A chart generator for the Dutch Alpino grammar

Daniël de Kok and Gertjan van Noord

June 10, 2009

Introduction

- Parsing: determining the grammatical structure of a sentence.
- Semantics: a parser can build a representation of meaning (semantics) as a side-effect of parsing a sentence.
- Generation: building natural language realizations representing given semantics.

Applications

- Checking a grammar: if a grammar is too permissive, using it with a generator will create ungrammatical sentences.
- Sentence fusion: combining the semantics of two (or more) sentences.
- Sentence compression: removing non-salient elements of a sentence.
- Machine translation: generating a sentence in a different language (interlingua or transfer-based MT).

Topics

- Description of the input formalism
- The Alpino chart generator
- Fluency ranking

Alpino chart generator

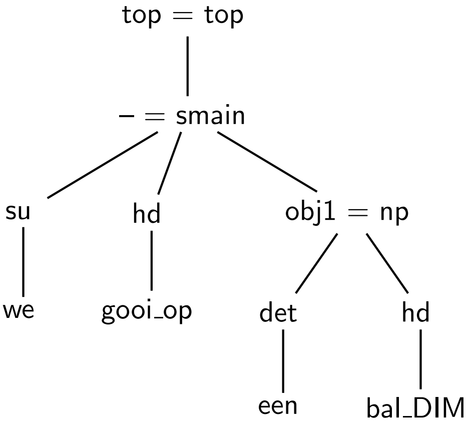
- Alpino is a wide-coverage parser for Dutch, with a lexicalized grammar in the tradition of HPSG.
- In the context of the STEVIN DAISY project, natural language generation components are being developed, such as a chart generator and models for fluency ranking.

Dependency trees

- The Alpino generator accepts dependency trees (DTs) that describe the grammatical dependency relations between lexical nodes, and the constituent categories dominating over lexical nodes.
- No information about the word order.
- Lexical nodes specify:
  - The sense (the word root, plus possibly some additional information to select a specific reading).
  - An Alpino part of speech tag.
  - Attributes that are used to specify additional requirements, such as the tense of a verb or the number of a noun.

Dependency trees (2)

wij gooiden een balletje op (literal: *we threw a (small) ball upwards*)



Lexical nodes

Sense	POS	Attributes
we	pron	
gooi_op	verb	tense='past'
een	det	
bal_DIM	noun	

Generates:

```
we gooiden een balletje op
we gooiden 'n balletje op
een balletje gooiden we op
'n balletje gooiden we op
```

Usefulness of dependency trees

Are dependency trees good enough for applications where generation is useful?

- Sentence fusion: Marsi and Krahmer (2005)
- Sentence compression: McDonald (2006), De Kok (2008, Ma Thesis)
- Machine translation: Dekang Lin (2006), Alshawi et al. (2000)

Generating from Alpino test suites

- We have produced DTs for most sentences in the Alpino test suite.
- Realizations can be generated for most DTs:

Suite	**Sentences**	$>=1$ **realization(s)**
g_suite	996	995
h_suite	991	970
i_suite	179	177
cdb	3872	3216
nlwikipedia-selection	7764	7657

Most of the remaining problems are related to productive lexicon rules that cannot be used in inverse direction.

The need for fluency ranking

- A grammar will often allow for more than one surface sentence (realization) to be generated.
- But not every realization is equally fluent.
- One example generated with the Alpino chart generator:

  omdat zijn rol toen echt wel uit was gespeeld
  omdat echt wel toen z'n rol was uit gespeeld
  omdat wel zijner rol echt waart uit gespeeld toen

- For a set of 7657 sentences from Wikipedia from 5 to 15 words, the average number of realizations, allowing minimal punctuation was 83.8
- So, we need good models to pick the most fluent realization from all realizations.

Fluency models

The Alpino generator implements two fluency models:

- An N-gram language model
- A maximum entropy model

N-gram language model

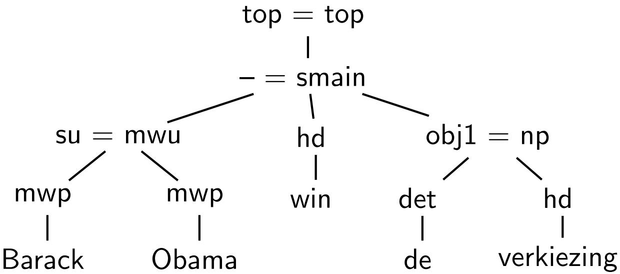
- Intuition: the realization that is the most likely to occur in a language, is the most fluent realization.
- We can estimate the sentence probability with an n-gram model:

  model: $p_n(w_n^k) = \prod_{i=1}^{k} p(w_i | w_{i-n+1}^{i-1})$

Disadvantages of n-gram models

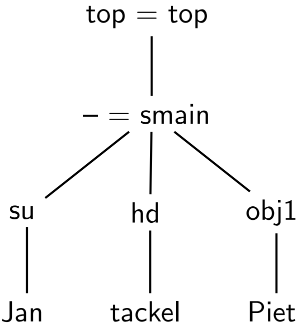
- Cannot capture dependencies that go beyond $n - 1$ span of history.
- Will often even fail to capture dependencies within its span due to data sparseness.
- Cannot directly capture structural characteristics.
- Of course, n-gram models still perform surprisingly well on many tasks.

su-obj1 order



de verkiezingen won Barack Obama (the elections won Barack Obama)
Barack Obama won de verkiezingen (Barack Obama won the elections)

su-obj1 volgorde



Piet tackelt Jan (Piet tackles Jan)
Jan tackelt Piet (Jan tackles Piet)

Maximum entropy models

- Train weights for features capturing various aspects of a realization, including structural information.
- Score realizations by multiplying (trained) feature weights and feature values.
- Integration with the parse disambiguation model of Alpino.

