

Wide-coverage Translation in GF

Krasimir Angelov

University of Gothenburg, Digital Grammars AB

August 18, 2017

1 Introduction

2 Free Language

- Resource Grammar
- Resource Lexicon
- Demo

3 Uncontrolled Language

- Demo

4 Summary

Get me out of here!

Can we escape from the controlled languages?

the bird stabs five

Clear

animals backs bad bellies big birds black bones breasts children
clouds cold correct days dirty dogs dry dull ears eggs eyes
feathers feet fingernails fires fish flowers forests fruits full good
grasses green guts hairs hands heads hearts heavy horns
knees lakes leaves legs lice livers long men moons mountains
mouths names narrow near necks new nights noses old
persons red rivers roads roots ropes rotten round seas seeds
sharp short skies skins small smooth snakes stars sticks stones
straight suns tails teeth thick thin tongues trees warm wet
white wide winds wings wives women worms years yellow

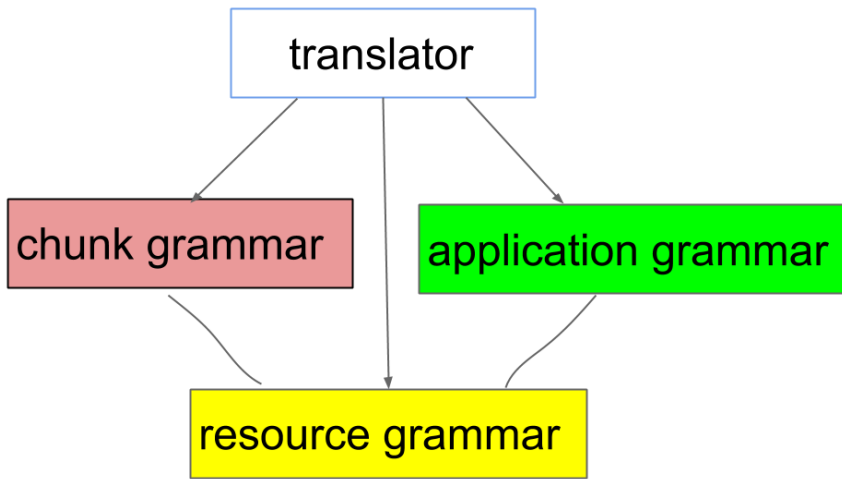
Grammar: bronzeage.pgf

From: BronzeAgeEng

- 1 Introduction
- 2 Free Language
 - Resource Grammar
 - Resource Lexicon
 - Demo
- 3 Uncontrolled Language
 - Demo
- 4 Summary

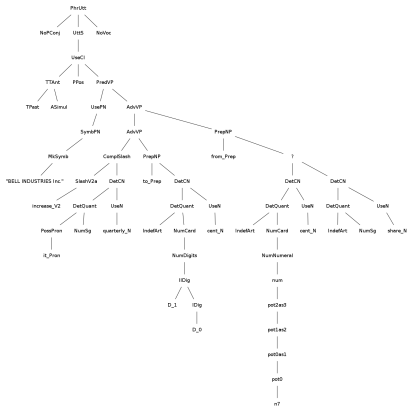
- 1 Introduction
- 2 Free Language
 - Resource Grammar
 - Resource Lexicon
 - Demo
- 3 Uncontrolled Language
 - Demo
- 4 Summary

The Components of the Grammar



Resource Grammar

- Level 1: Resource Grammar
 - morphology
 - word order
 - agreement



Statistical Disambiguation

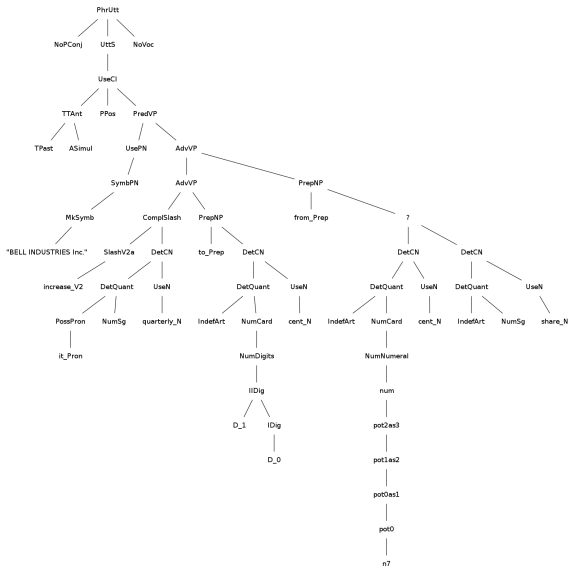
- In large grammars a single sentence can have millions of analyses
- Alternatives are ranked by their probabilities:

$$\begin{aligned} &P(\text{DetCN } (\text{DetQuant IndefArt NumSg}) (\text{UseN share..N}) \mid \text{NP}) \\ &= P(\text{DetCN} \mid \text{NP}) * P(\text{DetQuant IndefArt NumSg} \mid \text{Det}) * P(\text{UseN share..N} \mid \text{CN}) \\ &= P(\text{DetCN} \mid \text{NP}) * P(\text{DetQuant} \mid \text{Det}) * P(\text{IndefArt} \mid \text{Quant}) * P(\text{NumSg} \mid \text{Num}) \\ &\quad * P(\text{UseN} \mid \text{CN}) * P(\text{share..N} \mid \text{N}) \end{aligned}$$

