Beyond rationalism versus empiricism

Mark Johnson

Macquarie University
Sydney, Australia

COLING 2010
Be clear about your goals

- **Engineering is different to science**
  - science is about *insight* and *understanding*
  - engineering is about *making things work*
- **Be clear about what you’re trying to achieve**
  - this determines *what counts as success*
- The importance of a scientific insight is *not proportional to how useful it is*
  - you can bake a tasty cake without knowing chemistry!
- **Which knowledge is most useful depends on what your goals are!**
Capturing vs. covering a linguistic generalization

- **Capturing a generalization**: grammar accurately describes phenomenon at appropriate level, e.g., subject-verb agreement via PERSON and NUMBER features

- **Covering a generalization**: model covers common cases of a generalization, perhaps indirectly. E.g., head-to-head POS dependencies

- An “engineering” parser only needs to cover generalizations

- But feature design requires linguistic insight
  - *basic linguistic insights have greatest impact*
Capturing vs. covering a linguistic generalization

- **Capturing a generalization**: grammar accurately describes phenomenon at appropriate level, e.g., subject-verb agreement via PERSON and NUMBER features
- **Covering a generalization**: model covers common cases of a generalization, perhaps indirectly. E.g., head-to-head POS dependencies
- An “engineering” parser only needs to cover generalizations
- But feature design requires *linguistic insight*
  - *basic linguistic insights have greatest impact*
Capturing vs. covering a linguistic generalization

- **Capturing a generalization**: grammar accurately describes phenomenon at appropriate level, e.g., subject-verb agreement via PERSON and NUMBER features

- **Covering a generalization**: model covers common cases of a generalization, perhaps indirectly. E.g., head-to-head POS dependencies

- An “engineering” parser only needs to cover generalizations

- But feature design requires *linguistic insight*
  - *basic linguistic insights have greatest impact*
Capturing vs. covering a linguistic generalization

• **Capturing a generalization**: grammar accurately describes phenomenon at appropriate level, e.g., subject-verb agreement via PERSON and NUMBER features.

• **Covering a generalization**: model covers common cases of a generalization, perhaps indirectly. E.g., head-to-head POS dependencies.

• An “engineering” parser only needs to cover generalizations.

• But feature design requires *linguistic insight*
  
  ▶ basic linguistic insights have greatest impact
Capturing vs. covering a linguistic generalization

- **Capturing a generalization**: grammar accurately describes phenomenon at appropriate level, e.g., subject-verb agreement via PERSON and NUMBER features
- **Covering a generalization**: model covers common cases of a generalization, perhaps indirectly. E.g., head-to-head POS dependencies
- An “engineering” parser only needs to cover generalizations
- But feature design requires *linguistic insight*
  - *basic linguistic insights have greatest impact*
Capturing vs. covering a linguistic generalization

- **Capturing a generalization**: grammar accurately describes phenomenon at appropriate level, e.g., subject-verb agreement via PERSON and NUMBER features.

- **Covering a generalization**: model covers common cases of a generalization, perhaps indirectly. E.g., head-to-head POS dependencies.

- An “engineering” parser only needs to cover generalizations.

- But feature design requires **linguistic insight**.
  - **basic linguistic insights have greatest impact**.
Capturing vs. covering a linguistic generalization

- **Capturing a generalization**: grammar accurately describes phenomenon at appropriate level, e.g., subject-verb agreement via PERSON and NUMBER features.

- **Covering a generalization**: model covers common cases of a generalization, perhaps indirectly. E.g., head-to-head POS dependencies.

- An “engineering” parser only needs to cover generalizations.

- But feature design requires **linguistic insight**
  - *basic linguistic insights have greatest impact*
Capturing vs. covering a linguistic generalization

- **Capturing a generalization**: grammar accurately describes phenomenon at appropriate level, e.g., subject-verb agreement via PERSON and NUMBER features

- **Covering a generalization**: model covers common cases of a generalization, perhaps indirectly. E.g., head-to-head POS dependencies

- An “engineering” parser only needs to cover generalizations

- But feature design requires linguistic insight
  - basic linguistic insights have greatest impact
Capturing vs. covering a linguistic generalization

- **Capturing a generalization**: grammar accurately describes phenomenon at appropriate level, e.g., subject-verb agreement via PERSON and NUMBER features

- **Covering a generalization**: model covers common cases of a generalization, perhaps indirectly. E.g., head-to-head POS dependencies

- An “engineering” parser only needs to cover generalizations
- But feature design requires *linguistic insight*
  - *basic linguistic insights have greatest impact*
Capturing vs. covering a linguistic generalization

- **Capturing a generalization**: grammar accurately describes phenomenon at appropriate level, e.g., subject-verb agreement via PERSON and NUMBER features.

- **Covering a generalization**: model covers common cases of a generalization, perhaps indirectly. E.g., head-to-head POS dependencies.

- An “engineering” parser only needs to cover generalizations.

- But feature design requires *linguistic insight*.
  - *basic linguistic insights have greatest impact*.
Capturing vs. covering a linguistic generalization

- **Capturing a generalization**: grammar accurately describes phenomenon at appropriate level, e.g., subject-verb agreement via PERSON and NUMBER features.

- **Covering a generalization**: model covers common cases of a generalization, perhaps indirectly. E.g., head-to-head POS dependencies.

- An “engineering” parser only needs to cover generalizations.
- But feature design requires *linguistic insight*
  - *basic linguistic insights have greatest impact*
Capturing vs. covering a linguistic generalization

- **Capturing a generalization**: grammar accurately describes phenomenon at appropriate level, e.g., subject-verb agreement via PERSON and NUMBER features.
- **Covering a generalization**: model covers common cases of a generalization, perhaps indirectly. E.g., head-to-head POS dependencies.

- An “engineering” parser only needs to cover generalizations.
- But feature design requires *linguistic insight*
  - *basic linguistic insights have greatest impact*
Capturing vs. covering a linguistic generalization

• **Capturing a generalization**: grammar accurately describes phenomenon at appropriate level, e.g., subject-verb agreement via PERSON and NUMBER features

• **Covering a generalization**: model covers common cases of a generalization, perhaps indirectly. E.g., head-to-head POS dependencies

• An “engineering” parser only needs to cover generalizations

• But feature design requires *linguistic insight*
  - *basic linguistic insights have greatest impact*
Capturing vs. covering a linguistic generalization

• **Capturing a generalization**: grammar accurately describes phenomenon at appropriate level, e.g., subject-verb agreement via PERSON and NUMBER features

• **Covering a generalization**: model covers common cases of a generalization, perhaps indirectly. E.g., head-to-head POS dependencies

• An “engineering” parser only needs to cover generalizations

• But feature design requires **linguistic insight**
  - basic linguistic insights have greatest impact
Capturing vs. covering a linguistic generalization

- **Capturing a generalization**: grammar accurately describes phenomenon at appropriate level, e.g., subject-verb agreement via PERSON and NUMBER features

- **Covering a generalization**: model covers common cases of a generalization, perhaps indirectly. E.g., head-to-head POS dependencies

- An “engineering” parser only needs to cover generalizations

- But feature design requires *linguistic insight*
  - *basic linguistic insights have greatest impact*
Capturing vs. covering a linguistic generalization

- **Capturing a generalization**: grammar accurately describes phenomenon at appropriate level, e.g., subject-verb agreement via \textit{PERSON} and \textit{NUMBER} features.

- **Covering a generalization**: model covers common cases of a generalization, perhaps indirectly. E.g., head-to-head POS dependencies.

- An “engineering” parser only needs to cover generalizations.

- But feature design requires \textit{linguistic insight}
  - basic linguistic insights have greatest impact.
After all the low-hanging fruit is gone …

- Early statistical NLP focused on surface generalisations
  - but many of the simple ideas have been tried already
- Linguistic structure can help generalise better
  - e.g., Chelba and Jelinek “Structured language model” (aka *shift-reduce parser*)
    - theoretically most interesting ideas ≠ most useful ideas
      ⇒ try the simple stuff first!
- Look beyond theoretical linguistics to:
  - language acquisition, psycholinguistics
  - language typology, historical linguistics
  - neuroscience, genetics
- Our field still lacks many central insights
  - nobody knows where they’ll come from
    ⇒ it’s foolish for the field to put all our “theoretical eggs” in one basket!
Theoretical and computational linguistics have different goals

- A “parasitic gap” is a syntactic construction with one “filler” and multiple “gaps”

  Which book did you buy _ before reading _ ?

- Linguists have published many articles on parasitic gaps
- There are very few parasitic gaps in the PTB WSJ corpus
  \[ \Rightarrow \text{covering parasitic gaps won’t change your PTB f-score} \]
- Rare phenomena can be scientifically very important
  - Chomskyians argue that parasitic gaps must be innate because they are too rare to be learned
    and if you’re parsing a genre where parasitic gaps are common, you probably should pay attention to them!
Research is a gamble about the unknown

_Half the money I spend on advertising is wasted. The problem is: I don’t know which half._
— John Wanamaker

- Nobody knows what knowledge will turn out to be most important
  - that’s why it’s research!
- In an ideal world we’d all know everything . . .
  - but _time spent learning something is time not spent learning something else_
  - you are gambling that the knowledge you acquire today will be useful in tomorrow’s research
- It’s easy to identify grand goals . . .
  - but it takes genius to _identify a set of achievable steps that will reach a grand goal from where we are today_
Look forward, not backward!

- There are still deep scientific mysteries in our field; e.g., *compositionality*
  - how are trees be represented in the brain’s neural circuitry?
  - our statistical models reduce tree structures to finite-dimensional feature vectors of sufficient statistics
    - this is a lossy many-to-one mapping
  - the tree cannot be recovered from the feature vector
    - *are there more insightful mathematical models of compositional structures?*

- Understanding language and thought will probably require *synthesising and extending empiricist and rationalist insights* (and much more as well)

- Learn from the past, but look to the future!