The CMU Syntax-Augmented Machine Translation System: SAMT on Hadoop with N-best alignments

Andreas Zollmann, Ashish Venugopal, Stephan Vogel
{zollmann,ashishv,stephan.vogel}@cs.cmu.edu

October 21, 2008
Components of an MT system

- Statistical machine translations systems are component driven
  - Selection and preparation of parallel and monolingual corpora
  - Word alignment [Brown1993] on parallel corpora
  - Building n-gram language models from monolingual corpora
  - Phrase extraction and feature estimation from word alignment
  - Rule extraction (with optional parses) from phrase extraction
  - Translating with translation and language models (and more)
  - Training of feature weights via iterative translation and optimization

- The performance of each component has the potential to affect translation quality!
Addressing the problem of scale in MT

MapReduce training of Word Alignment Models


Randomized Monolithic Models


Distributed Monolithic Models


The SAMT machine translation pipeline

- Our flavor: Syntax Augmented Machine Translation (SAMT)
- Probabilistic Synchronous Context Free Grammars (PSCFGs)
- Rules with nonterminals are labeled based on parse trees
- Rules compose at nonterminals to form translations

\[ PRP \rightarrow \text{il he } \# \lambda_1 \cdots \lambda_n \]
\[ VB \rightarrow \text{va, go } \# \lambda_1 \cdots \lambda_n \]
\[ S \rightarrow \text{il ne VB}_1 \text{ pas , he does not VB}_1 \# \lambda_1 \cdots \lambda_n \]
\[ S \rightarrow \text{PRP}_1 \text{ ne VB}_2 \text{ pas , PRP}_1 \text{ does not VB}_2 \# \lambda_1 \cdots \lambda_n \]
Runtime challenges in SAMT

- Rule extraction runtime
- Resulting grammar on training data is very large
- Decoding can be significantly slower than phrase-based approaches
Considerations in porting SAMT to Hadoop

- Construct a phase-based pipeline for experimental reuse
- Keep memory requirements low and disk usage to a minimum
- Allocate and de-allocate machine on a per-phase basis
- Use existing code-base under Hadoop streaming only
MapReduce specifications

Map specification:
- MapInput: Input data to Map process (automatically split at line boundaries by Hadoop).
- MapOptions: Options to the Map process
- MapOutput: Key-value pairs output by Map process

Reduce specification:
- ReduceInput: Key-value pairs, all values share the same key
- ReduceOption: Options to Reduce process
- ReduceOutput: Unstructured output from Reduce process
- ReduceOutput(Side-effects): Additional files created by Reduce process
Phrase Extraction - Specification

Phrase Extraction
Create phrase pairs from alignment data

- MapInput: Input lines of the form $f, e, a(e, f), \pi(e)$
- MapOptions: Maximum extractable phrase length
- MapOutput: $key = ()$, $value = \langle f, e, Phrases(e, f), \pi(e) \rangle$
Rule Extraction - Specification

**RuleExtraction Map**

Generate PSCFG rules with nonterminal

- **MapInput**: Each line contains \( f, e, \text{Phrases}(e, f), \pi(e) \)
- **MapOptions**: Maximum \( \# \) of nonterminals per rule, maximum length of \( \gamma \), options to select rule NTs from \( \pi \)
- **MapOutput**: key = \( \text{ul}(\gamma) \) value = \( \langle \gamma, \alpha, \text{lhs}, 1 \rangle \) and key = \( \text{lhs} \) value = 1.