Penn Treebank Parse Tree

```
( (S
  (NP-SBJ (NNP BELL) (NNP INDUSTRIES) (NNP Inc.) )
  (VP (VBD increased)
    (NP (PRP$ its) (NN quarterly) )
    (PP-DIR (TO to)
      (NP (CD 10) (NNS cents) ))
    (PP-DIR (IN from)
      (NP
        (NP (CD seven) (NNS cents) )
        (NP-ADV (DT a) (NN share) ))))
  (. .) ))
```

Penn Treebank in GF



Chunking

- local agreement
- local reordering
- built in a couple of hours from the resource grammar

fun

```
PhrUtt : Utt -> Phr ;
```

```
ChunkPhr : Chunks -> Phr ;
```

```
OneChunk : Chunk -> Chunks ;
```

```
PlusChunk : Chunk -> Chunks -> Chunks ;
```

fun

```
AP_Chunk      : AP -> Chunk ;
```

```
S_Chunk       : S  -> Chunk ;
```

```
NP_Nom_Chunk  : NP -> Chunk ;
```

```
NP_Acc_Chunk  : NP -> Chunk ;
```

Some Problems on Level 1

climb up
| \
klättra upp
| \
из+качвам

The boy **climbed up** the mountain.
The boy **climbed** the mountain **up**.

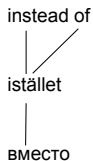
The prices **climbed up** quickly.
(different sense)

More Problems on Level 1

Some derivational morphology:



and simple multiword expressions:



Level 2: Non Compositional Translation

The RGL would be a perfect translation device if translation was compositional.

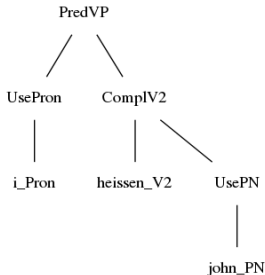
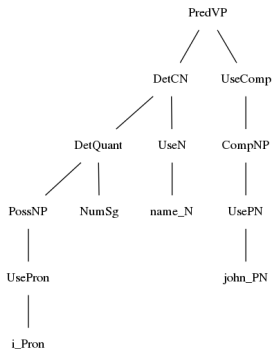
But compare:

- my name is John
- ich heisse John

Different languages use different linguistic devices for the same semantics.

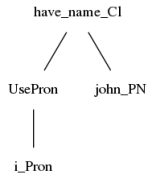
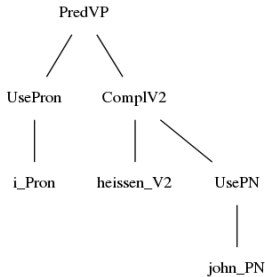
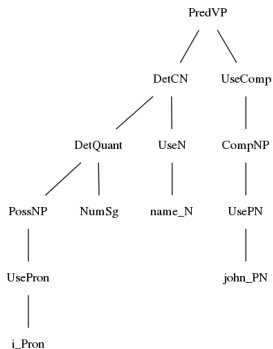
Three Different Trees for "My name is John"

My name is John:



Three Different Trees for "My name is John"

My name is John:



Semantic Predicates Defined Using the RGL

English:

```
lin have_name_C1 p n = PredVP (DetCN (PossNP p) (UseN name_N))  
                               (UseComp (CompNP (UsePN n)))
```

German:

```
lin have_name_C1 p n = PredVP p (CompV2 heissen_V2 (UsePN n))
```

Semantic Predicates Defined Using the RGL API

English:

```
lin have_name_C1 p n = mkC1 (mkNP (mkDet p) name_N) (mkNP n)
```

German:

```
lin have_name_C1 p n = mkC1 p (mkV2 heissen_V) (mkNP n)
```

French:

```
lin have_name_C1 p n = mkC1 p (reflV appeler_V) (mkNP n)
```

Semantic Predicates Defined Using the RGL API

In general stable cross lingual categories are:

- VP instead of V
- AP instead of A
- CN instead of N
- S in the worst case

An example from the Swedish Konstruktikon

```
is_wrong_VP : VP ;
```

- English: he is wrong

```
lin is_wrong_VP = UseComp (CompAP (PositA wrong_A))
```

- Swedish: han har fel

```
lin is_wrong_VP = ComplSlash (SlashV2 har_V2)  
                           (MassNP (UseN fel_N))
```

An example with Variable Parts

English: It makes sense
 It makes a lot of sense
 It makes some sense

```
lin makes_sense : Det -> VP ;
```

a More Typical Application Grammar

cat

Comment ; Item ; Kind ; Quality ;

fun

Pred : Item -> Quality -> Comment ;

This, That, These, Those : Kind -> Item ;

Mod : Quality -> Kind -> Kind ;

