Discriminative features in Reversible Stochastic Attribute-Value Grammars

Daniël de Kok, University of Groningen

EMNLP 2011 Workshop on Language Generation and Evaluation

Preview

- Preferences should be shared between parsing and generation, if we want a parser to be a able to recover the meaning that was the input of a generator.
- Reversible Stochastic Attribute-Value Grammars aim to integrate parsing and generation in one model.

Message #1: Reversible Stochastic Attribute-Value grammars are truly reversible.

Preview

Message #2: We can (and should) understand statistical models.

Attribute-value grammar

- Representation of lexical items and grammar rules as attribute-value structures
- Construction of derivations via unification
- Since unification is associate and commutative, attribute-value grammar can be used in two directions (parsing and generation)

Ambiguity

- Parsing a sentence can give multiple readings, not all equally likely
- Generating from a logical form can give multiple realizations, not all equally fluent

Solving ambiguity

- Models for parse disambiguation
- Models for fluency ranking
- For attribute-value grammar: feature-based models, such as maximum entropy models
- *Stochastic Attribute-Value Grammar* (SAVG)
- State of the art systems: separate models for parse disambiguation and fluency ranking
- Directional models

Reversibility

- We want:
  - A parser that can recover the meaning that was the input to a generator
  - A generator that can produce the sentence that was the input of a parser
- If not, communication will be difficult
- Consequently, preferences in parsing and generation should be shared

Subject/object fronting in Dutch

Consider the two possible readings of the sentence *Jan zag de man* (*Jan saw the man*):

- $[\textit{Jan}]_{su}$ *zag* $[\textit{de man}]_{obj}$
- $[\textit{Jan}]_{obj}$ *zag* $[\textit{de man}]_{su}$

Subject fronting is preferred in Dutch, consequently:

- In parse disambiguation we prefer reading of a fronted NP as a subject
- In fluency ranking we prefer realizations that have a fronted subject NP

Reversible models

- So, why use separate models for parse disambiguation and fluency ranking?
- Use one model for both tasks
- Reversible SAVG (De Kok et al., 2011)
- Shares preferences are shared between parsing and generation
- Performance of RSAVG does not differ significantly from models specific to parse disambiguation and fluency ranking

Reversible SAVG

$$p(d|c) = \frac{1}{Z(c)} exp \sum_i w_i f_i(c, d) \quad (1)$$

- Probability of a derivation d, given a set of constraints c
- These constraints are formed by the input (a sentence or logical form)
- During training a weight w_i is estimated for each feature f_i

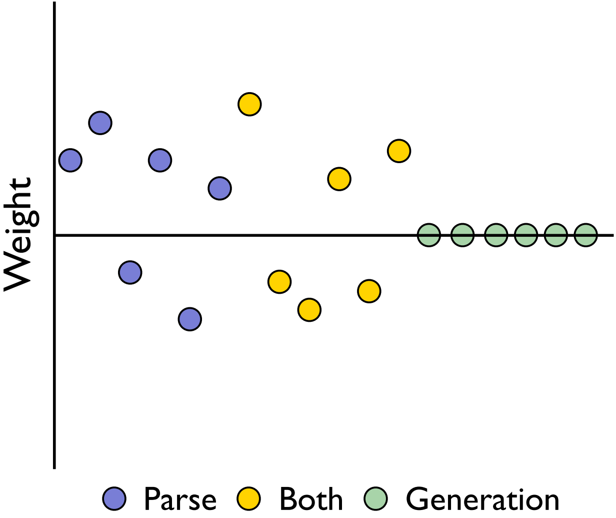
Three kinds of features

Features that are active during:

