Slovo, význam a počítač

19. 6. 2010

Jan Hajič, MFF UK
hajicj@gmail.com

O čem se bude mluvit

- Počítačová lingvistika (CL): O co se snažíme a o co se nesnažíme
- Vztah slovo-význam
- Vektorová sémantika a lexikálně-sémantické prostory
- Příklad: latentní sémantická analýza

O čem se bude mluvit

- CL: O co se snažíme a o co se nesnažíme
- Vztah slovo-význam
- Vektorová sémantika a lexikálně-sémantické prostory
- Příklad: latentní sémantická analýza

O co se snažíme?

- Cíl počítačové lingvistiky: „Vytvořit stroj, který bude schopen smysluplně komunikovat s člověkem v přirozeném jazyce.“
- Problém: přirozený jazyk není zdaleka tak precizní, jak by se počítačům líbilo
- Řešení (aspoň teď): především *statistické metody*
- Metody: vytváří se nějaký *model* jazyka a nad ním *algoritmus*, kterým se snažíme řešit nějaký problém
- Model je nějaká aproximace – chceme, aby byla dostatečně dobrá

O co se snažíme?

- *Správnost*
  - Problém: jak říct, nakolik je náš stroj dobrý?
- *Rozumná výpočetní složitost*
  - Aneb nechceme čekat na výsledky stovky let!
- *Adekvátnost*
  - Nakolik se náš model blíží tomu, jak se jazyk skutečně používá
  - Už i Google používá při automatickém překladu syntax...

Statistické metody

- Počítač se na komunikaci dívá jako na produkci posloupnosti znaků (písmen, slov, vět…)
- Chceme co nejlépe simulovat komunikaci lidskou (smysluplnou, v přirozeném jazyce)
- Tedy chceme vždycky vybrat nějakou správnou znaků
- Správnou ≈ co by tak řekl člověk?

Statistické metody

- Co by tak řekl člověk: jak to zjistit?
  - Přes jazyk jako systém pravidel: věta má přívlastek, předložka *von* v němčině se pojí se třetím pádem, atd.
  - Pomocí toho, co už kdy člověk řekl
    - Což vede právě ke statistickým metodám (mainstream od 90. let)

Jazykový model: příklad

- Pro automatické rozpoznávání jazyka: tzv. *trigramy* (trojice po sobě jdoucích znaků)
- Jazyk reprezentujeme tabulkou četnosti trigramů v trénovacích datech
- Při samotném rozpoznávání:
  - Uděláme si trigramovou tabulku pro cílový text
  - Najdeme „nejbližší“ jazykový model a na základě toho se rozhodneme, v jakém je text jazyce
  - „Blízkost“ je určena nějakou vhodnou matematickou funkcí

Jazykový model: příklad

- Správnost? Funguje to. :-)
- Výpočetní složitost? Rozumná.
- Adekvátnost?
  - Model zhruba zachycuje rozdílnou slovní zásobu a morfologii
  - Ukazuje se, že pro náš úkol to stačí

O co se nesnažíme

- Neříkáme, že „takhle jazyk ve skutečnosti funguje“!
- Nevyžadujeme stoprocentní správnost a adekvátnost

Kde se s CL můžete setkat

- Některé podproblémy a aplikace:
  - Strojový překlad (Machine Translation, MT)
  - Rozpoznávání mluvené řeči (Speech Recognition)
  - Vyhledávání (Information Retrieval, IR)
  - …

O co jde v tomhle referátu

- Jak naučit počítač zacházet se slovní zásobou?
- Jak reprezentovat význam slova?
- Myšlenka: použít *vektorovou sémantiku*
  - *„Význam slova odpovídá tomu, s jakými jinými slovy se používá.“*
  - Tedy: charakterizovat slovo jeho sousedy

O čem se bude mluvit

- Počítačová lingvistika (CL): O co se snažíme a o co se nesnažíme
- Vztah slovo-význam
- Vektorová sémantika a lexikálně-sémantické prostory
- Příklad: latentní sémantická analýza

Vztah slovo-význam

- Co je to slovo?
  - Chceme, aby slovo mělo význam
  - Řetězec písmen?
  - Základní tvar?
- Co je to význam?
  - Klasická definice: význam je to, co sdělujeme a *není* to jazyk (jazyk – označující, význam – označované)
- Má slovo vždy význam (a je nositelem významu vždy slovo)?

Vztah slovo-význam

- Neatomicita slova z hlediska významu
  - „Alice loves Bob.“ ≈ „Alice“ + „loves“ + „Bob“ + „.“?
- Není jasné, jestli můžeme významy rozkládat, aniž bychom přitom ztráceli informace
  - Někdy určitě nemůžeme (ustálené fráze)
- Význam slova nějak závislý na tom, na jaký *kontext* se podíváme?