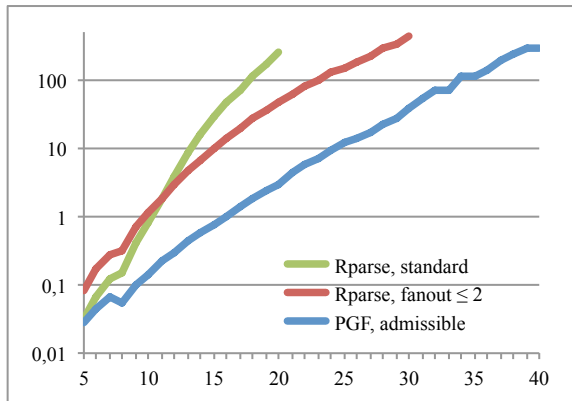
Wine, Cheese, Fish, Pizza : Kind ;

Very : Quality -> Quality ;

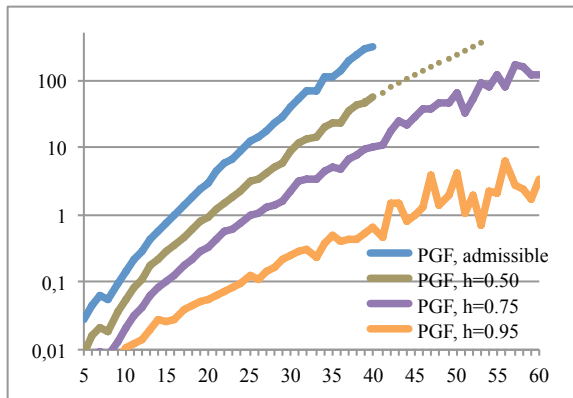
Fresh, Warm, Italian,

Expensive, Delicious, Boring : Quality ;

Comparison with RParse



Heuristics



- 1 Introduction
- 2 Free Language
 - Resource Grammar
 - Resource Lexicon
 - Demo
- 3 Uncontrolled Language
 - Demo
- 4 Summary

- A set of abstract language independent meanings

```
fun xylophone_N : N ;           -- 03721384 a percussion instrument w
                                wooden bars tuned to ...
fun arm_1_N : N ;              -- 05563770 a human limb; technically
                                the part of the superior
fun arm_2_N : N ;              -- 04565375 any instrument or
                                instrumentality used ...
fun account_for_V2 : V2 ;      -- 02635033 be the reason or
                                explanation for ...
fun studentFem_N : N ;         -- 10665698 a learner who is enrolled
                                in an educational ...
fun studentMasc_N : N ;        -- 10665698 a learner who is enrolled
                                in an educational ...
```

- Nouns, Verbs, Adjectives, Adverbs
 - Oxford Advanced Learners Dictionary
 - Princeton WordNet
 - Spelling variants (British/American/Others)
 - Harmonized with RGL
- Prepositions
 - PennTreebank
 - Wikipedia
- Verb Frames
 - PennTreebank

English Lexicon Example

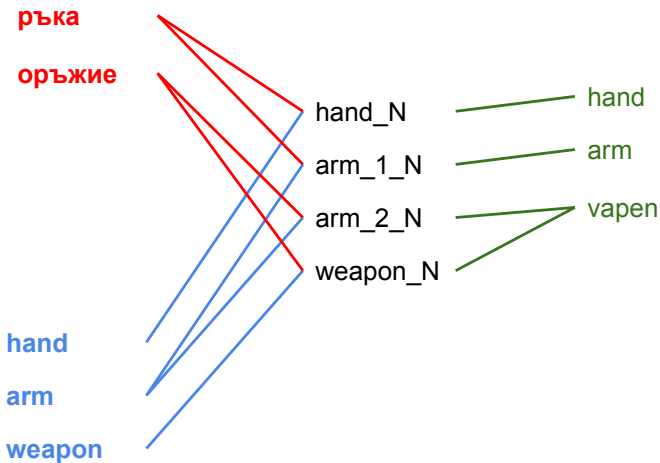
```
lin house_N = mkN "house" "houses";
lin play_V = mkV "play";
lin beautiful_A = compoundA (mkA "beautiful");
lin behind_Adv = mkAdv "behind";
lin instead_of_Prep = mkPrep "instead of";
lin theatre_N = variants {mkN "theatre";
                          mkN "theater"};
lin maharaja_N = variants {mkN "maharaja";
                           mkN "maharajah"};
lin ache_V = mkV "ache";
lin ache_for_V2 = prepV2 (mkV "ache") (mkPrep "for");
lin cod_liver_oil_N = mkN "cod-liver oil" ;
```

- Free Electronic Dictionaries (Bulgarian, Swedish)
- WordNet (Finnish, Russian)
- Universal WordNet (Bulgarian)
- Apertium (Bulgarian, Others?)
- Google Translate (Bulgarian, Swedish)
- Giza Phrase Tables (Bulgarian)
- PannLex (Thai)
- Manual Translation (Bulgarian, Chinese)
- Wiktionary (Most Other Languages)

- Smart Paradigms
- IrregXXX modules
- Free Morphological Lexicons
(OALD, Open Office, SALDO, KOTUS)

Learning Lexical Probabilities

There is no annotated corpus with abstract senses but we can learn the distribution by using EM and multilingual corpora.