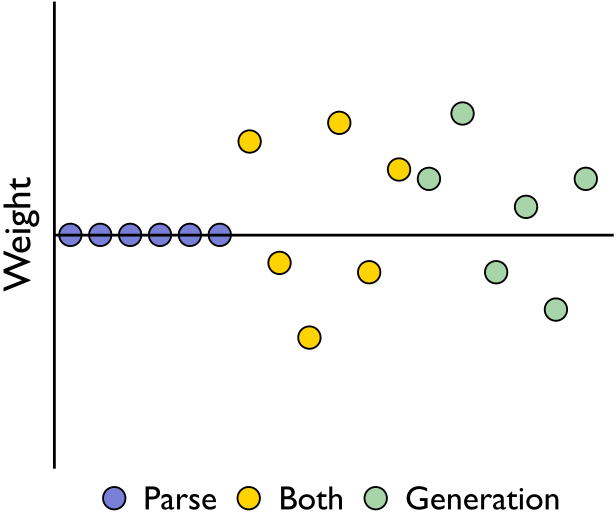
1. parsing
2. generation
3. both tasks

Uniform model

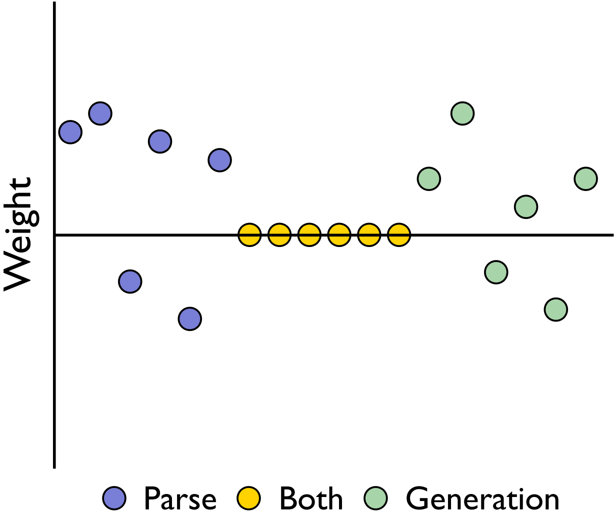
Disambiguation model



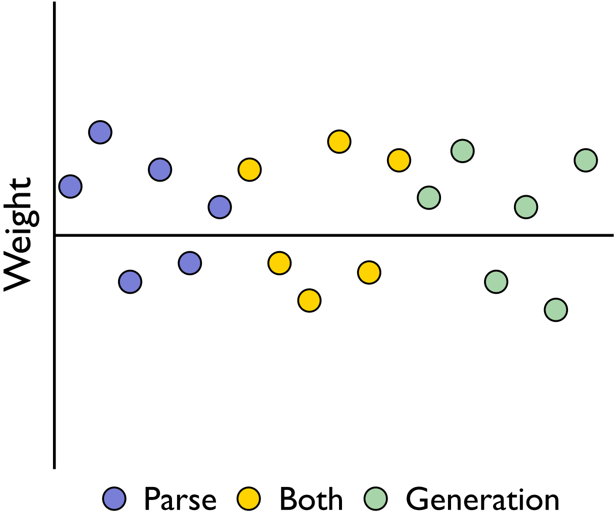
Fluency ranking model



Reversible model?



Or perhaps?



Research question

- Do reversible models use features used in both directions?
- If not, the model is not truly reversible

Methodology

- Find discriminative features in directional and reversible models using feature selection
- Calculate the contributions of the most discriminative features to the model
- Compare features by class, to detects shifts in feature use in reversible models, compared to directional models

Feature selection

- A good feature selection method (De Kok, 2010) should be able to remove:
  - Features that change of value sporadically
  - Features that correlate strongly with other features
  - Features with values that do not correlate with the ranking or classification
- For this experiment: a ranking of features

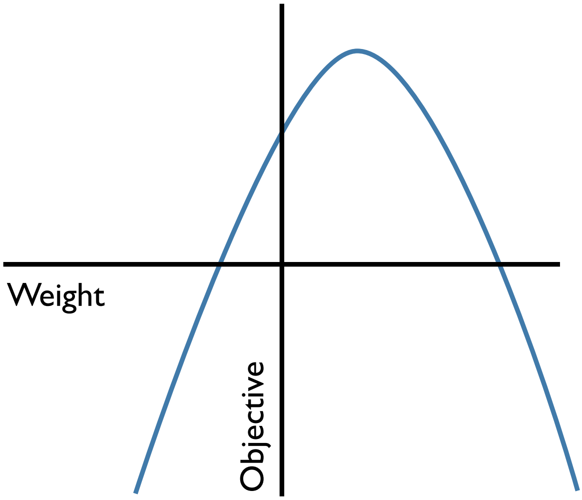
Selection methods

- Which selection method should be used?
- Three candidates that use maximum entropy modeling:
  - Grafting (Perkins et al., 2003)
  - Grafting-light (Zhu et al., 2010)
  - Gain-informed selection (Berger et al., 1996; De Kok, 2010)

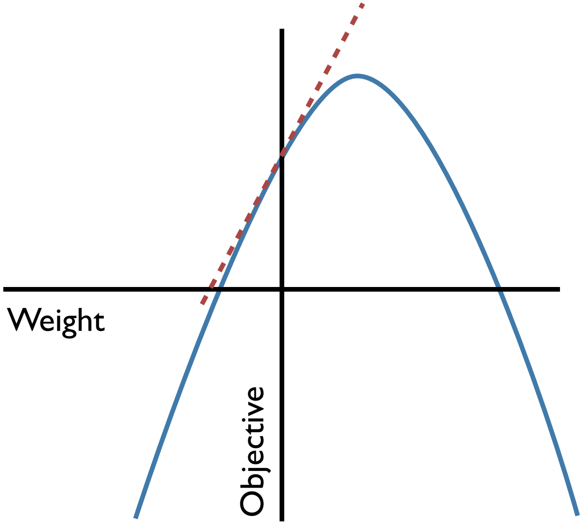
Grafting

1. Start with a uniform model
2. Pick the unselected feature with the highest gradient given the current model
3. Optimize the weights of selected features
4. Goto step 2, unless the threshold is reached

Grafting



Grafting



Grafting-light

- Same procedure as grafting
- Rather than performing a full optimization of the weights of selected features, perform one step of gradient descent

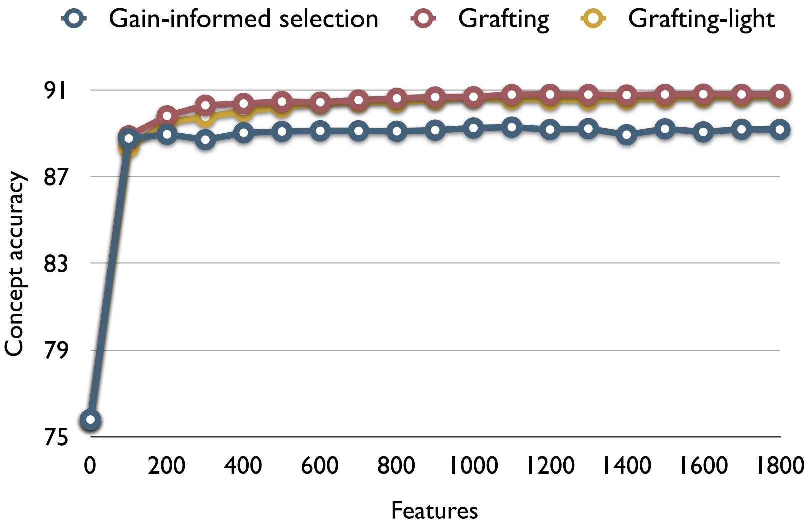
Gain-informed selection

1. Start with a uniform model
2. Pick the feature which provides the largest decrease of the objective function, given the current model
3. Optimize the weights of selected features
4. Goto step 2, unless the threshold is reached

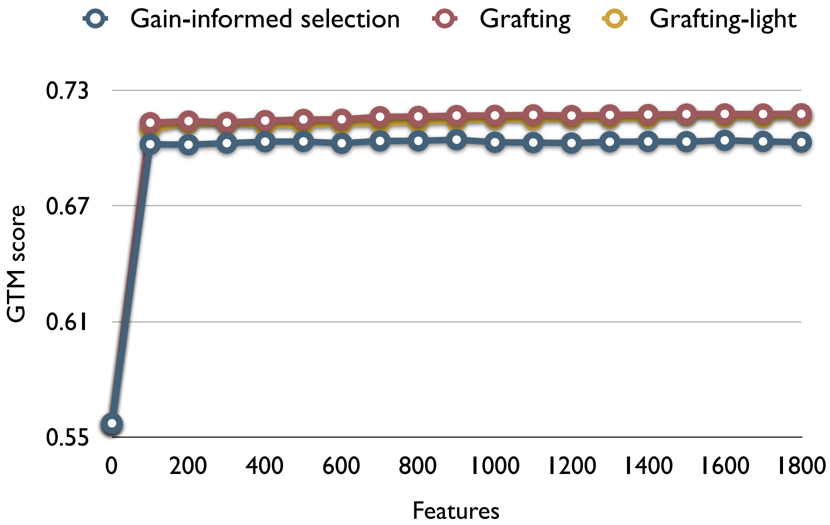
Evaluation

- Evaluated in the context of the Alpino parser and generator for Dutch (Van Noord, 2006; De Kok and Van Noord, 2010)
- Training: cdbl-part of the Eindhoven newspaper corpus (syntactic annotations from the Alpino Treebank)
- Evaluation: part of the Trouw 2001 newspaper (syntactic annotations from LASSY, part WR-P-P-H)
- Features before selection: 303872 (cutoff-2: 25578)

