Sample Selection for Parser Induction

Rebecca Hwa
University of Maryland
hwa@umiacs.umd.edu
Supervised Training

• Training examples contain hand-annotated structural information


• Annotation is labor intensive

• Minimize annotations
  – By using less annotation in each example
  – By using fewer examples
Sample Selection

• Interactive training session
  – Learner selects data to learn from
  – Teacher annotates data as needed

• Predict the benefit of annotation
  – *Training Utility Value (TUV)*
    • the improvement of the hypothesis if a candidate is labeled and added to the training set
  – Annotate the candidates with the highest TUVs
The Rest of this Talk

• Sample selection framework
• Toy example
  – Applying sample selection to PP-attachment
• Applying sample selection to parser induction
  – Expectation-based parser
  – History-based parser
• Experiments and discussions
• Conclusion
Sample Selection Algorithm

Initialize:
Train on small set of labeled examples to get initial hypothesis

Repeat
Using current hypothesis and an evaluation function, \( f \), pick \( n \) examples from unlabeled pool.
Ask human to label those \( n \) examples.
Add them to training set.
Re-train to get a new hypothesis.

Until (hypothesis good enough) or (human stops).
Approximation of TUVs

• True TUVs are not known
• Need relative ranking
• Ranking criteria for evaluation functions
  – Problem-space
    • Characteristic of the data distribution
  – Prediction of the model
    • High uncertainty (low likelihood)
  – Parameters of the model
    • Low confidence (high variance)
Application to PP-Attachment

• Unlabeled candidates:
  (delivered, speech, without, effort)
  (delivered, speech, without, content)

• Training examples:
  (v, delivered, speech, without, effort)
  (n, delivered, speech, without, content)

• Basic learner:
  – Backed-off model (Collins-Brooks, 1995)
PP-Attachment Model

delivered, speech, without, effort

delivered, speech, without, effort

delivered, without, effort

without, effort

speech, without

without
4 Proposed Evaluation Functions

- **Problem-space**
  - Pick candidates with new and frequent tuples

- **Prediction of the model**
  - Pick candidates that the model predicts are equally likely to attach to noun or verb

- **Parameters of the model**
  - Pick candidates whose tuple parameters values have a wide confidence interval

- **A hybrid of the above ranking criteria**
Experimental Results

![Graph showing test performance (%) vs. number of training examples for different methods: baseline, frequency, uncertainty, confidence, and hybrid. The graph illustrates the improvement in test performance with an increasing number of training examples.]
Lessons Learned

• Knowledge about the problem-space is helpful for very small training sets.
• Both uncertainty and confidence are effective ranking criteria
• It is better to combine both directly into an evaluation function than to treat each independently.
Application to Training Parsers

• Unlabeled candidates:

  Ford Motor Co. acquired 5 % of the shares in Jaguar PLC.

• Training examples:

  [[[S [NP Ford Motor Co. ] [VP acquired [NP [NP 5 % ] [PP of [NP [NP the shares] [PP in [NP Jaguar PLC]]]]]]]]]] . ]

• Basic learner:
  – PLTIG (Schabes and Waters, 1994; Hwa, 1998)
2 Proposed Evaluation Functions

• Uncertainty
  – Tree entropy
    • Compute the uncertainty over the distribution of parse trees.

• Confidence
  – Determine the sample variance of the parameters
Tree Entropy

- Probability distribution over all parse trees.

\[
\sum_{T_i \in \text{Trees}(O)} \Pr(T_i \mid O, G) = 1
\]

- Uniform distribution ⇒ very uncertain
- Spike distribution ⇒ very certain

\[
f_{te} = -\frac{1}{\log(|\text{Trees}(O)|)} \sum_{T_i \in \text{Trees}(O)} \Pr(T_i \mid O, G) \log \Pr(T_i \mid O, G)
\]
Confidence

• Determine the confidence level of each parameter
  – Compute sample variance
  – Simplifying assumption: compute the sample variance of a binomial distribution
    • Take each derivation step as a Bernoulli trial

• Compute the TUV of the sentence based on the parameters used
  – Sum over all derivations, normalized by sentence length
Inducing PLTIGs

• Candidate pool: 10 sets of 3600 sentences from sect. 02-09 of the WSJ corpus
  – Initial labeled training examples: 100 sentences
  – Examples added per iteration: 100 sentences
• Test: sect. 00 of the WSJ corpus
  – Metric: non-crossing bracket scores
• Comparison
  – Parser trained on data presented sequentially.
  – Parser trained on data selected by Tree Entropy.
  – Parser trained on data selected by Confidence.
PLTIG Experimental Result

![Graph showing parsing accuracy on test set (%) against number of brackets in the training set. The x-axis represents the number of brackets in the training set, ranging from 0 to 40,000. The y-axis represents parsing accuracy on the test set, ranging from 68% to 82%. There are four lines on the graph, each representing a different method: baseline (black diamonds), tree (magenta squares), entropy (green triangles), and confidence (yellow triangles). The graph shows an upward trend in parsing accuracy as the number of brackets in the training set increases.]
Inducing Collins Parsers

• Candidate pool: sect. 02-21 of the WSJ corpus
  – Initial labeled training examples: 500 sentences
  – Examples added per iteration: 100 sentences
• Test: sect. 00 of the WSJ corpus
  – Metric: combined precision & recall scores
• Comparison
  – Parser trained on data presented sequentially.
  – Parser trained on data selected by Tree Entropy.
Collins Parser Result

![Graph showing parsing accuracy on test set against the number of labeled constituents in the training set. The graph compares Baseline and Tree Entropy methods. The accuracy increases as the number of labeled constituents increases. At higher numbers, the accuracy stabilizes at around 86% for Baseline and 88% for Tree Entropy.]
Cross-Parser Training

• Are there a set of “best training sentences?”
  – Can examples selected for one parser be helpful for training another parser?
  – If so, can Tree Entropy find them?

• Comparison
  – Train a PLTIG parser with different data sets
    • Training data presented sequentially (baseline)
    • Training data chosen for the Collins parser
    • Training data selected by Tree Entropy (upper bound).
Cross-Parser Experimental Result

![Graph showing parsing accuracy on test set against number of labeled constituents in training set. The graph includes three lines: baseline, te(direct), and te (collins).]
Discussion

• Alternative interpretations of evaluation functions
  – Quantify the reliability of the outputs of a (partially trained) parser

• Applications
  – Select reliably parsed data for co-training
  – Filter out unreliable parser outputs
Example Application:
Cross-Language Noisy Treebank

Create noisy foreign language treebank by projecting high-quality English parse trees across word-aligned parallel text

- Want to filter out sentences with bad English parse trees from being projected
- Want to filter out bad projected parse trees from being included in the treebank
Conclusion

Sample selection can significantly reduce the number of labeled examples required to train NLP models

- For PP-attachment, using the *hybrid* evaluation function
  - Reduction of 44%
- For parsing, using the *tree entropy* evaluation function
  - Reduction of 36% for PLTIG
  - Reduction of 23% for Collins’ Model 2

Evaluation functions can have additional applications for other NLP frameworks