Computational linguistics and linguistics
Data-driven and generative approaches

Ann Copestake

Natural Language and Information Processing Group
Computer Laboratory
University of Cambridge

May 2008
Outline.

Introduction: two paradigms

Linguistically motivated computational grammars
  Overview
  DELPH-IN ERG and Matrix
  Ambiguity, parse ranking and robustness

Data-driven methods in computational linguistics
  An outline of annotation-based methodologies
  Learning articles
  Collocation
Introduction: two paradigms

Linguistically motivated computational grammars
Overview
DELPH-IN ERG and Matrix
Ambiguity, parse ranking and robustness

Data-driven methods in computational linguistics
An outline of annotation-based methodologies
Learning articles
Collocation
Formal linguistics and computational linguistics.

- Formal linguistics puts emphasis on *generativity*: words into phrases, phrases into utterances, compositional meaning.
- Syntactically-motivated semantics (i.e., compositional semantics), but not lexical semantics (*vote* is *vote*').
- Much computational linguistics in 1970s and 1980s followed this: compositional semantics plus inference. Reliance on AI.
- Post-1980s computational linguistics is much more data-driven. Limited/no external knowledge sources.
- Can these approaches be combined? What are the implications for linguistics?
Formal linguistics and computational linguistics.

- Formal linguistics puts emphasis on **generativity**: words into phrases, phrases into utterances, compositional meaning.

- Syntactically-motivated semantics (i.e., compositional semantics), but not lexical semantics (**vote** is **vote'**).

- Much computational linguistics in 1970s and 1980s followed this: compositional semantics plus inference. Reliance on AI.

- Post-1980s computational linguistics is much more **data-driven**. Limited/no external knowledge sources.

- Can these approaches be combined? What are the implications for linguistics?
Formal linguistics and computational linguistics.

- Formal linguistics puts emphasis on **generativity**: words into phrases, phrases into utterances, compositional meaning.
- Syntactically-motivated semantics (i.e., compositional semantics), but not lexical semantics (*vote* is *vote*').
- Much computational linguistics in 1970s and 1980s followed this: compositional semantics plus inference. Reliance on AI.
- Post-1980s computational linguistics is much more **data-driven**. Limited/no external knowledge sources.
- Can these approaches be combined? What are the implications for linguistics?
Formal linguistics and computational linguistics.

- Formal linguistics puts emphasis on *generativity*: words into phrases, phrases into utterances, compositional meaning.
- Syntactically-motivated semantics (i.e., compositional semantics), but not lexical semantics (*vote* is *vote*').
- Much computational linguistics in 1970s and 1980s followed this: compositional semantics plus inference. Reliance on AI.
- Post-1980s computational linguistics is much more data-driven. Limited/no external knowledge sources.
- Can these approaches be combined? What are the implications for linguistics?
Natural language interfaces to databases
(e.g., Copestake and Spärck Jones, 1989)

• Who owns a house in a street with parcels in Block 3/2?
• Which owners are in Market Place?
  i.e., Which owners own properties which are in Market Place?
  metonymy

Approach: analyse to produce semantic representation, map to
domain semantics, map to SQL.
Natural language interfaces to databases (e.g., Copestake and Spärck Jones, 1989)

- Who owns a house in a street with parcels in Block 3/2?
- Which owners are in Market Place?
  i.e., Which owners own properties which are in Market Place? metonymy

Approach: analyse to produce semantic representation, map to domain semantics, map to SQL.
Natural language interfaces to databases (e.g., Copestake and Spärck Jones, 1989)

- Who owns a house in a street with parcels in Block 3/2?
- Which owners are in Market Place?
  i.e., Which owners own properties which are in Market Place?  *metonymy*

Approach: analyse to produce semantic representation, map to domain semantics, map to SQL.
Introduction: two paradigms

Linguistically motivated computational grammars

Data-driven methods in computational linguistics

Natural language interfaces to databases (e.g., Copestake and Spärck Jones, 1989)

<table>
<thead>
<tr>
<th>OWNER</th>
<th>OOd OSurname OInits</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWNERSHIP</td>
<td>OWOOd OWPid</td>
</tr>
<tr>
<td>PARCEL</td>
<td>PPid PBid PStrnum PStrnam PLuc</td>
</tr>
<tr>
<td>PPark PDwell PFl PCityv PSqft</td>
<td></td>
</tr>
<tr>
<td>BLOCK</td>
<td>BBid BWid</td>
</tr>
<tr>
<td>WARD</td>
<td>WWid</td>
</tr>
</tbody>
</table>

- Who owns a house in a street with parcels in Block 3/2?
- Which owners are in Market Place?
  i.e., Which owners own properties which are in Market Place? metonymy

Approach: analyse to produce semantic representation, map to domain semantics, map to SQL.
Sentiment classification

- Task: scan documents for positive and negative opinions on people, products etc.
- Find all references to entity in some document collection: list as positive, negative (possibly with strength) or neutral.
- Simple classification, presented as summaries plus text snippets.
- Research (e.g., Pang et al, 2002): classify whole documents (movie reviews).
- Machine learning techniques, trained on rankings: based on words, often little or no parsing. Success rate 80+%. 
Ooooo. Scary.
The old adage of the simplest ideas being the best is once again demonstrated in this, one of the most entertaining films of the early 80’s, and almost certainly Jon Landis’ best work to date. The script is light and witty, the visuals are great and the atmosphere is top class. Plus there are some great freeze-frame moments to enjoy again and again. Not forgetting, of course, the great transformation scene which still impresses to this day.
In Summary: Top banana
Limited domain vs broad coverage language processing

- Until late 1980s: limited domain, often detailed semantics. Systems as agents.
- 2005–: question answering (aka ‘semantic search’), robust inference.
Data-driven and generative approaches.

- This talk: overview of some research in computational linguistics with emphasis on linguistic relevance.
- The course: generativity and conventionalisation in semantics and pragmatics.
- Underlying hypothesis: data-driven techniques work because they model some aspects of language that the classical approaches don’t.

A footnote: limited work on truly data-driven syntax. e.g., Penn Treebank parsing relies on annotation with (more-or-less) conventional parse trees.
This talk: overview of some research in computational linguistics with emphasis on linguistic relevance.

The course: generativity and conventionalisation in semantics and pragmatics.

Underlying hypothesis: data-driven techniques work because they model some aspects of language that the classical approaches don’t.

A footnote: limited work on truly data-driven syntax. e.g., Penn Treebank parsing relies on annotation with (more-or-less) conventional parse trees.
Data-driven and generative approaches.

- This talk: overview of some research in computational linguistics with emphasis on linguistic relevance.
- The course: generativity and conventionalisation in semantics and pragmatics.
- Underlying hypothesis: data-driven techniques work because they model some aspects of language that the classical approaches don’t.

A footnote: limited work on truly data-driven syntax. e.g., Penn Treebank parsing relies on annotation with (more-or-less) conventional parse trees.
Outline.

Introduction: two paradigms

Linguistically motivated computational grammars

Overview
DELPH-IN ERG and Matrix
Ambiguity, parse ranking and robustness

Data-driven methods in computational linguistics

An outline of annotation-based methodologies
Learning articles
Collocation
Formalisms and systems.

- Theoretical frameworks, generally hand-built grammars, partially hand-built lexicons, may have compositional semantics, may be bidirectional (parse and generate).
- LFG (PARC XLE system), TAG (U.Penn), CCG, GPSG, IBM/Microsoft, FUF/SURGE (generation only) . . .
- Last 10-15 years: stochastic parse ranking, automatic lexical acquisition, robustness, processing speed.
- Linguistically motivated grammars were dominant paradigm in 1980s: proportionally less significant now.
The DELPH-IN English Resource Grammar
(Flickinger et al)

- ERG demo at http://erg.emmtee.net.
- Broad-coverage, precise, bidirectional grammar for English, used in a number of projects.
- Approximately 80% - 90% coverage for corpora tried so far (after lexicon and any specific constructions added).
- Variety of strategies for adding lexicon automatically. Combine with ‘shallow’ analysers where robustness is required.
- Grammars for other languages developed partially on basis of the ERG (Japan, German, Matrix grammars).
Some corpora parsed with the ERG.

- **Verbmobil**: arranging meetings etc
  That doesn’t suit me very well
- **ecommerce**: ordering electronic goods
- **LOGON**: Norwegian hiking
- **SciBorg**: chemistry journals
  The synthesis of 2,8-dimethyl-6H,12H-5,11 methanodibenzo[b,f][1,5]diazocine (Troger’s base) from p-toluidine and of two Troger’s base analogs from other anilines . . .
The DELPH-IN Grammar Matrix (Bender et al)

- A toolkit for grammar development: used for small grammars of lots of languages (teaching), more substantial grammars of Spanish, Norwegian, Korean, Portuguese, French, Greek, Swedish, Italian (research).
- Aim: provide a core grammar that can be specialised for individual languages.
- Some typological distinctions automatically converted into grammar fragments.
- Potential use for field linguists.
Coordination in the Matrix (Drellishak and Bender)

Four dimensions:

1. kind of marking (lexical, morphological, none).
2. pattern of marking: a-, mono-, poly-, or omnisyndeton.
3. position of marking: before or after the coordinand.
4. phrase types covered: NP, NOM, VP, AP, etc.

