Summary of Workshop Goals

- Improve Chinese-English Statistical Machine Translation
  - Feature functions: simple → complicated (string-based, part of speech based, chunk based, tree based, tree-tree alignment based)
  - Finding feature functions by error analysis and feature hunting based on contrastive errors in the $n$-best devset
  - Rescoring using Risk Minimization, SVM/perceptron with margin, Min Bayes Risk
Weeks 2 and 3

- Data Preprocessing all over again

- Contrastive error analysis: parse trees of oracle vs. produced (Anoop)

- New oracle based on BLEU rather than WER+PER: BLEU of 45% instead of 38

- Moved to larger dev set (100x100 → 993x1000 → 5765x1000)

- More Features
Experiments: 993 Chinese sents, 1000-best English

<table>
<thead>
<tr>
<th>Blame</th>
<th>Experiment</th>
<th>Bleu Score (Dev Test (+/-1%))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Franz</td>
<td>baseline</td>
<td>30.05 (45)</td>
</tr>
<tr>
<td>Franz</td>
<td>oracle</td>
<td></td>
</tr>
<tr>
<td>Franz</td>
<td>Model1+POS-LM+paren</td>
<td>32.52 32.89</td>
</tr>
<tr>
<td>Anoop</td>
<td>Grand Feature Combination</td>
<td>30.73 32.74</td>
</tr>
<tr>
<td>Kenji</td>
<td>Model1</td>
<td>32.67 32.71 (w/ postprocess)</td>
</tr>
<tr>
<td>Franz</td>
<td>missingWords</td>
<td>32.03 32.18</td>
</tr>
<tr>
<td>Viren</td>
<td>8-gram-POS-LM</td>
<td>?? 32.10</td>
</tr>
<tr>
<td>Zhen</td>
<td>Parens+Quotes</td>
<td>32.28 32.08</td>
</tr>
<tr>
<td>Zhen</td>
<td>head-word-verb</td>
<td>32.43 31.95</td>
</tr>
<tr>
<td>Katherine</td>
<td>RefWordPenalty</td>
<td>32.29 31.19</td>
</tr>
<tr>
<td>Franz</td>
<td>parsing-prob</td>
<td>31.84 30.98</td>
</tr>
</tbody>
</table>
## Grand Feature Combination

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>Feature Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>word-level model</td>
<td>model1</td>
<td>IBM model1 prob (sum)</td>
</tr>
<tr>
<td>word-level model</td>
<td>wordpop-idf</td>
<td>prefer words that tend to appear in all 4 references</td>
</tr>
<tr>
<td>word-level model</td>
<td>MPMQ</td>
<td>matching parens, matching quotes</td>
</tr>
<tr>
<td>word-level model</td>
<td>rev-lm-1</td>
<td>reverse language model on training</td>
</tr>
<tr>
<td>word-level model</td>
<td>rev-lm-2</td>
<td>reverse language model on training and test</td>
</tr>
<tr>
<td>Part of speech</td>
<td>poslm</td>
<td>POS 8gram language model (Eng) prob returned by maxent POS tagger</td>
</tr>
<tr>
<td>Part of speech</td>
<td>POS-posterior</td>
<td>POS posterior pos probability returned by maxent POS tagger</td>
</tr>
<tr>
<td>Tree-based</td>
<td>MConn</td>
<td>balanced co-ordination in Eng tree</td>
</tr>
<tr>
<td>Tree-based</td>
<td>parserScoreDivLM</td>
<td>parser prob / trigram LM prob</td>
</tr>
<tr>
<td>Tree-based</td>
<td>rightBranch</td>
<td>prefers right branching tree</td>
</tr>
<tr>
<td>Tree-based</td>
<td>numPPs</td>
<td>num of prepositonal phrases</td>
</tr>
</tbody>
</table>
Ongoing Work

- Lexical dependency features (Drago)
- Tree-to-String alignments (Kenji)
- Tree-to-Tree alignments (Dan)
  (oracle expt: prefers oracle 60% of the time)
- Features based on alignment in N-best (Viren, Anoop, etc)
- Perceptron Training (Libin: stay tuned)
- Minimum Bayes Risk (Shankar: stay tuned)
Dependency features

- “small” dependency feature $d(w_1, w_2) = \log \frac{p(w_1 \rightarrow w_2)}{p(w_2 \rightarrow w_1)}$

- Example:
  - $f(”mark” \rightarrow ”achievement”) = 8$
  - $f(”achievement” \rightarrow ”mark”) = 4$
  - $d(”mark”,”achievement”) = \log(8/4) = \log 2$

- “big” dependency feature $D = f(d_i)$

Drago Radev
Example

china ’s 14 open border cities achievements remarkable
achievement->china
achievement->14
achievement->open
achievement->border
achievement->city
achievement->remarkable

china ’s 14 open border cities building remarkable achievements
city->china
city->14
city->open
city->border
city->build
build->achievement
achievement->remarkable

china ’s 14 open border cities , remarkable achievements
,->city
city->china
city->14
city->open
city->border
,->achievement
achievement->remarkable
Reranking with Linear Classifiers

- Assumption: The *weight vector* of the separating hyperplane and the *score metrics* are in the same direction.
  distance to the hyperplane = quality of a translation

- Large dimensional space. Allowing the use of various features.