Parse disambiguation



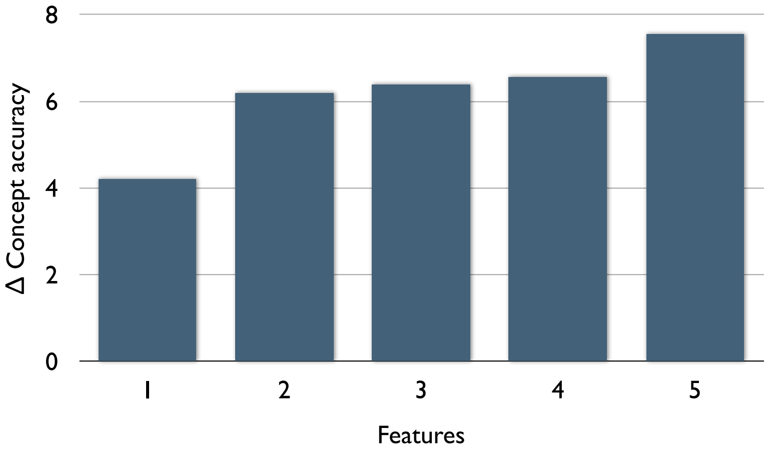
Fluency ranking



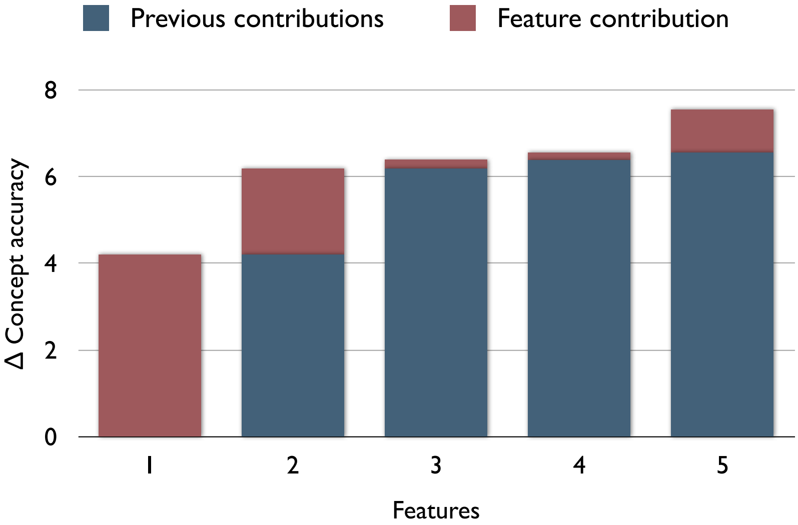
Our method of choice

- Grafting
- (If time is an issue, use grafting-light)

Feature contributions



Feature contributions



Feature contributions

- If e is an evaluation function and F a model, we can calculate the contribution of the i^{th} feature: $e(F_{0..i}) - e(F_{0..i-1})$
- If we select n features in total, then the overall improvement is: $e(F_{0..n}) - e(F_0)$
- Consequently, we can calculate the contribution of a feature to a model:

$$c(f_i) = \frac{e(p_{0..i}) - e(p_{0..i-1})}{e(p_{0..n}) - e(p_0)} \tag{2}$$

Feature classes

- We divide the features in the following classes:
  - Dependency (parsing)
  - Lexical (parsing)
  - N-gram (generation)
  - Rule (both)
  - Syntactic (both)
- We then calculate per-class feature contributions of the 300 most discriminative features

Per-class contributions in parse disambiguation

Class	**Directional**	**Reversible**
Dependency	21.53	13.35
Lexical	33.68	32.62
N-gram	0.00	0.00
Rule	37.61	47.35
Syntactic	7.04	6.26

Per-class contributions in fluency ranking

Class	Directional	Reversible
Dependency	0.00	0.00
Lexical	0.00	0.00
N-gram	81.39	79.89
Rule	14.15	15.75
Syntactic	3.66	4.39

Conclusions

- Grafting is the most effective selection method among the candidates for this task
- Models can be compressed enormously using feature selection, with very little loss in accuracy
- RSAVGs rely on features that are used in parsing and generation, even more than directional models

Thank you!