolo potrebné posúvať),
3.  Zvýraznenia používateľov – podľa ručného označovania dôležitých častí používateľmi.



10.4 Stratégie testovania

- prístup k testovaniu celého systému
 - testovanie zdola-nahor
 - testovanie zhora-nadol
 - jednofázové testovanie
 - sendvičové testovanie
 - testovanie porovnávaním



Testovanie zdola-nahor
angl. bottom-up testing

- vhodné, ak veľa modulov "nižšej" úrovne predstavuje všeobecné moduly, ktoré iné časti systému často používajú

Obr. 13. Zvýrazňovač dôležitosti fragmentov (vľavo hore).






2.5 Časť so zásuvnými modulmi

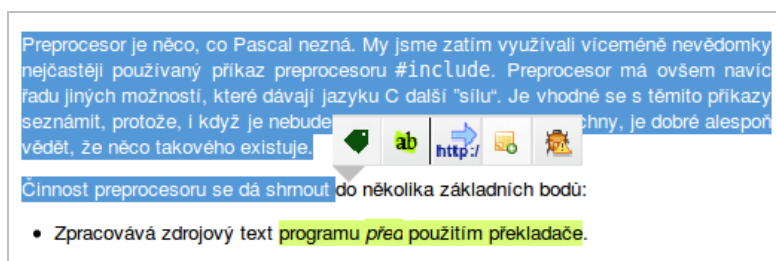
V pravej časti obrazovky sa nachádzajú zásuvné moduly systému. V aktuálnej verzii systému sa nachádzajú štyri moduly:

1. *Komponent pre zobrazenie skóre* – zobrazuje bodový zisk prihláseného študenta. Body sú študentovi pripisované v závislosti od aktivít, ktoré vykoná v súvislosti s rôznymi časťami systému (pridá tag do textu, nahlási chybu, pridá poznámku ...).
2. *Komponent pre zobrazovanie nahlásených chýb* – zobrazuje nahlásené chyby vo výučbových materiáloch od používateľov.
3. *Komponent pre zobrazenie a pridávanie externých odkazov* – zobrazuje externé zdroje, ktoré pridali študenti alebo vyučujúci k danému výučbovému materiálu. Umožňuje tiež pridať ďalšie externé zdroje.
4. *Komponent pre zobrazenie a pridávanie tagov* – zobrazuje tagy priradené k danému výučbovému materiálu. Umožňuje tiež pridať ďalšie tagy či už priamo jeho napísaním alebo označením časti textu, ktorý sa má použiť ako tag.

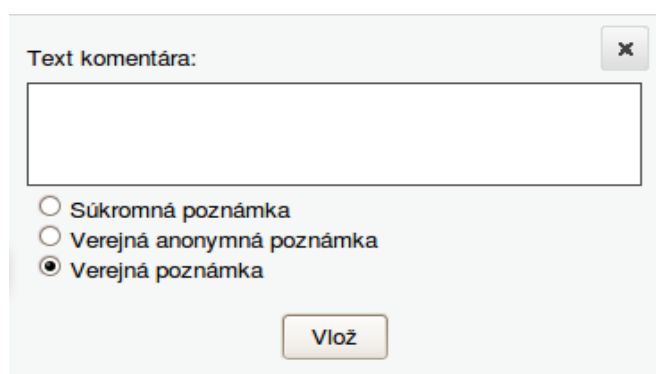
2.6 Pridávanie poznámok

Pridávať rôzne druhy poznámok sa v systéme dá k ľubovoľnému výučbovému materiálu (textom, otázkam a aj príkladom). Používateľ označí text, ku ktorému chce pridať poznámku. Následne si z kontextového menu môže vybrať jednu z možností (Obr. 14):

1.  *Pridanie tagu* – slúži na pridanie vyznačeného textu ako tag. Tagy sa zobrazujú zvýraznením pozadia otagovaného textu ako aj v komponente na pridávanie a zobrazovanie tagov.
2.  *Označenie textu* – vyznačenému textu sa zmení pozadie. Táto funkcia slúži na zvýraznenie dôležitých alebo inak zaujímavých častí textu.
3.  *Pridanie externého zdroju* – pomocou tejto voľby je možné k textu priradiť externý odkaz, ktorý sa bude zobrazovať v komponente na pridávanie a zobrazovanie externých odkazov.
4.  *Nový komentár* (Obr. 15) – slúži na pridanie poznámky k výučbovému materiálu. Poznámka sa zobrazuje na mieste, ku ktorému bola pridaná. Poznámka je viditeľná pre všetkých používateľov.
5.  *Nahlásenie chyby* – umožňuje pridať poznámku, ktorá upozorňuje ostatných študentov a učiteľa, že v danom výučbovom materiáli sa nachádza chyba.



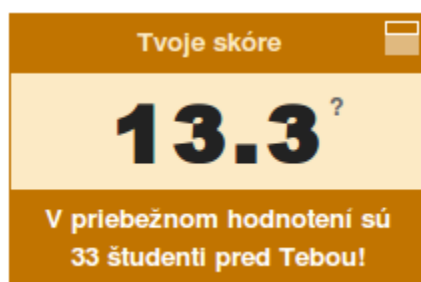
Obr. 14. Označení textu.



Obr. 15. Rozhranie pre pridanie komentáru.

2.7 Komponent pre zobrazovanie skóre

Komponent pre zobrazenie skóre (Obr. 16) slúži ako informatívny prvok pre študentov, ktorý sa využíva pri hre za účelom motivácie. Hra spočíva v získavaní bodov a snahe študentov dosiahnuť čo najvyššie skóre. Komponent zobrazuje aktuálnu výšku bodov, ktoré študent získal za rôzne akcie v systéme. Komponent zároveň zisťuje poradie študenta v celkovom bodovom poradí a tento údaj zobrazuje, za účelom motivácie študentov.





Obr. 16. Komponent pre zobrazenie skóre.

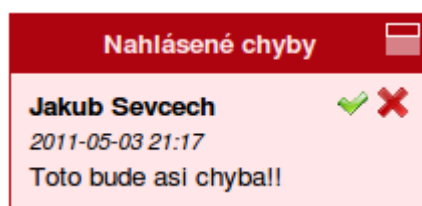
2.8 Komponent pre zobrazenie nahlásených chýb

Komponent pre zobrazenie nahlásených chýb (Obr. 17) slúži ako informatívny prvok. Komponent zobrazuje nahlásené chyby od používateľov v tabuľke v poradí:

používateľské meno autora nahlásenej chyby, čas nahlásenia chyby a text hlásenia resp. popis chyby. Po prechode myšou nad textom hlásenia, vo výučbovom texte sa zvýrazní časť textu, ku ktorému bola chyba nahlásená. V pravej hornej časti sú ku každej nahlásenej chybe zobrazené dve ikony:


-  Slúži na označenie chyby za vyriešenú, takto označená chyba sa ďalej nebude zobrazovať.
-  Slúži na zmazanie chybového hlásenia

Tieto ikony sú dostupné len pre používateľov s rolou *Administrátor* a *Učiteľ*.