Features templates (Velldal and Oepen 2006)

Velldal and Oepen proposed four feature templates for fluency ranking:

- *ngram_lm*: the n-gram language model score.
- *lds*: local derivation sub-trees, with optional grandparenting.
- *ldsb*: this template provides a back-off for *lds*, by reduction to one daughter.
- *tngramw*: n-grams of syntactic categories and the rightmost word.
- *tngram*: n-grams of syntactic categories (without a surface form).

Feature templates (Velldal 2007)

In addition, in his thesis Velldal proposes some feature templates that measure skewedness in the number of lexical nodes a constituent dominates over:

- *lds_dl*: local derivation subtrees, with binned frequencies of the number of lexical items each daughter dominates over.
- *lds_skew*: local derivation subtrees, with binned standard deviations of the number of lexical items each daughter dominates over.

Feature templates

- *lds_deps*: derivation tree node with a list of relations in its dt feature structure, ordered by the positions of their heads.
- Syntactic features from the Alpino parse disambiguation component:
  - Frame, stem/frame.
  - Ids of rules used in the derivation.
  - Topicalized/non-topicalized subject.
  - Long-distance dependencies.
  - Orderings in the middle-field.

Training/evaluation material

- The n-gram model was trained on 6.4 million sentences from the Twente News Corpus (89.7 million tokens).
- Testing and evaluation was performed on uncorrected DTs for 7763 sentences from the Dutch Wikipedia (August 2008) consisting of 5-15 words.

Training procedure

- Generate sentences from DTs in the training corpus, and extract features using feature templates. Realizations are scored by calculating the ROUGE-N score compared to the original sentence in the training corpus.
- For each DT 100 realizations are randomly selected for training.
- Features are filtered for relevance: a feature is relevant if it takes a different value for any two competing realizations for the same DT.
- Feature weights are estimated (using TADM).

Evaluation method

- Evaluation was performed using ten-fold cross-validation.
- The sentence as it was seen in the corpus is considered the most fluent (gold standard) realization.
- To compare realizations against the gold standard, the ROUGE-SU measure was used.
- Since we want to measure the performance of the fluency component, best match accuracy was used as the evaluation measure.

Results

Model	Accuracy	ROUGE-SU
Random selection	0.012	0.544
N-gram	0.390	0.674
Velldal	0.510	0.713
Velldal + lds deps	0.520	0.716
Velldal + lds deps + disambiguation	0.515	0.716

The catch...

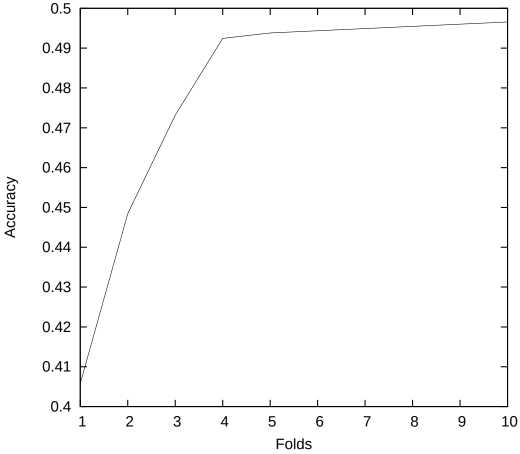
Gold standard: *de partij is zowel organiek als inhoudelijk niet verbonden met de beweging* (*the party is both organically as in content not associated with the movement*)

Fluency model: *de partij is zowel inhoudelijk als organiek niet verbonden met de beweging*

Gold standard: *ook concludeerde hij dat geluid een beweging van de stoffelijke lucht was* (*he also concluded that sound is a movement in material air*)

Fluency model: *hij concludeerde ook dat geluid een beweging van de stoffelijke lucht was*

MaxEnt learning curve



su-obj1 order revisited

n-gram language model:

```
Piet tackelt Jan|-46.873
Jan tackelt Piet|-46.873
```

maxent model:

```
Jan tackelt Piet|-78.885
Piet tackelt Jan|-78.626
```

Conclusions

- The Alpino chart generator can now generate from a substantial part of the Alpino test suites.
- Using a maximum entropy model for fluency ranking provided a substantial improvement over the n-gram language model.
- So far, adding parse disambiguation features or features modelling (dis)preferred dependency order did not provide a substantial improvement over the features proposed by Velldal.
- Software is available from: http://www.let.rug.nl/vannoord/alp/Alpino/