Statistical Machine Translation

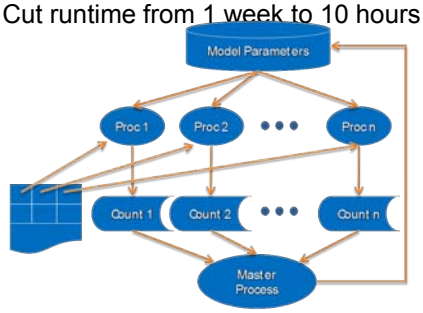
Parallel Processing for Large Data Situations

Qin Gao, Alok Parlikar, Nguyen Bach, Stephan Vogel (Language Technologies Institute & InterACT)

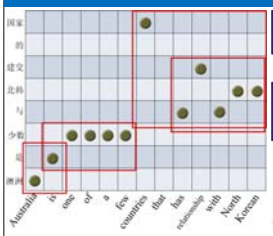
Word Alignment

Parallel GIZA++

- Distribute E-Step of EM Algorithm
- Run time:179 hours → 39 hours(11 jobs)
- IO bottleneck while reading/writing models on network FS
- Distributed word alignment
- Cut runtime from 1 week to 10 hours



Phrase Table Generation



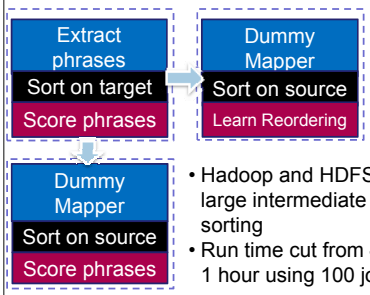
Extract phrases

Sort on source | Sort on target

Score phrases

Learn reordering

Map/Reduce Phrase Extraction



- Hadoop and HDFS handle large intermediate file and sorting
- Run time cut from 4 days to 1 hour using 100 jobs

Mapper | Sorting | Reducer

Back Translation

Israel , the once implacable enemy of the Arab cause , now seems to be slotted into this defensive structure .	Israel , the implacable enemy of once because of the Arabs , now seems to lie in this defensive structure .

En->Es

Israel , el enemigo implacable de una vez a causa de los Árabes , ahora parece colocarse en esta estructura defensiva.

Es->En

Discriminative N-Best List Reranking with Back Translation

Large amount of back translation sentences (total of 12 million words)

- Translation:400 CPU hours
- Feature computation: 222 CPU hours

Using the Intel Big Data cluster:

- Translation: 20 hours (20 nodes)
- Feature computation: 5 hours (50 nodes)

Parsing Training Data

Parse Trees → Syntactic Phrases

Parse Trees → Reordering Patterns

5M Pairs of sentences
↓
420 CPU-days for parsing

Parser could crash!

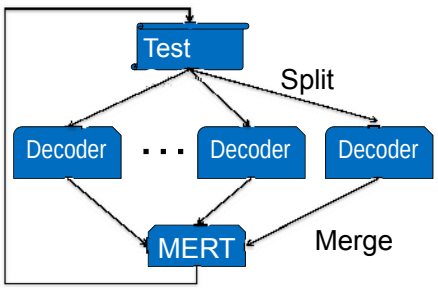
Hadoop's fault tolerance helps

Sentences parsed independently

Split the corpus into chunks, parse several chunks in parallel on cluster

Decoding and MERT

Translates one sentence at a time
Split up decoding into sub-processes, Collect the output for MERT



Filter the phrase table and language models on a per-sentence basis, beforehand.
- Each decoder instance loads faster
- Memory usage is kept in check

Tuning time: 12.5 hrs →70 mins using 50 nodes.

Speedup not linear: Loading models, MERT have significant overhead

Summary

Challenge for large data situations:
- Long training times: hundreds of CPU days)
- Large models: growing beyond 32GB

Solution:
- Parallelizing training and decoding
  - Parallel word alignment
  - Parallel parsing
  - Parallel phrase pair extraction and scoring
- Filtering of models per sentence

Remaining problems:
- IO bottleneck while reading/writing models on network FS
- Models in word alignment training are often too large to fit into memory.

SMT with Parallel Processing: We can now train more efficiently on much larger training sets

Carnegie Mellon

Research @ Intel

intel