Consistent compositional semantics for all variants. Implemented via types for constructions (e.g., coord-phrase): constructions for particular languages inherit from more general types (e.g., vp-top-coord-rule for Ono).
Ambiguity.

- Pre-1990s, most discussion related to lexical ambiguity, PP-attachment, etc.
  I saw the man with the telescope.
  Assumption: real world knowledge needed to resolve ambiguity.

- Some claim that corpus-based information is an approximation for AI and inference.

- But for actual large coverage grammars, even high precision ones, most ambiguity is due to unusual constructions.
  Police save a lobster from certain death.
  save as preposition, imperative etc.
Ambiguity.

- Pre-1990s, most discussion related to lexical ambiguity, PP-attachment, etc.
  I saw the man with the telescope.
  Assumption: real world knowledge needed to resolve ambiguity.

- Some claim that corpus-based information is an approximation for AI and inference.

- But for actual large coverage grammars, even high precision ones, most ambiguity is due to unusual constructions.
  Police save a lobster from certain death.
  save as preposition, imperative etc.
Ambiguity.

- Pre-1990s, most discussion related to lexical ambiguity, PP-attachment, etc.
  I saw the man with the telescope.
  Assumption: real world knowledge needed to resolve ambiguity.

- Some claim that corpus-based information is an approximation for AI and inference.

- But for actual large coverage grammars, even high precision ones, most ambiguity is due to unusual constructions.
  Police save a lobster from certain death.
  save as preposition, imperative etc.
Parse ranking.

- Build a treebank by automatically parsing sentences in a corpus and manually disambiguating (or, Penn Treebank, manually construct trees with aid of a robust grammar).
- Train parse ranking algorithm on the saved trees.
- ERG etc: Redwoods Treebank. Note need to update treebank for new versions of grammar.
- Redwoods experience: around 3000 sentences gives good model: many more would be needed for lexical dependencies (e.g., saw/telescope vs man/telescope).
Lexical probabilities.

- Word sense frequencies, subcategorisation possibilities etc generally have very skewed distributions.
  - *diet* (food) is 100 times more frequent than *diet* (parliament) in BNC.
  - believe NP VPinf very infrequent compared to believe NP, believe that S.
- Similar effect for subcat frames related by alternation.
Grammaticality and subcategorization (Manning 2003).

Judgements (from Pollard and Sag, 1994):

1. We consider Kim an acceptable candidate
2. We consider Kim quite acceptable
3. *We consider Kim as an acceptable candidate
4. *We consider Kim as quite acceptable

Corpus data:

1. The boys consider her as family and she participates in everything we do.
2. ‘We consider that as part of the job’, Keep said.
3. . . . he said he considers them as having championship potential.
Grammaticality and subcategorization (Manning 2003).

Judgements (from Pollard and Sag, 1994):

1. We consider Kim an acceptable candidate
2. We consider Kim quite acceptable
3. *We consider Kim as an acceptable candidate
4. *We consider Kim as quite acceptable

Corpus data:

1. The boys consider her as family and she participates in everything we do.
2. ‘We consider that as part of the job’, Keep said.
3. … he said he considers them as having championship potential.
Outline.

Introduction: two paradigms

Linguistically motivated computational grammars
  Overview
  DELPH-IN ERG and Matrix
  Ambiguity, parse ranking and robustness

Data-driven methods in computational linguistics
  An outline of annotation-based methodologies
  Learning articles
  Collocation
How to write a CL paper, option 1

1. Think of a phenomenon to work on: e.g., non-referential *it*.
2. Obtain a corpus, develop an annotation scheme and guidelines: e.g., referential vs extraposed vs . . .
3. Mark up some part of a corpus.
4. Calculate agreement between annotators.
5. If/when agreement is low: either a) return to 2 (good practice) or b) collapse categories until it’s OK (bad practice).
6. Think of some features, train a machine learning algorithm (or algorithms).
7. Report results which are better than a basic baseline method but lower than human agreement.
How to write a CL paper, option 2.

1. Take someone else’s marked up corpus.
2. Try a new machine learning method.
3. Report results which are better than the previous ones.

