Využití pravidlové disambiguace

(Jak disambiguátoři přicházejí o iluze)

johanka

April 16, 2008

Výchozí podmínky

Co máme k dispozici?

- Jazyk LanGR + poloboha Pavla Květoně
- Pravidla pro (částečnou) morfologickou disambiguaci (Niki Petkevič, Tomáš Jelínek)

O co se snažíme?

- Něco z toho dostat (ne nutně morfologickou disambiguaci)

Proč syntax?

- Mnohá disambiguační pravidla se v ní stejně vrtají a pak získanou informaci (zbytečně) zahodí
- Některá jsou "jasná" (identifikace podmětu a přísudku, předložkové fráze...)
- Jiná lze využít nepřímo či kolektivně (antecedent vztažného zájmena, cokoli v koordinaci..)
- LanGR snadno umožňuje i psaní ryze syntaktických pravidel

Přesná specifikace úlohy

- Ověření, zda lze stávající disambiguační pravidla využít pro parsing (úplný či částečný)
- Menší úpravy možné, doplňková pravidla možná, ale neměla by tvořit většinu díla
- V žádném případě nepsat od začátku nový parser! (Takových už je dost a lepších.)

Jak funguje disambiguace? (138. opakování)

- Načítá text po větách, provede či načte morfologickou analýzu
- Na každou větu pouští v cyklu pravidla, dokud se něco děje (ubývají tagy)
- Pravidla mají část konfigurační (kde se uplatní) a akční (co tam mají udělat)
- Pravidla se dělí na negativní (mažou nepřípustné tagy) a pozitivní (vybírají přípustné tagy)

Technické řešení syntaktické extenze

- Syntax běží zároveň s disambiguací (záleží ovšem na konkrétních pravidlech)
- LanGR nebuduje žádné interní datové stuktury, ale umí "vyhodit hlášku"
- Vyhodíme tedy hlášku, kdykoli máme (na základě namatchované konfigurace) pocit, že nějaká dvě slova jsou v určitém syntaktickém vztahu
- Následně sebereme všechny hlášky pro danou větu a zkusíme z nich něco postavit

Konkrétní vykonaná činnost

- Ohledána všechna pravidla (cca 2000), cca 400 z nich shledáno pro syntax vhodnými a rozšířeno, napsáno cca 80 vlastních "záchytných" ryze syntaktických pravidel
- Postprocessing (závěrečná "semílačka") v Perlu, poměrně netriviální
- Output: závislosti PDT a-layer compatible - ne nutně, hlavně kvůli evaluaci
- Přirozeným výsledkem je les, lze si vynutit ošklivý strom
- Evaluace, porovnání s jinými parsery + kombinovatelnost

Detaily posprocessingu

- Na vstupu dostane hlášky typu *CONNECT 1 2 Desc: Subj, Dom: P [Opt: level bezpečnosti]*
- V závislostním pojetí bere řídící členy jako kandidáty na otce, ale výsledek není přímočarý
- Některé závislosti vznikají z více vztahů (vztažná věta)
- Asi 30 různých dalších úprav (propagování vztahu skrz koordinaci apod.)
- Při sporu se méně bezpečné informace zahazují, ale ne hned
- Průběžná detekce cyklů
- Velká výhoda: zachování levelů bezpečnosti - každá výsledná závislost má stanoveno, jak moc jí věříme!

Čísla (PDT 1.0 etest)

Data	Coverage	Rules	Charniak	Collins
All	100 %	65.37 %	84.24 %	82.29 %
Very Safe	35.64 %	**93.35 %**	93.65 %	93.07 %
Safe	22.69 %	80.42 %	86.72 %	85.26 %
Unsafe	21.45 %	50.63 %	79.86 %	77.73 %
Untouched	20.20 %	14.79 %	69.39 %	64.78 %

Proč je to úplně blbě? (I)

- Pozitivní pravidla nepřinesou nic nového
  - Zdeněk má lepší :)
  - Všichni mají stejné problémy a statistika je vždy o krok napřed
- Negativní pravidla pozitivním nepomůžou
  - Lze je (případně) zkusit jako hinty někomu jinému
  - Nebo jako Karlsson a jeho parta - zkusit všechno a seškrtávat

Proč je to úplně blbě? (II)

- Disambiguace má malé pokrytí - trvalý problém!
- Nepomůže ani vmezeření statistického taggeru
- Systém je regulární :((= jednoprůchodový. Nelze vypustit rozpoznaný kus věty a znova začít ohledávat.)
- Na tohle každý parser jednou dojede (IMHO i tagger)
- Bezkontextovost lze dohackovat, ale dost brutálně a s nejistým výsledkem

Mezihra

Drtivá disambiguační čísla

Opáčko z minulého týdne (únor 2006, PDT 2.0 d-test):

- morfologie: precision 25.72 %, recall 99.40 %
- root (bezpečná pravidla): precision 54.95 %, recall 98.84 %
- všechna pravidla: precision 70.11 %, recall 97.94 %
- perceptron Anny K.: accuracy 95.50 %

Jak použít disambiguaci k disambiguaci?

