Statistical Approach towards Deep Lexical Acquisition

Yi Zhang

Department of Computational Linguistics & Phonetics
Saarland University, Germany

August 18, 2005

Outline

Outline

1. Motivation
   - Grammar Coverage
   - Case Study: Manual Lexical Extension
2. Previous Work in Automated DLA
   - Unification-Based Approach
   - Data-Driven Approach
3. DLA as Classification Task
   - Maximum Entropy Model for DLA
   - Importing Lexicon from WordNet
4. Conclusion and Future Work
   - Conclusion
   - Future Work

Coverage Problem with Deep Grammars

- Broad coverage linguistically deep processing is desirable for advanced NL applications.
- State-of-the-art deep grammars can only achieve moderate coverage.

Coverage test of ERG on BNC

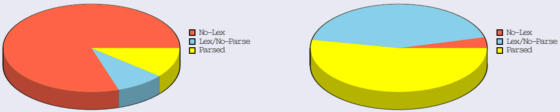
[Baldwin et al (2004)]

- Full lexical coverage for 32% strings
- Of these, parse generated for 57% (83% correct)
- For parsing failure
  - Missing lexical entries 26%
  - Missing constructions 17%
  - Garbage strings 17%
  - Others 40%

Case Study: Manual Lexical Extension

- Corpus *“Shanghai”*: 1600 English sentences/strings about tourism in Shanghai (similar to the “rondane” corpus in LOGON).
- Discover new word/MWE; map it to one of the leaf lexical types in ERG

Coverage before & after lexical extension



- Lexical extension is crucial for broad coverage text processing
- Manual extension requires adequate linguistic sufficiency, and is time consuming
- New lexicon incorporated into ERG

Outline

1. Motivation
   - Grammar Coverage
   - Case Study: Manual Lexical Extension
2. Previous Work in Automated DLA
   - Unification-Based Approach
   - Data-Driven Approach
3. DLA as Classification Task
   - Maximum Entropy Model for DLA
   - Importing Lexicon from WordNet
4. Conclusion and Future Work
   - Conclusion
   - Future Work

Unification-Based Approach

Erbach (1990)

- Parse the sentence with the unknown words
- Collect the lexical information from the syntactic structure of the parse
- Create new lexical entries according to the collected lexical information

Barg and Walther (1998)

- *Generalizable* and *Revisable* information

Fouvry (2003)

- Use external sources to reduce the computational complexity

Data-Driven Approach

Brent (1991)

To learn the SFs of verbs from untagged text (shallow)

Baldwin (2005)

Bootstrap deep lexicon from secondary language resources with the help of shallow processing tools

Problems

- Unification based approach
  - Grammar dependent
  - Underspedified lexical entries overgenerate
- Data-driven approach
  - Most of the approaches focus on some specific aspect of the lexicon (SF for verbs, countability for nouns, etc)
  - All relies on the availability of secondary language resources

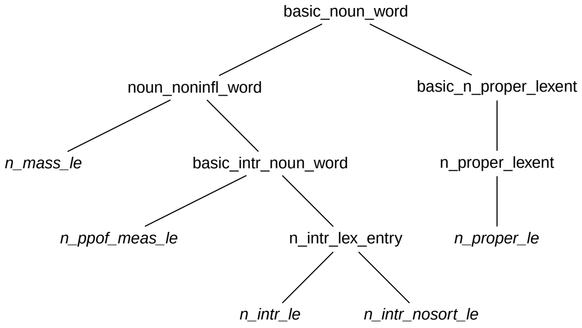
Ideas

Use the treebank generated by the grammar to learn statistical models for DLA.

