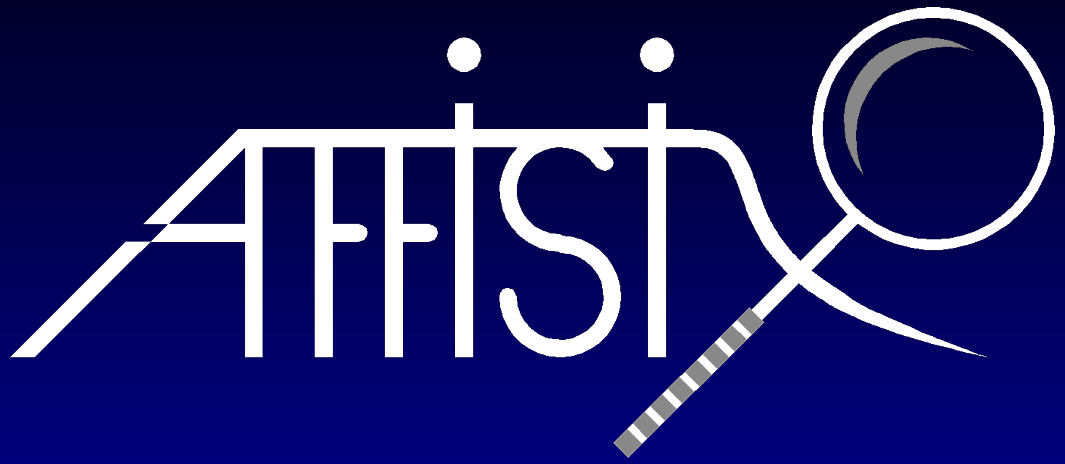


Automatická segmentace slov s pomocí nástroje Affisix

Michal Hrušecký, Jaroslava Hlaváčová

Michal@Hrusecky.net, Hlavacova@ufal.mff.cuni.cz

Motivace

Při zpracování přirozeného jazyka

- nikdy nemůžeme mít ve slovníku všechna slova
- i o slovech, která neznáme, chceme být schopni něco říct

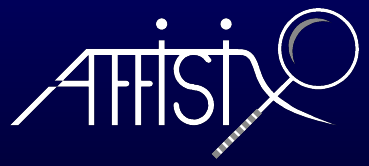
Jak vznikají slova?

- přidáním jiné předpony
- přidáním jiné přípony
- ...

Předpony často neovlivňují mluvnické kategorie

$\Rightarrow$ mohlo by se hodit znát dělení slova (budeme-li to umět automaticky)

Algoritmy



Úvodní definice

- nepotřebujeme opravdové lingvistické předpony a přípony
- potřebujeme něco, co se tak jen chová

Předpony

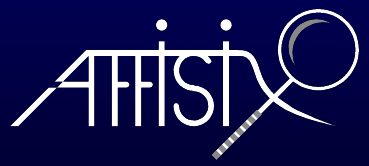
- předpona je společný začátek mnoha slov
- předponu můžeme připojit na začátek slova a vznikne nám slovo nové

Přípony

- přípona je společný konec mnoha slov

Algoritmy:

