Contextual Models for Tree Disambiguation

Maximilian Ludvigsson Oscar Kalldal

Chalmers University of Technology

Presentation, November 2017

Outline

Problem

Tree Disambiguation

Probabilistic Parsing

Estimation

Model
- Example
- Expanding to syntactic bigram

Evaluation

Conclusion

Problem

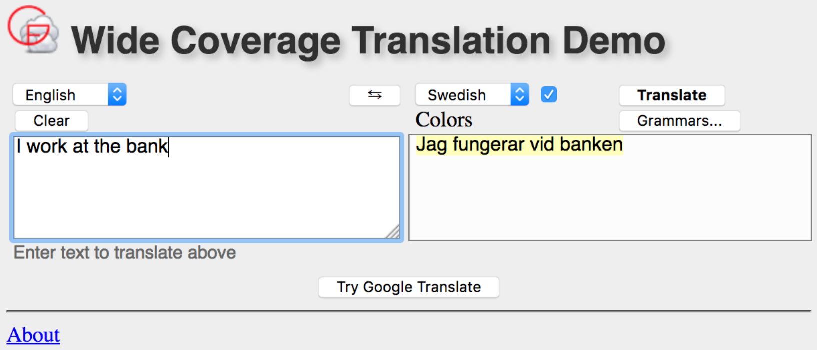
Why?

- The GF parser fails: “I work at the bank”
- Reranking to disambiguate

How?

- Language independent
- Easy to extend

Problem


Wide Coverage Translation Demo
English
Swedish
Translate
Clear
Colors
Grammars...
I work at the bank
Jag fungerar vid banken
Enter text to translate above
Try Google Translate
About

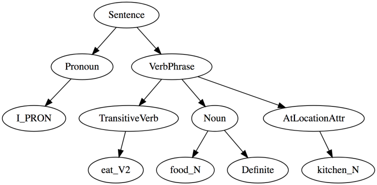
Ambiguity

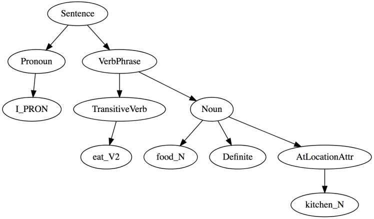
Two kinds of ambiguity

- Syntactic
- Lexical

Syntactic Ambiguity

"I eat the food in the kitchen"





(1)

wo	zai	chufang	chi	fan
I	[at loc.]	kitchen	eat	food.

'I eat the food in the kitchen'

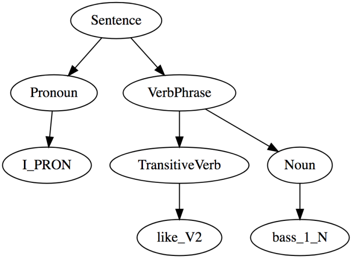
(2)

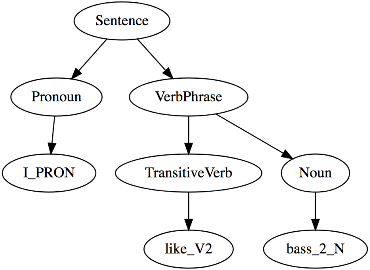
wo	chi	zai	chufang	de	fan
I	eat	[at loc.]	kitchen	[attr.]	food

'I eat the food in the kitchen'

Lexical Ambiguity

"I like bass"

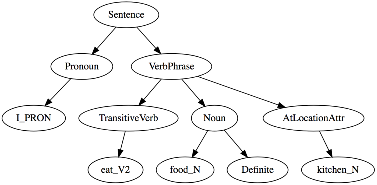


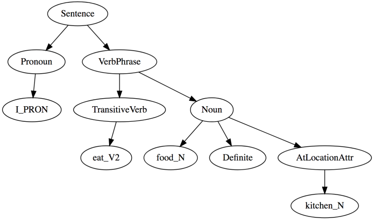


(3) Jag gillar aborre
I like bass
'I like bass'

(4) Jag gillar bas
I like bass
'I like bass'

