Podobnost kontextů ve velkých textových korpusech

Informatické kolokvium

Karel Pala Pavel Rychlý

Centrum zpracování přirozeného jazyka

13. března, 2007

Obsah

Co je to NLP?

- NLP je zkratka pro “Natural Language Processing”,
- česky “(počítačové) zpracování přirozeného jazyka”.
- Který jazyk je přirozený?
  - ***přirozený jazyk*** je jazyk, který vznikl přirozeným vývojem a lidé ho používají k běžné komunikaci (čeština, angličtina, . . .)
  - *umělý/formální jazyk* je uměle navržený, explicitně definovaný jazyk, např. programovací jazyky, matematické formalismy, . . .

Co znamená ***zpracovávat*** počítačově přirozený jazyk?

- snažit se popsat formálně (formalizovaně) přirozený jazyk tak, aby s ním bylo možné (alespoň do určité míry) strojově manipulovat, například:
  - automaticky analyzovat strukturu jazykových výrazů
  - překládat počítačem texty mezi různými jazyky
  - zachytit významy obsažené v textech, vyhledávat je
  - zlepšit různé aplikace znalostmi o jazyku (např. information retrieval, rozhraní k databázím/informačním systémům)
- hlavní motivací je plnohodnotná (obousměrná) komunikace s počítačem v přirozeném jazyce (což je jedna z hlavních úloh v oblasti AI)

Motivace

Použití velkých textových korpusů

- korpusový manažer Manatee/Bonito
- Sketch Engine
- OUP www.askoxford.com/oec
- Manatee se užívá v ÚČNK, ÚJČ, SNK, Berlínské akademii, pro maďarský, slovinský, chorvatský, ruský korpus
- učení se cizím jazykům

Jak se počítačově zpracovává přirozený jazyk?

- **korpus** – rozsáhlý soubor textů
  - umožňuje nám nahlédnout, jak se jazyk používá
  - lze na něm zkoumat různé pravidelnosti/zákonitosti (která slova se často používají spolu s jinými slovy atp.)
- **statistické metody a strojové učení**
  - máme-li dostatečně rozsáhlá ručně zpracovaná data, je možné v nich obsaženou “znalost”/informaci použít pomocí metod statistiky a strojového učení na zpracování nových dat
- a mnoho dalšího ...

Velikost korpusu

- **korpus** – rozsáhlý soubor textů
- Co znamená *rozsáhlý*?
- první koprusy: 1 milion slov
  - příliš malé pro zajímavější výsledky
  - dostačující pro globální statistiky
  - délka věty/slova, nejčastější slova
- nyní běžně stovky milionů slov
  - průměrná rychlost čtení je 125–225 slov za minutu
  - $200 * 60 * 18 = 216000$ slov za den (18 hodin)
  - $\rightsquigarrow$ 79 milionů za rok (365 dní)
  - dost velká slovní zásoba
- pro větší jazyky jsou nyní dostupné giga-korpusy
  - více než miliarda slov
  - zhruba 50 let čtení při 4 hodinách denně
  - málokdo dokáže přečíst více

Mohou počítače porozumět volnému textu?

- v korpusech máme dostatek infomací
- zpracování informací na počítači: manipulace se symboly
- potřeba definovat různé úrovně *elementů/symbolů*
  - znak (UTF-8)
  - slovo (znaky mezi mezerami, oddělovači)
  - lemma (základní tvar)
  - význam

významy

Slovo (a jeho některé části) jsou základními nositeli významu

- slovo bez kontextu – žádný význam, mnoho potenciálních významů
- stejné slovo v různých kontextech – různé významy
- ▸ příklad “korpus”
- slovo v *podobných* kontextech – stejný význam
- co to je kontext?

Co to je kontext?

Kontext jsou slova v okolí klíčového slova.

- Jaké okolí?:
  - následující slovo
  - předcházející slovo
  - okno, +1 až +5
  - okno, -5 až -1
- Ne všechna slova v okolí jsou důležitá.
- Jak určíme důležitost?
  - nejčastější kolokace – ale to je “the”
  - (statisticky) nejvýznamnější – jaký vzorec?

