Hierarchical Structure in Natural Language

- Words are hierarchically organized in syntactic constituents — tree structure
- Part of Speech (POS) and Non-Terminal (NT) tags identify the type of constituent
- Lexicalized annotation of intermediate nodes in the tree

Identifying the syntactic structure ≡ Parsing

✔ Automatic parsing of natural language text is an area of active research

Microsoft Research
Speech.Net
Exploiting Syntactic Structure for Language Modeling
Ciprian Chelba, Frederick Jelinek

- Hierarchical Structure in Natural Language
- **Speech Recognition: Statistical Approach**
- **Basic Language Modeling:**
  - Measures for Language Model Quality
  - Current Approaches to Language Modeling

- **A Structured Language Model:**
  - Language Model Requirements
  - Word and Structure Generation
  - Research Issues
  - Model Performance: Perplexity results on UPenn-Treebank
  - Model Performance: Perplexity and WER results on WSJ/SWB/BN

- **Any Future for the Structured Language Model?**
  - Richer Syntactic Dependencies
  - Syntactic Structure Portability
  - Information Extraction from Text
Speech Recognition — Statistical Approach

\[
\hat{W} = \arg\max_{W} P(W|A) = \arg\max_{W} P(A|W) \cdot P(W)
\]

- \(P(A|W)\) acoustic model: channel probability;
- \(P(W)\) language model: source probability;
- search for the most likely word string \(\hat{W}\).

✔ due to the large vocabulary size — tens of thousands of words — an exhaustive search is intractable.

Microsoft Research
Speech.Net
Basic Language Modeling

Estimate the source probability

\[ P(W), \quad W = w_1, w_2, \ldots, w_n \]

from a training corpus — millions of words of text chosen for its similarity to the expected utterances.

Parametric conditional models:

\[ P_\theta(w_i/w_1 \ldots w_{i-1}), \theta \in \Theta, w_i \in \mathcal{V} \]

- \( \Theta \) parameter space
- \( \mathcal{V} \) source alphabet (vocabulary)

✔ Source Modeling Problem
Measures for Language Model Quality

Word Error Rate (WER)

TRN:  UP  UPSTATE  NEW  YORK  SOMEWHERE  UH  OVER  OVER  HUGE  AREAS
HYP:  UPSTATE  NEW  YORK  SOMEWHERE  UH  ALL  ALL  THE  HUGE  AREAS

1  0  0  0  0  0  0  1  1  1  1  0  0

:4 errors per 10 words in transcription; WER = 40%

Evaluating WER reduction is computationally expensive.

Perplexity (PPL)

\[ PPL(M) = \exp \left( -\frac{1}{N} \sum_{i=1}^{N} \ln [P_M(w_i | w_1 \ldots w_{i-1})] \right) \]

✔ different than maximum likelihood estimation: the test data is not seen during the model estimation process;

✔ good models are smooth:

\[ P_M(w_i | w_1 \ldots w_{i-1}) > \epsilon \]
Current Approaches to Language Modeling

Assume a Markov source of order $n$; equivalence classification of a given context:

$$[w_1 \ldots w_{i-1}] = w_{i-n+1} \ldots w_{i-1} = h_n$$

**Data sparseness:** 3-gram model $(w_i|w_{i-2}, w_{i-1})$

- approx. 70% of the trigrams in the training data have been seen once.
- the rate of new (unseen) trigrams in test data relative to those observed in a training corpus of size 38 million words is 32% for a 20,000-words vocabulary;

**Smoothing:** recursive linear interpolation among relative frequency estimates of different orders $f_k(\cdot), k = 0 \ldots n$ using a recursive mixing scheme:

$$P_n(u|z_1, \ldots, z_n) = \lambda(z_1, \ldots, z_n) \cdot P_{n-1}(u|z_1, \ldots, z_{n-1}) + (1 - \lambda(z_1, \ldots, z_n)) \cdot f_n(u|z_1, \ldots, z_n),$$

$$P_{-1}(u) = \text{uniform}(U)$$

**Parameters:**

$$\theta = \{\lambda(z_1, \ldots, z_n); \text{count}(u|z_1, \ldots, z_n), \forall(u|z_1, \ldots, z_n) \in T\}$$
Exploiting Syntactic Structure for Language Modeling

- Hierarchical Structure in Natural Language
- Speech Recognition: Statistical Approach
- Basic Language Modeling:

🔗 A Structured Language Model:
- Language Model Requirements
- Word and Structure Generation
- Research Issues:
  * Model Component Parameterization
  * Pruning Method
  * Word Level Probability Assignment
  * Model Statistics Reestimation
- Model Performance: Perplexity results on UPenn-Treebank
- Model Performance: Perplexity and WER results on WSJ/SWB/BN
A Structured Language Model

- Generalize trigram modeling (local) by taking advantage of sentence structure (influence by more distant past)
- Use exposed heads $h$ (words $w$ and their corresponding non-terminal tags $l$) for prediction:

$$P(w_i|T_i) = P(w_i|h_{-2}(T_i), h_{-1}(T_i))$$

$T_i$ is the partial hidden structure, with head assignment, provided to $W_i$. 

