

Posudek bakalářské práce

Matematicko-fyzikální fakulta Univerzity Karlovy

Autor práce Daniela Jurášová
Název práce Automatické generovanie hrebeňoviek
Rok odevzdání 2021
Studijní program Informatika
Studijní obor Obecná informatika

Autor posudku Mgr. Michal Novák, Ph.D. Oponent
Pracoviště Ústav formální a aplikované lingvistiky

K celé práci

lepší OK horší nevyhovuje

	lepší	OK	horší	nevyhovuje
Obtížnost zadání		X		
Splnění zadání		X		
Rozsah práce <i>... textová i implementační část, zohlednění náročnosti</i>			X	
<p>Práce formálně splňuje zadání, které je však formulováno velice obecně. Generování hřebenovek bylo možné řešit různými způsoby, přičemž autorka si vybrala jeden z těch jednodušších. Tím je procházení databáze WordNet, což je lexikálně-sémantická ontologie, ze které jsou na základě hypero-/hyponymických vztahů vybrána tematicky podobná slova do křížovky. Jako legenda jsou použity glosy, jež jsou rovněž obsaženy u většiny položek databáze WordNet. Očekával bych, že řešitelka kromě této metody alespoň prozkoumá i nějakou složitější. Např. pomocí slovních vektorů, které jsou mimochodem v zadání zmíněny odkazem na doporučenou literaturu (Mikolov, 2013).</p> <p>Zesložitění mohlo být docíleno i rozšířením použitých jazykových zdrojů, např. zahrnutím slovníku Wiktionary. Je pravda, že řešitelka kromě anglické verze WordNetu zahrnula i slovenskou, kde se rozhodla vyřešit mapování jednotlivých hesel, které z důvodu nekompatibility verzí nešlo spojit jednoduše přes identifikátor záznamů. Otázkou však je, jestli je toto slučování nutné. Kdyby se tyto ontologie používaly odděleně, ztratili bychom tím informaci o některých vztazích mezi slovy? Tato nebo podobná analýza mi v práci chybí.</p>				

Textová část práce

lepší OK horší nevyhovuje

	lepší	OK	horší	nevyhovuje
Formální úprava <i>... jazyková úroveň, typografická úroveň, citace</i>		X		
Struktura textu <i>... kontext, cíle, analýza, návrh, vyhodnocení, úroveň detailu</i>		X		
Analýza			X	
Vývojová dokumentace	X			
Uživatelská dokumentace	X			

Textová část je z mého pohledu silnějším místem práce. Text je dobře strukturován, jazyková a typografická úroveň odpovídají tomu, co se očekává od bakalářského studenta. Detailně a přitom přehledně je zpracována zejména vývojová dokumentace, slovně popisující každou důležitou metodu. Uživatelská dokumentace je rovněž přehledná, což je ale částečně důsledkem velké jednoduchosti samotné aplikace. Oceňuji kapitolu „8 Odozva uživatelů“, která hodnotí přínosnost aplikace na základě dotazníkového průzkumu. To není u infromatických prací úplně běžné.

Na druhou stranu mi v práci chybějí analytičtější pohledy. Jediná analýza se zabývá tím, kterou položku použít na mapování anglické a slovenské verze WordNetu místo nekompatibilních identifikátorů. Odpověď je však víceméně zřejmá už z charakteru dat. Naopak otázka, jestli je vůbec nutno tyto databáze slučovat, se v práci nijak neřeší. Některá implementační rozhodnutí by si rovněž zasloužila analýzu, protože z mého pohledu nejsou optimální, a opomenutí bližšího vysvětlení spíše naznačuje, že řešitelka si tyto aspekty neuvědomuje (viz níže).