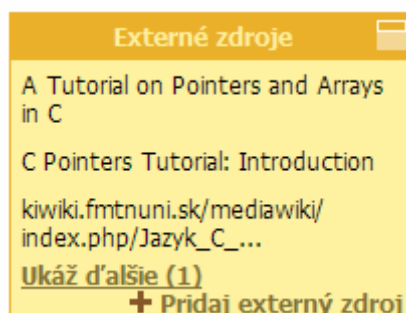


Obr. 17. Komponent pre zobrazenie nahlásených chýb.

2.9 Komponent pre prácu s externými zdrojmi

Komponent pre prácu s externými zdrojmi umožňuje používateľom vkladať odkazy na stránky s obsahom súvisiacim s obsahom aktuálne zobrazeného vzdelávacieho objektu. Používatelia môžu taktiež tieto odkazy hodnotiť. Rovnako ako u iných typov poznámok, pri vkladaní externých zdrojov môže používateľ špecifikovať ich viditeľnosť – teda či sú súkromné, anonymné alebo verejné. Formulár na vkladanie odkazu na externý zdroj sa zobrazí po kliknutí na odkaz „ Pridaj externý zdroj“.

V komponente sú zobrazené všetky externé zdroje vložené k momentálne zobrazenému vzdelávaciemu objektu. Po zobrazení objektu sú v komponente zobrazené iba tri externé zdroje s najvyšším hodnotením od používateľov. Všetky externé zdroje používateľ zobrazí kliknutím na odkaz „Ukáž ďalšie“.

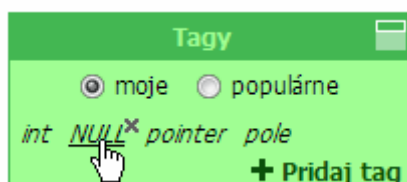


Obr. 18. Komponent pre prácu s externými zdrojmi.

2.10 Komponent pre zobrazenie a pridávanie tagov

Komponent pre zobrazenie a pridávanie tagov umožňuje používateľom pridávať tagy ku vzdelávacím objektom, prehliadať ich a odstraňovať. Používatelia môžu pri vkladaní tagu špecifikovať, či bude vložený ako verejný, anonymný alebo súkromný. Tag môže používateľ odstrániť nadídením nad tag a kliknutím na tlačidlo „X“.

Používateľ môže prepínať medzi zobrazením jeho vložených tagov a populárnych tagov. Populárnym sa tag vtedy, pokiaľ je verejný a je k vzdelávaciemu objektu vložený minimálne tromi používateľmi.

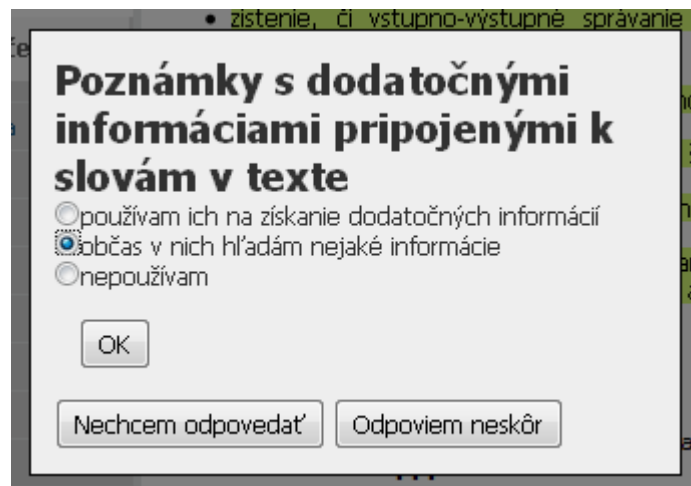


Obr. 19. Komponent pre zobrazenie a pridávanie tagov.

2.11 Adaptívne systémové otázky

System môže v prípade vhodnej situácie vygenerovať a zobraziť otázku (Obr. 20), slúžiacu väčšinou na získanie odozvy od používateľa na funkcie systému. Používateľ odpovie vyplnením odpovede podľa typu otázky (výber jednej možnosti, vpísanie reakcie a pod.) a stlačením tlačidla *OK* odpoveď odošle. Ak používateľ nechce na otázku odpovedať, môže stlačením tlačidla *Nechcem odpovedať* alebo *Odpoviem neskôr* odpoveď zrušiť, respektíve odložiť.

V prípade odmietnutia odpovede sa práve zobrazená otázka deaktivuje a už sa nezobrazí (čo však nevyklučuje, že systém neskôr nevygeneruje rovnakú otázku nanovo, ak to bude vhodné). V prípade odloženia odpovede sa otázka prestane zobrazovať, ale o krátku chvíľu sa zobrazí znova. Niektoré otázky sú špeciálne zamerané na aktuálny okamih a preto sa už znova nezobrazia, ani ak ich používateľ iba odloží tlačidlom *Odpoviem neskôr*.



Obr. 20. Príklad systémovej otázky. Zobrazí sa v popredí pred výučbovým objektom.

C Inštalačná príručka

Pre nainštalovanie a spustenie systému ALEF ako aj webovej služby na vytváranie poznámok je potrebné mať nainštalované:

- interpretér jazyka Ruby verzie 1.8.7,
- Rails 2.3.5.
- MySQL 5

zvyšné kroky sú špecifické pre jednotlivé aplikácie

Inštalácia a spustenie systému ALEF

Vytvorte novú MySQL databázu a importujte do nej priložený súbor alef.sql

Na toto je možné napríklad v príkazovom riadku zadať príkaz:

```
mysql -user root -p meno_databazy < alef.sql
```

Rozbaľte archív alef.zip do adresára z ktorého chcete aplikáciu spúšťať. Následne je potrebné nastaviť prístupové údaje v súbore config/database.yml, ktorý sa nachádza v adresári, do ktorého ste rozbalili archív alef.zip

- pri voľbe „common“ je potrebné nastaviť používateľa a heslo na prístup k databáze prípadne ostatné potrebné údaje
- pri voľbe „development“ je potrebné nastaviť meno databázy, do ktorej ste importovali súbor alef.sql

Následne je potrebné v súbore config/is_server prepísať hodnotu true na false.