Microsoft Research
Speech.Net
Language Model Requirements

- Model must operate left-to-right: \[ P(w_i/w_1 \ldots w_{i-1}) \]
- In hypothesizing hidden structure, the model can use only word-prefix \( W_i \), i.e., not the complete sentence \( w_0, \ldots, w_i, \ldots, w_{n+1} \) as all conventional parsers do!
- Model complexity must be limited; even trigram model faces critical data sparseness problems
- Model will assign joint probability to sequences of words and hidden parse structure:
  \[ P(T_i, W_i) \]
the contract ended with a loss of the contract, 7 cents.
The contract ended with a loss of 7 cents.
ended VBD loss NP with IN a DT loss NN of IN the DT contract NN 7 CD cents NNS
ended with a loss of the contract 7 cents.

...; null; predict cents; POS tag cents; adjoin-right-NP; adjoin-left-PP;
ended with a loss of the contract 7 cents

null; predict cents; POStag cents; adjoin-right-NP; adjoin-left-PP; ...; adjoin-left-VP'; null; ...;
Word and Structure Generation

\[ P(T_{n+1}, W_{n+1}) = \]

\[ \prod_{i=1}^{n+1} \left( P(w_i|h_{-2}, h_{-1}) \cdot P(g_i|w_i, h_{-1}.tag, h_{-2}.tag) \cdot P(T_i|w_i, g_i, T_{i-1}) \right) \]

- The **predictor** generates the next word \( w_i \) with probability \( P(w_i = v|h_{-2}, h_{-1}) \)
- The **tagger** attaches tag \( g_i \) to the most recently generated word \( w_i \) with probability \( P(g_i|w_i, h_{-1}.tag, h_{-2}.tag) \)
- The **parser** builds the partial parse \( T_i \) from \( T_{i-1}, w_i, \) and \( g_i \) in a series of **moves** ending with **null**, where a parser move \( a \) is made with probability \( P(a|h_{-2}, h_{-1}) \);
  \[ a \in \{ \text{adjoin-left, NTtag}, \text{adjoin-right, NTtag}, \text{null} \} \]
Research Issues

- Model component parameterization — equivalence classifications for model components:
  
  \[ P(w_i = v|h_{-2}, h_{-1}), P(g_i|w_i, h_{-1}.tag, h_{-2}.tag), P(a|h_{-2}, h_{-1}) \]

- Huge number of hidden parses — need to prune it by discarding the unlikely ones

- Word level probability assignment — calculate \( P(w_i/w_1 \ldots w_{i-1}) \)

- Model statistics estimation — unsupervised algorithm for maximizing \( P(W) \) (minimizing perplexity)
Pruning Method

Number of parses $T_k$ for a given word prefix $W_k$ is $|\{T_k\}| \sim O(2^k)$;

Prune most parses without discarding the most likely ones for a given sentence

Synchronous Multi-Stack Pruning Algorithm

- the hypotheses are ranked according to $\ln(P(W_k, T_k))$
- each stack contains partial parses constructed by the same number of parser operations

The width of the pruning is controlled by:

- maximum number of stack entries
- log-probability threshold
Pruning Method

0 parser op
k predict.

... 

p parser op
k predict.

... 

P_k parser
k predict.