- Inseparable in a space with 12 or 20 dimensions
  - Do not use all the translations in the training.
  - *λ-trick*: a new dimension for each sample (Herbrich 02)
  - It still doesn’t work. Each sentence requires different *bias*. 
Variants of Perceptrons for Reranking

• **Alg 1**: Pairwise translations as samples.
  - $\text{class}(T_{ij}, T_{ik}) \in \{-1, +1\}$, for two translations $T_{ij}$ and $T_{ik}$ of a sentence $i$.
  - $T_{ij}$ and $T_{ik}$ in top and bottom $p\%$.
  - Fast implementation.

• **Alg 2**: Multi-bias perceptron with margin
  - Input: translations $(x_{ij}, y_{ij})_{i=1..s, j=1..n}$, where $x_{ij}$ in top or bottom $p\%$.
  - Goal: $\mathbf{w}, b_1, b_2, ..., b_s$.
  - Update: If $y_{ij}(\mathbf{w}^t x_{ij}) \leq \tau$, then $\mathbf{w}^{t+1} = \mathbf{w}^t + \eta y_{ij} x_{ij}$, and $b_i^{t+1} = b_i^t + \eta y_{ij} R^2$.
  - Converge: $t \leq 2(s + 1)((\frac{R}{M})^2) + \frac{\tau}{\eta M^2}$, if the training data is separable with margin $M$.
  - $\mathbf{w}$ is global for all sentences, providing the direction of the score metric for ranking.
Reranking results on 993 dataset (Alg 2 + $\lambda$-trick: 31.66; Baseline: 30.06)
Minimum Bayes-Risk (MBR) Decoders for SMT

• MBR decoders have been shown to optimize performance in speech and language processing tasks using application dependent loss functions

• Goal: Design MBR decoders to optimize translation quality under various loss functions that measure translation performance

  – We assume that we have available
    * A Baseline Translation/Language Model to give the scores $P((E, A)|F)$
    * A set $\mathcal{E}$ of most likely translations of $F$
      For each translation $E'$ in this set, we have word-to-word alignments between $E'$ and $F$ and a parse-tree $T'$.
    * A parse-tree $T_F$ for the source sentence $F$
    * A Loss function $L((E, A, T), (E', A', T'); T_F)$ that measures the quality of $E'$ wrt $E$ using information from word sequences $(E, E')$, alignments $(A, A')$ and their parse-trees $(T, T')$ and $T_F$.

  – MBR Decoder
    $$(\hat{E}, \hat{A}, \hat{T}) = \arg\min_{(E', A', T') \in \mathcal{E}} \sum_{(E, A) \in \mathcal{E}} L((E, A, T), (E', A', T'); T_F) P((E, A)|F)$$

  – $\mathcal{E}$ is an N-best List in these experiments

Shankar Kumar

Syntax for SMT Team Update
Performance of the MBR Decoder

- Development Set: 993 sentences, 1000-best lists, Baseline from ISI
  - Loss functions defined at the sentence level

<table>
<thead>
<tr>
<th>Decoder</th>
<th>BLEU</th>
<th>mWER</th>
<th>mPER</th>
<th>Parse-Error</th>
<th>NIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>30.73</td>
<td>67.31</td>
<td>43.22</td>
<td>92.8331</td>
<td>9.165</td>
</tr>
<tr>
<td>MBR Decoder</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLEU</td>
<td>31.12</td>
<td>67.47</td>
<td>42.95</td>
<td>-</td>
<td>9.254</td>
</tr>
<tr>
<td>WER</td>
<td>30.93</td>
<td>66.68</td>
<td>42.69</td>
<td>-</td>
<td>9.214</td>
</tr>
<tr>
<td>PER</td>
<td>30.86</td>
<td>67.01</td>
<td>42.18</td>
<td>-</td>
<td>9.278</td>
</tr>
<tr>
<td>Parse-Loss</td>
<td>30.77</td>
<td>67.34</td>
<td>43.22</td>
<td>92.8328</td>
<td>9.176</td>
</tr>
</tbody>
</table>

- Test Set: 878 sentences, 1000-best lists - Best system using Minimum Error Training with all syntactic feature functions

<table>
<thead>
<tr>
<th>Decoder</th>
<th>BLEU</th>
<th>mWER</th>
<th>mPER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>32.9</td>
<td>61.72</td>
<td>38.31</td>
</tr>
<tr>
<td>MBR-BLEU</td>
<td>33.2</td>
<td>61.7</td>
<td>38.29</td>
</tr>
</tbody>
</table>