V príkazovom riadku sa nastavte na adresár, kde ste rozbalili archív alef.zip a zadajte nasledujúce príkazy, ktoré nainštalujú potrebné gem balíčky:

```
gem install nokogiri mysql
rake gems:install
```

V tejto chvíli by mal byť ALEF pripravený na spustenie. Spustiť ho môžete príkazom:

```
ruby script/server
```

Tento príkaz sa znovu spúšťa pomocou príkazového riadku nastaveného na adresár kde ste rozbalili súbor alef.zip

ALEF je teraz dostupný na adrese <http://localhost:3000/> prostredníctvom vášho webového prehliadača.

V prípade neúspechu je možné si vyskúšať aplikáciu na adrese <http://alef.fiit.stuba.sk>

Inštalácia a spustenie webovej služby na vytvorenie poznámok

Vytvorte novú MySQL databázu. Rozbaľte adresár WebService.zip do adresára odkiaľ chcete aplikáciu spúšťať. Z príkazového riadka nastaveného na adresár kde ste rozbalili súbor WebService.zip spustíte nasledujúce príkazy na nainštalovanie potrebných gem balíčkov:

```
gem install activerdf
gem install activerdf_sparql
gem install ruby-readability
gem install mysql
gem install nokogiri
rake gems:install
```

Následne je potrebné nastaviť prístupové údaje v súbore config/database.yml, ktorý sa nachádza v adresári, do ktorého ste rozbalili archív WebService.zip. Pri voľbe „development“ je potrebné nastaviť meno a prístupové údaje k vytvorenej MySQL databáze.

Následne je potrebné vytvoriť v databáze potrebnú štruktúru tabuliek. Na toto je potrebné spustiť príkaz:

```
rake db:schema:load RAILS_ENV="development"
```

V tejto chvíli by mala byť webová služba pripravená na spustenie. Spustiť ju môžete príkazom:

```
ruby script/server
```

Tento príkaz sa znovu spúšťa pomocou príkazového riadku nastaveného na adresár kde ste rozbalili súbor WebService.zip

Webová služba je teraz dostupná na adrese <http://localhost:3000/> prostredníctvom vášho webového prehliadača.

V prípade neúspechu je možné si vyskúšať aplikáciu na adrese

<http://leela.fiit.stuba.sk/word-definition>

**D Príspevok publikovaný na konferencii
IIT.SRC 2011**

Automatic Web Content Annotation

Jakub ŠEVCECH*

Slovak University of Technology
Faculty of Informatics and Information Technologies
Ilkovičova 3, 842 16 Bratislava, Slovakia
Sevo_jakub@yahoo.fr

Abstract. Today we are facing with information overload. It is thus important to provide information to a reader in such a way, that it can be used effectively and can be the most useful. In this paper we propose a method for automatic extending the information content by adding annotations to keywords in the text of web pages. The method is designed to be able to insert annotations into the text written in Slovak. Information to fill annotations is obtained through publicly available services for information retrieval. Since most of services for text analysis works primarily in English, a method for creating annotations uses text of web page translated into English. To assign annotations to the words in the original text, we propose a method for mapping equivalent words between the Slovak text and its translation into English.

1 Introduction

When reading a web page, visitor often encounters a word or phrase, he does not understand, or would require some additional information about this term. This situation occurs more frequently if the page contains technical or explanatory text, such as for example in digital libraries [2] and various educational sites [11]. Common scenario that follows when a website visitor encounters an unknown expression is the following: the user opens a new tab in his web browser, displays his favorite search engine and starts to search for an explanation of this term. This scenario has several drawbacks for the user and for the author of the document as well. Website visitor have to stop work with document, because without understanding the term, he cannot continue. Moreover, searching for information on unknown word means additional effort for the document reader. After finding answers, the reader at best returns to the original site and continues reading the document. Again, he must make an additional effort to recall or to re-discover, what he worked at before he left the document. In the worst case the reader does not return any more to the original page so the author has lost one reader.

One of the solutions to this situation is, if an annotation was attached to that unknown word. Such annotation can immediately explain to user what the word means. Currently several rather popular tools for manual document annotation exist [8]. They often have the possibility of creating the annotations collaboratively. These tools provide a number of supporting instruments to users, through which they can add annotations into document and then share them among themselves. However, with the existing amount of information and documents on the Internet, it is impossible

* Bachelor degree study programme in field: Informatics
Supervisor: Professor Mária Bielíková, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

for document users to annotate them all. Furthermore, it is extremely difficult to ensure sufficient quality. Methods and tools for automatic annotation the documents can help in this issue.

In this paper we propose a method able to automatically add annotations to the keywords in the text of web pages. We consider the annotation a definition of the word, or a link to other web page associated with that word. The method consists of several steps, processing of document, creation of annotations and finally their visualization. By document processing is generally thought search of words or phrases, for which it is necessary to attach an annotation. For search for information to fill the annotations, we propose use of publicly available services, just like in [5] is proposed use of various publicly available services for keyword extraction, to create user model.

Suitable domain for the validation of methods for automatic document annotation is the domain of educational systems. When using educational system, annotations added to the keywords in the text are particularly important. Therefore, we decided to verify the proposed method in a learning system, ALEF [11] that is currently used at the Slovak University of Technology for support of learning programming and software engineering.

2 Related work

The basic idea of annotation is that it is a kind of information attached to the document. Agosti and Ferro classify annotations according to two basic criteria [3]. They divide annotations into metadata that describe the information in the document and into the information extending document with additional content. The second possible view is based on how the annotations are attached to the document. Annotations may be part of the document such as information in the header file, that describes it or they may be stored in a separate file and linked with document.

There are two basic types of tools adding annotations into documents. Tools that do not focus directly on creating annotations, but on encouraging readers of the documents to create annotations and on tools that are trying to create annotations automatically and try to connect them to the document. Manually created annotations have most frequently the form of highlighted parts of the text, or they are additional textual or multimedia information connected to parts of the document. For example Diigo¹ and Annotea [8] are such tools.

Diigo is a popular service that enables to create various kinds of annotations on the website and allows sharing annotations between users of the service. Annotea is a toolkit that aims to create semantic layer over the document with help of annotations created by readers of the document. Annotations are stored in RDF format separately from the annotated document.

Instruments that focus on automatic creation of annotations must solve most of the problems that tools for manual document annotation deal with. For example, attaching notes to document and storing them. In addition automatic tools have to solve the activities that are otherwise left to the user who creates the annotation. Thus, the automatic tools must ensure the placement of such annotations to the correct location in the document and must find information to fill annotations.