... 

word predictor
and tagger

null parser transitions
parser adjoin/unary transitions
Word Level Probability Assignment

The probability assignment for the word at position $k + 1$ in the input sentence must be made using:

$$P(w_{k+1} / W_k) = \sum_{T_k \in S_k} P(w_{k+1} / W_k T_k) \cdot \rho(W_k, T_k)$$

- $S_k$ is the set of all parses present in the stacks at the current stage $k$
- interpolation weights $\rho(W_k, T_k)$ must satisfy:

$$\sum_{T_k \in S_k} \rho(W_k, T_k) = 1$$

in order to ensure a proper probability over strings $W^*$:

$$\rho(W_k, T_k) = P(W_k T_k) / \sum_{T_k \in S_k} P(W_k T_k)$$
Model Parameter Reestimation

Need to re-estimate model component probabilities such that we decrease the model perplexity.

\[ P(w_i = v|h_{-2}, h_{-1}), P(g_i|w_i, h_{-1}.tag, h_{-2}.tag), P(a|h_{-2}, h_{-1}) \]

Modified Expectation-Maximization (EM) algorithm:

- We retain the \( N \) “best” parses \( \{T^1, \ldots, T^N\} \) for the complete sentence \( W \)
- The hidden events in the EM algorithm are restricted to those occurring in the \( N \) “best” parses
- We seed re-estimation process with statistics gathered from manually parsed sentences
Language Model Performance — Perplexity

- Training set: UPenn Treebank text; 930Kwds; manually parsed;
- Test set: UPenn Treebank text; 82Kwds;
- Vocabulary: 10K — out of vocabulary words are mapped to <unk>
- incorporate trigram in word PREDICTOR:

\[ P(w_i|W_i) = (1 - \lambda) \cdot P(w_i|h_{i-2}, h_{i-1}) + \lambda \cdot P(w_i|h_{i-1}, w_{i-2}), \lambda = 0.36 \]

<table>
<thead>
<tr>
<th>Language Model</th>
<th>L2R Perplexity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DEV set</td>
</tr>
<tr>
<td></td>
<td>no int</td>
</tr>
<tr>
<td>Trigram ( P(w_i</td>
<td>w_{i-2}, w_{i-1}) )</td>
</tr>
<tr>
<td>Seeded with Treebank ( P_0(w_i</td>
<td>h_{i-2}, h_{i-1}) )</td>
</tr>
<tr>
<td>Reestimated ( P(w_i</td>
<td>h_{i-2}, h_{i-1}) )</td>
</tr>
</tbody>
</table>
Language Model Performance — Wall Street Journal

- Training set: WSJ0 “Treebank”-ed text; \(~\approx\) 20Mwds automatically parsed using Ratnaparkhi’s MaxEnt parser trained on UPenn-Treebank text (mismatch);
- Test set: DARPA’93 HUB1 3.4kwds, 213 sentences;
- Vocabulary: 20k open, standard
- incorporate trigram in word PREDICTOR:

\[
P(w_i|W_i) = \lambda \cdot P(w_i|w_{i-1}, w_{i-2}) + (1 - \lambda) \cdot P(w_i|h_{-2}, h_{-1}), \lambda = 0.4
\]

3gram trained on CSR text, 40Mwds
- \(A^*\) lattice decoder

<table>
<thead>
<tr>
<th>Language Model</th>
<th>L2R Perplexity</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DEV set</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TEST set</td>
<td></td>
</tr>
<tr>
<td></td>
<td>no int</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3-gram int</td>
<td></td>
</tr>
<tr>
<td>Trigram</td>
<td>33</td>
<td>147.8</td>
</tr>
<tr>
<td>Initial SLM (E0)</td>
<td>39.1</td>
<td>151.9</td>
</tr>
<tr>
<td>Reestimated SLM (E3)</td>
<td>34.6</td>
<td>144.1</td>
</tr>
</tbody>
</table>

Microsoft Research
Speech.Net
Language Model Performance — Switchboard

- Training set: Switchboard “Treebank”-ed text; 2.29 Mwds; automatically parsed using SLM;
- Test set: Switchboard “Treebank”-ed text; 28 Kwds (WS97 DevTest), 2427 sentences;
- Vocabulary: 22K closed over test set;
- incorporate trigram in word PREDICTOR:

\[ P(w_i|W_i) = \lambda \cdot P(w_i|w_{i-1}, w_{i-2}) + (1 - \lambda) \cdot P(w_i|h_{-2}, h_{-1}), \lambda = 0.6 \]

- A* lattice decoder

<table>
<thead>
<tr>
<th>Language Model</th>
<th>L2R Perplexity</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DEV set</td>
<td>TEST set</td>
</tr>
<tr>
<td></td>
<td>no int 3-gram</td>
<td>no int 3-gram int</td>
</tr>
<tr>
<td>Trigram</td>
<td>22.53</td>
<td>68.56</td>
</tr>
<tr>
<td>Initial SLM (E0)</td>
<td>23.94</td>
<td>72.09</td>
</tr>
<tr>
<td>Reestimated SLM (E3)</td>
<td>22.70</td>
<td>71.04</td>
</tr>
</tbody>
</table>