Metoda čtverců



Čtverec

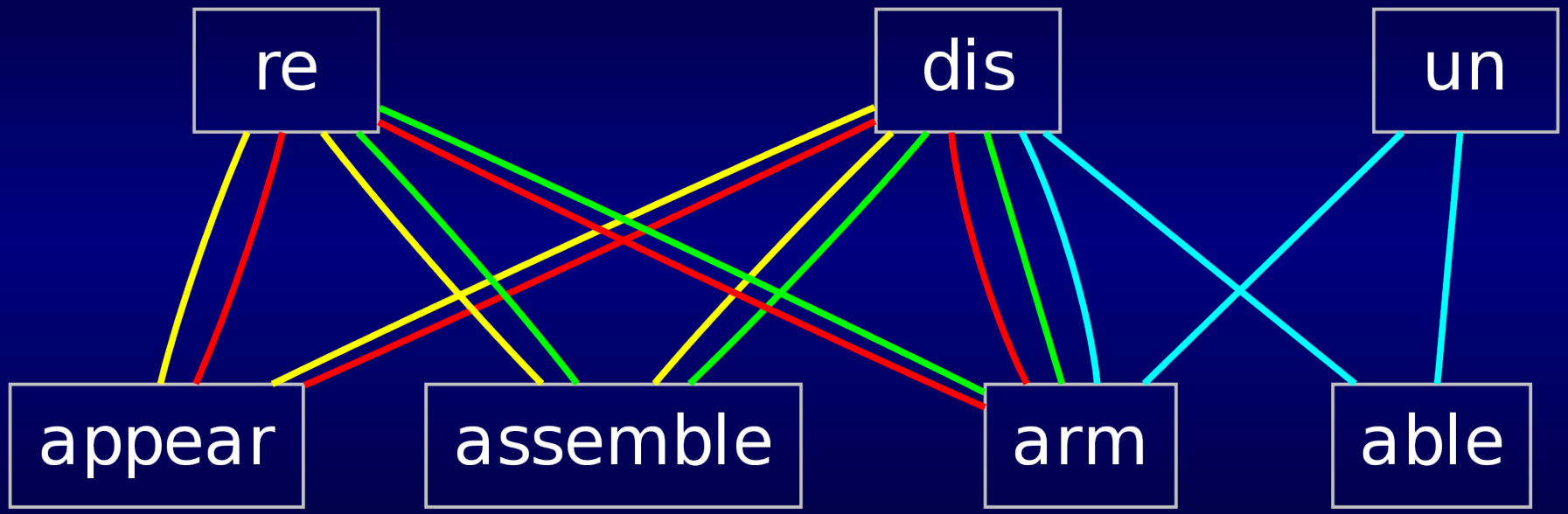
- předpony můžeme na začátku slova zaměňovat
- přípony můžeme na konci slova zaměňovat

$\Rightarrow$ Definujeme čtverec:

Nechť *a,b,c,d* jsou řetězce, Ω množina slov z jazyka a :: značí operaci spojení. Potom $<a,b,c,d>$ tvoří čtverec, právě tehdy pokud:

- a::c je platné slovo (a::c $\in \Omega$)
- a::d je platné slovo (a::d $\in \Omega$)
- b::c je platné slovo (b::c $\in \Omega$)
- b::d je platné slovo (b::d $\in \Omega$)

Čtverec - příklad



Segmentace

Obecně

- zkusíme rozdělit slovo
- spočítáme, v kolika čtvercích se obě části vyskytují
- hodně čtverců $\Rightarrow$ dělení je dobrý kandidát pro dělení slova

Segmentace

Obecně

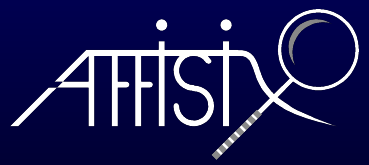
- zkusíme rozdělit slovo
- spočítáme, v kolika čtvercích se obě části vyskytují
- hodně čtverců $\Rightarrow$ dělení je dobrý kandidát pro dělení slova

Konkrétně

- zkusíme rozdělit slovo
- u předpon spočítáme, v kolika čtvercích se vyskytuje počáteční část
- u přípon spočítáme, v kolika čtvercích se vyskytuje koncová část
- hodně čtverců $\Rightarrow$ dělení je dobrý kandidát na předponu/příponu

Algoritmy:

Metoda entropie



Entropie

- Vyjadřuje míru nejistoty
- Definována následovně:

$$H(s) = -\sum_{s_i \in S} p(s_i|s) \log_2 p(s_i|s)$$

kde S je množina jevů, které mohou nastat po jevu s.

Dopředná entropie

- Předpona je společný začátek mnoha slov
  - $\Rightarrow$ je těžké předpovědět konec slova
  - $\Rightarrow$ budeme-li sledovat písmena jdoucí po sobě, entropie bude vysoká
- Dopřednou entropii H_f definujeme jako:

$$H_f(r) = -\sum_{s_i;r::s_i\in\Omega} p_f(s_i|r)\log_2 p_f(s_i|r)$$

kde $p_f(s_i|r)$ označuje pravděpodobnost, že slovo začínající r bude pokračovat s_i.

Zpětná entropie

- Přípona je společný konec mnoha slov
  - $\Rightarrow$ je těžké předpovědět začátek slova
  - $\Rightarrow$ budeme-li sledovat písmena jdoucí po sobě od konce, entropie bude vysoká
- Zpětnou entropii H_b definujeme jako:

$$H_b(r) = -\sum_{s_i; s_i :: r \in \Omega} p_b(s_i|r) \log_2 p_b(s_i|r)$$

kde $p_b(s_i|r)$ označuje pravděpodobnost, že slovo končící r bude začínat s_i.

