Programming Assignment 3

Implementing a data index and more

Jaka Kokošar, Danijel Maraž, and Toni Kocjan

Fakulteta za Računalništvo in Informatiko UL dm9929@student.uni-lj.si, jk0902@student.uni-lj.si, tk3152@student.uni-lj.si

Povzetek The article covers the work done in the scope of the third programming assignment as part of the subject web information extraction and retrieval.

Keywords: Data Processing Indexing Retrieval

1 Introduction

After having collected web pages in the first assignment and thoroughly stripped them down to the data we are interested in in the second we were ready to continue in the final step which is constructing a data index and implementing the function of querying.

2 Data Processing

Glavna funkcija *preprocess* prejme kot argument rezultat funkcije *text*, ki sama prejme našo surovo html vsebino in tej odstrani nepotrebne tehnične html oznake. Preprocess nato:

- S funkcijo *remove_punctuation* odstrani ločila in več
- Z *nltk.tokenize.word_tokenize* pretvori v vrsto besednih značk
- Odstrani se značke, ki niso alfabetične
- Vse velike začetnice se pretvorijo v male
- Odstrani se značke, ki so *stopword*
- S pomočjo pretvorbe v podatkovno strukturo množice se odstranijo duplikati

Nato ta vrne seznam ostalih značk.

3 Indexing

Funkcija *initiating_indexing* se požene in začne meriti čas gradnje indeksa. Na koncu izpiše na standardni izhod celoten porabljen čas. Glavno nalogo indeksiranja opravlja razred *BetterThanGoogle* nad katerim se pokliče funkcijo *create_index* in se mu kot argument poda relativno pot do datotek za sestavo indeksa.

3.1 BetterThanGoogle

3.2 Create_index

Funkcija kot argumenta prejme instanci razredov *Preprocessor* (ta služi za procesiranje besedila po opisu iz poglavja Data Processing) in *DBHandler* (ta služi za interakcijo z bazo). Nato pridobimo ime datoteke ter vsebino s pomočjo naše abstrakcije korpusov (*file_name* in *document*). Za tem iteriramo skozi vsak par ter spremenljivko *document* ustrezno obdelamo z razredom *preprocessor* (glej *__call__* od preprocessor). Za vsako značko, ki jo vrne preprocessor:

- Najdemo vse njene pojavitve v besedilu (*find_occurrences*)
- Pod pogojem, da smo našli vsaj eno pojavitev se izvede faza vnosa v indeks
- V indeks vnesemo ime značke, ime datoteke, število pojavitev značke, ter niz posameznih pojavitev ločen z vejico

Vredno je tudi omeniti, da program v log ves čas izpisuje koliko datotek je obdelal do sedaj in koliko mu jih še manjka.

4 Data Retrieval

Funkcija *initiating_search* prejme niz za katerega želimo iskati pojavitve v trenutnem indeksu. Ta ustvari nov objekt *SearchEngine* in mu poda instanco *DBHandler*. Potem se za dejansko iskanje na ustvarjenem objektu kliče funkcijo *perform_query*. Nato program izpiše na zaslon rezultate in čas porabljen za poizvedbo.

4.1 SearchEngine

Razred *SearchEngine* kot argument prejme *db_handler* (preko katerega se izvajajo vse interakcije z bazo).

Perform_query Funkcija kot argument prejme niz besed ločenih s presledki. Nato iz teh ustvari seznam besed, ter vsaki besedi velike črke zamenja z malimi. Za vsako se nato naredi poizvedba v bazi iz tabele *Posting* in s funkcijo *_find_occurrences_in_file* in več zankami ustvari terko *QueryResults(beseda, snippeti pojavitev besede)*. Na koncu funkcija vrne drugo terko s številom pojavitev na prvem mestu in seznamom terk *QueryResults* na drugem.