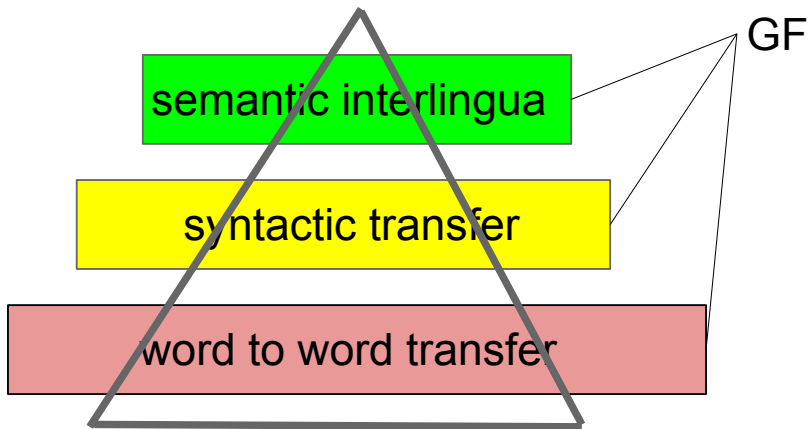


- Currently only context-free disambiguation model
- All alternatives are listed in probability order
- Work on context-dependent disambiguation is ongoing

| | | |
|----------------|-----------|------------------------|
| aim_at_V2 | bank_1_N | 6.067170072244503e-75 |
| aim_at_V2 | bank_2_N | 0.40476189961606895 |
| along_Prep | bank_1_N | 2.2727272727272742e-2 |
| account_for_V2 | weather_N | 7.679211977545736e-34 |
| account_for_V2 | time_1_N | 6.42858812965317e-34 |
| account_for_V2 | tense_N | 1.1176033841179318e-82 |

- There are still many errors in the dictionaries. English, Swedish and Bulgarian seems to be in the best shape.
- In some languages words are checked in Frequency Order

- 1 Introduction
- 2 Free Language
 - Resource Grammar
 - Resource Lexicon
 - Demo
- 3 Uncontrolled Language
 - Demo
- 4 Summary



Colors in Translation

what is your wife's name

vad heter din fru

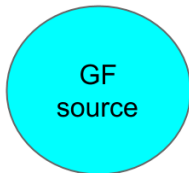
the vice president kicked the
bucket

skruvstädspresidenten
sparkade hinken

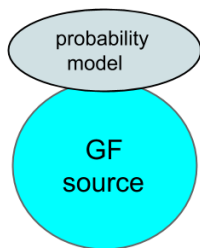
long time no see

lång tid nej ser

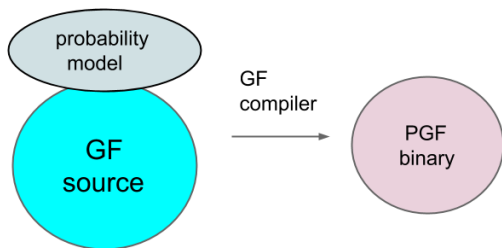
Architecture of the App



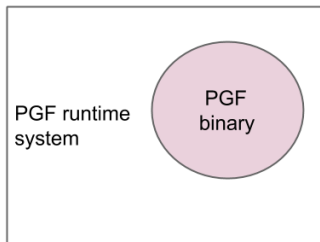
Architecture of the App



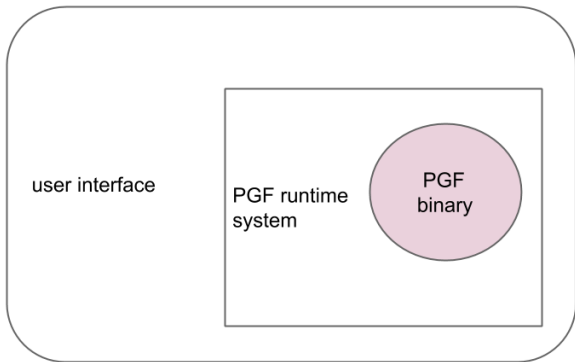
Architecture of the App



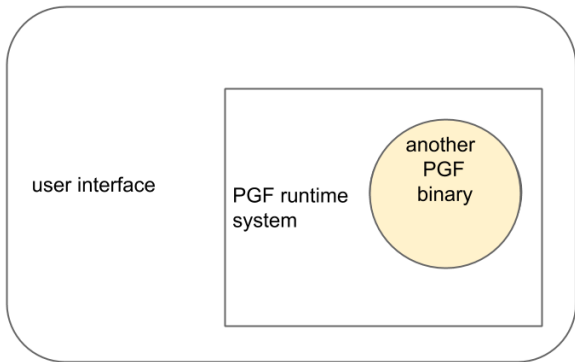
Architecture of the App



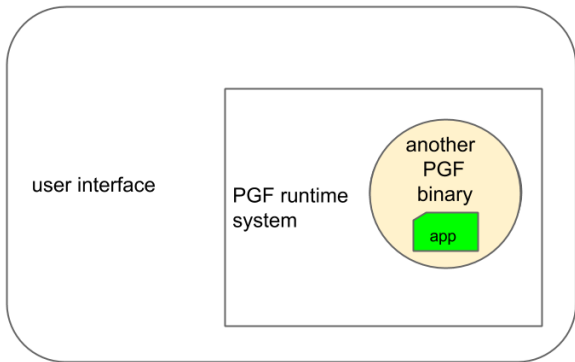
Architecture of the App



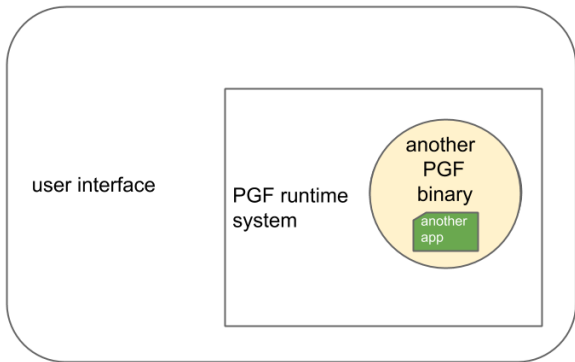
Architecture of the App



Architecture of the App



Architecture of the App



- 1 Introduction
- 2 Free Language
 - Resource Grammar
 - Resource Lexicon
 - Demo
- 3 Uncontrolled Language
 - Demo
- 4 Summary

Building the Spine

huvud

?3 = headBodypart

[1]

Building the Spine

huvud

```
?5 = showBodypartCommand ?3  
?3 = headBodypart
```

[1]

Building the Spine

huvud

?1 = commandPhrase ?5

?5 = showBodypartCommand ?3

?3 = headBodypart

[1]

Building the Spine

huvud

?2 = bleedBodypartStatement ?0 ?3

?2 = painBodypartStatement ?0 ?3

?1 = commandPhrase ?5

?5 = showBodypartCommand ?3

?3 = headBodypart

[1]

Building the Spine

huvud

?4 = whenQuestion ?2
?4 = whetherQuestion ?2
?1 = howlongHaveQuestionPhrase ?2
?1 = statementHaveNotPhrase ?2
?1 = statementHavePhrase ?2
?1 = statementHaveTimePhrase ?2 ?0
?1 = statementNotPhrase ?2
?1 = statementPhrase ?2
?1 = statementTimePhrase ?2 ?0
?2 = bleedBodypartStatement ?0 ?3
?2 = painBodypartStatement ?0 ?3
?1 = commandPhrase ?5
?5 = showBodypartCommand ?3
?3 = headBodypart

[1]

Building the Spine

huvud

?1 = questionHavePhrase ?4
?1 = questionHaveTimePhrase ?4 ?0
?1 = questionPhrase ?4
?4 = whenQuestion ?2
?4 = whetherQuestion ?2
?1 = howlongHaveQuestionPhrase ?2
?1 = statementHaveNotPhrase ?2
?1 = statementHavePhrase ?2
?1 = statementHaveTimePhrase ?2 ?0
?1 = statementNotPhrase ?2
?1 = statementPhrase ?2
?1 = statementTimePhrase ?2 ?0
?2 = bleedBodypartStatement ?0 ?3
?2 = painBodypartStatement ?0 ?3
?1 = commandPhrase ?5
?5 = showBodypartCommand ?3
?3 = headBodypart

[1]

Building the Spine

huvud

?1 = questionHavePhrase ?4
?1 = questionHaveTimePhrase ?4 ?0
?1 = questionPhrase ?4
?4 = whenQuestion ?2
?4 = whetherQuestion ?2
?1 = howlongHaveQuestionPhrase ?2
?1 = statementHaveNotPhrase ?2
?1 = statementHavePhrase ?2
?1 = statementHaveTimePhrase ?2 ?0
?1 = statementNotPhrase ?2
?1 = statementPhrase ?2
?1 = statementTimePhrase ?2 ?0
?2 = bleedBodypartStatement ?0 ?3
?2 = painBodypartStatement ?0 ?3
?1 = commandPhrase ?5
?5 = showBodypartCommand ?3
?3 = headBodypart

ont

[1] ?1 = thisWillHurtALittlePhrase [1]
?1 = thisWillHurtPhrase [1]
?1 = thisWillNotHurtPhrase [1]
?2 = painBodypartStatement ?0 ?0 [1]
?2 = painOrganStatement ?0 ?0 [1]
?3 = painSymptom [1]