**RuleExtraction Reduce**

Discard rare rules and compute features for each rule

- **ReduceInput**: All rules that share the same \( \text{ul}(\gamma) \)
- **ReduceOptions**: Minimum occurrence counts for lexical and nonlexical rules, \( \min p(trg, \text{lhs}|src) \)
- **ReduceOutput**: Uniqued rules with features: unlabelled source frequency, labelled source frequency and rule frequency.
### Phrase Extraction - Runtimes

<table>
<thead>
<tr>
<th>System</th>
<th>Map (mins)</th>
<th>Reduce (mins)</th>
<th>Compressed Output (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IWSLT hier</td>
<td>0.1</td>
<td>NA</td>
<td>6</td>
</tr>
<tr>
<td>IWSLT syntax</td>
<td>0.1</td>
<td>NA</td>
<td>8</td>
</tr>
<tr>
<td>230M hier</td>
<td>2</td>
<td>NA</td>
<td>2627</td>
</tr>
<tr>
<td>230M syntax</td>
<td>2</td>
<td>NA</td>
<td>3576</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System</th>
<th>Map (mins)</th>
<th>Reduce (mins)</th>
<th>Compressed Output (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IWSLT hier</td>
<td>1.5</td>
<td>1.5</td>
<td>232</td>
</tr>
<tr>
<td>IWSLT syntax</td>
<td>2</td>
<td>4</td>
<td>527</td>
</tr>
<tr>
<td>230M hier</td>
<td>3 hrs 20</td>
<td>1 hr</td>
<td>1753</td>
</tr>
<tr>
<td>230M syntax</td>
<td>4 hrs 10 mins</td>
<td>2 hrs 20 mins</td>
<td>2478</td>
</tr>
</tbody>
</table>

**Table:** Wall-clock time for Map and Reduce steps, using 40 processors for each resource condition
Rule Filtering - Specification

RuleFiltering Map: Partition rules based on applicability for each sentence in test corpus

- MapInput: Rules from Rule Extraction stage (single source as key with multiple rules as values)
- MapOptions: test set (source-side) corpus to filter rules
- MapOutput: key = sno value = ⟨lhs, γ, α, φ⟩ such that all words in the γ are in sentence number sno in the source corpus

RuleFiltering Reduce: Add features and system rules to produce sentence-specific grammar file

- ReduceInput: All rules and special counts for a single sentence
- ReduceOptions: Additional models to generate features in φ
- ReduceOutput: Rules with fully formed φ for a single
LM Filtering - Specification

LM Filtering Map

Filter n-grams from LM based on applicability for each sentence in test corpus

- MapInput: Each line is a line from an ARPA format LM
- MapOptions: Access to a $sno \rightarrow vocabulary$ map from the filtering stage (loaded into memory)
- MapOutput: key = $sno$ value = $t_1 \cdots t_n$ if every $t_i$ is in the target vocabulary of $sno$.

LM Filtering Reduce

Create sentences specific ARPA-format LMs

- ReduceInput: All n-grams that are compliant with a single sentence’s vocabulary
- ReduceOutput: Statistics over n-grams are computed and output as a header to form a complete ARPA LM
Decoding - Specification

Decoding Map
Generate an N-Best list of translations for each source sentence

- MapInput: A single sentence to translate per line
- MapOptions: Options typically passed to a decoder to run translation. We also specify a path to a HDFS directory containing per-sentence translation and language models.
- MapOutput: key = sno value = n-best list
Minimum Error Rate Training

Minimum Error Rate Training, Och, 2003

Input: N-Best lists
Output: Parameters $\lambda$ that maximize automatic evaluation metrics
Multiple initial configurations are important

- In MapReduce, it is not possible to tell reducer $X$ to use parameter set $X$
- We output $\langle \text{params}, \text{data} \rangle$ as key value pairs
- Each Reducer receives one parameter set and associated data
Explicit BOS/EOS modelling

- Treat BOS and EOS as regular words:
- $S \rightarrow <s> \text{NP } \text{VP } </s> \# <s> \text{VP } \text{NP} . </s>$
- $S \rightarrow <s> \text{PRP } \text{MD } \text{VP } </s> \# <s> \text{MD } \text{PRP } \text{VP}? </s>$

