Exemplar-Based Syntax

Rens Bod
Institute for Logic, Language and Computation
University of Amsterdam
Outline

• Examples vs. Rules in linguistics

• Data-Oriented Parsing: productivity from examples

• What linguistic phenomena can DOP explain?
  - preference among alternatives
  - grammaticality decision tasks
  - language acquisition...

• Conclusions
Intro: Linguistic Theories Provide

**Representations**

```
S
 /   \\  
|   NP |
|     |
|   PP |
|   / \\|
|  NP  |
|   /  \|
| DET N |
|   /   \|
| the dog |
```

Formal encoding of grammatical relations

**Rules**

```
S -> NP VP
NP -> NP PP
NP -> det N
PP -> P NP
VP -> V
N -> dog, hill
V -> ...
```

determine representations for all possible utterances

Usual goal: minimal, nonredundant set
The "Competence Hypothesis"

- Language user *applies* internalized rules to produce internal representations

- Language user *acquires* maximally general rules by abstraction of linguistic experience guided by universal principles and constraints
Competence-based models don't obviously account for

- Robustness, interpretations of ill-formed sentences
- Degrees of acceptability, core vs periphery
- Preference among alternatives
- Language changes over time
- Frequency effects in production and comprehension
Alternative View: Representations only, no rules

- Language user acquires *examples of representations* from syntactic experience.

- Language user applies *operations on representations* to produce representations for new utterances, which are added to memory.

- "Rules" perhaps appear in the scientific discourse, but are not part of native speaker's "competence"
DOP: Productivity from examples

(following Bod, Scha, Kaplan, Sima'an, Collins, and others: *Data Oriented Parsing*)

Given a language corpus annotated with representations...

1. Divide representations into fragments
2. Combine fragments to process a new sentence
3. Update corpus with the new sentence

Original model defined on surface phrase-structure trees (Bod 1992)

But there are also DOP models for TAG, HPSG, LFG, ... analyses
Illustration of DOP ("Tree-DOP")
(Bod 1992, 1998)

Given an extremely simple corpus consisting of two structures:

```
S
  | NP
  | John
  | VP
  | V
  | likes
  | NP
  | Mary

S
  | NP
  | Peter
  | VP
  | V
  | hates
  | NP
  | Susan
```
Some fragments (i.e. subtrees) of this corpus

NP
  | NP
  | Mary
  | Susan

S
  | NP
  | VP
  | hates

S
  | NP
  | VP
  | VP
  | V
  | NP
  | likes

S
  | NP
  | VP
  | V
  | NP
  | likes
  | Mary
  | etc.
Combine fragments to derive structures for new sentences

In DOP, "ο" is left-most substitution
Another derivation of the same structure:
She wanted the dress on the rack.

She saw the dog with the telescope.
Corpus

Decompositie

etc.
she wanted the dress on the rack

she saw the dog with the telescope

she saw the dress

she saw the dog

etc.
she wanted the dress on the rack

she saw the dog on the telescope

she saw the dog with the telescope

etc.
Probabilities in DOP

• Large fragments may be statistically significant though they are "linguistically" redundant (e.g. *What time is it?* vs. *How late is it?*)

• Probability of a sentence and a parse tree can be compositionally computed from the probabilities of the fragments that make it up

• By putting various restrictions on the fragments, DOP can in principle instantiate any known probabilistic grammar

• DOP's probabilities are derived from fragment frequencies but can be adjusted by a monotonically decreasing *recency* function
Probability of...

- a subtree: its relative frequency in the distribution of subtrees with same root category
- a derivation: product of its subtree probabilities
- a parse tree: sum of the probabilities of all its derivations
- a sentence: sum of the probabilities of all its parse trees

Comprehension: compute most probable meaning given a sentence (and update corpus)

Generation: compute most probable sentence given a meaning (and update corpus)
Linguistic phenomena that DOP can deal with

- Preference among alternative structures
- Grammaticality decision task in sentence processing
- S-shaped curve in language change
- Gradience of grammaticality judgments
- Language acquisition
1. **Preference among alternatives: ambiguity resolution**

- Language users prefer more probable structures over less probable ones in case of ambiguity.