Problémy metody entropie

- Dopředná entropie je vždy na začátku slova vysoká
- Dopředná entropie je vždy na konci slova nízká
- Měla by postupně klesat

$\Rightarrow$ Jen hodnota na určení vhodnosti dělení nestačí

Slovo *odpředchvíle*

	o	d	p	ř	e	d	c	h	v	...
H_f		2.55	2.92	2.16	1.43	0.00	1.55	0.00	0.00	...

Diferenční entropie

⇒ Potřebujeme proměnlivou mez

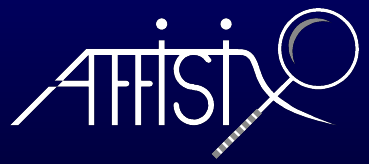
⇒ Lze použít změnu entropie jako měřítko

- Nehledáme místa s vysokou entropií, ale místa, kde entropie naproti očekávání roste

Slovo *odpředchvíle*

	o	d	p	ř	e	d	c	h	v	...
H_f		2.55	2.92	2.16	1.43	0.00	1.55	0.00	0.00	...
H_{df}		0.00	0.38	-0.76	-0.73	-1.43	1.55	-1.55	0.00	...

Affisix



O programu

- open source
- podporuje vyhledávání předpon, přípon i vnitřních částí slov
- dva módy
  - collector
    - Výstupem je seznam segmentů podle zadaných podmínek
  - filter
    - Označí ve vstupním textu segmenty podle zadaných podmínek
- podporuje vyhodnocování výrazů
  - lze snadno testovat různé podmínky/kombinace metod
- podporuje proměnné a více průchodů
  - lze vyjádřit ještě složitější podmínky

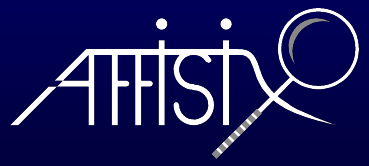
Ukázka konfigurace - Dopředná entropie

```
mode:                collector
precision:           8
cond_prefix_end:     10
cond_prefix_comb:    ">(@(res;fentr;avg;fentr(j));1)"
cond_prefix_format:  "%n: %w - %{fentr}\n"
```

Ukázka konfigurace - Kombinace entropií a XML výstup

```
mode:                   collector
precision:              8
cond_prefix_end:        10
cond_prefix_comb:       ">(@(res;comb;avg;
                            +(fentr(j);dfentr(j)));1)"
cond_prefix_format:     "<%n value=\"%{comb}\">%w</%n>\n"
```

Výsledky



Data a značení

Data

- data ze SYN2000 (lehce pročištěné)
- testováno na předpony
- výsledky hodnoceny člověkem

Značení

- $||H_f||$ - normalizovaná dopředná entropie
- $||H_{df}||$ - normalizovaná dopředná diferenční entropie
- $||S_f||$ - normalizovaný počet dopředných čtverců
- $C = ||H_f|| + ||H_{df}|| + ||S_f||$

