Analyzátor klíčových slov - Popis řešení

Bc. Martin Ševčík, Masarykova univerzita 28/02/2019

Definice úlohy

Cílem zadání je vytvořit skript v jazyce Perl, který z **textového souboru** předaného na standardní vstup **vyextrahuje klíčová slova** a tato **předá na standardní výstup**, vždy **jedno klíčové slovo** na **jednom řádku**. Cílem je vyextrahovat z dokumentu klíčová slova vhodná **pro hledání podobných dokumentů**. Klíčová slova je třeba na výstupu **seřadit dle důležitosti**. Řešení lze omezit pouze na **české** texty a na jednoslovní klíčová slova. Při řešení lze použít **perlové moduly z CPANu**. Vše ostatní musí být součástí řešení. Váš skript bude kontrolován na **linuxovém počítači.**

Analýza problému a rozbor zadání

- Vstup ve formě textového dokumentu text.txt – vstupní text bude česky, je nutné dodržet kódování utf-8, aby se text správně načetl.
- Extrakce klíčových slov určených pro další vyhledávání podobných textů – definujeme klíčové slovo jako slovo, které je významově v souvislosti s textem konkrétní (není významově obecné) a alespoň částečně vystihuje konkrétní myšlenku textu jako celku.
- Klíčová slova předaná na výstup budou v češtině, každé na novém řádku a budou seřazena podle důležitosti – musí být dodrženo kódování utf-8, je třeba vymyslet, jak vypsat každé slovo na nový řádek, ale ještě před výpisem seřadit vybraná slova dle důležitosti, kterou je nutné definovat. Výpis slov je také potřeba uložit do nově vytvořeného textového souboru klicovaslova.txt, který by mohl pro případné hledání podobných dokumentů sloužit jako zdroj podobnosti.
- Celý skript musí být kompatibilní s os linux a lze pro jeho vytvoření využít vhodné CPAN moduly, které pravděpodobně mohou uvedení celého skriptu do provozu výrazně usnadnit.

Návrh řešení

- V celém skriptu je nutné povolit kódování UTF-8, které je třeba aplikovat i při načítání textu ze zdrojového souboru pomocí parametru ":encoding(utf8)".
- Je nutné sestavit samostatnou databázi nebo seznam stop slov, která jsou příliš obecná a budou z textu před určením klíčových slov pomocí regulárních výrazů vyfiltrována.

- V tomto konkrétním případě bude nejschůdnějším řešením určit důležitost klíčového slova podle toho, kolikrát se ve vyfiltrovaném textu vyskytuje. K řešení tohoto problému se nabízí použití metod sort(), zápis totožných slov (určených pomocí regulárních výrazů) v textu do array a následné spočítání a porovnání indexů. Vzhledem k složitějšímu procesu bude zpočátku nejvhodnější zkontrolovat, zda neexistuje CPAN modul, který by tento proces řešil.
- Výsledná klíčová slova seřazená od nejdůležitějšího bude nejlepší zapsat do array a pak je po jednom vypsat pomocí cyklu foreach do terminálu a do textového souboru.

Algoritmizace

Nejdříve je nutné ověřit, zda je v systému nainstalován CPAN modul Lingua::EN::Keywords, bez kterého nebude skript fungovat. Program tento problém řeší tak, že zavolá externí instalační shell skript nacházející se ve stejné složce:

Listing 1: Příkaz pro spuštění instalačního skriptu – Analyzátor klíčových slov.

```
1 system("/bin/bash ./instalator_modulu.sh") == 0
2   or die "Skript nelze spustit";
```

Tento skript otevře CPAN shell a ověří, případně se pokusí nainstalovat chybějící moduly, a to ideálně bez nutnosti přístupu root uživatele pomocí modulu local::lib:

Listing 2: Instalační skript instalator_modulu.sh – Analyzátor klíčových slov.

```
1  #!/usr/bin/sh
2  eval `perl -I ~/perl5/lib/perl5 -Mlocal::lib`
3  echo 'eval `perl -I ~/perl5/lib/perl5 -Mlocal::lib`' >> ~/.profile
4  echo 'export MANPATH=$HOME/perl5/man:$MANPATH' >> ~/.profile
5  perl -MCPAN -e shell <<END_OF_CPAN_COMMANDS
6  install local::lib
7  END_OF_CPAN_COMMANDS
8  echo "################################"
9  perl -MCPAN -Mlocal::lib -e 'CPAN::install(LWP)'
10 echo "################################"
11 perl -MCPAN -e shell<<END_OF_CPAN_COMMANDS
12 install Lingua::EN::Keywords
13 END_OF_CPAN_COMMANDS
14 echo "################################"
```

Po instalaci potřebných modulů se instalační skript instalator_modulu.sh ukončí. Skript Klicova_slova.pl pokračuje dále, a načte vstupní textový soubor v kódování utf-8.