Word Sketch

Jednostránkový souhrn chování slova ▸ try online

Word Sketch

Jak jej lze vytvořit

- Velký vyvážený korpus
- Vyhledáme závislé prvky (subjects, objects, heads, modifiers etc)
- Seznam kolokací pro každou gramatickou relaci
- Statistika pro třídění každého seznamu

The Sketch Engine

- Vstup:
  - libovolný korpus, libovolný jazyk
  - lemmatizovaný, značkovaný
  - specifikace gramatických relací
- Výstup:
  - Word sketches
  - Thesaurus
  - Dotazovací systém

Koeficient výnačnosti

- počty výskytů $(word_1, gramrel, word_2)$
- $AScore(w_1, R, w_2) =$
  $\log \frac{||w_1, R, w_2|| \cdot ||*, *, *||}{||w_1, R, *|| \cdot ||*, *, w_2||} \cdot \log(||w_1, R, w_2|| + 1)$

Koeficient podobnosti

- porovnání profilů slov w_1 a w_2
- pouze důležité (význačné) kontexty
- jaký je překryv
- počty $(word_1, (gramrel, word_i))$ a $(word_2, (gramrel, word_i))$

$$Sim(w_1, w_2) = \frac{\sum_{(tup_i, tup_j) \in \{tup_{w_1} \cap tups_{w_2}\}} AS_i + AS_j - (AS_i - AS_j)^2/50}{\sum_{tup_i \in \{tup_{w_1} \cup tup_{w_2}\}} AS_i}$$

Velikosti dat

Velikosti korpusů, jejich slovníků a počty slov v kontextech

Korpus	Velikost	Slov	Lemat	Různé k.	Všechny k.
BNC	111m	776k	722k	23m	63m
SYN2000	114m	1,65m	776k	19m	58m
OEC	1,12g	3,67m	3,12m	84m	569m
Itwac	1,92g	6,32m	4,76m	67m	587m

Velikosti slovníků i počty různých kontextů rostou sublineárně s velikostí korpusu.

Velikost matice

- Podobnost všech dvojic lemmat
- Matice velikosti N^2, kde N je 700k – 5m
- Počet prvků v řádech tera (10^{12})
- Matice je naštěstí velice řídká
- Většina hodnot je 0 nebo “skoro” 0
- Dokonce většina celých řádků/sloupců je prázdných

Praktické velikosti dat

- Výpočet pouze pro slova s minimální četností
- Lépe limitovat počty kontextů než prostých výskytů
- Z kontextů brát pouze statisticky významné

Korpus	MIN	Lemmat	KWIC	CTX
BNC	1	152k	5.7m	608k
BNC	20	68k	5.6m	588k
OEC	2	269k	27.5m	994k
OEC	20	128k	27.3m	981k
OEC	200	48k	26.7m	965k
Itwac	20	137k	24.8m	1.1m

Praktické velikosti dat

- Matice velikosti N^2, kde N je 50k – 200k
- Počet prvků v řádech giga (10^{10})
- Hodnota každého prvku vznikne aplikací funkce podobnosti na vektory délky K = 500k – 1m.
- Přímočarý algoritmus pro výpočet celé matice má časovou složitost $O(N^2K)$.
- Složitost je polynomiální, ale algoritmus je prakticky nepoužitelný pro dané rozsahy hodnot.
- Odhadované doby výpočtu jsou v měsících až letech.
- Heuristiky snižují velikosti N a K na úkor přesnosti výsledných hodnot.
- Doba výpočtu je potom v řádech dnů s chybou 1–4%.

Efektivní algoritmus

- I menší matice je velice řídká
- Není potřeba počítat podobnost pro slova, která nemají nic společného,
- tedy nemají žádný společný kontext.
- Hlavní cyklus algoritmu tedy nevedeme přes slova, ale přes kontexty.

Efektivní algoritmus

- Vstup: seznam všech možných slov v kontextech, $\langle w, r, w' \rangle$, s četnostmi výskytů v korpusu
- Výstup: matice podobností slov $sim(w_1, w_2)$

for $\langle r, w' \rangle$ **in** CONTEXTS:
    WLIST = set of all w where $\langle w, r, w' \rangle$ exists
    for w_1 **in** WLIST:
        for w_2 **in** WLIST:
            $sim(w_1, w_2) += f(frequencies)$

Optimalizace

- Pokud $|WLIST| > 10000$, daný kontext přeskočíme.
- Matici $sim(w_1, w_2)$ během výpočtu nedržíme celou v paměti.
- Opakovaný běh hlavního cyklu pro omezený rozsah w_1.
- Místo $sim(w_1, w_2) += x$ generujeme na výstup $\langle w_1, w_2, x \rangle$.
- Výstupní seznam potom setřídíme a sčítáme jednotlivé x.
- Využití TMMS (Two Phase Multi-way Merge Sort) s průběžným sčítáním.
- Místo několika stovek GB třídíme jednotky GB.

Výsledky

- Algoritmus je řádově rychlejší než přímočarý algoritmus. (18 dnů $\times$ 2 hodiny)

Korpus	MIN	Lemmat	KWIC	CTX	čas
BNC	1	152k	5.7m	608k	13m 9s
BNC	20	68k	5.6m	588k	9m 30s
OEC	2	269k	27.5m	994k	1h 40m
OEC	20	128k	27.3m	981k	1h 27m
OEC	200	48k	26.7m	965k	1h 10m
Itwac	20	137k	24.8m	1.1m	1h 16m

- Bez omezení přesnosti.
- Možnost snadné paralelizace.