Phrase Table Training For Precision and Recall: What makes a Good Phrase and a Good Phrase Pair?

Yonggang Deng, Jia Xu, Yuqing Gao

IBM T.J Watson Research Center …

Presented by Vamshi Ambati

Context of the Paper

- Phrase Tables are the bread and butter of SMT
- Main questions in creation of one-
  - Which phrase pairs do we extract?
  - How do we parameterize them?
- Desirable characteristics -
  - Precision: Extracted translation pairs should be accurate (Johnson et al. 2007)
  - Recall: Extract as many valid pairs as possible (Deng and Byrne 2005)

Related Work

- Which phrase pairs do we extract?
  - Phrases from Alignment Matrix (Och and Ney 2003)
  - More symmetrization (Koehn et.al 2003)
  - Joint word-alignment and Phrases Extraction (Marcu and Wong 2002, Wu 95)
  - Miscellaneous Phrases from Alignment Matrix (Och and Ney 2003)
  - Syntax based extraction (Yamada and Knight 2002, Lavie et.al 2008)

Related Work

- How do we parameterize them?
  - Relative frequency estimation (Och and Ney 2003)
  - Lexical Weighting IBM1 or 4 (Koehn et.al 2003)
  - Smoothing phrase tables (Foster et.al 2006)
  - Additional features to reduce over-estimation (Zhao et.al 2004, Tillmann and Zhang 2006)

Algorithm 1 A Generic Phrase Training Procedure

Train Model-1 and HMM word alignment models
for all sentence pair $(\mathbf{e}, \mathbf{f})$ **do**
    Identify candidate phrases on each side
    for all candidate phrase pair (E, F) **do**
        Calculate its feature function values f_k
        Obtain the score $q(E, F) = \sum_{k=1}^{K} \lambda_k f_k(E, F)$
    end for
    Sort candidate phrase pairs by their final scores q
    Find the maximum score $qm = \max q(E, F)$
    for all candidate phrase pair (E, F) **do**
        If $q(E, F) \geq qm - \tau$, dump the pair into the pool
    end for
end for
Built a phrase translation table from the phrase pair pool
Discriminatively train feature weights λ_k and threshold τ

Algorithm Highlights

- Prepare IBM Model 1, HMM lexicons that support feature extraction
- List all the n-grams to a predefined length
- Extract features for all possible phrase pairs
- Score each phrase with a log-linear model
- Select best pairs by thresholding the combined score at a cut-off
- Discriminatively learn the weights for log-linear model and cut-off threshold

Feature Functions:
Model-based Phrase Pair Posterior

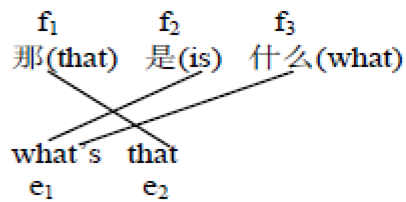
$$A_{(i_1,i_2)}^{(j_1,j_2)} = \{\mathbf{a} : a_j \in [i_1, i_2] \text{ iff } j \in [j_1, j_2]\}$$

$$P_\theta(e_{i_1}^{i_2} \to f_{j_1}^{j_2} | \mathbf{e}, \mathbf{f}) \quad = \quad \frac{\sum_{\mathbf{a} \in A_{(i_1,i_2)}^{(j_1,j_2)}} f(\mathbf{a}, \mathbf{f} | \mathbf{e}; \theta)}{\sum_{\mathbf{a}} f(\mathbf{a}, \mathbf{f} | \mathbf{e}; \theta)} \quad (1)$$

- Sum over hidden alignments – which ones?
- Combining bidirectional posteriors as a geometric mean
- IBM Model-1 vs HMM

Feature Functions:

Bilingual Information Metric



	e_1^1	e_1^2	e_2^2	$H_{BL}(f_{j_1}^{j_2})$
f_1^1	0.0006	0.012	0.89	0.08
f_1^2	0.0017	0.035	0.343	0.34
f_1^3	0.07	0.999	0.0004	0.24
f_2^2	0.03	0.0001	0.029	0.7
f_2^3	0.89	0.006	0.006	0.05
f_3^3	0.343	0.002	0.002	0.06
$H_{BL}(e_{i_1}^{i_2})$	0.869	0.26	0.70	

$$H_{BL}(e_{i_1}^{i_2} | \mathrm{e}, \mathrm{f}) = H(\hat{P}_{\theta_{HMM}}(e_{i_1}^{i_2} \rightarrow *))$$

$$H(P) = -\sum_x P(x) \log P(x)$$

Feature Functions:

Monolingual Information Metric

- Predictive Uncertainty
- Ex: 'we want to have a table near the window'

$$H_{LM}(w_1^{n-1}) = H(P(\cdot|w_1^{n-1})).$$

$$PU(w_1^N, i) = H_{LMF}(w_1^i) + H_{LMB}(w_N^{i+1})$$

Feature Functions:
Word Alignments Induced Metric

- Within phrase pair consistency ratio (WPPCR)
- Computed using Viterbi Alignments
- Viterbi case: WPPCR=1
- Soft case: WPPCR is low for precise phrases