- Samotná nic moc (k "ryzí homonymii" má výsledek daleko)
- Zvednutí čísel v konečném čase nereálné (výkonnost systému je stabilizovaná s občasnými drobnými výkyvy směrem k vyšší precision či recallu)
- Heuristiky zklamaly očekávání (chybují příliš a uplatňují se málo)
- Jediná naděje je v kombinaci :)
- Nutná, ale nikoli postačující podmínka: musíme umět různé věci!

Jak mohou pravidla pomoci statistice?

Možností není zase tolik:

- Pustíme předem
- Pustíme zadem
- Sdrátujeme dohromady
- Ještě něco jiného?

Předení a drátování

- Všichni pořád básní o "ručně psaných featurách"
- Efektivně to ale znamená skoro totéž jako "pustit předem" (jen bychom tagy nemazali, ale označili)
- A má to i stejný zádrhel - velmi nízké uplatnění pravidel na zcela nedisambiguovaném (tedy defaultním) vstupu
- Tedy puštění předem funguje, nechybuje, ale nic moc nedělá (dle výběru pravidel od nepatrného zlepšení po patrné zhoršení :))
- Potěšující pozorování: Annin tagger není citlivý na redukování tagů na vstupu, netřeba přetrénovávat (naopak)

Pravidla po statistice

- Hodně chyb oprávněně chytí
- Víc jich ale nareportují neoprávněně a ještě víc neodhalí vůbec
- Proč 90 % chyb taggeru vůbec nenajdeme, toť otázka :) (neznámá slova?)
- Smazaný tag téměř vždy znamená problém ve větě, ale ledva v polovině případů je to přímo na místě onoho tagu (přecitlivělost na překlepy, gramatické chyby...)
- Pravidla po statistice nelze tedy v žádném případě použít jako automatický detektor a opravovač chyb

Co teď? - Úkoly hodné celých mužů

1. Proč se pravidla v preprocessingu nechytají?
   - Kvůli slovnědruhové homonymii!
   - No tak jim s ní pomůžeme, ne?
2. Dají se nějak využít přecitlivělá pravidla v postprocessingu?
   - Dají. Třeba na odlišení slušně a méně slušně vypadajících/označkovaných dat.

Pravidla po rozetnutí slovního druhu

Pracovní postup (+ výsledky na d-testu):

1. Morfologická analýza (25.72 %/recall 99.40 %)
2. (opt) Pečlivě zvolená pravidla (zatím neděláme)
3. Tagger (95.50 %)
4. Vrátíme všechny tagy se stejným subposem, jako má vybraný tag (30.11 %, recall 98.87 %)
5. Pravidla (zatím root) (61.87 %/98.61 %) (vs. 54.95 %/98.84 %)
6. Tagger (95.73 % !) (vs. 95.50 %)

Ověření: pouze kroky: 1, 3, 4, 6 dávají jen 95.46 % - samotné rozfázování téměř nemá vliv

Pozorování: pokles recallu mezi 4 a 5 je nepatrný - není třeba redukovat množinu pravidel, naopak je možné zkusit ji rozšiřovat

Unsupervised vize (náčelníkova :))

Pracovní postup (+ výsledky na dtestu):

1. Ojetí hromady dat (SYN*) více taggery (3?) (95.50/93.97/94.37)
2. Vybrání pouze těch vět, kde se všechny na všem shodnou (33 % dat, 98.40 (m. recall 99.64))
3. (opt) Vyházení těch vět, kde se něco nelíbí pravidlům (30 % dat, 98.64 (m. recall 99.71))
4. Použití výsledku jako nových trénovacích dat pro taggery (???)

Postranní čísýlka

Accuracy/podíl na datech:

- Morče + pravidla 96.1/80
- Morče + Hajič 97.94/43 (+pravidla 98.25/38)
- Morče + Krbec 97.77/47
- Hajič + Krbec 96.97/45
- alespoň dva taggery 95.89/89

K zamyšlení: Šlo by to i s parsery? A šlo by k tomu použít negativních pravidel?

Prozatímní závěry

- Parsování pozitivními pravidly byl blbej nápad
- Negativní pravidla zatím čekají na možné syntaktické využití
- Pravidla nelze žádným triviálním způsobem využít ke zlepšení situace na poli taggerů
- Existují však způsoby netriviální :)
- Úkol do diskuse/ke kafi/na doma: Máte nějaké další nápady?