- This so-called ambiguity problem is very hard:

  Many sentences from the Wall Street Journal treebank have more than *one million* different parse trees (Charniak 1997).

- The problem of *ambiguity* is a hot topic in natural language processing, computer vision and computational musicology.
Examples of Ambiguity

List the sales of products in 1973
How to solve ambiguity?

The Likelihood Principle (Helmholtz 1910)

"Perceptual input will be organized into the most probable structure consistent with the input"

How can the likelihood principle be tested in NLP?

1) Divide a manually created Treebank into a training set and a test set
2) Use the training set to extract the subtrees/rules and their probabilities
3) Compute the most probable tree structure for each test set sentence
4) Compare the results with the trees in the test set
The so-called Tree-DOP model has various shortcomings:

1. Tree-DOP does not make use of syntactic features, grammatical functions, semantic forms etc

2. Many syntactic and semantic dependencies are not reflected directly in a surface tree

- Does the DOP idea that "all fragments count" also hold for linguistically richer structures, such as used in LFG, HPSG, etc?


These models show: any subtree restriction decreases the accuracy!
For example, LFG-DOP uses following corpus-representations:

- c-structure
- f-structure
- a mapping $\phi$ between them

Diagram:

```
S -> NP  VP
   |    |
   Kim eats

PRED    'Kim'
TENSE    PRES
NUM    SG
PRED    'eat(SUBJ)'
SUBJ
```
Two types of fragments in LFG-DOP

1. **Ungeneralized** fragments:

   - NP
     - PRED 'Kim'
     - NUM SG
   - VP
     - SUBJ
     - TENSE PRES
     - PRED 'eat(SUBJ)'

2. **Generalized** fragments -- generated by a "Discard" decomposition operation which constructs generalizations by deleting attribute-value pairs

   - NP
     - VP
     - S
     - SUBJ
     - TENSE PRES
     - PRED 'eat(SUBJ)'

   etc!
Composition operation: lefmost substitution of c-structures followed by recursive unification of f-structures:

A derivation for an LFG-DOP representation $R =$

Sequence of fragments for which the iterative application of the composition operation produces $R$
Probability model

• Probability of a representation $R$ is the sum of the probabilities of its derivations $D$:

$$P(R) = \sum_{D \text{ derives } R} P(D)$$

• Probability of a derivation $D = \langle f_1, f_2 \ldots f_k \rangle$ is the product of the competition probabilities of its fragments $f_i$:

$$P(\langle f_1, f_2 \ldots f_k \rangle) = \prod_i \text{CP}(f_i | \text{CS}_i)$$

• Competition probability of a fragment $f$ is its probability in the distribution $\text{CS}$ of composable fragments at a certain step in the derivation

$$\text{CP}(f | \text{CS}) = \frac{P(f)}{\sum_{f' \in \text{CS}} P(f')}$$
Experimental Evaluation of LFG-DOP

• Two manually LFG-annotated corpora are available:

  Verbmobil corpus:  540 LFG-representations

  Homecentre corpus: 980 LFG-representations

• We used 10 training/test set splits:
  sentences from the test set were parsed by fragments from the training set

• For computational reasons we restricted the maximum fragment depth to 4
Comparing different fragment sizes

DOP Hypothesis is reinforced by LFG representations

<table>
<thead>
<tr>
<th>Size</th>
<th>Exact Match</th>
<th>Precision</th>
<th>Recall</th>
<th>SemPrecision</th>
<th>SemRecall</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30.6%</td>
<td>74.2%</td>
<td>72.2%</td>
<td>83.3%</td>
<td>80.8%</td>
</tr>
<tr>
<td>≤2</td>
<td>34.1%</td>
<td>76.2%</td>
<td>74.5%</td>
<td>86.9%</td>
<td>82.7%</td>
</tr>
<tr>
<td>≤3</td>
<td>35.6%</td>
<td>76.8%</td>
<td>75.9%</td>
<td>87.8%</td>
<td>85.3%</td>
</tr>
<tr>
<td>≤4</td>
<td>35.9%</td>
<td>77.5%</td>
<td>76.4%</td>
<td>88.1%</td>
<td>86.7%</td>
</tr>
</tbody>
</table>

