Low Cost Portability for Statistical Machine Translation Based on N-gram Coverage

Matthias Eck, Stephan Vogel and Alex Waibel
MT Summit X (September 2005)

UW MTRG - April 26, 2005
presented by Sarah Petersen
Introduction

• Challenge: Statistical MT relies on having lots of parallel training data
• In some situations, we need to limit the amount of data
• Expensive to create parallel corpus
• Computing resources limited
• Solution: intelligently select sentences based on n-gram coverage
Outline

- Motivation and goals
- Previous work
- Approach
- Experiments
- Future work
- Discussion
Motivation: Why Limit Training Data?

- Reducing translation cost: especially important for minority languages
- Translation on small devices:
  - Limited memory requires limited corpus
- Standard SMT: may obtain speed, efficiency improvements
Goals

• Decrease data requirements while maintaining competitive results

• Test data is not known at training time

• Main idea:
  • Sort sentences by importance
  • Account for sentence length
Previous Work

Active learning

• e.g. parsing (Hwa, 2004), ASR (Kam and Meyer, 2002)
• Translation Model Adaptation (Hildebrand et al., 2005)
• But test data is known here
Approach

Sentence-sorting algorithm

• For all sentences not in the sorted list:
  • Calculate weight of each sentence
  • Find sentence S w/ highest weight
  • Add S to sorted list
• Note that this is a greedy algorithm
Weighting Sentences

- Weight depends on previously selected sentences
- Word and phrase (n-gram) coverage is critical for good SMT performance
- Assign sentence weight based on how many new n-grams the sentence contributes
First Weighting Approach

• For n-grams up to trigrams, count # of previously unseen n-grams

\[ \text{weight}_j(\text{sentence}) = \sum_{n=1}^{j} #(\text{unseen n-grams}) \]

• This approach prefers longer sentences

• Translators charge by the word, not the sentence

• Longer sentences make training more difficult
New Version

- Normalize by sentence length
  \[ weight_j(sentence) = \frac{\sum_{n=1}^{j} \#(\text{unseen n-grams})}{|sentence|} \]

- "New n-grams per word to translate"

- Can use \(|sentence|^2 \) to prefer even shorter sentences

- In general:
  \[ weight_{i,j}(sentence) = \frac{\sum_{n=1}^{j} \#(\text{unseen n-grams})}{|sentence|^i} \]
English-Spanish Experiments

- **Training data**: BTEC corpus, travel domain
- **123,416 sentences**
  - **English**: 903,525 words
  - **Spanish**: 852,362 words
- **Test data**: 500 sentences, medical domain
- **MT system**: Vogel et al, 2003, 2004
Baseline Systems

- All sentences:
  - NIST 4.19, BLEU 0.141
- Subsets selected in original order in corpus:
  - Steep increase in scores up to 400k words, then levels off
  - Coverage goes up linearly
Coverage

• Results for weight$_{0,j}$ (not normalized by length)
  • Unigram only - coverage increases very quickly
  • Less steep for higher-order n-grams, but still much steeper than baseline
Translation Results: $\text{weight}_{0,j}$

- **Unigram only ($\text{weight}_{0,1}$)**
  - Steep increase above baseline until $\sim 200k$ words
  - Then below baseline - likely that baseline has decent coverage and better LM
- **Bi/trigrams ($\text{weight}_{0,2}$ and $\text{weight}_{0,3}$)**
  - Above or at baseline
  - $\text{weight}_{0,2}$ is slightly better than $\text{weight}_{0,3}$
Translation Results, continued: weight_{0,j}

- Uni/bigram version:
  - NIST score of 4.0 at 200k words (5% worse than using all data)
  - NIST score of 4.1 at 320k words (2% worse than using all data)
Translation Results: \( \text{weight}_{1,j} \)

- Unigram only (\( \text{weight}_{1,1} \)) - very similar to previous
- Uni/bigram (\( \text{weight}_{1,2} \)) and trigram (\( \text{weight}_{1,3} \)) are better than \( \text{weight}_{0,j} \)
  - NIST score of 4.0 at 170k words
  - NIST score of 4.1 at 220k words
  (\( \text{weight}_{2,j} \) was worse than \( \text{weight}_{1,j} \))
Relative Improvement

• Greatest relative improvement over baseline occurs at 40k words

• Translations are still poor, but many are good enough to be useful in some contexts
Thai-English Experiments

- weight$_{1,2}$
- Medical domain
- Training data - 59,191 sentences
  - English: 457,736 words
  - Thai: 422,692 words
- Test data: 496 sentences
Thai-English Results

- Details in Tables 2 and 3 (p. 233)
- System trained on 10k sentences outperforms NIST score of 43k and 38k sentence baseline systems
- 30k sentences (half the training data)
  - NIST score 2% below best
  - BLEU score 7% below best
Future Work

• Could include n-gram frequency in sentence score:
  • Favor freq. words to get better coverage of the training corpus
  • Favor infreq. words to get higher information gain

• Current system tries to cover every n-gram once: could try to cover every n-gram k times

• Could be more important to cover unigrams earlier and bigrams later

• For some cases (e.g. small devices) could consider target language stats, too
Discussion

• Do the weights level off? How does this correlate with results?
• What happens if you’ve covered all the n-grams and still have sentences left?
• Using optimal English/Spanish weighting scheme for Thai/English - does this assumption make sense?