Outline

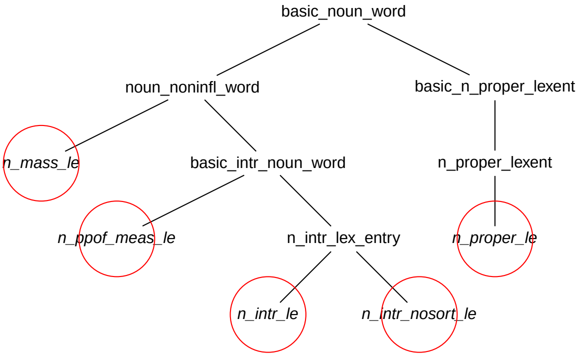
DLA as Classification Task

- The lexical entries can be constructed with the lexeme and one of the atomic types
- DLA assigns an atomic type to each unknown word/lexeme



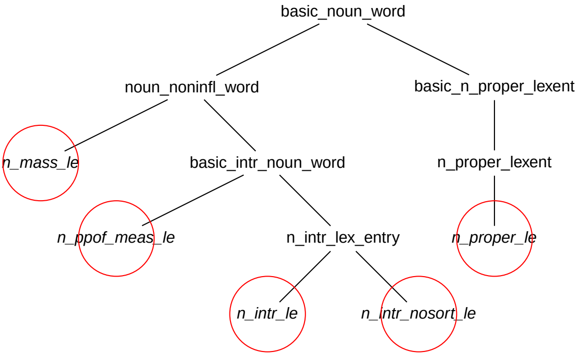
DLA as Classification Task

- The lexical entries can be constructed with the lexeme and one of the atomic types
- DLA assigns an atomic type to each unknown word/lexeme



DLA as Classification Task

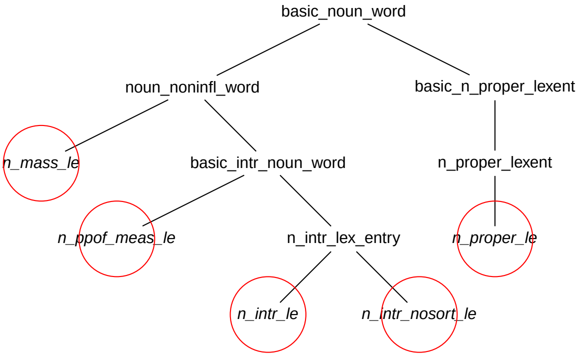
- The lexical entries can be constructed with the lexeme and one of the atomic types
- DLA assigns an atomic type to each unknown word/lexeme

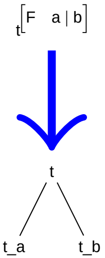


$_t\left[\text{F} \quad a \mid b\right]$

DLA as Classification Task

- The lexical entries can be constructed with the lexeme and one of the atomic types
- DLA assigns an atomic type to each unknown word/lexeme





Tagger-based Model

- Use general purpose POS tagger
  - *TnT*: HMM-based trigram tagger [Brants (2000)]
  - *MXPOST*: ME-based tagger [Ratnaparkhi (1996)]
- Use atomic lexical types as tag-set
- Train tagger with corpus annotated with lexical types
- Tag the input sequence and use the tagger output for unknowns to create new lexical entries

Maximum Entropy Model

- General feature representation
- Capable of handling large feature set
- No independence assumption between features

$$p_\Lambda(t|x) = \frac{\exp(\sum_i \lambda_i f_i(x,t))}{\sum_{t' \in T} \exp(\sum_i \lambda_i f_i(x,t'))}, \Lambda = \{\lambda_i\}$$

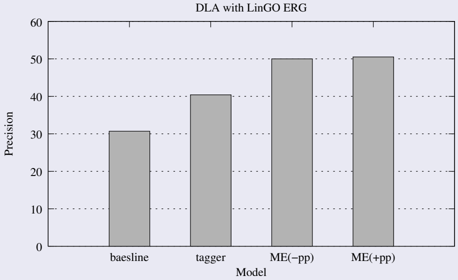
Classification Features

- Morphological features
  - Prefix/Suffix
- Syntactic features
  - Adjacent words/lexical types
  - Partial parse chart/chunks
  - Dependency head/daughters/labels
- Semantic features
  - (R)MRS fragments