**Verbmobil corpus**

<table>
<thead>
<tr>
<th>Size</th>
<th>Exact Match</th>
<th>Precision</th>
<th>Recall</th>
<th>SemPrecision</th>
<th>SemRecall</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>31.3%</td>
<td>75.0%</td>
<td>71.5%</td>
<td>84.8%</td>
<td>80.7%</td>
</tr>
<tr>
<td>≤2</td>
<td>36.3%</td>
<td>77.1%</td>
<td>74.7%</td>
<td>87.4%</td>
<td>84.5%</td>
</tr>
<tr>
<td>≤3</td>
<td>37.8%</td>
<td>77.8%</td>
<td>76.1%</td>
<td>89.0%</td>
<td>86.1%</td>
</tr>
<tr>
<td>≤4</td>
<td>38.4%</td>
<td>80.0%</td>
<td>78.6%</td>
<td>90.5%</td>
<td>87.4%</td>
</tr>
</tbody>
</table>

**Homecentre corpus**
2. Grammaticality decision tasks in sentence processing

One assumption of DOP and exemplar-based approaches is that all previously heard sentences are stored (for some time). Can this be tested?

**Common wisdom:**

Semantically transparent, non-idiosyncratic sentences are **not** stored in memory.

(e.g. Sachs 1967; Murphy and Shapiro 1994, **but see:** Tomasello 2003).

**Aim of our psycholinguistic experiments (with U. of Leeds):**

To show that this common wisdom is wrong for (frequent) three-word sentences.
Experimental Set-Up:

Participants:

- 50 native speakers of English (undergraduates from the University of Leeds)

Materials:

- 50 frequent three-word (subject-verb-object) sentences were selected from the British National Corpus (BNC)
- All sentences were **semantically transparent** and **non-idiosyncratic**
- All sentences consisted of **uninflected monomorphemic** words
- For each sentence, three additional sentences were created:
  
  by substituting either subject, verb or object by a simplex and roughly equally frequent word of the same category and length (with the same subcategorization scheme)

- The additional sentences were syntactically and semantically correct but had a **low** frequency.
Example:

"I like it" has a relatively high log-frequency of 6.8 in the BNC (using natural logarithms)

By substituting the verb "like" (log-freq = 10.6) by the roughly equally frequent verb "keep" (log-freq = 10.8) we get the low-frequency sentence "I keep it" (log-freq = 2.9).

Pseudo-sentences and Filler-sentences:

For each of the test sentences, an ungrammatical pseudosentence was derived by randomly changing the word order.

Procedure

Participants had to decide as quickly as possible whether a presented word string (on a computer screen) was an English sentence or not.
Analysis and Summary of Results

• For each stimulus and participant we calculated the mean reaction times (RT) and error scores

Main Result:

• By-participant and by-stimulus analyses of variance showed:
  
  high-frequency sentences were reacted to faster than low-frequency sentences.

• The RT-differences cannot be explained by differences in lexical frequency or complexity, since lexical frequency and complexity were kept constant.

• The RT-differences can neither be explained by differences in parsing time, since the test sentences had the same subject-verb-object pattern and surface structure.

• Conclusion: Frequent sentences must somehow be stored in memory.
Modeling: A DOP-based Multi-Route Race Model

- We developed a **DOP-based multi-route race model** that can account for the observed reaction times with one free parameter: the speed of the parsing route.