Vztah slovo-význam

- Jak mají slova významy?
  - Kolik významů má slovo (a kolik slov má význam)?
  - Homonymie vs. polysémie vs. další rozdíly

- Hypotézy:
  - co slovo, to význam
  - co text, to význam
  - co kolokace, to význam

Vztah slovo-význam

- No ale slovo ve skutečnosti pokaždé znamená něco trochu jiného
  - *Alice loves Bob* vs. *Alice loves tennis*
  - *Alice loves Bob* vs. *Alice loves Cindy*
  - *Alice loves Bob* vs. *Alice loves Bob*

- Tedy: slovo jako jistá *aproximace* nějakého „oblaku významů“

Slovo jako aproximace

- Hezké z hlediska CL
- Jak tuto aproximaci najít?

- Upřesníme hypotézu význam ≈ kontexty a použijeme ji jako model

Vztah slovo-kontext-význam

- Význam každé pasáže jedinečný
- Význam slova jako jistá kombinace významů všech pasáží v našem korpusu, ve kterých se vyskytuje
- „Slovník“:

  Alice(X loves Bob:4, X loves tennis:1, X loves Cindy:1, …)

  loves(Alice X Bob:4, Alice X tennis:1, Alice X Cindy:1, …)

Vztah slovo-kontext-význam

- Od reprezentace významu pomocí pasáží, ve kterých se popisované slovo vyskytuje, k reprezentaci pomocí ostatních slov jako takových:

  Alice(Bob:4, loves:6, tennis:1, Cindy:1, …)

  loves(Alice:6, Bob:4, tennis:1, Cindy:1, …)

- Dostáváme slovo jako *vektor*

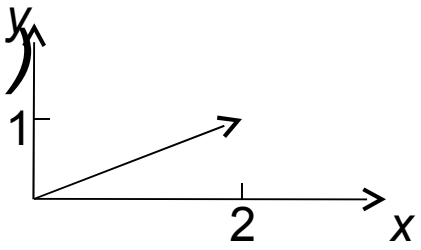
O čem se bude mluvit

- Počítačová lingvistika (CL): O co se snažíme a o co se nesnažíme
- Vztah slovo-význam
- Vektorová sémantika a lexikálně-sémantické prostory
- Příklad: latentní sémantická analýza

Odbočka k vektorům

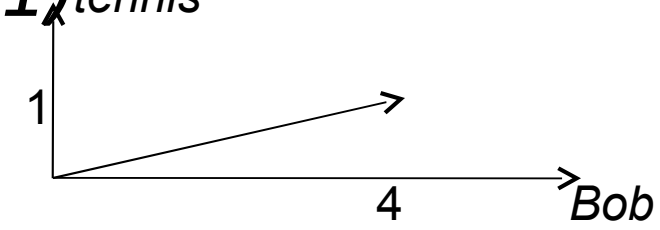
- Z analytické geometrie:

u(2;1)



analogicky můžeme mít:

Alice(4;1)



Vektorová sémantika

- Význam slova tedy reprezentujeme jako *vektor* v prostoru, jehož rozměry jsou ostatní slova/významy
  - slovo je jakási „nálepka“ pro význam

Lexikálně-sémantické prostory

- Máme tedy určitý „významový prostor“, jehož rozměry jsou slova
- Význam každého slova je reprezentován jedním bodem
  - Ovšem pozor: toto je aproximace! (jisté „významové těžiště slova“)
  - Ve skutečnosti slova mají různě ostré hranice – a spíše nejasnější
    - obecně: čím používanější slovo, tím neostřejší

Lexikálně-sémantické prostory

- Vzdálenosti: můžeme měřit úplně stejně jako v normálních vektorových prostorech (třeba v rovině)

- Jak je tato reprezentace významu adekvátní?
  - Tzn. pracujeme se slovy skutečně jako s nějakými oblastmi ve významovém prostoru?
- Důležité: jestli L-S prostor souhlasí s teorií *aktivace*

Aktivace

- Teorie aktivace se snaží vysvětlit, jakým způsobem volíme slova

- Aktivační trojúhelníky:
  - „Attend“: classes, meetings, weddings… school
  - „School“: teachers, students… classes

  - Takže: {attend, school} -> classes

Aktivace

- Klíčová slova:
  - Aktivace
  - Aktivační síla
  - Asociační vzdálenost
  - Asociační dráha
- No a všechno tohle bychom v L.-S. prostorech uměli simulovat ☺

Lexikálně-sémantické prostory

- Potíže:
  - Ohromný počet rozměrů (co slovo, to rozměr!)
  - Neumí v této podobě zachycovat syntaktické vztahy
  - Nedokáže jít až k úplnému správnému jazykovému vyjádření

O čem se bude mluvit

- Počítačová lingvistika (CL): O co se snažíme a o co se nesnažíme
- Vztah slovo-význam
- Vektorová sémantika a lexikálně-sémantické prostory
- Příklad: Latentní sémantická analýza

Latentní sémantická analýza

- 1998: Deerwester, Dumais, Furnas, Harshman, Landauer, Lochbaum, Streeter
- Metoda konstruování L-S prostoru založená na *singulární dekompozici matic (SVD)*
- Umí zachycovat hlubší souvislosti než jednoduché souvýskytové frekvenční analýzy
  - Výsledky až o 30 % f-score lepší (hlavně recall)

Latentní sémantická analýza

- Použití:
  - Vyhledávání (!!!)
  - Automatické známkování
  - Automatická sumarizace textů
  - Základ pro rozlišování polysémie
  - …

Latentní sémantická analýza

- Data: libovolný korpus
  - Alespoň 20 000 různých slov ve 20 000 různých pasážích
  - Největší LSA systémy: asi 500 milionů slov (zhruba takovému objemu je vystaven americký středoškolák)
  - Nepotřebuje žádné další zpracovávání (morfologické značkování apod.)