†The WER improvement is significant at level 0.008 according to a sign test at sentence level
25-best rescoring WER was 40.6%
Language Model Performance — Broadcast News

- Training set: ≈ 14Mwds; automatically parsed using Ratnaparkhi’s MaxEnt parser trained on UPenn-Treebank text (mismatch);
- Test set: DARPA’96 HUB4 devtest;
- Vocabulary: 64K open
- incorporate trigram in word PREDICTOR:

\[
P(w_i|W_i) = \lambda \cdot P(w_i|w_{i-1}, w_{i-2}) + (1 - \lambda) \cdot P(w_i|h_{-2}, h_{-1}), \lambda = 0.4
\]

- 3gram trained on CSR text, 100Mwds
- A* lattice decoder

<table>
<thead>
<tr>
<th>Language Model</th>
<th>L2R Perplexity</th>
<th>WER-F0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DEV set</td>
<td>TEST set</td>
</tr>
<tr>
<td></td>
<td>no int</td>
<td>3-gram int</td>
</tr>
<tr>
<td>Trigram</td>
<td>35.4</td>
<td>217.8</td>
</tr>
<tr>
<td>Initial SLM (E0)</td>
<td>57.7</td>
<td>231.6</td>
</tr>
<tr>
<td>Reestimated SLM (E3)</td>
<td>40.1</td>
<td>221.7</td>
</tr>
</tbody>
</table>

Microsoft Research
Speech.Net
Exploiting Syntactic Structure for Language Modeling

Ciprian Chelba, Frederick Jelinek

Acknowledgments:

- this research was funded by the NSF grant IRI-19618874 (STIMULATE);
- thanks to Eric Brill, William Byrne, Sanjeev Khudanpur, Harry Printz, Eric Ristad, Andreas Stolcke and David Yarowsky for useful comments, discussions on the model and programming support;
- also thanks to:
  Bill Byrne, Sanjeev Khudanpur, Mike Riley, Murat Saraclar for help in generating the SWB, WSJ and BN lattices;
  Adwait Ratnaparkhi for making available the MaxEnt WSJ parser;
  Vaibhava Goel, Harriet Nock and Murat Saraclar for useful discussions about lattice decoding;
Exploiting Syntactic Structure for Language Modeling

- Hierarchical Structure in Natural Language
- Speech Recognition: Statistical Approach
- Basic Language Modeling:
  - Measures for Language Model Quality
  - Current Approaches to Language Modeling
- A Structured Language Model:
  - Language Model Requirements
  - Word and Structure Generation
  - Research Issues
  - Model Performance: Perplexity results on UPenn-Treebank
  - Model Performance: Perplexity and WER results on WSJ/SWB/BN

Any Future for the Structured Language Model?

- Richer Syntactic Dependencies
- Syntactic Structure Portability
- Information Extraction from Text
Richer Syntactic Dependencies

Ciprian Chelba, Peng Xu (CLSP)

Is it beneficial to enrich the syntactic dependencies in the SLM?

- 3 simple ways to enrich the syntactic dependencies by modifying the binarization of parse trees:
  - opposite
  - same
  - both

- perplexity and WER results on UPenn Treebank and Wall Street Journal
“Opposite” Enriching Scheme

predicted word

null

predicted word

null

adjoin_{left,right}

cent

POStag cents; adjoin-right-NP+CD; adjoin-left-PP+NP; ...; adjoin-left-VP’+PP; null; ...;
Enriched Language Model Performance — Perplexity

- Training set: UPenn Treebank text; 930Kwds; manually parsed;
- Test set: UPenn Treebank text; 82Kwds;
- Vocabulary: 10K — out of vocabulary words are mapped to <unk>
- incorporate trigram in word PREDICTOR:

\[
P(w_i|W_i) = (1 - \lambda) \cdot P(w_i|h_{-2}, h_{-1}) + \lambda \cdot P(w_i|w_{i-1}, w_{i-2}), \lambda = 0.6
\]

<table>
<thead>
<tr>
<th>Model</th>
<th>Iter</th>
<th>(\lambda = 0.0)</th>
<th>(\lambda = 0.6)</th>
<th>(\lambda = 1.0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>3</td>
<td>158.75</td>
<td>148.67</td>
<td>166.63</td>
</tr>
<tr>
<td>opposite</td>
<td>3</td>
<td>150.83</td>
<td>144.08</td>
<td>166.63</td>
</tr>
<tr>
<td>same</td>
<td>3</td>
<td>155.29</td>
<td>146.39</td>
<td>166.63</td>
</tr>
<tr>
<td>both</td>
<td>3</td>
<td>153.30</td>
<td>144.99</td>
<td>166.63</td>
</tr>
<tr>
<td>opposite+h_{-3}.NT</td>
<td>3</td>
<td>153.60</td>
<td>144.40</td>
<td>166.63</td>
</tr>
</tbody>
</table>
Enriched Language Model Performance — WER

