Pracownia 2 z wyszukiwania informacji

Prowadzący: dr hab. Tomasz Jurdziński

Michał Rychlik, Paweł Murias

Wrocław, dnia 18 czerwca 2011 r.

1 Instalacja

Do porawnego działania programu potrzebny jest intepreter języka python (program powstawał przy użyciu pythona w wersji 2.7, ale powinien być również zgodny z pythonem w wersjach od 2.4 włącznie). Jedyna zewnętrzna biblioteka, jaką należy zainstować to progressbar (do pobrania $http://pypi.python.org/pypi/progressbar/2.2$).

2 Instrukcja użytkownika

Aplikacja złożona jest z dwóch części:

- Pomocniczej
- Zasadnicznej

Część pomocnicza, którą reprezentuje program helpers.py filtruje pliki podane jej jako parametry uzyskania, pozostawiając jedynie informacje dotyczące interesujących artykułów. Tworzy ona również vector page rank dla wszystkich dokumentów i zapisuje go na dysku. Próba wywołania tej części ze zbyt małą liczbą parametrów zakończy się niepowodzeniem, zostanie również wyświetlony komunikat o tym jakie i w jakiej kolejności pliki powinno się podawać jako parametry.

Część zasadnicza jest reprezentowana przez program main.py, który jako swoje parametry pobiera plik z zapytaniami oraz plik z artykułami. Ponownie, podanie zbyt małej liczby argumentów skutkuje wyświetleniem odpowiedniego komunikatu. Skrypt main.py wypisuje wyniki swojej pracy na standardowe wyjście.

3 Opis użytych algorytmów

Dokumenty reprezentowane są w modelu przestrzeni wektorowej z wagami tf-idf. W czasie tworzenia indeksu przeglądane są kolejno wszystkie słowa z danego artykułu, a następnie indeksowane są ich wszystkie możliwe formy bazowe znalezione w morfologiku. "Przy okazji" obliczane są wartości tf oraz na koniec przetwarzania artykułu, również df. Następnie wektory z wartościami tf-idf są dodawane do specjalnego słownika, w którym kluczami są termy

występujące w dokumentach, tak aby łatwo było uzyskać wszystkie wektory, w których dane słowo występuje jako term.

Zastosowaną miarą podobieństwa wektorów jest miara kosinusowa. W czasie obsługi zapytania, zapytanie zamieniane jest na wektor z wagami tf-idf dla poszczególnych jego termów. Następnie wektor ten jest porównywany ze wszystkimi dokumentami zawierającymi jaką kolwiek formę bazową wszystkich termów w nim występujących. Tak uzyskane vectory (odpowiedzi) są sortowane malejąco po wartości podobieństwa (zmodyfikowanej o page rank danego dokumentu).

Do obliczania page ranku dokumentów zawartych w zbiorze wykorzystano metodę iteracyjną. Algorytm ten jest uruchamiany dokładnie raz (razem z filtrowaniem interesujących dokumentów). Metoda iteracyjna została tak zmodyfikowana, aby każdy jej krok działał w czasie proporcjonalnym do ilości niezerowych elementów w macierzy procesu Markowa oraz ilości elementów wektora wynikowego). Obliczenia kończą się kiedy dwa kolejne wektory które uzyskujemy otrzymują note 0.99 w skali od 0 do 1 realizowanej przez miarę kosinusową. Tak obliczona wartość page rank dla artykułu jest mnożona razy 10 i dodawana do podobieństwa tego artykułu i zapytania.

4 Implementacja

Aplikacja została w całości napisana w języku python.

5 Wyniki testów

Wyniki testów dla pliku *pytania.txt* znajdują się w pliku pytania-results.txt

Szybkość zbieżności metody iteracyjnej obliczania page rank została potwierdzona.

Proces obliczania kolejnych wektorów pośrednich zakończył się (po spełnieniu kryterium opisanego w punkcie Opis użytych algorytmów) po 4 krokach.