To find the location to which it is appropriate to assign the annotation are frequently used methods from the field of Natural Language Processing (NLP), capable of processing plain text and make its parsing, machine translation, named entity extraction, and so on. A variety of ATR [12] algorithms is used too to obtain the keywords from text.

There are several ways in searching for information to fill annotations. One way is to use pretreated database, to retrieve information to fill annotations. Martin Adam [1] used this method in instrument Pannda. This approach is only used if created annotations are from specific domain. If we wanted to create annotations from different domains it is necessary to create a new data base.

Other way to obtain information to fill annotations is to use freely available services for finding information, like tools as Gnosis², DictionaryTooltip³ and many others make it.

1 Diigo, <http://www.diigo.com/>

2 Gnosis, <http://www.opencalais.com/Gnosis>

3 DictionaryTooltip, <http://www.dictionarystip.com/>

3 Method for document annotation

We proposed a method, which creates annotations for keywords in the text of web page. Input into the annotation process of web page is the body of the page in Slovak language. Creating annotations for this page consists of several steps:

1. Elimination of redundant parts of web page and selection of text to be annotated
2. Search for candidate words for assignment of annotations
3. Search for information to fill the annotations
4. Insertion of the annotations into web page

Before searching the annotations, it is necessary to analyze the document and find the words to which it is appropriate to assign the annotations (steps 1 and 2). As a first step it is necessary to remove redundant parts of the web page as various navigation elements, advertisements, etc. It is necessary to select from the web page body only text that speaks about the main content of the page and only this text will be postponed for further processing.

The second step in the analysis of document is search for candidate words for assignment of annotations. We compared the quality of keywords extracted by service *AlchemyAPI* for three types of texts. Slovak text, text translated by words into English and text translated into English using machine translation. Result for Slovak text and the by words translated text were of poor quality, compared to the third text. We decided therefore to use machine translation to preprocess the text before keyword extraction. Otherwise the quality of extracted keywords and hence the quality of annotations would decrease significantly.

After text translation, the document is prepared for a search for words that are appropriate to assign the annotation. It is necessary to decide what properties should have these words. We could look for example for foreign words, technical words, keywords or words that belong to some list or are significant in other way. The method for extraction of these candidate words depends on the purpose of annotations.

The proposed method for annotation creation uses publicly available services for information search such as Google Search or SlideShare⁴, to obtain information to fill annotations. These services can provide different kinds of information and also of different quality according to the required purpose. The annotations may take form of definitions of words, links to related sites, or it may be a variety of multimedia content like video or an image.

In the step 2 we extracted candidate words from translated text. To connect annotations created in step 3 to the correct words in original text, we have to find mappings between extracted candidate words and their equivalents in original text. In this paper we concentrate on proposal of method for mapping words from translated text back to words in original text.

3.1 Mapping candidate words

Before the actual search of candidate words, the text is translated into English, therefore it is necessary to find a mapping between found keywords which are in English and words in the original text. In later steps when we connect annotations to specific position in the text, we need this mapping to determine to which word annotation belongs to.

Most of methods used for the mapping of words in parallel texts, use various statistical methods [6], but also methods based on the positions of words in the text [4] and methods based on dictionaries are used too.

We propose a method for connecting equivalent words between text and its translation using bilingual dictionary. In the design of the method, we focused on the fact that the Slovak is flexive language, thus the same words can have a lot of different shapes. For mapping of such words we need a dictionary of all possible shapes of words that occur in Slovak language. Such a dictionary

⁴ SlideShare, <http://www.slideshare.net/>

would be huge, would be difficult to work with and even harder to create. We used therefore a much smaller vocabulary, in which is located every word as a single shape and for connection of different shapes of words we used a method similar to the method used by Garabík in his Slovak morphology analyzer in [7]. He used sequences of Levenshtein edit operations [9] to find inflection paradigms that transform word in basic form to its inflected form.

Levenshtein distance between two words is a minimal number of Levenshtein edit operations necessary to convert a string of characters to another. In this conversion are allowed three operations: insertion of character, removal of character and replacement of character by another one. We also adjusted the cost of individual Levenshtein operations depending on the position in word where operation took place. We take into account the fact that if the letter is changed in the root of a word, thus the beginning of the word, the meaning of the word changes significantly. So at the beginning of a word the operation has higher cost than in the rest of the word. We also take into account the fact that the difference in the shapes of words are just differences in the words suffix, so we let the cost of operations linearly decrease in the last characters of the word.

The first step in the mapping words between the text and its translation is the removal of stop-words in both texts. Stop-words, are all words that do not assume significance and hence cannot be selected as words that need an attached annotation, during the search for candidate words to add annotations. Such words are for example conjunctions or articles.

In the process of mapping words we assume that equivalent sentences are in the same order in both original and the translated text. With this assumption we browse through sentences in the text and we associate the word with its equivalent in the equivalent sentence in second text. The algorithm for this mapping is as follows:

```

original_sentences = split_to_sentences(original_text)
translated_sentences = split_to_sentences(translated_text)

for i = 0 to |original_sentences| do
  for each word ow in original_sentences[i] do
    for each word tw in translated_sentences[i] do
      translations = find_translations_for(tw)
      for each translation t in translations do
        if levenshtein_distance(t,ow)<threshold
          declare_mappings(ow,tw)
        end
      end
    end
  end
end
end
end

```

For every sentence in the text we go through all of the words in that sentence. For each word in translated sentence we find its translations using the bilingual dictionary. For all translations of that word we calculate Levenshtein distance between the translation and each word in a sentence from the original text. If we find such a translation of a word that have Levenshtein distance to the word in the original sentence smaller than a threshold, we declare these words as equivalent translations, thus mapped words.

4 Evaluation

We experimented with the aim to verify the success rate of the proposed method for mapping equivalent words between the Slovak text and its translation into English. As a test sample we used part of the textbook with the theme of software engineering. After removing stop-words, the test sample consisted of 1 928 words.

We implemented the proposed method along with two enhancements. The first improvement maps the words that failed to connect using the basic method. In this improvement we assume that in most cases if two words are adjacent in one sentence, they will be adjacent in the translated sentence too. We therefore proposed a method that passes the mapped words in sentence and if both mappings have unmapped neighbouring words, they are linked and declared as mappings.

Second improvement resolves the different shapes of words (even in non flexive languages such as English). We preprocess every entry in the dictionary used in mapping process, so that all English words in this dictionary are stemmed using Porter algorithm [10]. Stemming reduces words to their stem (base form), that is the same for all morphologically related words. Then in process of word mapping, we use preprocessed dictionary to find translations of stemmed words.