Building the Spine

huvud

?1 = questionHavePhrase ?4
?1 = questionHaveTimePhrase ?4 ?0
?1 = questionPhrase ?4
?4 = whenQuestion ?2
?4 = whetherQuestion ?2
?1 = howlongHaveQuestionPhrase ?2
?1 = statementHaveNotPhrase ?2
?1 = statementHavePhrase ?2
?1 = statementHaveTimePhrase ?2 ?0
?1 = statementNotPhrase ?2
?1 = statementPhrase ?2
?1 = statementTimePhrase ?2 ?0
?2 = bleedBodypartStatement ?0 ?3
?2 = painBodypartStatement ?0 ?3
?1 = commandPhrase ?5
?5 = showBodypartCommand ?3
?3 = headBodypart

ont

[1] ?4 = whoSymptomQuestion ?3
?2 = symptomStatement ?0 ?3
?1 = thisWillHurtALittlePhrase [1]
?1 = thisWillHurtPhrase [1]
?1 = thisWillNotHurtPhrase [1]
?2 = painBodypartStatement ?0 ?0 [1]
?2 = painOrganStatement ?0 ?0 [1]
?3 = painSymptom [1]

Building the Spine

huvud

?1 = questionHavePhrase ?4
?1 = questionHaveTimePhrase ?4 ?0
?1 = questionPhrase ?4
?4 = whenQuestion ?2
?4 = whetherQuestion ?2
?1 = howlongHaveQuestionPhrase ?2
?1 = statementHaveNotPhrase ?2
?1 = statementHavePhrase ?2
?1 = statementHaveTimePhrase ?2 ?0
?1 = statementNotPhrase ?2
?1 = statementPhrase ?2
?1 = statementTimePhrase ?2 ?0
?2 = bleedBodypartStatement ?0 ?3
?2 = painBodypartStatement ?0 ?3
?1 = commandPhrase ?5
?5 = showBodypartCommand ?3
?3 = headBodypart

[1]

ont

?1 = howlongHaveQuestionPhrase ?2
?1 = statementHaveNotPhrase ?2
?1 = statementHavePhrase ?2
?1 = statementHaveTimePhrase ?2 ?0
?1 = statementNotPhrase ?2
?1 = statementPhrase ?2
?1 = statementTimePhrase ?2 ?0
?4 = whenQuestion ?2
?4 = whetherQuestion ?2
?4 = whoSymptomQuestion ?3
?2 = symptomStatement ?0 ?3
?1 = thisWillHurtALittlePhrase [1]
?1 = thisWillHurtPhrase [1]
?1 = thisWillNotHurtPhrase [1]
?2 = painBodypartStatement ?0 ?0 [1]
?2 = painOrganStatement ?0 ?0 [1]
?3 = painSymptom [1]

Building the Spine

huvud

?1 = questionHavePhrase ?4
?1 = questionHaveTimePhrase ?4 ?0
?1 = questionPhrase ?4
?4 = whenQuestion ?2
?4 = whetherQuestion ?2
?1 = howlongHaveQuestionPhrase ?2
?1 = statementHaveNotPhrase ?2
?1 = statementHavePhrase ?2
?1 = statementHaveTimePhrase ?2 ?0
?1 = statementNotPhrase ?2
?1 = statementPhrase ?2
?1 = statementTimePhrase ?2 ?0
?2 = bleedBodypartStatement ?0 ?3
?2 = painBodypartStatement ?0 ?3
?1 = commandPhrase ?5
?5 = showBodypartCommand ?3
?3 = headBodypart

[1]

ont

?1 = questionHavePhrase ?4
?1 = questionHaveTimePhrase ?4 ?0
?1 = questionPhrase ?4
?1 = howlongHaveQuestionPhrase ?2
?1 = statementHaveNotPhrase ?2
?1 = statementHavePhrase ?2
?1 = statementHaveTimePhrase ?2 ?0
?1 = statementNotPhrase ?2
?1 = statementPhrase ?2
?1 = statementTimePhrase ?2 ?0
?4 = whenQuestion ?2
?4 = whetherQuestion ?2
?4 = whoSymptomQuestion ?3
?2 = symptomStatement ?0 ?3
?1 = thisWillHurtALittlePhrase [1]
?1 = thisWillHurtPhrase [1]
?1 = thisWillNotHurtPhrase [1]
?2 = painBodypartStatement ?0 ?0 [1]
?2 = painOrganStatement ?0 ?0 [1]
?3 = painSymptom [1]

Merge

| | | |
|------------------------------------|-----------------------------------|-----|
| ?1 = howlongHaveQuestionPhrase ?2 | ?1 = thisWillHurtPhrase | [1] |
| ?1 = statementHaveNotPhrase ?2 | ?1 = thisWillNotHurtPhrase | [1] |
| ?1 = statementHavePhrase ?2 | ?2 = bleedBodypartStatement ?0 ?3 | |
| ?1 = statementHaveTimePhrase ?2 ?0 | ?2 = painBodypartStatement ?0 ?3 | [1] |
| ?1 = statementNotPhrase ?2 | ?2 = painOrganStatement ?0 ?0 | [1] |
| ?1 = statementPhrase ?2 | ?2 = symptomStatement ?0 ?4 | |
| ?1 = statementTimePhrase ?2 ?0 | ?3 = headBodypart | [1] |
| ?1 = questionHavePhrase ?5 | ?4 = painSymptom | [1] |
| ?1 = questionHaveTimePhrase ?5 ?0 | ?5 = whenQuestion ?2 | |
| ?1 = questionPhrase ?5 | ?5 = whetherQuestion ?2 | |
| ?1 = commandPhrase ?6 | ?5 = whoSymptomQuestion ?4 | |
| ?1 = thisWillHurtALittlePhrase | ?6 = showBodypartCommand ?3 | |