- Training set: WSJ0 “Treebank”-ed text; \( \approx 20 \) Mwds automatically parsed using Ratnaparkhi’s MaxEnt parser trained on UPenn-Treebank text (mismatch);
- Initial parses binarized and enriched using the **opposite** scheme
- Enrich CONSTRUCTOR context with the \( h_{-3}.NT \) tag
- Test set: DARPA’93 HUB1 3.4kwd, 213 sentences;
- Vocabulary: 20k open, standard
- incorporate trigram in word **PREDICTOR**:

\[
P(w_i|W_i) = \lambda \cdot P(w_i|w_{i-1}, w_{i-2}) + (1 - \lambda) \cdot P(w_i|h_{-2}, h_{-1})
\]

3gram trained on CSR text, 40Mwds
- N-best rescoring

<table>
<thead>
<tr>
<th>Model</th>
<th>Iter</th>
<th>Interpolation weight</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.0</td>
<td>0.2</td>
</tr>
<tr>
<td>baseline SLM WER, %</td>
<td>13.1</td>
<td>13.1</td>
</tr>
<tr>
<td>opposite SLM WER, %</td>
<td>12.7</td>
<td>12.8</td>
</tr>
<tr>
<td>opposite+( h_{-3}.NT ) SLM WER, %</td>
<td>12.3</td>
<td>12.4</td>
</tr>
</tbody>
</table>
Syntactic Structure Portability

Is the knowledge of syntactic structure as embodied in the SLM parameters portable across domains?

- ATIS-III corpus
- Training set: 76k words
- Test set: 9.6k words
- Vocabulary: 1k; OOV rate: 0.5%

Initial Statistics:

- parse the training data (approximately 76k words) using Microsoft’s NLPwin and then initialize the SLM from these parse trees
- use the limited amount of manually parsed ATIS-3 data (approximately 5k words)
- use the manually parsed data in the WSJ section of the Upenn Treebank.
Syntactic Structure Portability: Perplexity Results

✔ regardless of initialization method, further N-best EM reestimation iterations are carried out on the entire training data (76k wds)

- incorporate trigram in word PREDICTOR:

\[ P(w_i|W) = \lambda \cdot P(w_i|w_{i-1}, w_{i-2}) + (1 - \lambda) \cdot P(w_i|h_{-2}, h_{-1}), \lambda = 0.6 \]

<table>
<thead>
<tr>
<th>Initial Stats</th>
<th>Iter</th>
<th>$\lambda = 0.0$</th>
<th>$\lambda = 0.6$</th>
<th>$\lambda = 1.0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLPwin parses</td>
<td>0</td>
<td>21.3</td>
<td>16.7</td>
<td>16.9</td>
</tr>
<tr>
<td>NLPwin parses</td>
<td>13</td>
<td>17.2</td>
<td>15.9</td>
<td>16.9</td>
</tr>
<tr>
<td>SLM-atis parses</td>
<td>0</td>
<td>64.4</td>
<td>18.2</td>
<td>16.9</td>
</tr>
<tr>
<td>SLM-atis parses</td>
<td>13</td>
<td>17.8</td>
<td>15.9</td>
<td>16.9</td>
</tr>
<tr>
<td>SLM-wsj parses</td>
<td>0</td>
<td>8311</td>
<td>22.5</td>
<td>16.9</td>
</tr>
<tr>
<td>SLM-wsj parses</td>
<td>13</td>
<td>17.7</td>
<td>15.8</td>
<td>16.9</td>
</tr>
</tbody>
</table>
Syntactic Structure Portability: WER Results

- rescoring N-best (N=30) lists generated by the Microsoft Whisper speech recognizer. The 1-best WER — baseline — is 5.8%. The best achievable WER on the N-best lists generated this way is 2.1% — ORACLE WER