Using these advancements we have created four functions:

1. basic function,
2. function improved by taking into account the positions of words in the text,
3. function using a stemmed dictionary and
4. function applying both enhancements.

The results of experiments on these functions are summarized in Table 1. The functions are in the table in the order in which we have enumerated them.

Table 1: Results of experiments on word mapping method and its enhancements

Function	Correct	Incorrect	More	Mapped Words / all words
1	810	51	14	45.38%
2	901	538	195	84.85%
3	993	60	21	63.59%*
4	1040	405	178	96.08%*

* When functions using preprocessed dictionary were tested, some words in the sentences were stemmed into the same stem, so the total number of by sentence unique stems is only 1689 against total number of words 1928

In evaluating the success rate of the function for mapping of equivalent words, we recorded the number of correctly mapped words, incorrectly mapped number words and number of such assignments, where the correct words were mapped, but to the same word were mapped other, incorrect words. The last column of Table 1 shows the ratio of all mapped words (both correctly and incorrectly) to all words in the test sample.

We see that the ratio of correctly mapped and all mapped words in the basic function (function 1) is more than 90 % words. However, the number of all mapped words is only a little more than 45 % of all words in sample. Similar result was also achieved in function using a preprocessed dictionary (function 3), the number of correctly mapped words is still more than 90 % of all mapped words, but portion of mapped words to all words increased slightly. Both functions taking into account the position of unmapped words in a sentence (functions 2 and 4) reached the ratio of mapped words to all words more than 80 % but the number of errors in mapping is disproportionately increased.

Function working with both improvements is able to find the largest number of mappings, but it produces many errors. The best ratio between number of found mappings and number of errors is achieved when basic function with stemmed dictionary is used.

5 Conclusion and future work

We proposed a method for automatic creation of annotations for web pages in the Slovak language. Created annotations are formed by additional information associated with the keyword in the text. This additional information is obtained through publicly available services for

information retrieval. Annotations are in form of word definitions and in form of links to resources related to annotated keyword.

During the processing of the document we used the document translated into English, so we created and verified a method for mapping of words between Slovak text and its translation into English. For this method, we developed two improvements that increase the ratio of mapped words to the total number of words in the text.

In further work we plan to create a module into the education system ALEF developed in our university and used in several educational courses [12] to display the annotations to students. Furthermore, we intend to design a method for personalization of annotations on the basis of explicit evaluation of annotations and their contents by users and on the basis of users interaction with annotations. We currently work on creating annotations for the texts for the course Principles of software engineering in the education system ALEF, which will serve to verify the quality of annotations as well as the usefulness of their personalization.

Acknowledgement: This work was partially supported by the Cultural and Educational Grant Agency of the Slovak Republic, grant No. KEGA 028-025STU-4/2010.

References

- [1] Adam M.: An approach to automated on-line annotation, In: *Proc. of Research Project Workshop Tools for Acquisition*, (2007), pp. 20–25
- [2] Agosti M., Ferro N.: Annotations: Enriching a Digital Library, In: *Research and Advanced Technology for Digital Libraries*, (2003), vol. 2769, pp. 88-100
- [3] Agosti M., Ferro N.: A formal model of annotations of digital content, In: *Transactions on Information Systems (TOIS)*, (2007), vol. 26, no. 1, pp. 3+
- [4] Barbu A.-M.: A Positional Linguistics-Based System for Word Alignment., In: *Lecture Notes in Computer Science*, (2004), vol. 3206, pp. 23-30
- [5] Barla, M., Bieliková, M.: Ordinary Web Pages as a Source for Metadata Acquisition for Open Corpus User Modeling. In *Proc. of IADIS WWW/Internet 2010*. IADIS Press (2010), pp. 227-233.
- [6] Gale W., Church K.: Identifying word correspondences in parallel texts., In: *In Fourth DARPA Workshop on Speech and Natural Language*, (1991), pp. 152-157
- [7] Garabík R.: Slovak morphology analyzer based on Levenshtein edit operations, In: *Proc. of Workshop on Intelligent and Knowledge oriented Technologies*, (2006), pp. 2-5
- [8] Kahan J., Koivunen M.-R.: Annotea: An open RDF infrastructure for shared Web annotations, In: *Proc. of the 10th Int. Conf. on World Wide Web*, ACM Press, (2001), pp. 623-632
- [9] Levenshtein V. I.: Binary Codes Capable of Correcting Deletions, Insertions and Reversals, In: *Soviet Physics Doklady*, (1966) vol. 10, pp. 707.+,
- [10] Porter M. F.: An algorithm for suffix stripping, In: *Program* (1980) vol. 14, no. 3, pp. 130-137
- [11] Šimko M., Barla M. Bieliková M.: ALEF: A Framework for Adaptive Web-based Learning 2.0., In: *Proc. of IFIP Advances in Information and Communication Technology, , Key Competencies in the Knowledge Society*, Springer, Vol. 324 (2010), pp. 367–378
- [12] Zhang, Z., Iria, J., Brewster, C., Ciravegna, F.: A Comparative Evaluation of Term Recognition Algorithms. In: *Proc. of the 6th Int. Conf. on Language Resources and Evaluation*, (2008), pp. 2108-2113.

E Príspevok zaslaný na konferenciu Web Intelligence 2011

Fourth International Workshop on Web Information Retrieval Support Systems

22. – 27. August 2011

Lyon (France)

Zaslané dňa 4.4.2011

Automatic Annotation of Non-English Web Content

Jakub Ševcech and Mária Bieliková

Institute of Informatics and Software Engineering, Slovak University of Technology in Bratislava, Slovakia

E-mail: sevcech08@student.fiit.stuba.sk, bielik@fiit.stuba.sk

Abstract—Nowadays we are facing the daily information overload. It is thus difficult to get exactly information we need. It often happens that while reading, we find a word we do not understand. We would need an explanation or additional information about this word. For this purpose annotations in the Web environment are created and attached to such words. In this paper we propose a method for automatic extending the content available on the Web by adding annotations to selected terms (keywords) in the text. The method is designed to be able to insert annotations into the text written in Slovak with a potential to be language independent. Annotations themselves are obtained through publicly available services providing information retrieval. We adapt created annotations taking into account implicit feedback from users in form of clickthrough data. We evaluate proposed method in the environment of an educational web-based system.