Experiments

- IWSLT 2006 Chinese-English : 40K
- Tune parameters (phrase-score and decoding) on 06dev set
- Test on 04dev,04test, 05test,06test
- Decoder:
  - Stack based decoder
  - Pharoah-style features (14?)
- LM:
  - Trigram, Kneser-Ney smoothing

Translation Results

BLEU Scores

Table	04dev	04test	05test	06dev	06test
HMM	0.367	0.407	0.473	0.200	0.190
Model-4	0.380	0.403	0.485	0.210	0.204
New	0.411	0.427	0.500	0.216	0.208

METEOR Scores

Table	04dev	04test	05test	06dev	06test
HMM	0.532	0.586	0.675	0.482	0.471
Model-4	0.540	0.593	0.682	0.492	0.480
New	0.568	0.614	0.691	0.505	0.487

Table 3: Translation Results

Phrase table size vs Quality

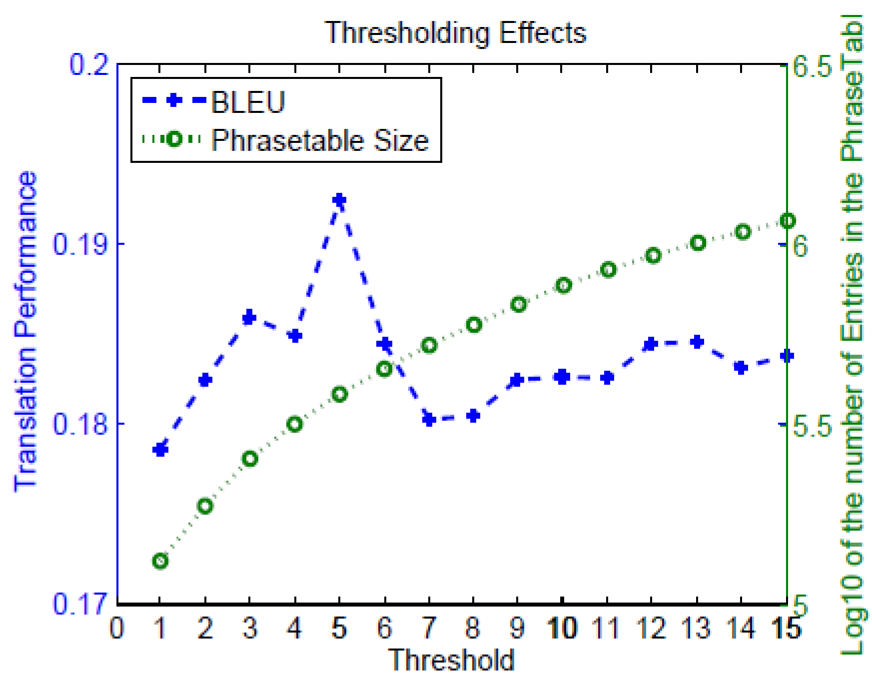


Figure 1: Thresholding effects on translation performance and phrase table size

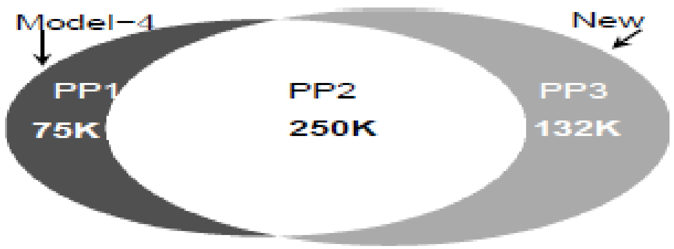
Effect of Features

Features	04dev	04test	05test	06dev	06test
basic	0.393	0.406	0.496	0.205	0.199
+align	0.401	0.429	0.502	0.208	0.196
+align_BLT	0.411	0.427	0.500	0.216	0.208

Table 4: Translation Results (BLEU) of discriminative phrase training approach using different features

- Word-alignment seems to be a crucial feature

Effect of Recall



Features	04dev	04test	05test	06dev	06test
PP2	0.380	0.395	0.480	0.207	0.202
PP1+PP2	0.380	0.403	0.485	0.210	0.204
PP2+PP3	0.411	0.427	0.500	0.216	0.208
PP1+PP2+PP3	0.412	0.432	0.500	0.217	0.214

Table 5: Translation Results (BLEU) of Different Phrase Pair Combination

- How are s

Discussion

- Training
  - Decoder features were not trained along with phrase features
- Recall vs. Features vs. Parameterization
- Threshold to filter phrase table
  - What is the right way to do this
- Is this a Joint -
  - phrase extraction+ word alignment
  - Phrase extraction + decoding