Implementační část práce

lepší OK horší nevyhovuje

Kvalita návrhu ... architektura, struktury a algoritmy, použité technologie				X
Kvalita zpracování ... jmenné konvence, formátování, komentáře, testování	X			
Stabilita implementace		X		

Naopak návrh a implementační rozhodnutí jsou nejslabším místem práce. Práce sestává ze dvou oddělených aplikací: 1) utilita pro předzpracování WordNetu, 2) GUI aplikace pro samotnou hru. Vnitřní architektura aplikací je však už poměrně strohá a odpovídá spíše procedurálnímu návrhu než objektovému návrhu, který by byl zejména pro GUI aplikaci určitě vhodnější. I když procedurálně jsou logika, zobrazování a interakce odděleny dostatečně, všechen kód je obsažen v rámci jediné třídy, což zejména u GUI aplikace hodnotím negativně. Kromě jedné všeobjímající třídy se v každé z aplikací používá třída Record, reprezentující jednu položku v databázi WordNet. Ta je však v rozporu s pravidly o znovupoužitelnosti kódu definovaná dvakrát, pro každou aplikaci samostatně (a trochu jinak).

I když pro danou úlohu není nutno používat komplexní datové struktury a algoritmy, i ty jednoduché jsou někdy navrženy neefektivně. Např. už zmíněné mapování dvou WordNetů o velikosti M a N je implementováno v časové složitosti $O(M*N)$, i když s pomocí datové struktury Dictionary by šlo složitost snížit na $O(M+N)$. Podobně na první pohled neefektivně je vytvářen seznam tematických slov. Pro každé dosud nalezené tematické slovo se hledají jeho hypo-/hyperonymá tak, že se znovu prochází celý WordNet reprezentovaný jako seznam položek. Přitom by bylo možné položky propojit už ve fázi inicializace na dva průchody seznamem a samotné hledání by tak probíhalo v konstantním čase. Vysloveně do očí bijící je reprezentace křížovky jako seznamu stringů, kde každý string je konkatenací anglického a slovenského slova a glosy s dvojtečkou jako oddělovačem. V drtivé většině případů je potřeba přistupovat k jednotlivým částem stringu, ne ke stringu jako celku. Pokaždé je tak nutno rozdělit string na položky pomocí metody Split. Proč řešitelka nepoužila datový typ Struct nebo Tuple? Neefektivnost řešení neměla na mém počítači zásadní vliv na rychlost aplikace, ale při použití větších databází nebo méně výkonného hardwaru by se určitě projevilo. Je možné, že tato obecně méně efektivní řešení jsou v tomto konkrétním případě výhodnější. V práci však postrádám na toto téma jakoukoli diskuzi, což naznačuje, že řešitelka si tento problém vůbec neuvědomila.

Dalším nedostatkem je chybějící parametrizace. Cesty k datovým souborům jsou zadány přímo v kódu, takže např. při použití případné novější verze WordNetu je nutno extrahovaný datový soubor pojmenovat stejně nebo celou aplikaci znovu zkompileovat. Rovněž není možno měnit hodnotu hloubky prohledávání databáze, která je zadrátována přímo v kódu na 5.

Pozitivně hodnotím časté komentování kódu a přehledné pojmenování proměnných a metod.

Co se týče použitelnosti, GUI aplikace je jednoduchá a přehledná, funguje bez zjevných chyb. Z mého pohledu je i graficky hezká, i když většina uživatelů ji v dotazníkovém průzkumu hodnotila průměrně. Nedostatkem v UI je zadávání slov do křížovky, kdy je nutné každé písmenko potvrdit klávesou ENTER a pokud dané políčko obsahuje víc než jeden znak, uživatel je varován vyskakovacím oknem. Uživatelsky přijatelnějším řešením by bylo každý další znak rovnou vepsat do dalšího nižšího políčka křížovky, což v kapitole 8.3 připouští i samotná autorka. Dalším drobným nedostatkem je, že pohybem po křížovce pomocí klávesových šipek je možné se dostat i do neviditelných políček komponenty DataGridView.

Celkové hodnocení	Dobře
Práci navrhuji na zvláštní ocenění	Ne

Datum: 21. června 2021

Podpis