Tree probabilities

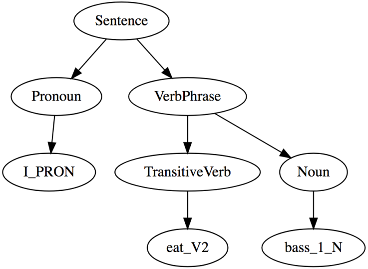


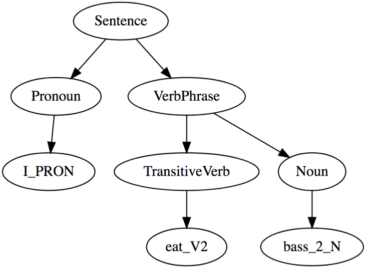


Higher probability

Lower probability

Tree Probabilities

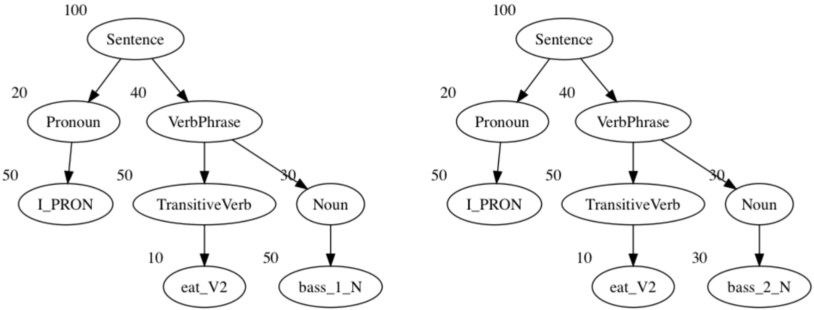




High probability **Very low probability**

Context-Free Model

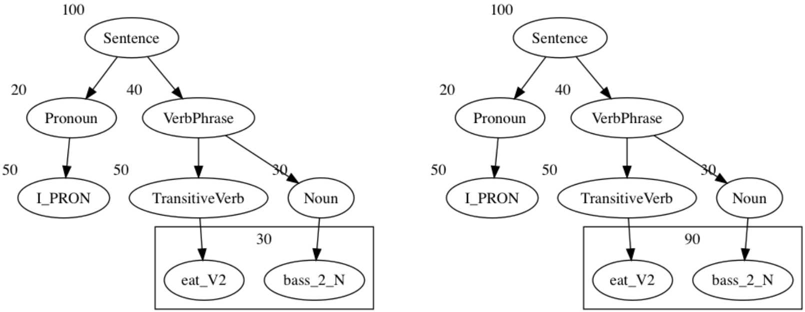
Log-probability score for each node



Lower probability! **Higher probability!**

Context-Aware Model

Probability scores accounts for potentially long range interactions, eating fish is probable but eating instruments is not!

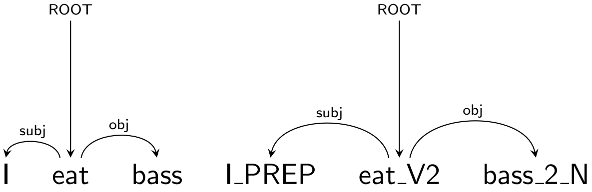


This is what we want!

Finding Relevant Interactions

- (eat, bass) carries strong signal
- (I, bass) intuitively carries much weaker signal
- How do we find the right interactions?
- Solution: lexicalize on head in UD tree

UD and GF2UD



- Universal Dependencies - language independent dependency grammars
- Deterministic mapping from AST to *Abstract Dependency Tree* in UD style
- Finds a new tree over only the constant functions
- Lexicalization: condition probability score on parent in ADT

Assigning the correct probabilities

- Estimate frequencies from corpora or treebanks
- Many UD-resources, gold-treebanks, high quality parsers
- These resources are still lexically ambiguous!
- Solution: estimate probabilities from resources in multiple languages and use expectation maximization

Example

Toy problem

Languages	English, Swedish
English Vocabulary	*play, gamble*
Swedish Vocabulary	*leka, spela*
Latent Space	*child_play, instrument_play, gamble*
	$\mathbf{1}$ $\mathbf{2}$ $\mathbf{3}$

Dictionary:

$$\text{play} = \{\mathbf{1}, \mathbf{2}\}$$
$$\text{gamble} = \{\mathbf{3}\}$$

$$\text{spela} = \{\mathbf{2}, \mathbf{3}\}$$
$$\text{leka} = \{\mathbf{1}\}$$

Example

Corpus

Latent Space	*child_play*,	*instrument_play*,	*gamble*
	1	**2**	**3**

English Corpus play play play gamble

Swedish Corpus spela spela leka leka

Example

Gold

Latent Space	*child_play*,	*instrument_play*,	*gamble*
	1	**2**	**3**

It would be easy if we knew the underlying meaning (maximum likelihood)

English Corpus	play	play	play	gamble
	1	**1**	**2**	**3**
Swedish Corpus	leka	leka	spela	spela
	1	**1**	**2**	**3**

$$P(\mathbf{1}) = 0.5$$
$$P(\mathbf{2}) = 0.25$$
$$P(\mathbf{3}) = 0.25$$

Algorithm

Assuming unique linearizations

Expectation step:

$$\hat{c}_{sij}^{t} = \frac{c_{si}\pi_j^{t-1}}{\sum_{k:y_k \in D_{si}} \pi_k^{t-1}}$$

Maximization step:

$$\pi_j^t = \sum_{s,i} \frac{\hat{c}_{sij}^{t}}{N}$$

Example

First iteration

English Corpus	play	play	play	gamble
	$\{\mathbf{1,2}\}$	$\{\mathbf{1,2}\}$	$\{\mathbf{1,2}\}$	$\mathbf{3}$
Swedish Corpus	spela	spela	leka	leka
	$\{\mathbf{2,3}\}$	$\{\mathbf{2,3}\}$	$\mathbf{1}$	$\mathbf{1}$