Keywords-Web annotation; keywords; keywords mapping; adaptive annotations

I. INTRODUCTION AND RELATED WORKS

While reading a web page, the visitor often encounters a word or phrase, he does not understand, or would require some additional information about this term. This situation occurs more frequently if the page contains technical or explanatory text, such as for example in digital libraries [3] and various educational sites [6]. Common scenario that follows when a website visitor encounters an unknown expression is the following: the user opens a new tab in his web browser, displays his favorite search engine, and starts to search for a definition or an explanation of this word. This scenario has several drawbacks for the user and for the author of the website as well. Website visitor have to stop work with the document and have to move attention to work with other sources in order to search for additional information. Often he does not return to the original site anymore, so we lose the reader.

One of the solutions to this situation would be an annotation [4] attached to this word. Such annotation can immediately provide explanation of unknown word. Content annotation is an active area of research on the Web. Moreover, with advent of the Web 2.0 many applications facilitating annotations were developed.

There are two basic types of tools for adding annotations into documents. Firstly, tools that do not focus directly on

creating annotations, but on supporting readers of the documents in creating annotations exist. Examples of such tools are Diigo¹ or AnnotatEd [7]. These tools provide a number of supporting instruments to users, through which they can manually add annotations into a web document and then share them. However, with the existing amount of information and documents on the Internet, it is impossible for users of these documents to annotate them all.

The second type, are annotation tools that aim at creating annotations automatically. The process of creating annotations can be divided into two basic parts, positioning of annotation to right place and search for information to fill the annotations. To find a location to which it is appropriate to assign the annotation various ATR algorithms [10] or different approaches from the field of Natural Language Processing (NLP) are used. However, satisfactory results are currently achieved only for English texts. To overcome this problem it is possible to use machine translation to translate this text into English. We suppose that for this task existing translating mechanisms are sufficient, even though they are still not perfect as we need mainly *important* words in the sentence such as nouns and verbs and these are translated sufficiently in most cases. Since we need terms explanation in the original text, we have to find then the mappings between extracted keywords and their equivalents in the original text.

There are several ways in searching for information to fill annotations. One way is to use pretreated database to retrieve information to fill annotations. Adam [2] used this method in its instrument Pannda. This approach can be only used if we create annotations for specific domain. Considering this approach, we need specific database for each particular domain, which is not applicable generally for the Web. Wikipedia Miner toolkit [1] uses similar approach, but it links keywords in text with corresponding articles in Wikipedia. Thanks to Wikipedia this approach is applicable basically for every domain. Other way to obtain information to fill annotations is to use freely available services for information retrieval, as tools Gnosis², DictionaryTooltip³ and many others do it. Using such

¹ Diigo, <http://www.diigo.com/>

² Gnosis, <http://www.opencalais.com/Gnosis>

³ DictionaryTooltip, <http://www.dicionarytip.com/>

services, annotations can be created for every domain and they can provide any type of information that is gathered through these services.

In this paper we propose a method for automatic annotations of documents in non-English language and adaptive presentation of the created annotations. We consider the annotation a definition of the word, or a set of links to other web pages related to that word found by existing services for information retrieval.

II. METHOD FOR WEB CONTENT ANNOTATION OVERVIEW

Our method consists of four steps:

1. Elimination of redundant parts of web page and selection of text to be annotated
2. Search for candidate words for annotations
3. Search for information to fill the annotations
4. Adaptation and visualization of annotations

Before searching the annotations, it is necessary to analyze the document and find the words to which it is appropriate to assign the annotations (steps 1 and 2). As a first step it is necessary to remove redundant parts of the web page as various navigation elements, advertisements, etc. It is necessary to select from the web page body only text that speaks about the main content of the page, and only this text is postponed for further processing. In our current implementation we use for task the Readability⁴ service.

The second step is a search for candidate words for the assignment of annotations. Currently, machine analysis of text achieves satisfactory results only for English texts. We believe that this is sufficient for several languages including Slovak language and therefore we translate analyzed text into the English. Without the use of these results, the quality of found keywords and hence the quality of annotations would decrease significantly. For search for keywords it is possible to use many different algorithms and services [5, 10]. To connect created annotations to the correct words in the original text, we proposed a method for finding mappings between extracted candidate words and their equivalents in original text. This method is key element for annotation acquisition for various languages.

For annotation creation we use publicly available services for information search (see evaluation section for more details). These services provide different type of information and also of different quality according to the required purpose. The annotations may take form of words definitions, links to related sites, or multimedia (like video or an image). We use services providing definitions of keywords and services providing links to web pages related to the keyword.

Finally, created annotations are visualized to the user. Before actual visualization we reorder the list of annotations according his implicit feedback.

We evaluated our method within an educational framework ALEF [6], where created annotations are presented to the students in form of a list of links to related web pages and in form of definitions of keywords occurred in the learning objects presented by the ALEF. These annotations are presented along with other types of annotations present in ALEF. Students can tag or comment learning objects, they can highlight text for better remembering, annotate text in form of links to external sources or annotate text by simple questions.

Our method is designed to be applicable for any kind of content. It heavily depends on quality of used services. Dependency on other services can be seen as a disadvantage. However, it enables wider applicability and concentration on other important issues such as language independence or adaptive presentation of acquired annotations.

III. MAPPING CANDIDATE WORDS

We connect equivalent words in two texts (original and its translation) using bilingual dictionary. We primarily consider Slovak language, which is flexive language with many shapes of words and can represent (considering its syntax) rather large group of languages. Our concern was effectiveness. Effective processing of huge amounts of texts is more important as having exact stemming method. Our hypothesis is that we can work on morphological level on the level of strings with sufficient accuracy considering task of mapping words for the annotation process.

We use a dictionary, in which is located every word as a single shape and for connection of different shapes of words we used a method similar to method used by Garabík in his Slovak morphology analyzer in [8]. We used the comparison of words using Levenshtein distance. Levenshtein distance of two words is a minimal number of Levenshtein edit operations necessary to convert a string of characters to another. During this conversion three operations are allowed: insertion of character, removal of character and replacement of character by another one.

We also adjusted the cost of individual Levenshtein operations depending on the position in word where the operation took place. We take into account the fact that if a letter is changed in the root of a word, the meaning of the word changes significantly. We raise then the cost of the operation at the beginning of the word. We also take into account the fact that the difference in the shapes of words are just differences in the words suffix, so we let the cost of operations linearly decrease in last characters of the word.

The first step of mapping words between a text and its translation is the removal of stop-words in both texts. In the process of mapping words we assume that equivalent sentences appear in the same order in both original and the translated text. With this assumption we browse through each pair of the sentence and its translation. We move through the words in the translated sentence and we seek for translations of each word in a bilingual dictionary, just like

⁴ Readability, <https://www.readability.com/>

if we tried to translate the translated sentence word by word back to Slovak. We then compare each translation from dictionary using Levenshtein distance with every word in the original sentence. If calculated distance of two words is less than established threshold, we declare them to be mapped.