- Only allow the ones spanning full sentence
BOS/EOS Modelling: Results

<table>
<thead>
<tr>
<th>System</th>
<th>Dev. BLEU</th>
<th>2007 BLEU</th>
<th>2008 BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>IWSLT Hier.</td>
<td>0.278</td>
<td>0.360</td>
<td>0.427</td>
</tr>
<tr>
<td>IWSLT Hier. with full-sentence rules</td>
<td>0.277</td>
<td>0.367</td>
<td>0.460</td>
</tr>
<tr>
<td>IWSLT Syntax</td>
<td>0.296</td>
<td>0.335</td>
<td>0.430</td>
</tr>
<tr>
<td>IWSLT Syntax with full-s. rules</td>
<td>0.301</td>
<td>0.361</td>
<td>0.440</td>
</tr>
</tbody>
</table>

- Not much impact on development-set performance
- Impressive increases in BLEU score on the test sets
Using $N$-best alignments in the pipeline

- extract $N$-best alignments in each direction
- select top $N$ from $N^2$ bidirectional alignment pairs according to $p(\langle a_f, a_r \rangle) = (p_f(a_f) \times p_r(a_r))^\alpha$
- Renormalize: $\hat{p}(a_i) = p(a_i) / \sum_{j=1}^{N} p(a_j)$
- rule $r$’s total count for the sentence pair $\langle f, e \rangle$ is thus:

$$\sum_{i=1}^{N} \hat{p}(a_i) \cdot \begin{cases} 1 & \text{if } r \text{ can be extracted from } e, f, a_i \\ 0 & \text{otherwise} \end{cases}$$ (1)
## Results

<table>
<thead>
<tr>
<th>System</th>
<th># Rules (per sent.)</th>
<th>Dev</th>
<th>2007 BLEU</th>
<th>2008 BLEU</th>
<th>2007 Time (s)</th>
<th>2008 Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax N = 1</td>
<td>400K</td>
<td>0.309</td>
<td>0.355</td>
<td>0.453</td>
<td>8108</td>
<td>8367</td>
</tr>
<tr>
<td>Syntax N = 5</td>
<td>680K</td>
<td>0.322</td>
<td>0.374</td>
<td>0.470</td>
<td>15376</td>
<td>15577</td>
</tr>
<tr>
<td>Syntax N = 10</td>
<td>900K</td>
<td>0.313</td>
<td>0.382</td>
<td>0.467</td>
<td>19298</td>
<td>19469</td>
</tr>
<tr>
<td>Syntax N = 50</td>
<td>1500K</td>
<td>0.316</td>
<td>0.370</td>
<td>0.478</td>
<td>29500</td>
<td>30894</td>
</tr>
<tr>
<td>Hier N = 1</td>
<td>10K</td>
<td>0.277</td>
<td>0.367</td>
<td>0.460</td>
<td>895</td>
<td>1451</td>
</tr>
<tr>
<td>Hier N = 5</td>
<td>12K</td>
<td>0.286</td>
<td>0.374</td>
<td>0.472</td>
<td>906</td>
<td>1476</td>
</tr>
<tr>
<td>Hier N = 10</td>
<td>13K</td>
<td>0.291</td>
<td>0.382</td>
<td>0.477</td>
<td>944</td>
<td>1516</td>
</tr>
<tr>
<td>Hier N = 50</td>
<td>14K</td>
<td>0.282</td>
<td>0.384</td>
<td>0.463</td>
<td>979</td>
<td>1596</td>
</tr>
</tbody>
</table>

- $N$-best alignments help
- Syntax more than 2 BP better than Hier on dev. set, but inconclusive on test sets
- wall-clock times for Syntax $N = 10$: 2820 s ('07); 1200 s ('08);
- wall-clock times for Syntax $N = 50$: 5520 s ('07); 2280 s ('08);
Conclusions

- Developed a Hadoop-based platform for SMT experimentation
- Use of MapReduce permits experimentation with wider pipelines, such as integrating Nbest alignment evidence
- High variance in IWSLT test set BLEU scores makes results difficult to interpret conclusively
- System is open-source: www.cs.cmu.edu/~zollmann/samt