Listing 3: Perl skript načítá vstup – Analyzátor klíčových slov.

```
use Lingua::EN::Keywords;
use utf8;
my $file = "text.txt";
my $document = do {
  local $/ = undef;
  open my $fh, "<:encoding(UTF-8)", $file
  or die "could not open $file: $!";
  <$fh>;
};
```

Následuje vyfiltrování importovaného textu od obecných slov pomocí regulárního výrazu. Jako zdroj českých stop slov posloužil seznam na webu https://countwordsfree.com/.

Listing 4: Odfiltrování stop slov a znaků – Analyzátor klíčových slov.

```
...
$document =~ s/[.]+//gui;
$document =~ s/[?]+//gui;
$document =~ s/[:]+//gui;
$document =~ s/[^A-Za-z]\ba\b//gui;
$document =~ s/[^A-Za-z]\baby\b//gui;
$document =~ s/[^A-Za-z]\bahoj\b//gui;
$document =~ s/[^A-Za-z]\baj\b//gui;
...
```

Po vyfiltrování textu definuje skript array @keywords, který je pomocí metody keywords() modulu Lingua::EN::Keywords naplněn pěti nejdůležitějšími klíčovými slovy dle frekvence výskytu.

Listing 5: Volání metody keywords() modulu Lingua::EN::Keywords – Analyzátor klíčových slov.

```
my @keywords = keywords($document);
```

Listing 6: Modul Lingua::EN::Tagger extrahuje 5 nejdůležitějších klíčových slov – Zdroj: https://metacpan.org/source/SIMON/Lingua-EN-Keywords-2.0/Keywords.pm.

```
package My::Tagger;
@My::Tagger::ISA=qw(Lingua::EN::Tagger);
my %known_stems;
sub stem {
    my ( $self, $word ) = @_;
    return $word unless $self->{'stem'};
    return $known_stems{ $word } if exists $known_stems{$word};
    my $stemref = Lingua::Stem::En::stem( -words => [ $word ] );

```

```
10      $known_stems{ $word } = $stemref->[0] if exists $stemref->[0];
11  }
12
13  sub stems { reverse %known_stems; }
14
15  package Lingua::EN::Keywords;
16  use Lingua::EN::Tagger;
17  require 5.005_62;
18  use strict;
19  use warnings;
20
21  my $t = My::Tagger->new(longest_noun_phrase => 5,weight_noun_phrases=>0);
22
23  require Exporter;
24  our @ISA = qw(Exporter);
25  our @EXPORT = qw( keywords);
26  our $VERSION = '2.0';
27  sub keywords {
28      my %wl = $t->get_words(shift);
29      my %newwl;
30      $newwl{unstem($_)} += $wl{$_} for keys %wl;
31      return (sort { $newwl{$b} <=> $newwl{$a} } keys %newwl)[0..5];
32  }
33  sub unstem {
34      my %cache = $t->stems;
35      my $word = shift;
36      return $cache{$word} || $word;
37  }
```

V závěru skript spustí dva foreach cykly, které při každé iteraci vypíší na nový řádek jedno z pěti klíčových slov uložených v @keywords seřazených dle důležitosti, a to nejdříve do nového souboru klicovaslova.txt, a poté i do konzole.

Listing 7: Výstup skriptu – Analyzátor klíčových slov.

```
1  foreach (@keywords) {
2    binmode(STDOUT, ":utf8");
3    print "$_\n";
4  };
5
6  my $filename = 'klicovaslova.txt';
7  open(my $fh, '>:encoding(UTF-8)', $filename) or die "Could not open file '↩
       $filename' $!";
8
9  foreach (@keywords) {
10   binmode(STDOUT, ":utf8");
```

```
11    print $fh "$_\n";
12  };
13
14  close $fh;
```

Poté se skript ukončí.

Nedokonalosti a podněty ke zlepšení pro případnou další verzi

1. Úsporněji implementovat seznam stop slov.
2. Instalační skript pro CPAN moduly je na některých systémech nefunkční a je třeba moduly instalovat manuálně.
3. Nahradit činnost modulu vlastním algoritmem, který by mohl určovat důležitost klíčových slov na základě kritérií zadaných uživatelem.
4. Odhalit a odstranit příčinu upozornění "Wide character in print..."