<table>
<thead>
<tr>
<th>Initial Stats</th>
<th>Iter</th>
<th>$\lambda = 0.0$</th>
<th>$\lambda = 0.6$</th>
<th>$\lambda = 1.0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLPwin parses</td>
<td>0</td>
<td>6.4</td>
<td>5.6</td>
<td>5.8</td>
</tr>
<tr>
<td>NLPwin parses</td>
<td>13</td>
<td>6.4</td>
<td>5.7</td>
<td>5.8</td>
</tr>
<tr>
<td>SLM-atis parses</td>
<td>0</td>
<td>6.5</td>
<td>5.6</td>
<td>5.8</td>
</tr>
<tr>
<td>SLM-atis parses</td>
<td>13</td>
<td>6.6</td>
<td>5.7</td>
<td>5.8</td>
</tr>
<tr>
<td>SLM-wsj parses</td>
<td>0</td>
<td>12.5</td>
<td>6.3</td>
<td>5.8</td>
</tr>
<tr>
<td>SLM-wsj parses</td>
<td>13</td>
<td>6.1</td>
<td>5.4</td>
<td>5.8</td>
</tr>
</tbody>
</table>

✔ The model initialized on WSJ parses outperforms the other initialization methods based on in-domain annotated data, achieving a significant 0.4% absolute and 7% relative reduction in WER
Conclusions

✔ original approach to language modeling that takes into account the hierarchical structure in natural language

✔ devised an algorithm to reestimate the model parameters such that the perplexity of the model is decreased

✔ showed improvement in both perplexity and word error rate over current language modeling techniques

✔ model initialization is very important

✔ code and data is available at http://www.research.microsoft.com/~chelba

Future Work

✘ better parametrization/statistical modeling tool in model components, especially PRE-DICTOR and PARSER; potential improvement in PPL from guessing the final best parse is large.
Information Extraction Using the Structured Language Model

Ciprian Chelba, Milind Mahajan

- Information Extraction from Text
- SLM for Information Extraction
- Experiments
Information Extraction from Text

- Information extraction viewed as the recovery of a two level semantic parse $S$ for a given word sequence $W$
- Sentence independence assumption: the sentence $W$ is sufficient for identifying the semantic parse $S$

```
Schedule meeting with Megan Hokins about internal lecture at two thirty p.m.
```

GOAL: Data driven approach with minimal annotation effort: clearly identifiable semantic slots and frames
SLM for Information Extraction

Training:

**Initialization**  Initialize SLM as a syntactic parser from treebank

**Syntactic Parsing**  Train SLM as a matched constrained parser and parse the training data: boundaries of semantic constituents are matched

**Augmentation**  Enrich the non/pre-terminal labels in the resulting treebank with semantic tags

**Syntactic+Semantic Parsing**  Train SLM as an L-matched constrained parser: boundaries and tags of the semantic constituents are matched

Test:

- **Syntactic+Semantic Parsing**  of test sentences; retrieve the semantic parse by taking the semantic projection of the most likely parse:

\[ S = SEM(\arg\max_{T_i} P(T_i, W)) \]
Experiments

MiPad data (personal information management)

- training set: 2,239 sentences (27,119 words) and 5,431 slots
- test set: 1,101 sentences (8,652 words) and 1,698 slots
- vocabularies: WORD: 1,035wds, closed over test data; FRAME: 3; SLOT: 79;

<table>
<thead>
<tr>
<th>Training Iteration</th>
<th>Error Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Training Slot</td>
</tr>
<tr>
<td>Stage 2</td>
<td>Stage 4</td>
</tr>
<tr>
<td>Baseline</td>
<td>43.41</td>
</tr>
<tr>
<td>0, MiPad/NLPwin</td>
<td>0</td>
</tr>
<tr>
<td>0, UPenn Trbnk</td>
<td>0</td>
</tr>
<tr>
<td>1, UPenn Trbnk</td>
<td>1</td>
</tr>
<tr>
<td>1, UPenn Trbnk</td>
<td>2</td>
</tr>
</tbody>
</table>

- baseline is a semantic grammar developed manually that makes no use of syntactic information
- initialize the syntactic SLM from in-domain MiPad treebank (NLPwin) and out-of-domain Wall Street Journal treebank (UPenn)
- 3 iterations of N-best EM parameter reestimation algorithm
Conclusions

✔ Presented a data driven approach to information extraction that outperforms a manually written semantic grammar

✔ Coupling of syntactic and semantic information improves information extraction accuracy, as shown previously by Miller et al., NAACL 2000

Future Work

✘ Use a statistical modeling technique that makes better use of limited amounts of training data and rich conditioning information — maximum entropy

✘ Aim at information extraction from speech: treat the word sequence as a hidden variable, thus finding the most likely semantic parse given a speech utterance