Variants: learn categories, lexicon etc.
How to write a CL paper, option 2.

1. Take someone else’s marked up corpus.
2. Try a new machine learning method.
3. Report results which are better than the previous ones.

Variants: learn categories, lexicon etc.
Playing the annotation game.

- Generally, relatively little knowledge of computer science or programming needed: scripting language to extract features (e.g., perl), standard machine learning packages (e.g., WEKA).
- Choice of phenomenon, development of annotation scheme, feature selection, error analysis: should all be informed by linguistics.
- What does and doesn’t work may be theoretically interesting.
Avoiding annotation.

- Limiting annotation: active learning, semi-supervised methods.
- Derive training data from existing corpora.
- Generation tasks: make assumptions about likely input, see if corpus data can be replicated in this respect (e.g., adjective ordering, learning articles).
- Data-driven semantics: vector space models (e.g., clustering.)
Can we predict the articles associated with noun phrases?

- translation from Chinese into English, fixing telegraphese
- a/the/no determiner

Methodology:
1. take an English corpus with noun phrases marked
2. training data: look at distributions of determiners based on features (e.g., head of NP)
3. test data: remove all instances of the and a
4. use model to predict the vs a vs no determiner on test data
(S
  (NP-SBJ
    (NP (NNP Pierre) (NNP Vinken))
  )
  (, ,)
  (ADJP
    (NP (CD 61) (NNS years))
    (JJ old)
  )
  (, ,)
  (VP (MD will)
    (VP (VB join)
      (NP (DT the) (NN board))
      (PP-CLR (IN as)
        (NP (DT a) (JJ nonexecutive) (NN director))
        (NP-TMP (NNP Nov.) (CD 29))
      )
    )
  )
  (.
   .
  )
)
Articles in the Penn Treebank

(S
  (NP-SBJ
    (NP null (NNP Pierre) (NNP Vinken) )
    (, ,)
  )
  (ADJP
    (NP null (CD 61) (NNS years) )
    (JJ old) )
  (, ,) )
  (VP (MD will)
    (VP (VB join)
      (NP (DT the) (NN board) )
      (PP-CLR (IN as)
        (NP (DT a) (JJ nonexecutive) (NN director) )))
    (NP-TMP null (NNP Nov.) (CD 29) )))
  (. .) )
(S
  (NP-SBJ
    (NP  (NNP Pierre) (NNP Vinken) )
    (, ,)
    (ADJP
      (NP  (CD 61) (NNS years) )
      (JJ old) )
    (, ,) )
  (VP (MD will)
    (VP (VB join)
      (NP  (NN board) )
      (PP-CLR (IN as)
        (NP  (JJ nonexecutive) (NN director) ))
      (NP-TMP  (NNP Nov.) (CD 29) )))
  (. .) )

Articles in the Penn Treebank
Articles in the Penn Treebank

(S
  (NP-SBJ
    (NP null (NNP Pierre) (NNP Vinken) )
    (, ,)
  (ADJP
    (NP null (CD 61) (NNS years) )
    (JJ old) )
  (, ,) )
  (VP (MD will)
    (VP (VB join)
      (NP (DT the) (NN board) )
      (PP-CLR (IN as)
        (NP (DT a) (JJ nonexecutive) (NN director) ))
      (NP-TMP null (NNP Nov.) (CD 29) )))
  (. .) )
Introduction: two paradigms

Linguistically motivated computational grammars

Data-driven methods in computational linguistics

Machine learning results

Minnen, Bond and Copestake (1999), using Wall Street Journal:

- baseline — always guess ‘no article’: score 70%
- head of NP as feature: score 80%
- combined features: score 83%

Ezekiel (2003):

- add ‘discourse’ features: has the head noun been seen in this discourse?
- no improvement observed . . .
Magnitude adjectives and non-physical-solid nouns. (Copestake, 2005)

Distributional data from the British National Corpus (100 million words)