Experiment with ERG

- ERG June 2004
- Redwoods Treebank (5th)
- 10-fold cross validation

Results



Importing lexicon from WordNet

Assumption

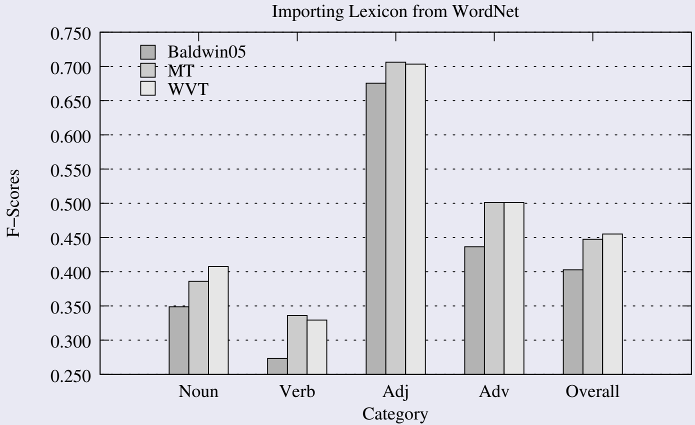
There is a strong correlation between the semantic and syntactic similarity of words. [Levin (1993)]

Fact

Above 90% of the synsets in WordNet (2.0) share at least one lexical type among all included words

Importing Lexicon from WordNet

Results



Outline

Conclusion

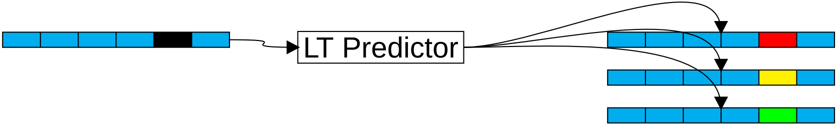
- Cross validation on Redwoods shows about 50% precision with the ME model.
- Experiment on small domain texts shows precision above 80% with very small training set (about 1.5K sentences).
- The method is language independent, and requires minimum extra language resource.

Future Work

- Embedding the DLA module into the grammar engineering platform.
- Use parse result as feedback to enhance the precision.

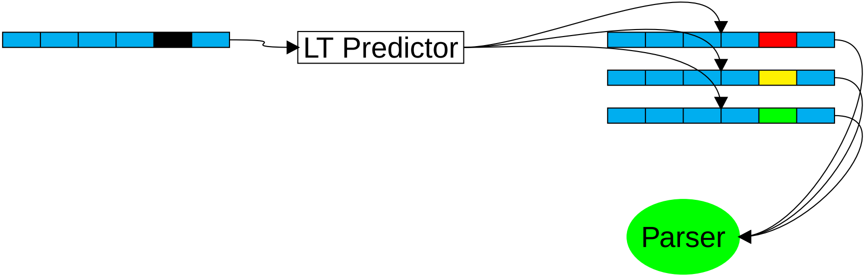
Future Work

- Embedding the DLA module into the grammar engineering platform.
- Use parse result as feedback to enhance the precision.



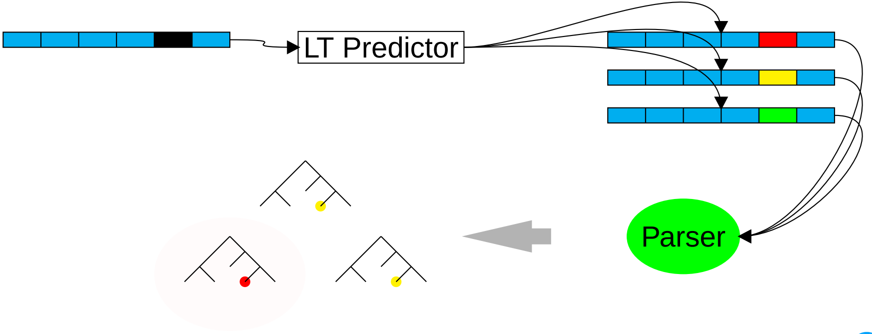
Future Work

- Embedding the DLA module into the grammar engineering platform.
- Use parse result as feedback to enhance the precision.



Future Work

- Embedding the DLA module into the grammar engineering platform.
- Use parse result as feedback to enhance the precision.



Future Work

- Embedding the DLA module into the grammar engineering platform.
- Use parse result as feedback to enhance the precision.