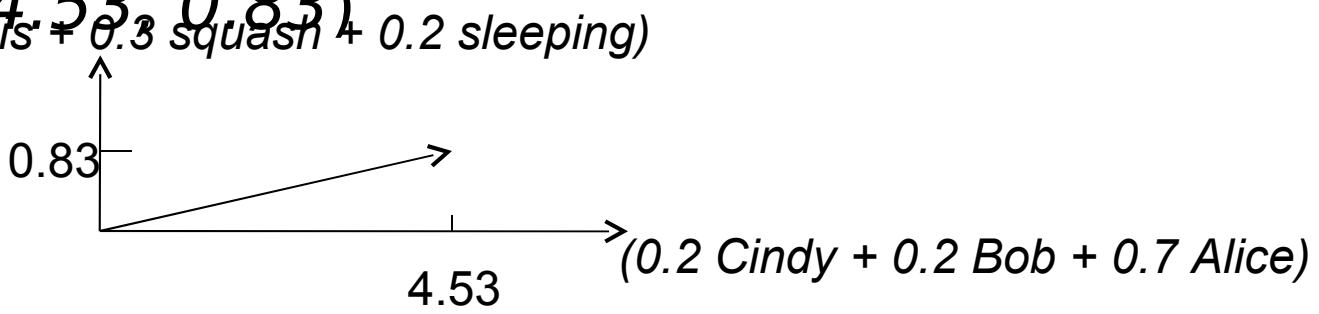
Latentní sémantická analýza

- Algoritmus:
  1) Vyplnit matici (≈ tabulku) výskytů zkoumaných slov v datech – pasážích
  2) (ne nutně) přepočítat hodnoty podle důležitosti
     - tf-idf transformace
  3) Provést SVD
  4) Hrát si s výslednými maticemi

Latentní sémantická analýza

- Redukce počtu rozměrů uvnitř SVD: vybereme jenom ty nejrelevantnější souřadnice

loves(4.53, 0.83)



- Rozměru potom odpovídá jistý *koncept* (prototyp?)

Příklad

- Landauer, Foltz, Laham (1998)
- Data:

  c1: *Human* machine *interface* for ABC *computer* applications
  c2: A *survey* of *user* opinion of *computer system response time*
  c3: The *EPS user interface* management *system*
  c4: *System* and *human system* engineering testing of *EPS*
  c5: Relation of *user* perceived *response time* to error measurement
  m1: The generation of random, binary, ordered *trees*
  m2: The intersection *graph* of paths in *trees*
  m3: *Graph minors* IV: Widths of *trees* and well-quasi-ordering
  m4: *Graph minors*: A *survey*

  (nadpisy článků)

Příklad

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

Příklad

- Zrekonstruovaná matice po SVD:

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	0.16	0.40	0.38	0.47	0.18	-0.05	-0.12	-0.16	-0.09
interface	0.14	0.37	0.33	0.40	0.16	-0.03	-0.07	-0.10	-0.04
computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
user	0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19
system	0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.20	-0.11
survey	0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
trees	-0.06	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	0.66
graph	-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
minors	-0.04	0.25	-0.10	-0.21	0.15	0.22	0.50	0.71	0.62

Příklad

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	0.16	0.40	0.38	0.47	0.18	-0.05	-0.12	-0.16	-0.09
interface	0.14	0.37	0.33	0.40	0.16	-0.03	-0.07	-0.10	-0.04
computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
user	0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19
system	0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.20	-0.11
survey	0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
trees	-0.06	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	0.66
graph	-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
minors	-0.04	0.25	-0.10	-0.21	0.15	0.22	0.50	0.71	0.62

Podobnosti:
r(human,user) = -0.38
r(human,minors) = -0.24

Podobnosti:
r(human,user) = 0.94
r(human,minors) = -0.83

- Co nám LSA říká?
  - *(trees,m4)* = 0.66 … „V pasáži obsahující *graph* a *minors* je šance 0.66, že se tam také vyskytne slovo *trees*.”

Důsledky LSA pro lingvistiku

- Zjevně je možné se pomocí statistických metod naučit z textu daleko víc, než co v něm je napsáno
  - „John is Mary’s father.” “Mary is Peter’s sister.”

    → “John is Peter’s father.”

A to je vše, přátelé.