Merge

| | | |
|------------------------------------|-----------------------------------|-----|
| ?1 = howlongHaveQuestionPhrase ?2 | ?1 = thisWillHurtPhrase | [1] |
| ?1 = statementHaveNotPhrase ?2 | ?1 = thisWillNotHurtPhrase | [1] |
| ?1 = statementHavePhrase ?2 | ?2 = bleedBodypartStatement ?0 ?3 | [1] |
| ?1 = statementHaveTimePhrase ?2 ?0 | ?2 = painBodypartStatement ?0 ?3 | [2] |
| ?1 = statementNotPhrase ?2 | ?2 = painOrganStatement ?0 ?0 | [1] |
| ?1 = statementPhrase ?2 | ?2 = symptomStatement ?0 ?4 | |
| ?1 = statementTimePhrase ?2 ?0 | ?3 = headBodypart | [1] |
| ?1 = questionHavePhrase ?5 | ?4 = painSymptom | [1] |
| ?1 = questionHaveTimePhrase ?5 ?0 | ?5 = whenQuestion ?2 | |
| ?1 = questionPhrase ?5 | ?5 = whetherQuestion ?2 | |
| ?1 = commandPhrase ?6 | ?5 = whoSymptomQuestion ?4 | |
| ?1 = thisWillHurtALittlePhrase | ?6 = showBodypartCommand ?3 | [1] |

Merge

| | | |
|------------------------------------|-----------------------------------|-----|
| ?1 = howlongHaveQuestionPhrase ?2 | ?1 = thisWillHurtPhrase | [1] |
| ?1 = statementHaveNotPhrase ?2 | ?1 = thisWillNotHurtPhrase | [1] |
| ?1 = statementHavePhrase ?2 | ?2 = bleedBodypartStatement ?0 ?3 | [1] |
| ?1 = statementHaveTimePhrase ?2 ?0 | ?2 = painBodypartStatement ?0 ?3 | [2] |
| ?1 = statementNotPhrase ?2 | ?2 = painOrganStatement ?0 ?0 | [1] |
| ?1 = statementPhrase ?2 | ?2 = symptomStatement ?0 ?4 | [1] |
| ?1 = statementTimePhrase ?2 ?0 | ?3 = headBodypart | [1] |
| ?1 = questionHavePhrase ?5 | ?4 = painSymptom | [1] |
| ?1 = questionHaveTimePhrase ?5 ?0 | ?5 = whenQuestion ?2 | |
| ?1 = questionPhrase ?5 | ?5 = whetherQuestion ?2 | |
| ?1 = commandPhrase ?6 | ?5 = whoSymptomQuestion ?4 | [1] |
| ?1 = thisWillHurtALittlePhrase [1] | ?6 = showBodypartCommand ?3 | [1] |

Merge

| | | |
|------------------------------------|-----------------------------------|-----|
| ?1 = howlongHaveQuestionPhrase ?2 | ?1 = thisWillHurtPhrase | [1] |
| ?1 = statementHaveNotPhrase ?2 | ?1 = thisWillNotHurtPhrase | [1] |
| ?1 = statementHavePhrase ?2 | ?2 = bleedBodypartStatement ?0 ?3 | [1] |
| ?1 = statementHaveTimePhrase ?2 ?0 | ?2 = painBodypartStatement ?0 ?3 | [2] |
| ?1 = statementNotPhrase ?2 | ?2 = painOrganStatement ?0 ?0 | [1] |
| ?1 = statementPhrase ?2 | ?2 = symptomStatement ?0 ?4 | [1] |
| ?1 = statementTimePhrase ?2 ?0 | ?3 = headBodypart | [1] |
| ?1 = questionHavePhrase ?5 | ?4 = painSymptom | [1] |
| ?1 = questionHaveTimePhrase ?5 ?0 | ?5 = whenQuestion ?2 | [2] |
| ?1 = questionPhrase ?5 | ?5 = whetherQuestion ?2 | [2] |
| ?1 = commandPhrase ?6 | ?5 = whoSymptomQuestion ?4 | [1] |
| ?1 = thisWillHurtALittlePhrase [1] | ?6 = showBodypartCommand ?3 | [1] |

