Specyfikacja Funkcjonalna

Marcin Matłacz

24 marca 2015

Informacje Ogólne
Program zostanie napisany w języku C z użyciem standardowych bibliotek. Uruchomienie programnu będzie się odbywać za pomocą linii komend. Jedyna interackcja użytkownika z programem to uruchomienie z odpowiednimi parametrami. Forma wyświetlanych komunikatów to tekst.

Moduły programu i zależności między nimi

Moduł główny (main):
Główny moduł programu. Odpowiedzialny z zinterpretowanie poleceń użytkownika i przekazanie ich do odpowiednich modułów.
Otrzymuje:

- nazwy plików bazowych(max 2) i/lub nazwy plików pośrednich(max 2); (na raz można podać 2 pliki bazowe lub 2 pliki pośrednie lub 1 plik pośredni i 1 plik bazowy)
- nazwa generowanego pliku tekstowego;
- liczbe słów i/lub liczbę akapitów;

W przypadku podania dwóch plików, program połączy je w jeden plik pośredni, na podstawie którego zostaną wygenerowany zostanie tekst.

Moduł główny będzie się komunikował z modułami:

- analizator;
- generator tekstu;
- moduł odczytujący pliki pośrednie;

Struktury:
Moduł wykorzystuje struktury ngrams

Moduł analizy tekstów bazowych, analizator (analyzer):
Moduł odpowiada za przeanalizowanie tekstu bazowego i zamienienie go na ngramy. Wydobywa informacje o ilości wystąpień ngramów i słów.
Podział informacji o ngramach:

- spis prefixów;
- do każdego prefixu dołączony jest spis sufixów następujących po nim;

Struktury:

- struct ngrams{
  struct prefix **prefixes;
  int number_of_prefixes;
  int number_of_words;
  };

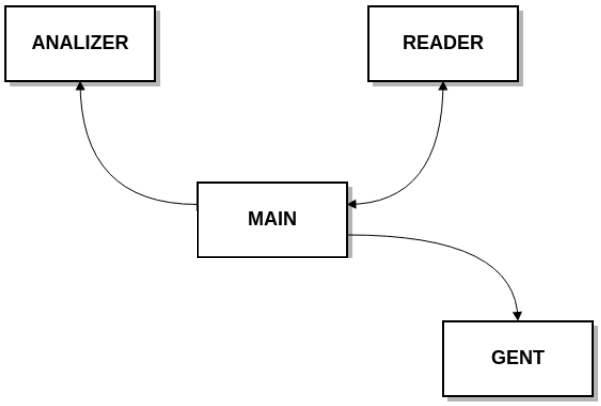
Funkcje:

- void add_ngram(struct ngrams *data, char **text, int i, int n);
  dodaje ngram do struktury data. text - tekst bazowy, i - pozycja w tablicy text pierwszego słowa ngramu, n - długość ngramu.
- void add_sufix(struct prefix *pointer, char *text)
  dodaje sufix text do istniejącego prefixu pointer.
- struct ngrams *analize(char *basefilename, int n);
  funkcja otrzymuje nazwę pliku bazowego basefilename i stopień ngramu n.
  zwraca wskaźnik do struktury zawierającej ngramy z tekstu bazowego basefilname.

Generator tekstu (gent):
Moduł zajmuje się generacją tekstu o zadanej liczbie słów i/lub akapitów.
Jeżeli obie wartości są zerowe, użyta zostanie domyślna liczba słów. Jeżeli jeden z parametrów jest zerowy to zosatnie zignorowany i moduł wygeneruje tekst na podtawie drugiego z nich. Moduł otrzymuje z main wskźnik na strukturę z danymi do generacji tekstu. Wygenerowany plik tekstowy zostanie zapisany z nazwą output_name.
Struktury:
Te same co w analizatorze;
Funkcje:
int gent(struct ngrams *intermediate_file, int words, int paragraphs,
char *output_name);
Zwraca 0 gdy powiedzie się generacja tekstu a 1 gdy zakończy się z błędem.
Generator statystyk (gens):
Moduł otrzymuje z main informacje potrzebne do generacji statystyk.
Struktury:
Te co w analizatorze.
Funkcje:
int gens(struct ngrams *plik_posredni, char *output_name);
Zwraca 0 w przypadku powodzenia a 1 w przypadku zakończenia działania z błędem.
Zapisuje statystyki w postci pliku tekstowego o nazwie output_name.
Moduł łączący pliki pośrednie (merger):
Moduł zajmuje się odczytem ngramów występujących w plikach pośrednich, porównaniem ngramów, dopisaniem nowych ngramów i sufiksów z pliku drugiego do pierwszego. Zapisuje plik pośredni powstały z połączenia w postaci pliku tekstowego.
Struktury:
Te same co w analizatorze.
Funkcje:
int(struct ngrams *plik_posr1, struct ngrams *plik_posr2);
przepisuje ngramy nie wystepujące w plik_posr1 z plik_posr2.
dopisuje sufixy z plik_posr2 do występujących już prefixów z plik_posr1.
Moduł odzczytujący pliki pośrednie (reader):
Zajumje się przekształceniem pliku pośredniego w formie pliku tekstowego do postaci struktury która jest możliwa do przetworzenia przez inne moduły.
Struktury:
Te same co w analizatorze.
Funkcje:
struct ngrams *reader(char **intermediate_file);
funkcja odczytuje ngarm:

- odczytuje prefix i dodaje do struktury;
- odczytuje sufixy danego prefixu i dodaje do struktury;

Diagram modułów:



Strzałki oznaczają w którym kierunku odbywa się przesyłanie danych.
(z pominięciem komunikatów 0, 1)