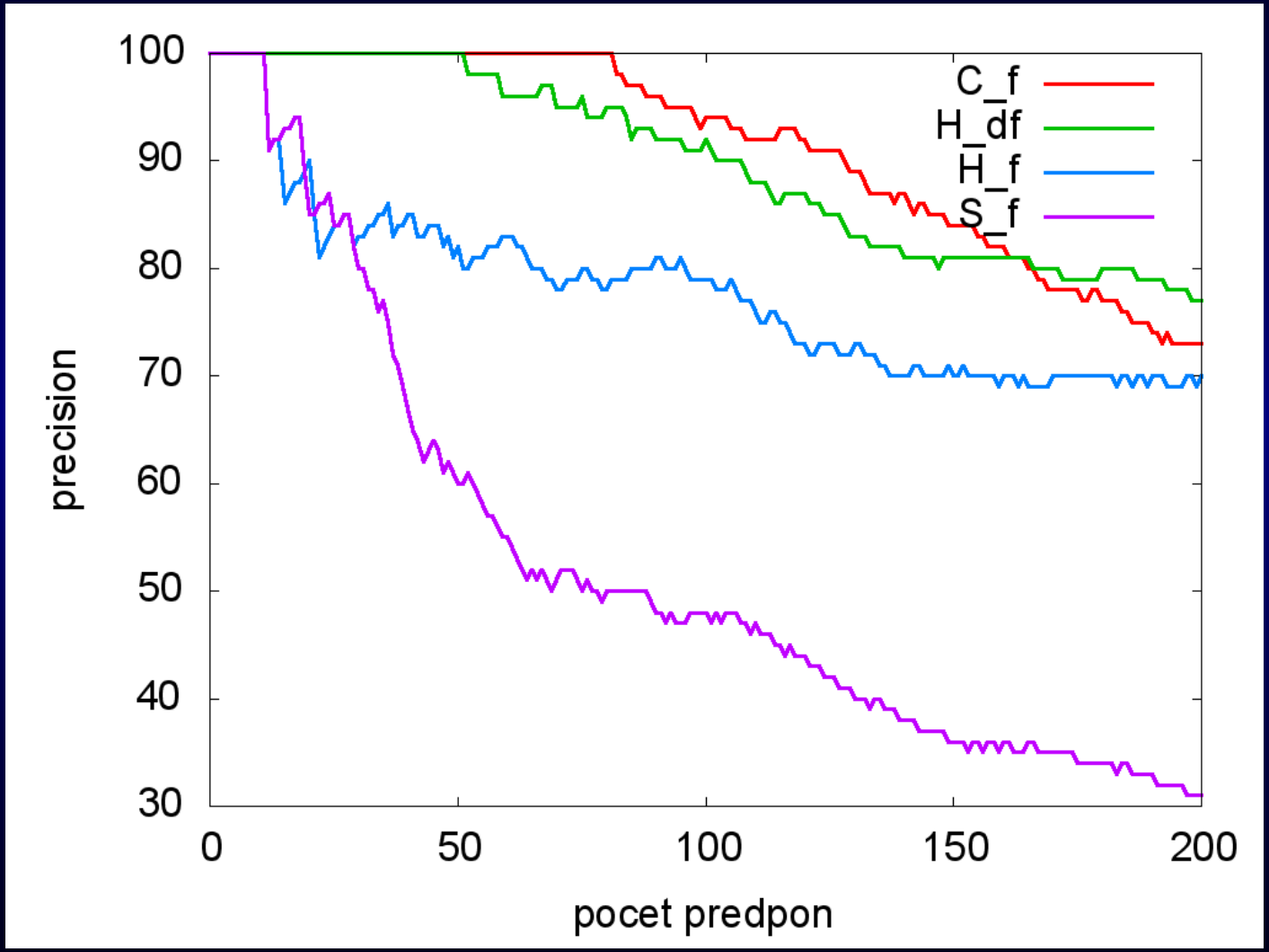
Výsledky vyhledávání předpon

	prefix	C_f	$\|\|H_f\|\|$	$\|\|H_{df}\|\|$	$\|\|S_f\|\|$
1.	super	2.682	0.967	0.978	0.737
2.	pseudo	2.602	0.963	0.961	0.677
3.	mikro	2.532	0.921	0.917	0.693
4.	sebe	2.505	0.898	0.849	0.757
5.	rádoby	2.480	0.983	1.000	0.496
6.	deseti	2.437	0.917	0.865	0.653
7.	mimo	2.423	0.968	0.801	0.653
8.	hyper	2.420	0.906	0.894	0.619
9.	anti	2.393	0.942	0.706	0.744
10.	roz	2.390	0.922	0.530	0.937

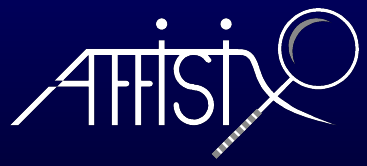
Srovnání precission jednotlivých metod (předpony)

metoda	10	50	100	150	200
C_f	100%	100%	94%	84%	73%
H_{df}	100%	100%	92%	81%	77%
H_f	100%	82%	79%	70%	70%
S_f	100%	60%	48%	36%	31%

Srovnání precission jednotlivých metod (předpony)



Plány do budoucna



Plány do budoucna

- další metody
- další zdokonalování nástroje Affisix
- pokusy s těmito novými vlastnostmi
- pokusy s jinými jazyky
- pokusy s reálnými aplikacemi
  - další pokusy se strojovým překladem

Děkuji za pozornost