Merge

| | | | |
|------------------------------------|-----|-----------------------------------|-----|
| ?1 = howlongHaveQuestionPhrase ?2 | [2] | ?1 = thisWillHurtPhrase | [1] |
| ?1 = statementHaveNotPhrase ?2 | [2] | ?1 = thisWillNotHurtPhrase | [1] |
| ?1 = statementHavePhrase ?2 | [2] | ?2 = bleedBodypartStatement ?0 ?3 | [1] |
| ?1 = statementHaveTimePhrase ?2 ?0 | [2] | ?2 = painBodypartStatement ?0 ?3 | [2] |
| ?1 = statementNotPhrase ?2 | [2] | ?2 = painOrganStatement ?0 ?0 | [1] |
| ?1 = statementPhrase ?2 | [2] | ?2 = symptomStatement ?0 ?4 | [1] |
| ?1 = statementTimePhrase ?2 ?0 | [2] | ?3 = headBodypart | [1] |
| ?1 = questionHavePhrase ?5 | [2] | ?4 = painSymptom | [1] |
| ?1 = questionHaveTimePhrase ?5 ?0 | [2] | ?5 = whenQuestion ?2 | [2] |
| ?1 = questionPhrase ?5 | [2] | ?5 = whetherQuestion ?2 | [2] |
| ?1 = commandPhrase ?6 | [1] | ?5 = whoSymptomQuestion ?4 | [1] |
| ?1 = thisWillHurtALittlePhrase | [1] | ?6 = showBodypartCommand ?3 | [1] |

Filter

| | | | |
|------------------------------------|-----|----------------------------------|-----|
| ?1 = howlongHaveQuestionPhrase ?2 | [2] | | |
| ?1 = statementHaveNotPhrase ?2 | [2] | | |
| ?1 = statementHavePhrase ?2 | [2] | | |
| ?1 = statementHaveTimePhrase ?2 ?0 | [2] | ?2 = painBodypartStatement ?0 ?3 | [2] |
| ?1 = statementNotPhrase ?2 | [2] | | |
| ?1 = statementPhrase ?2 | [2] | | |
| ?1 = statementTimePhrase ?2 ?0 | [2] | ?3 = headBodypart | [1] |
| ?1 = questionHavePhrase ?5 | [2] | ?4 = painSymptom | [1] |
| ?1 = questionHaveTimePhrase ?5 ?0 | [2] | ?5 = whenQuestion ?2 | [2] |
| ?1 = questionPhrase ?5 | [2] | ?5 = whetherQuestion ?2 | [2] |
| | | ?6 = showBodypartCommand ?3 | [1] |

Extraction

```
howlongHaveQuestionPhrase (painBodypartStatement ? headBodypart)
statementHaveNotPhrase (painBodypartStatement ? headBodypart)
statementHavePhrase (painBodypartStatement ? headBodypart)
statementHaveTimePhrase (painBodypartStatement ? headBodypart) ?
statementNotPhrase (painBodypartStatement ? headBodypart)
...
questionHavePhrase (whenQuestion (painBodypartStatement ? headBodypart))
questionHavePhrase (whetherQuestion (painBodypartStatement ? headBodypart))
questionPhrase (whenQuestion (painBodypartStatement ? headBodypart))
questionPhrase (whetherQuestion (painBodypartStatement ? headBodypart))
...
```

- Query: huvud ont
- Suggestions:
 - hur länge har ?0 haft **ont** i **huvudet**
 - ?0 har inte haft **ont** i **huvudet**
 - ?0 har haft **ont** i **huvudet**
 - ...

Ranking

| | |
|---|------------|
| howlongHaveQuestionPhrase (painBodypartStatement ? headBodypart) | [0.064651] |
| statementHaveNotPhrase (painBodypartStatement ? headBodypart) | [0.076847] |
| statementHavePhrase (painBodypartStatement ? headBodypart) | [0.109109] |
| statementHaveTimePhrase (painBodypartStatement ? headBodypart) ? | [0.076249] |
| statementNotPhrase (painBodypartStatement ? headBodypart) | [0.109109] |
| ... | |
| questionHavePhrase (whenQuestion (painBodypartStatement ? headBodypart)) | [0.071202] |
| questionHavePhrase (whetherQuestion (painBodypartStatement ? headBodypart)) | [0.100887] |
| questionPhrase (whenQuestion (painBodypartStatement ? headBodypart)) | [0.145479] |
| questionPhrase (whetherQuestion (painBodypartStatement ? headBodypart)) | [0.207390] |
| ... | |

Ranking

| | |
|--|-------------------|
| howlongHaveQuestionPhrase (painBodypartStatement ? headBodypart) | [0.064651] |
| statementHaveNotPhrase (painBodypartStatement ? headBodypart) | [0.076847] |
| statementHavePhrase (painBodypartStatement ? headBodypart) | [0.109109] |
| statementHaveTimePhrase (painBodypartStatement ? headBodypart) ? | [0.076249] |
| statementNotPhrase (painBodypartStatement ? headBodypart) | [0.109109] |
| ... | |
| questionHavePhrase (whenQuestion (painBodypartStatement ? headBodypart)) | [0.071202] |
| questionHavePhrase (whetherQuestion (painBodypartStatement ? headBodypart)) | [0.100887] |
| questionPhrase (whenQuestion (painBodypartStatement ? headBodypart)) | [0.145479] |
| questionPhrase (whetherQuestion (painBodypartStatement ? headBodypart)) | [0.207390] |
| ... | |

- 1 Introduction
- 2 Free Language
 - Resource Grammar
 - Resource Lexicon
 - Demo
- 3 Uncontrolled Language
 - Demo
- 4 Summary

- Resource Grammar/Lexicon
- Statistical Parsing
- Approximate Search in a Controlled Language