Wyszukiwanie informacji Lato 2011

Dokumentacja wyszukiwarki boolowskiej

Tomasz Jurdzinski *Aleksander Balicki, Tomasz Maciejewski*

1 Instalacja

Pliki projektu wgrywamy do jednego folderu.
Do podfolderu `data/` wgrywamy pliki źródłowe morfologika i wikipedii.

2 Użytkowanie

¡DOROBIC FAJNE UI¿

3 Opis użytych algorytmów i struktur danych

3.1 Tworzenie indeksu

Proces tworzenia wykorzystuje ideę MapReduce.

3.1.1 Faza map

Na początku przechodzimy przez plik wikipedii po linii i wyrażeniem regularnym wyznaczamy słowa. Dla każdej znormalizowanej formy słowa. tworzymy parę (*sowo*, $nr_dokumentu$, *pozycja*) i dodajemy ją do pliku tymczasowego `WORDS` jako jedną linię.

3.1.2 Faza reduce

Po przejściu przez cały plik wikipedii sortujemy plik `WORDS`, stabilnie, po pierwszym słowie, tym sposobem mamy zachowaną kolejność wystąpień dokumentów i pozycji w ramach artykułu. Przechodzimy teraz przez posortowany plik `WORDS.sorted`, i dla każdego trójliterowego prefiksu (lub krótszego, jeżeli całe słowo jest krótsze niż 3 litery) tworzymy tablice hashującą z listą postingową (odpowiednio skompresowaną lub nie). Zapisujemy tą tablice do pliku z użyciem biblioteki do serializacji. W ten sam sposób najpierw serializujemy morfologika, aby potem móc szybko normalizować słowa.

3.2 Wyszukiwanie

Sposób wyszukiwania interaktywnego, to szczególny przypadek wyszukiwania w formie wsadowej. Wyszukiwarka w formie wsadowej, po wczytaniu wszystkich zapytań, gromadzi z nich słowa, gru-

pując po prefiksie (hash z prefixami jako klucze i pythonowymi setami słów).

3.3 Normalizacja

Dla słowa w, odczytujemy plik zależny od trójliterowego prefiksu w i sprawdzamy, czy słowo jest w tym słowniku, jeśli tak zwracamy wszystkie jego formy bazowe, odpowiednio po operacji stemmingu lub nie.

3.4 Spamiętywanie

3.5 Struktury danych

1. dict() - pythonowa wbudowana tablica hashująca
2. set() - pythonowa wbudowana implementacja zbioru, też bazowana na tablicy hashującej
3. OrderedDict() - pythonowa implementacja tablicy hashującej pamiętająca kolejność dodanych kluczy

4 Opis użytych bibliotek

4.1 marshal

4.2 gzip

4.3 pstats

4.4 unittest

5 Opis testów