<table>
<thead>
<tr>
<th></th>
<th>importance</th>
<th>success</th>
<th>majority</th>
<th>number</th>
<th>proportion</th>
<th>quality</th>
<th>role</th>
<th>problem</th>
<th>part</th>
<th>winds</th>
<th>support</th>
<th>rain</th>
</tr>
</thead>
<tbody>
<tr>
<td>great</td>
<td>310</td>
<td>360</td>
<td>382</td>
<td>172</td>
<td>9</td>
<td>11</td>
<td>3</td>
<td>44</td>
<td>71</td>
<td>0</td>
<td>22</td>
<td>0</td>
</tr>
<tr>
<td>large</td>
<td>1</td>
<td>1</td>
<td>112</td>
<td>1790</td>
<td>404</td>
<td>0</td>
<td>13</td>
<td>10</td>
<td>533</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>high</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>92</td>
<td>501</td>
<td>799</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>90</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>major</td>
<td>62</td>
<td>60</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>272</td>
<td>356</td>
<td>408</td>
<td>1</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>big</td>
<td>0</td>
<td>40</td>
<td>5</td>
<td>11</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>79</td>
<td>79</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>strong</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>8</td>
<td>0</td>
<td>3</td>
<td>132</td>
<td>147</td>
<td>0</td>
</tr>
<tr>
<td>heavy</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>198</td>
</tr>
</tbody>
</table>
**Introduction: two paradigms**

Linguistically motivated computational grammars

Data-driven methods in computational linguistics

---

**Adjectives: selected examples.**

<table>
<thead>
<tr>
<th>BNC frequencies:</th>
<th>number</th>
<th>proportion</th>
<th>quality</th>
<th>problem</th>
<th>part</th>
<th>winds</th>
<th>rain</th>
</tr>
</thead>
<tbody>
<tr>
<td>large</td>
<td>1790</td>
<td>404</td>
<td>0</td>
<td>10</td>
<td>533</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>high</td>
<td>92</td>
<td>501</td>
<td>799</td>
<td>0</td>
<td>3</td>
<td>90</td>
<td>0</td>
</tr>
<tr>
<td>big</td>
<td>11</td>
<td>1</td>
<td>0</td>
<td>79</td>
<td>79</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>heavy</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>198</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Acceptability judgements:</th>
<th>number</th>
<th>proportion</th>
<th>quality</th>
<th>problem</th>
<th>part</th>
<th>winds</th>
<th>rain</th>
</tr>
</thead>
<tbody>
<tr>
<td>large</td>
<td>*</td>
<td></td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>high</td>
<td></td>
<td>*</td>
<td>?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>big</td>
<td></td>
<td></td>
<td>?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>heavy</td>
<td>?</td>
<td>*</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Investigated the distribution of *heavy, high, big, large, strong, great, major* with the most common co-occurring nouns in the BNC.

Nouns tend to occur with up to three of these adjectives with high frequency and low or zero frequency with the rest.

50 nouns in BNC with the extended use of *heavy* with frequency 10 or more, 160 such nouns with *high*. Only 9 with both: *price, pressure, investment, demand, rainfall, cost, costs, concentration, taxation*

Clusters: e.g., weather precipitation nouns with *heavy*. Note *heavy shower* (weather, not bathroom).
Hypotheses about distribution.

- 'abstract' heavy, high, big, large, strong, great, major all denote magnitude (in a way that can be made formally precise)
- distribution differences due to collocation, soft rather than hard constraints
- adjective-noun combination is semi-productive
- denotation and syntax allow heavy esteem etc, but speakers are sensitive to frequencies, prefer more frequent phrases with 'same' meaning
Collocation in general

- Collocation effects prevent some otherwise grammatical phrases from being fully acceptable.
- Collocation is partially but not completely arbitrary: maybe analogy-governed rather than rule-governed.
- Language learners find collocations difficult: e.g., for magnitude adjectives, tend to default to *big* (frequent error in language learner corpus).
Introduction: two paradigms

Linguistically motivated computational grammars

Data-driven methods in computational linguistics

Two quotations.

All continuities, all possibilities of infinitesimal gradation, are shoved outside of linguistics in one direction or the other. Joos (1950)

It must be recognized that the notion ‘probability of a sentence’ is an entirely useless one, under any known interpretation of this term. (Chomsky 1969)

If you believe this, then you won’t like modern computational linguistics!
Two quotations.

All continuities, all possibilities of infinitesimal gradation, are shoved outside of linguistics in one direction or the other. Joos (1950)

It must be recognized that the notion ‘probability of a sentence’ is an entirely useless one, under any known interpretation of this term. (Chomsky 1969)

If you believe this, then you won’t like modern computational linguistics!
Conclusion.

- Computational linguists work together with linguists on ‘traditional’ models including morphology, syntax and compositional semantics.
- Computational linguistics has demonstrated that probabilistic models can provide better models of language than purely symbolic ones.
- Hypothesis: this is not simply due to performance, world knowledge, pragmatics or other non-linguistically relevant effects.
- So: if ‘linguistics’ means the study of language, then this is part of linguistics . . .
- Huge amounts of work remain to be done at a theoretical and methodological level.