Forest MIRA

Forest rescoring in Joshua for MIRA training

Team Members
Matt Post (JHU)
Gideon Wenniger (Amsterdam)
Forest MIRA

Margin-infused relaxed algorithm

Crammer & Singer (2003, 2006)

Large scale discriminative tuner; maximizes model score difference between “hope” and “fear” translations

Chiang et al. (2008, 2009); Chiang (2012)

Batch implementation in Moses (kbmira)

Cherry & Foster (2012)
MIRA

Works by maximizing the margin between hope and fear items

hope items maximize

model score + $\lambda$ BLEU score

fear items maximize

model score – $\lambda$ BLEU score
Problem

Moses’ kbmira extracts hope and fear items from the k-best list
Visualization
(Chiang, 2012)

The k-best list is not representative of the model space!

![Graph showing visualization](image)
Visualization
(Chiang, 2012)

The k-best list is not representative of the model space!
Visualization
(Chiang, 2012)

The k-best list is not representative of the model space!
Goal

Implement forest rescoring in Joshua (Moses) for proper hope/fear updating in Cherry & Foster’s kbmira implementation
Sub-goals

• BLEU approximation that factors over the hypergraph
• Forest rescoring and extraction
• Write many-featured feature function
Factorizable BLEU

$$BP \times \exp \left( \sum_{n=1}^{n} \frac{1}{4} \log p_n \right)$$

Don’t clip counts for each $p_n$

Scale reference length to proportion of input consumed

$$\lambda = \frac{\text{(span width)}}{\text{(sentence length)}}$$

$$BP = \frac{\text{elen}}{\lambda \cdot \text{reflen}}$$
Scoring hyperedges

Decoder combines chart items by adding model scores (vectors of scores)

\[
\text{score}(e') = \text{score}(e_1) + \text{score}(e_2)
\]

Doesn’t work with BLEU!

\[
\text{BLEU}(e') \neq \text{BLEU}(e_1) + \text{BLEU}(e_2)
\]
Instead, we have to store the sufficient statistics on each hyperedge

\[
\begin{array}{ccccccc}
\text{counts} & n=1 & n=2 & n=3 & n=4 & \text{len} \\
2 & 1 & 0 & 0 & 2 \\
\end{array}
\]

These we can sum together and then use to compute \sim\text{BLEU}
Computing ~BLEU

Terminal productions: accumulate all ngrams

... barack obama, president ...
Terminal productions: accumulate all ngrams

... barack obama, president ...

[X] → barack obama

1 2 3 4 len
Computing \(\sim\) BLEU

Terminal productions: accumulate all ngrams

\[
[X] \rightarrow \text{barack obama} \\
\begin{array}{cccc}
1 & 2 & 3 & 4 & \text{len} \\
2 & & & & \\
\end{array}
\]

... barack obama, president ...
Computing ~BLEU

Terminal productions: accumulate all ngrams

... barack obama, president ...

\[ [X] \rightarrow \text{barack obama} \]

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>len</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Computing ~BLEU

Terminal productions: accumulate all ngrams

...barack obama, president ...

[X] → barack obama

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>len</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Computing ~BLEU

Terminal productions: accumulate all ngrams

... barack obama, president ...

\[
\begin{array}{c|c|c|c|c|c}
  & 1 & 2 & 3 & 4 & \text{len} \\
\hline
[X] & 2 & 1 & 0 & 0 & \\
\end{array}
\]
Computing $\sim$ BLEU

Terminal productions: accumulate all ngrams

... barack obama , president ...
Computing $\sim$BLEU

Mixed productions: accumulate all ngrams that overlap tail node boundaries
Computing ~BLEU

... barack obama , president ...
Computing ~BLEU

... barack obama, president ...
Computing ~BLEU

... barack obama , president ...

We count only ngrams in the new rule or crossing the boundary
Computing $\sim$ BLEU

... barack obama , president ...

barack obama | , president

We count only ngrams in the new rule or crossing the | boundary
Computing ~BLEU

... barack obama , president ...

We count only ngrams in the new rule or crossing the boundary
Computing \textasciitilde BLEU

... barack obama , president ...

\begin{itemize}
\item \texttt{[X]} \rightarrow \texttt{[X]} , president
\begin{tabular}{c|c|c|c|c|c}
1 & 2 & 3 & 4 & \text{len} \\
2 & 2 & & & \\
\end{tabular}
\end{itemize}

\begin{itemize}
\item \texttt{[X]} \rightarrow \texttt{barack obama}
\begin{tabular}{c|c|c|c|c|c}
1 & 2 & 3 & 4 & \text{len} \\
2 & 1 & 0 & 0 & 2 \\
\end{tabular}
\end{itemize}

\texttt{barack obama} | , president

\textit{We count only ngrams in the new rule or crossing the | boundary}
Computing ~BLEU

... barack obama , president ...

```
[ ] → [ ] , president
    1 2 3 4 len
    2 2 2

[ ] → barack obama
    1 2 3 4 len
    2 1 0 0 2
```

(barack obama | , president)

We count only ngrams in the new rule or crossing the | boundary
Computing \~\text{BLEU}

\[ \text{barack obama} \mid , \text{president} \]

We count only ngrams in the new rule or crossing the \mid boundary
Computing \sim \text{BLEU}

... barack obama, president ...

We count only ngrams in the new rule or crossing the boundary
Computing ~BLEU

... barack obama, president ...

We count only ngrams in the new rule or crossing the boundary.

combined
Rescoring

Once we have the BLEU scores on all edges, we apply standard k-best three times

- Extract the model best (per usual)
- Rescore with model + ~BLEU (hope)
- Rescore with model – ~BLEU (fear)
Experiments

German-English Europarl + Common Crawl

Tuning on newstest2010

Testing on newstest2012

Running MIRA for 8 iterations, test model with best tuning score

10 features
Tuning

- MIRA
- Forest MIRA (k=100)
- Forest MIRA (k=300)
## Test

<table>
<thead>
<tr>
<th>Model</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIRA</td>
<td>23.6</td>
</tr>
<tr>
<td>Forest MIRA (k=300)</td>
<td>23.1</td>
</tr>
<tr>
<td>Forest MIRA (k=100)</td>
<td>22.9</td>
</tr>
</tbody>
</table>
Problems

Can’t use left-state minimization! We need the full state and don’t want the complexity of state splitting

This means that we tune without left-state minimization, adding it back in for test, introducing a mismatch between models
Missing

Haven’t included rolling / decaying BLEU “pseudo corpus” stats

Haven’t tested on large feature sets (where we expect to see the most benefits, cf. Cherry (2013))
Summary

Shows promise

Implementation is a little messy but not difficult