IV. ANNOTATION ADAPTATION

Annotation adaptation is based on an implicit feedback derived from users' behavior. When the user interacts with the content of the annotation, we gather implicit feedback in the form of fact that the user clicked on presented element of the annotation and that he did not click on the other. Elements of the annotation content are presented in a list, while the user is affected by its arrangement. We therefore do not assign the same weights to clicks on the elements placed in various positions in the presented list.

We interpret a click as a statement that the clicked element is better or more relevant than other element offered at the same time. Similar approach for interpreting implicit feedback is used in [9], where five strategies for interpreting these statements are proposed.

We use the following for reordering list of annotations:

1. **Click > Skip Above**, where the element user clicked is better than all the elements listed on higher positions and which user did not click.
2. **Click > No-Click Next**, where the element user clicked is better than the immediately following element which user did not click.

Based on these strategies and users' feedback, we get a set of statements about the quality of provided information. These statements are then used to rearrange the content of the annotations. We consider these statements as oriented edges of graph, where elements of content of annotation are the nodes of the graph. The statements gathered from users' feedback can repeat for the same elements and moreover they can be contradictory. We therefore combine repetitive edges to one, where the number of combined edges is stored in the weight of the result edge.

Subsequently we use adapted PageRank algorithm that takes into account the weights of the edges and we calculate the rating of nodes. Using PageRank we ensure that contradictory statements are taken into account. When we arrange nodes by their decreasing rating, we obtain the new order for elements of content of the annotation.

V. EVALUATION

We experimented with the aim to evaluate the success rate of the proposed method for mapping equivalent words between the Slovak text and its translation into English. As a test sample we used part of the textbook for the course Principles of software engineering. After removing stop-words, the test sample consisted of 1 928 words.

We implemented the proposed method along with two enhancements. The first improvement maps the words that failed to connect using the basic method. In this

improvement we assume that in most cases, if two words are adjacent in one sentence, they will be adjacent in the translated sentence too. We therefore proposed a method that passes the mapped words in sentence and if both mappings have unmapped neighbouring words, they are linked and declared as mappings.

Second improvement resolves the different shapes of words even in non flexive languages such as English. We preprocess every entry in the dictionary used in mapping process, so that all English words in this dictionary are stemmed using Porter algorithm. Stemming reduces words to their stem that is the same for all morphologically related words. In process of words mapping we use preprocessed dictionary to find translations of stemmed words.

Using these advancements we created four functions:

- basic function,
- function taking into account positions of unmapped words,
- function using a stemmed dictionary and
- function applying both enhancements.

The results of experiments on these functions are summarized in Table 1 and Table 2.

Table 1. Success rate of mapping functions for equivalent words.

<i>Function</i>	<i>Correct</i>	<i>Incorrect</i>	<i>More</i>
Basic	92.75%	5.82%	1.60%
1st enhancement	55.14%	32.92%	11.93%
2nd enhancement	92.45%	5.58%	1.95%
Both enhancements	64.07%	24.95%	10.96%

Table 2. Number of mapped words compared to total number of words.

<i>Function</i>	<i>Mapped words number / Total words number</i>
Basic	45.38%
1st enhancement	84.85%
2nd enhancement	63.59%
Both enhancements	96.08%

In evaluating the success rate of the function for mapping of equivalent words (Table 1), we recorded the number of correctly mapped words, incorrectly mapped words and number of assignments, where the correct words were mapped, but to the same word were mapped other, incorrect words. Table 2 shows the ratio of all mapped words to all words in the test sample.

We see that the ratio of correctly mapped and all mapped words in the basic function is more than 90 % of mapped words. However, the number of all mapped words is only a little more than 45 % of all words in sample. Similar result was also achieved in function using a preprocessed dictionary. The number of correctly mapped words is still more than 90 % of all mapped words, but portion of mapped words to all words increased slightly. Both functions taking into account the position of unmapped

words in a sentence (function with first enhancement and with both enhancements) reached the ratio of mapped words to all words more than 80 % but the number of errors in mapping is disproportionately increased.

Function working with both improvements is able to find the largest number of mappings, but it produces many errors. The best ratio between number of found mappings and number of errors is achieved when basic function with stemmed dictionary is used. This ratio can be even better, if better dictionary is used.

Next, we focused on evaluation of quality of information we gathered through publicly available services. We selected 16 texts from the field of software engineering. In these texts, we extracted keywords using the AlchemyAPI service. Then we gathered additional information using Google Search, SlideShare, Dbpedia and DictService. Google Search and SlideShare took as an argument the keyword and returned list of hyperlinks to related resources. DictService returned definitions from different dictionaries for the keyword. When searching for keywords using AlchemyAPI, we were able to search for concepts of the processed document. With these concepts we received a link to corresponding resource in Dbpedia. We used SPARQL to seek for websites related to that resource.

Volunteers were then asked to say whether they will find the gathered information useful, when they read the analysed text and do not understand the keyword, or if they need further information about this keyword. The results we obtained are summarized in Table 3 where the ratio of relevant and irrelevant information gathered by different services for information retrieval is shown.

Table 3. Relevancy of gathered information.

Service	Relevant	Irrelevant
Google Search	70.01%	26.98%
DBpedia	63.29%	31.90%
DictService	59.64%	40.36%
SlideShare	26.32%	72.79%

Small amount of links returned by evaluated services, was corrupted, thus it was not possible to evaluate its relevance (we omit them here). We observed big differences in relevancy of returned information between the compared services. The quality of the annotations thus heavily depends on services used to gather information.

It is necessary to properly choose services to search for the annotation content. None of the evaluated services reached success rate approaching 100%, so there remains place for improvement of their utilization, as well as space to emphasize the relevant information. We approach this problem by reordering information according implicit feedback. Drawback is that we first need users to click and in such way evaluate the annotations. However, this can be overcome rather quickly considering power of collaborating users also in connection of specific settings of educational

systems, where the users obviously have motivation to go through the resources.

VI. CONCLUSIONS

We proposed a method for automatic creation of annotations for web pages in the Slovak language. We proposed the method for mapping equivalent words between the text and its translation. Our method is not constrained by the Slovak language. It is open to other languages with similar structure. It enables effective use of results achieved for keyword extraction in English, which is well elaborated and still evolving. We evaluated our approach and confirmed that this current quality of language translation is sufficient for such task. This brings new possibilities for analysis and enhancement of non-English web sites.