- Initial guess: $P_0(\mathbf{1}) = P_0(\mathbf{2}) = P_0(\mathbf{3}) = 0.33$
- Expectation:

$$c_1 = 1.5 + 2 = 3.5$$
$$c_2 = 1.5 + 1 = 2.5$$
$$c_3 = 1 + 1 = 2$$

- Maximization:

$$P_1(\mathbf{1}) = c_1/c_{\text{total}} = 3.5/8 \approx 0.44$$
$$P_1(\mathbf{2}) = c_2/c_{\text{total}} = 2.5/8 \approx 0.31$$
$$P_1(\mathbf{3}) = c_3/c_{\text{total}} = 2/8 \approx 0.25$$

Example

Second iteration

English Corpus	play	play	play	gamble
	$\{\mathbf{1,2}\}$	$\{\mathbf{1,2}\}$	$\{\mathbf{1,2}\}$	$\mathbf{3}$
Swedish Corpus	spela	spela	leka	leka
	$\{\mathbf{2,3}\}$	$\{\mathbf{2,3}\}$	$\mathbf{1}$	$\mathbf{1}$

- Expectation:

$$c_1 = 1.8 + 2 = 3.8$$
$$c_2 = 1.2 + 1.1 = 2.3$$
$$c_3 = 1 + 0.9 = 1.9$$

- Maximization:

$$P_2(\mathbf{1}) = c_1/c_{\text{total}} = 3.8/8 \approx 0.48$$
$$P_2(\mathbf{2}) = c_2/c_{\text{total}} = 2.3/8 \approx 0.29$$
$$P_2(\mathbf{3}) = c_3/c_{\text{total}} = 1.9/8 \approx 0.24$$

Example

10th iteration

- A few more iterations:

$$P_{10}(\mathbf{1}) \approx 0.497$$
$$P_{10}(\mathbf{2}) \approx 0.256$$
$$P_{10}(\mathbf{3}) \approx 0.247$$

Expanding to bigram

Latent Space $\{\underset{\mathbf{1}}{\textit{play\_1}},\ \underset{\mathbf{2}}{\textit{play\_2}}\} \times \{\underset{\mathbf{3}}{\textit{ball}},\ \underset{\mathbf{4}}{\textit{flute}}\}$

$$\text{(play, ball)} = \{(\mathbf{1}, \mathbf{3}), (\mathbf{2}, \mathbf{3})\}$$
$$\text{(play, flute)} = \{(\mathbf{1}, \mathbf{4}), (\mathbf{2}, \mathbf{4})\}$$

Expanding to bigram

Latent Space $\{\underset{\mathbf{1}}{\textit{play\_1}},\ \underset{\mathbf{2}}{\textit{play\_2}}\} \times \{\underset{\mathbf{3}}{\textit{game\_1}},\ \underset{\mathbf{4}}{\textit{game\_2}}\}$

$$\text{(play, game)} = \{(\mathbf{1},\mathbf{3}),(\mathbf{2},\mathbf{3}),(\mathbf{1},\mathbf{4}),(\mathbf{2},\mathbf{4})\}$$

Wordnet dictionaries

- Over 117 000 synsets, 150 000 words in english wordnet
- Not all synsets have unique linearization
- Requires us to store both possible linearizations for every synset as well ass possible synsets for each word
- Requires a lot of memory for bigrams
- Just the Noun-Verb pairs takes more memory than our computers can handle

Evaluation

Qualitative: Performance on a set of hand-crafted test sentences.

Quantitative: WSD performance on a large set of annotated sentences.

Evaluation

Qualitative

"He works at the bank"

GF parser	Rerank	Interpretation
26.398	16.542	he *labors* at the *bank institution*
26.398	45.383	he *functions* at the *bank institution*
29.458	16.802	he *labors* at the *river bank*
29.458	37.562	he *functions* at the *river bank*

Evaluation

Quantitative

Trainomatic data

Model	**Success rate**	**OOV**
7 languages	75%	-
No English	49%	65%
Only English	67%	12%
Ambiguous sentences	55 114	
Total # of sentences	834 468	

Wordnet examples

Model	**Success rate**	**OOV**
7 languages	72%	-
No English	42%	33%
Only English	66%	8%
Ambiguous sentences	739	
Total # of sentences	48 247	

Real World Challenges

Parameter Space

- 46 000 GF unigrams
- 120 000 Wordnet unigrams
- 17 000 000 GF bigrams
- 225 000 000 Wordnet bigrams

Data

- CoNLL 2017 Shared Task
- $\approx$ 140 GB gzipped data
- more than 10^9 sentences
- English, Dutch, Bulgarian, Finnish, French, Swedish, Hindi

Summary

Strengths

- Identifies simple relations
- Generalizes across languages
- Explainable

Weaknesses

- Large parameter space
- OOV
- Dependent on a good dictionary

Summary and Outlook

We have...

- Implemented the EM-algorithm
- Formalized the underlying mathematical model
- Gotten it to work with large scale data sources
- Adapted the model to Wordnet (non-unique linearization)

Outlook

- Smoothing (Kneser-Ney)
- N-gram composition
- Higher order models
- Wordnet-graph based reduction of parameters

Thanks

Special thanks to...

- Prasanth
- Krasimir

Links

Code http://github.com/okalldal/gf-exjobb

Blog http://github.com/okalldal/gf-exjobb/wiki/Journal