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

C. Dyer, A. Cordova, A. Mont, and J. Lin. Fast, easy, and cheap: Construction of statistical machine translation models with mapreduce. In Proc. of the Workshop on SMT, ACL, 2008.

Randomized Monolithic Models

D. Talbot, T. Brants. Randomized Language Models via Perfect Hash-functions. In Proc. of the ACL, 2008

Distributed Monolithic Models

Ying Zhang, Almut Silja Hildebrand, and Stephan Vogel. Distributed Language Modeling for N-best List Re-ranking. In Proc. of EMNLP, 2006

T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean. Large language models in MT. Proc. of EMNLP-CoNLL, 2007

The SAMT machine translation pipeline

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

$$
\begin{aligned}
PRP &\rightarrow \text{il he } \# \ \lambda_1 \cdots \lambda_n \\
VB &\rightarrow \text{va, go } \# \ \lambda_1 \cdots \lambda_n \\
S &\rightarrow \text{il ne } \text{VB}_1 \text{ pas , he does not } \text{VB}_1 \ \# \ \lambda_1 \cdots \lambda_n \\
S &\rightarrow \text{PRP}_1 \text{ ne } \text{VB}_2 \text{ pas , PRP}_1 \text{ does not } \text{VB}_2 \ \# \ \lambda_1 \cdots \lambda_n
\end{aligned}
$$

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

RuleExtraction Reduce

Discard rare rules and compute features for each rule

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

Phrase Extraction - Runtimes

System	Map(mins)	Reduce(mins)	Compressed Output(MB)
IWSLT hier	0.1	NA	6
IWSLT syntax	0.1	NA	8
230M hier	2	NA	2627
230M syntax	2	NA	3576

System	Map(mins)	Reduce(mins)	Compressed Output(M
IWSLT hier	1.5	1.5	232
IWSLT syntax	2	4	527
230M hier	3 hrs 20	1 hr	1753
230M syntax	4 hrs 10 mins	2 hrs 20 mins	2478

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

Rule Filtering - Specification

RuleFiltering Map: Partion 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 $= \langle \text{lhs}, \gamma, \alpha, \phi \rangle$ 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 setences specific ARPA-format LMs

- ReduceInput: All n-grams that are compliant with a single sentence's vocabulary
- ReduceOuput: 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 λ 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 $\langle params, data \rangle$ 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 $\rightarrow$ $<s>$ NP VP $</s>$ # $<s>$ VP NP . $</s>$
- S $\rightarrow$ $<s>$ PRP MD VP $</s>$ # $<s>$ MD PRP VP ? $</s>$
- Only allow the ones spanning full sentence

BOS/EOS Modelling: Results

System	Dev. BLEU	2007 BLEU	2008 BLEU
IWSLT Hier.	0.278	0.360	0.427
IWSLT Hier. with full-sentence rules	0.277	0.367	0.460
IWSLT Syntax	0.296	0.335	0.430
IWSLT Syntax with full-s. rules	0.301	0.361	0.440

- 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: $\hat{p}(a_i) = p(a_i) \Big/ \sum_{j=1}^{N} p(a_j)$
- rule r's total count for the sentence pair $\langle f, e \rangle$ is thus:

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

Results

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

- 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