We evaluated the quality of information gathered from publicly available services for information retrieval. We showed that the quality of created annotations heavily depends on the quality of used services. Both experiments were performed independently of the educational system in which the proposed method is implemented.

We proposed a method for adaptation of annotations, based on user implicit feedback in form of clickthrough data. We used adapted PageRank algorithm to find ratings of annotation content elements, where edges of analyzed graph were created using implicit feedback form users.

REFERENCES

- [1] D. Milne and I. H. Witten, "Learning to link with wikipedia," in Proc. of the 17th ACM Conf. on Information and Knowledge Management. New York, NY, USA: ACM, 2008, pp. 509-518.
- [2] M. Adam, "An approach to automated on-line annotation," in *Proc. of research project workshop Tools for Acquisition, Organization and Presenting of Information and Knowledge*, 2007, pp. 20–25.
- [3] M. Agosti and N. Ferro, "Annotations: Enriching a digital library," in *Research and Advanced Technology for Digital Libraries*, ser. LNCS, Springer, 2003, vol. 2769, pp. 88-100.
- [4] M. Agosti and N. Ferro, "A formal model of annotations of digital content," *ACM Trans. Inf. Syst.*, Nov. 2007, vol. 26, no. 1, pp. 3+.
- [5] M. Barla and M. Bieliková, "Ordinary Web Pages as a Source for Metadata Acquisition for Open Corpus User Modeling," in *Proc. of IADIS WWW/Internet 2010*. IADIS Press, 2010, pp. 227-233.
- [6] M. Šimko, M. Barla, and M. Bieliková, "ALEF: A framework for adaptive Web-Based learning 2.0," in *Key Competencies in the Knowledge Society*, ser. IFIP Advances in Inf. and Communication Technology, Springer, 2010, vol. 324, ch. 36, pp. 367-378.
- [7] R. Farzan and P. Brusilovsky, "AnnotatEd: A social navigation and annotation service for web-based educational resources," in *New Rev. Hypermedia Multimedia*, vol. 14, no. 1, pp. 3-32, Jan. 2008.
- [8] R. Garabík, "Slovak morphology analyzer based on Levenshtein edit operations," in *Proc. of Workshop on Intelligent and Knowledge oriented Technologies*, 2006, pp. 2-5.
- [9] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay, "Accurately interpreting clickthrough data as implicit feedback," in *Proc. of the 28th annual int. ACM SIGIR conf.* New York, NY, USA: ACM, 2005, pp. 154-161.
- [10] Z. Zhang, J. Iria, C. Brewster, and F. Ciravegna, "A comparative evaluation of term recognition algorithms," in *The 6th Int. Conf. on Language Resources and Evaluation*, 2008, pp. 2108-2113.

F Diplom získaný v medzinárodnej súťaži eLearning v praxi v rámci konferencie ICETA 2010

Autorský kolektív:

Mária Bieliková, Marián Šimko, Michal Barla, Pavel Michlík,
Martin Labaj, Vladimír Mihál, Maroš Unčík, Jakub Ševcech



CENA MINISTRA ŠKOLSTVA, VEDY, VÝSKUMU A ŠPORTU SR

v medzinárodnej súťaži

eLearning v praxi

v rámci konferencie ICETA 2010

pre projekt s mimoriadnym prínosom pre vzdelávanie

Stará Lesná 29. október 2010

Eugen Jurzyca

minister školstva, vedy, výskumu a športu Slovenskej republiky

ICETA 2010
International Conference

Súťaž „E-learning v praxi“

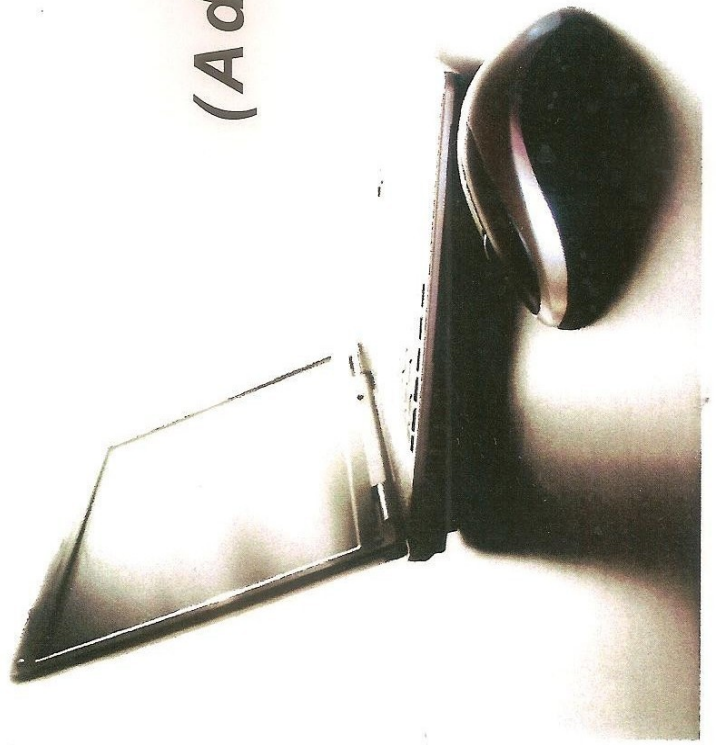
Súťaž je organizovaná ako súčasť 8. ročníka medzinárodnej konferencie ICETA 2010
27. október 2010, Stará Lesná, Slovensko

DIPLOM

1. miesto v kategórii
Podporný materiál pre on-line vzdelávanie

ALEF

(Adaptive Learning Framework)



Ildikó Pšenáková
Predsedkyňa poroty

František Jakab
Predseda
organizačného výboru

G Obsah elektronického média

- Bakalárska práca – elektronická verzia dokumentu
- Implementácia
 - Webová služba – zdrojový kód webovej služby na vytváranie poznámok
 - ALEF – zdrojový kód rámca ALEF
- Testovacie údaje
 - Mapovanie slov – údaje použité na testovanie metódy na mapovanie slov
 - Overenie kvality informácií na naplnenie poznámok – súbory s informáciami ohodnotenými expertmi
- Obraz databázy rámca ALEF – obraz databázy použitej pri vyhodnocovaní experimentov
- Konferencia IIT.SRC – článok a poster na konferenciu IIT.SRC 2011
- Konferencia WI – článok zaslaný na konferenciu Web Intelligence 2011
- Výstup služby na vytvorenie poznámok – ukážka XML súboru s vytvorenými poznámkami získaný pomocou webovej služby