- Our model is a generalization of the **dual-route race model** for words by Baayen et al. (1997):

  While Baayen et al.'s model needs only two routes, our model has several routes:
  
  1. full-parsing route  
  2. full-form-retrieval (or direct) route  
  3. all routes that lie between full-parsing and full-form retrieval

**The Exemplar Hypothesis:** Every sub-string, of arbitrary size, can function as a unit
Example: Routes for the sentence "I love you"

Although the direct route is the shortest route, it is not necessarily the fastest, as we will see.

We abstract from the actual tree structures, since these are the same for all sentences.
How can we model the reaction time of a route? (1)

• We will model the response latency of a unit by its frequency, assuming that

  (1) human processing is sensitive to the logarithm of the frequency of a unit rather than to its absolute frequency (cf. Shapiro 1969; Scarborough et al. 1977).

  (2) the time required for a unit to reach threshold activation level is inversely proportional to its log-frequency (cf. Baayen et al. 1997)

• Then the predicted latency of a unit $u$ is:

$$t(u) = \frac{1}{1 + \log f(u)}$$
How can we model the reaction time of a route? (2)

The processing time of a certain route \( <u_1,\ldots, u_n> \), where \( 1 \leq n \leq 3 \), is defined as the sum of the latency times of its units \( u_i \) plus the time required by the parsing process \( \Delta p \):

\[
\begin{align*}
t( <u_1,\ldots, u_n> ) &= \sum_{i=1}^{n} \frac{1}{1 + \log f(u_i)} + \Delta p \\
\end{align*}
\]

We assume that \( \Delta p \) is a constant, since the surface structure is the same for all test sentences and this surface structure may function as a template.

Our modeled reaction time for a sentence is determined by the fastest route:

\[
\text{RT} = \min_t [t( <u_1,\ldots, u_i> )]
\]
I like it

Total response latency predicted by the model:

direct route: $t = 0.070$

intermediate route: $t = 0.119$

intermediate route: $t = 0.116$

full-parsing route: $t = 0.161$
Model times for the sentence "I test you":

\[
\begin{align*}
\log f(I) &= 18.5 \quad \rightarrow \quad t(I) = 0.051 \\
\log f(test) &= 16.4 \quad \rightarrow \quad t(test) = 0.058 \quad \text{(note that test has the same freq. as love)} \\
\log f(you) &= 18.2 \quad \rightarrow \quad t(you) = 0.052 \\
\log f(I \text{ test}) &= 7.3 \quad \rightarrow \quad t(I \text{ test}) = 0.120 \\
\log f(test \text{ you}) &= 7.9 \quad \rightarrow \quad t(test \text{ you}) = 0.112 \\
\log f(I \text{ test you}) &= 4.4 \quad \rightarrow \quad t(I \text{ test you}) = 0.186
\end{align*}
\]

Here, the full-parsing route wins

**Modeled reaction time** = 0.161 + \(\Delta p\). **Observed mean reaction time** = 0.873 s

(NB: There are also sentences for which an intermediate route wins, e.g. "We love you", "I keep it", "They need it".)
For a parsing time $\Delta p$ of 330 ms, the results match up to a constant: observed RT = model time + 380 ms (=response execution time?)
3. Language acquisition

• The problem of language learning in DOP is the problem of defining an initial corpus

• Two possibilities have been proposed:

  1. *Nativist*: the initial corpus contains the set of universal representations that hold for all languages and all linguistic phenomena (Bod 1998; Bod et al. 2003)

  2. *Empiricist*: the initial corpus is empty, and all structure is bootstrapped by distributional statistics (e.g. van Zaanen 2000; Manning & Klein 2002; Zuidema 2004)

• DOP can in principle work with both an empiricist and a nativist methodology
Conclusions

• Language displays all the hallmarks of an exemplar-based system: grammaticality judgments, grammaticality decision tasks, preference among alternatives, language change, etc.

• A probabilistic model like DOP can predict a large number of gradient linguistic phenomena:

• Our results suggest that:

  Knowledge of language should be understood not as a minimal set of rules or constraints but as a redundant set of language experiences that changes slightly every time a new utterance is produced or perceived

• Further reading: