Christof Monz

Applied Language Technology
Translation Modeling

Today's Class

- Refined alignment strategies
- Phrase extraction
- Computing phrase translation probabilities
- Translation model pruning

Word Alignment

- Word alignment aims to find the word-to-word translations between parallel sentence pairs
- The most likely alignment for a given sentence pair is known as the Viterbi alignment
- Common methods are:
  - IBM Model 1 (only translation probabilities)
  - IBM Model 2 (+ distortion probabilities)
  - IBM Model 3 (+ fertility)
  - IBM Model 4 (+ relative reordering)
  - IBM Model 5 (resolves deficiency issue)
  - HMM Alignment
- In practice, models are combined: IBM-1 (5x), IBM-2 (5x), HMM (5x), IBM-3 (5x), IBM-4 (5)

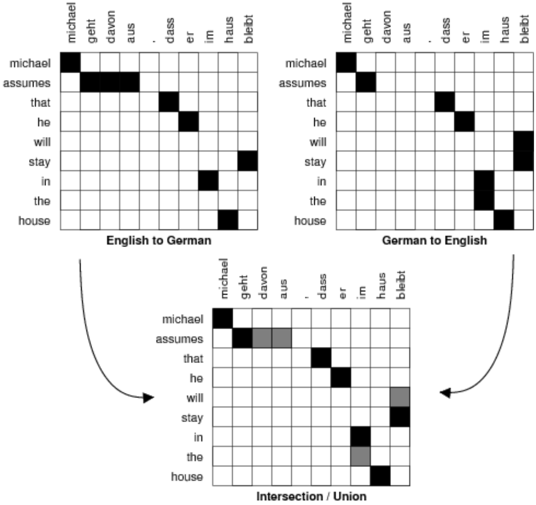
Refined Alignment

- IBM and HMM model are directional (e-to-f or f-to-e)
- IBM and HMM model alignments are one-to-many
- Viterbi alignments of IBM and HMM models are noisy
- Refine alignments by combining the Viterbi alignments from both directions
  - results in symmetric alignments
  - many-to-many alignments
  - higher quality alignments

Alignment Quality

- Assume a manually annotated test set with sure (S) and probable (P) alignments
- Precision: $\frac{|A \cap P|}{|A|}$
- Recall: $\frac{|A \cap S|}{|S|}$
- F-Measure: $\frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$
- Alignment error rate (AER): $1 - \frac{|A \cap P| + |A \cap S|}{|S| + |P|}$
- Correlation with MT quality
  - AER correlates poorly
  - F-Measure correlates somewhat

Refined Alignment



Refined Alignment

Refined Alignment

```
GROW-DIAG():
  iterate until no new points added
    for english word e = 0 ... en
      for foreign word f = 0 ... fn
        if ( e aligned with f )
          for each neighboring point ( e-new, f-new ):
            if ( ( e-new not aligned or f-new not aligned ) and
                 ( e-new, f-new ) in union( e2f, f2e ) )
              add alignment point ( e-new, f-new )
FINAL(a):
  for english word e-new = 0 ... en
    for foreign word f-new = 0 ... fn
      if ( ( e-new not aligned or f-new not aligned ) and
           ( e-new, f-new ) in union( e2f, f2e) )
        add alignment point ( e-new, f-new )
```

Refined Alignment

- Different alignment refinement strategies result in different alignment density (number of links)

  intersection
  < grow-diag
  < grow-diag-final
  < union

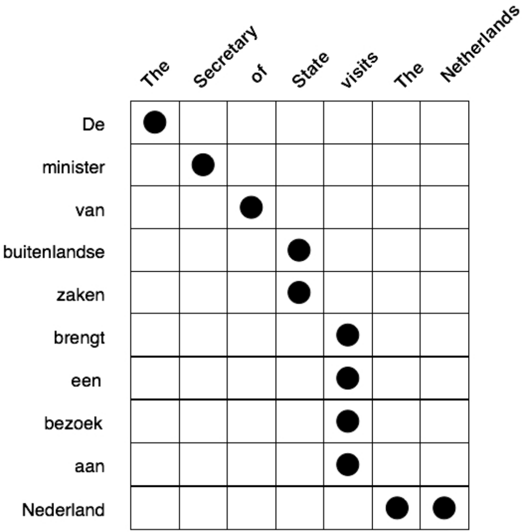
- Refined alignment does result in higher alignment quality (in particular grow-diag and grow-diag-final)

Extracting Phrases

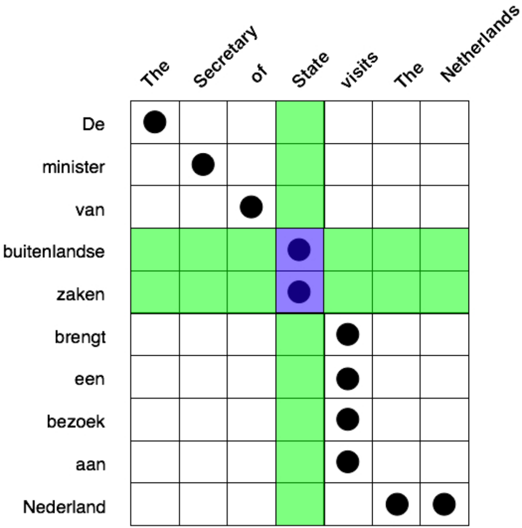
- All phrases that are consistent are extracted
- A phrase pair $(\bar{e},\bar{f})$ is consistent with an alignment A if and only if

1. No English words in the phrase pair are aligned to words outside it
   $\forall e_i \in \bar{e}(e_i,f_j) \in A \Rightarrow f_j \in \bar{f}$
2. No foreign words in the phrase pair are aligned to words outside it
   $\forall f_j \in \bar{f}(e_i,f_j) \in A \Rightarrow e_i \in \bar{e}$
3. The phrase pair contains at least one alignment link
   $\exists e_i \in \bar{e},f_j \in \bar{f} s.t.(e_i,f_j) \in A$

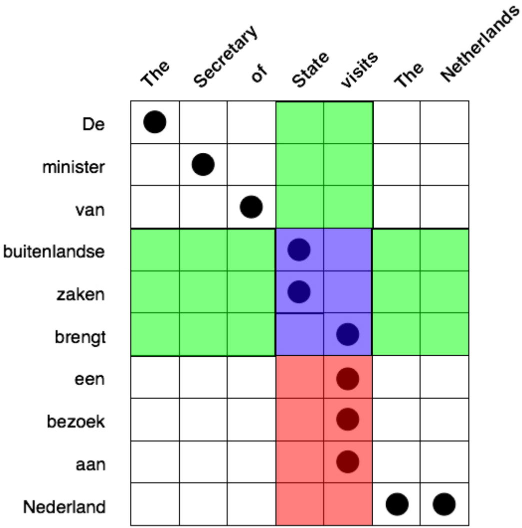
Phrase Extraction



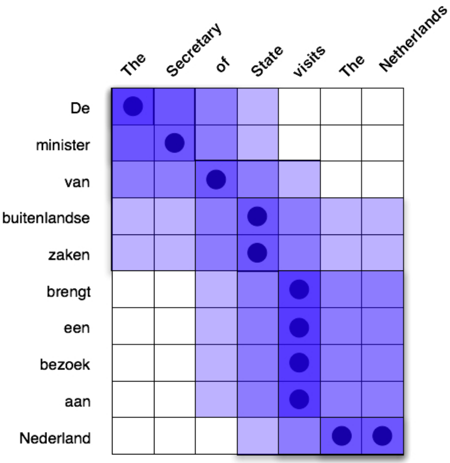
Phrase Extraction



Phrase Extraction



Phrase Extraction



De	The
minister	Secretary
De minister	The Secretary
van	of
buitenlandse zaken	State
van buitenlandse zaken	of State
De min. van b. zaken	The Sec. of State
brengt een bezoek aan	visits
Nederland	The Netherlands
brengt een bezoek aan N.	visits The N.
b. zaken brengt een b. aan	State visits
van b. z. brengt een b. aan	of State visits
b. zaken brengt een b. aan N.	State vis. The N.

Phrase Extraction

- Almost all approaches to phrase extraction are based on binary alignment links
  - A word pair (f_j, e_i) is aligned or not
- Alignment strength is not necessarily binary
  - The actual word translation probabilities $p(f_j|e_i)$ can be low
  - An alignment can be present in one direction but not the other
  - Multiple alignment approaches could be combined
- Can be expressed as weighted alignments
- Phrase extraction needs to be adjusted
  - How to redefine consistency in a weighted alignment framework?
  - See work by Liu et al. (EMNLP, 2009)

Scoring Phrase Translations

- Phrase extraction: collect all phrases from all sentence pairs in the data
- Phrase pair scoring: assign probabilities to phrase translations
- Score by relative frequency:

$$p(\bar{e}|\bar{f}) = \frac{\text{count}(\bar{e},\bar{f})}{\sum_{\bar{e'}} \text{count}(\bar{e'},\bar{f})}$$

Actual Phrase Pairs

يوح نا ب ولس الثاني ||| john paul ii , of ||| 1

يوح نا ب ولس الثاني ||| john paul ii , ||| 1

يوح نا ب ولس الثاني ||| john paul ii ? ||| 1

يوح نا ب ولس الثاني ||| john paul ii appeared ||| 1

يوح نا ب ولس الثاني ||| john paul ii praised ||| 1

يوح نا ب ولس الثاني ||| john paul ii sets ||| 1

يوح نا ب ولس الثاني ||| john paul ii ||| 0.761905

يوح نا ب ولس الثاني ||| john paul the second canonized ||| 0.5

يوح نا ب ولس الثاني ||| john paul the second ||| 1

Discontinuous Phrases

- Phrase-based SMT uses continuous phrases only
- For some language pairs (such as Chinese-English) it has been shown that allowing for discontinuous phrases helps
- Extraction:
  - Extract long continuous phrases: $(f_1 \ldots f_J, e_i \ldots e_I)$
  - If $(f_i \ldots f_j, e_k \ldots e_l)$, is an extracted phrase itself:
  - $(f_1 \ldots f_{i-1} X_1 f_{j+1} \ldots f_J, e_i \ldots e_{k-1} X_1\, e_{l+1} \ldots e_I)$
  - where X_1 is a gap variable
  - where $1 \leq i,j \leq J$ and $1 \leq k,l \leq I$
- Some constraints:
  - Not more than two gap variables per rule
  - Maximum length of gap (i.e., $j - i < n$)
  - No rules where the source side is of the form: $f_1 \ldots f_{i-1} X_1\, X_2 f_{j+1} \ldots f_J$ or $f_1 \ldots f_{i-1} X_2\, X_1 f_{j+1} \ldots f_J$

Discontinuous Phrase Pairs

- Discontinuous phrase pairs in general require different decoding strategies
  - Chart-parsing with synchronous grammars
- Some discontinuous phrase pairs can be used in phrase-based MT:
  - Phrase-based systems generate translation from left to right
  - $(f_1 \ldots f_{i-1} X_1 f_{j+1} \ldots f_J, e_i \ldots e_{k-1} X_1)$
  - I.e., all target gaps occur to the very right of the target phrase
  - Still requires different treatment of distortion
- Translation models with discontinuous phrase pairs are about an order of a magnitude larger than continuous translation models
  - Even after pruning out discontinuous rules r with $c(r) < 3$

Phrase Translation Probabilities

- Extracted phrases themselves are noisy
  - Due to alignment errors
- Phrase translation probabilities are unreliable
  - particularly for low frequency pairs (maximum likelihood estimates)
- Ad-hoc solution: remove phrase pairs with low counts
  - Smaller phrase table
  - Substantial loss of coverage (Zipfian distribution)
- Alternative solution: consider additional scores

Phrase Translation Probabilities

- Sentence translation probability based on Bayes

  $\text{trans}(f) = \text{argmax}_e \; p(f|e) \; p(e)$
- Also consider direct translation probabilities: $p(e|f)$

  $\text{trans}(f) = \text{argmax}_e \; p(f|e) \; p(e|f) \; p(e)$
- Not Bayesian anymore: Turn into a log-linear model:
  - $\text{trans}(f) = \text{argmax}_e \; \exp(\sum_{m=1}^{M} \lambda_m h_m(e,f))$
- Where, e.g.,
  - $h_1(e,f) = \log \; p(f|e)$
  - $h_2(e,f) = \log \; p(e|f)$
  - $h_3(e,f) = \log \; p(e)$
- How do we estimate λ_m? (See later class on optimization)

Actual Phrase Pairs

يوح نا ب ولس الثاني ||| john paul ii , of ||| 1 0.04

يوح نا ب ولس الثاني ||| john paul ii , ||| 1 0.04

يوح نا ب ولس الثاني ||| john paul ii ? ||| 1 0.04

يوح نا ب ولس الثاني ||| john paul ii appeared ||| 1 0.04

يوح نا ب ولس الثاني ||| john paul ii praised ||| 1 0.04

يوح نا ب ولس الثاني ||| john paul ii sets ||| 1 0.04

يوح نا ب ولس الثاني ||| john paul ii ||| 0.761905 0.64

يوح نا ب ولس الثاني ||| john paul the second canonized ||| 0.5 0.04

يوح نا ب ولس الثاني ||| john paul the second ||| 1 0.08

Phrase Translation Probabilities

- So far we treated phrases as atomic units
- Translation probabilities are based on co-occurrence counts of entire phrases
- Smoother distributions can be obtained by considering smaller units
  - Smaller units (words) $\rightarrow$ larger, more reliable counts
- Lexical weighting

Lexical Weighting

- $\text{lex}(\bar{e}|\bar{f},a) = \prod_{i=1}^{\text{length}(\bar{e})} \frac{1}{|j:(i,j)\in a|} \sum_{\forall(i,j)\in a} w(e_i|f_j)$
- Where $w(e_i|f_j)$ is the word translation probability
  - can be estimated based on the Viterbi word alignments
  - also known as Koehn, Marcu, and Och (KMO)

$$\begin{aligned}\text{lex}(\bar{e}|\bar{f},a) = {} & w(\text{does}|\text{NULL}) \times \\ & w(\text{not}|\text{nicht}) \times \\ & \frac{1}{3}(w(\text{assume}|\text{geht}) + w(\text{assume}|\text{davon}) + w(\text{assume}|\text{aus}))\end{aligned}$$

Actual Phrase Pairs

يوح نا ب ولس الثاني ||| john paul ii , of ||| 1 9.788e-05 0.04 8.052e-05

يوح نا ب ولس الثاني ||| john paul ii , ||| 1 9.789e-05 0.04 0.00087

يوح نا ب ولس الثاني ||| john paul ii ? ||| 1 9.789e-05 0.04 1.846e-05

يوح نا ب ولس الثاني ||| john paul ii appeared ||| 1 9.788e-05 0.04 4.206e-07

يوح نا ب ولس الثاني ||| john paul ii praised ||| 1 0.0002 0.04 3.554e-05

يوح نا ب ولس الثاني ||| john paul ii sets ||| 1 5.033e-05 0.04 6.674e-05

يوح نا ب ولس الثاني ||| john paul ii ||| 0.761905 9.788e-05 0.64 0.007141

يوح نا ب ولس الثاني ||| john paul the second canonized ||| 0.5 0.0003 0.04 2.98e-06

يوح نا ب ولس الثاني ||| john paul the second ||| 1 6.210e-05 0.08 0.01410

Lexical Weighting

- Lexical weighting requires phrase-internal alignment links
- Alternatively IBM-Model 1 alignment probabilities can be used to compute word-based translation probabilities:
  - $p(\bar{e}|\bar{f}) = \frac{\varepsilon}{(J+1)^I} \prod_{i=1}^{I} \sum_{j=1}^{J} w(e_i|f_j)$
  - where $I = \text{length}(\bar{e})$, $J = \text{length}(\bar{f})$
  - ε is a small constant, sometimes $\varepsilon = p(I|J)$
  - $w(e_i|f_j)$ is the IBM-1 translation probability
  - $w(e_i|f_j)$ can also be based on relative frequencies taken from the Viterbi alignment

Lexical Weighting

- Another alternative is Zens and Ney's noisy-OR lexical weighting scheme:
  - $p(\bar{e}|\bar{f}) = \prod_{i=1}^{I}(1 - \prod_{j=1}^{J}(1 - w(e_i|f_j)))$
  - $w(e_i|f_j)$ is the IBM-1 translation probability
  - $w(e_i|f_j)$ can also be based on relative frequencies taken from the Viterbi alignment
- Noisy-OR based on Pearl's work on Bayesian networks
- For each p_i we compute the probability that it is not the case that p_i is not 'generated' by any of the words in $\bar{f}$

Translation Model Smoothing

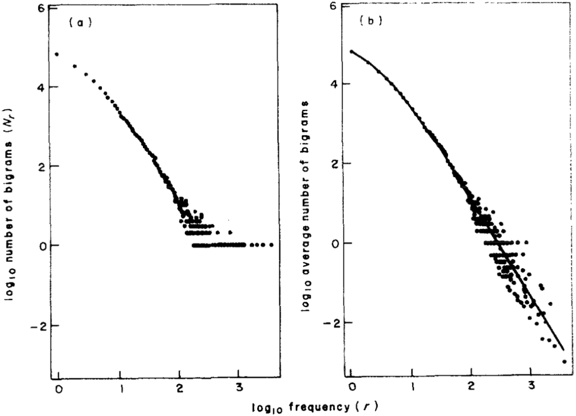
- Lexical translation weights obtain smoother distributions by using smaller, more frequent units (words)
- Alternatively, one can smooth the phrase translation probabilities directly
- Smoothing typically aims to address the problem of estimating the probability of unseen events
  - Set aside probability mass for unseen events
  - Remove probability mass from seen events
  - Do we have unseen events in the context of translation models?

Good-Turing Smoothing

- Compute modified counts based on relative counts
  - $gt(c) = (c+1)\ \frac{n_{c+1}}{n_c}$
  - where n_c is the number of events (phrase pairs) that occur c times
  - typically, $n_{c'} < n_c$ for $c' > c$
- Problem: n_c becomes sparse and unreliable for large c
  - For example, if $n_{100,451} = 1$ but $n_{100,452} = 0$, then $gt(100,451) = 0$!
- Solution: Apply linear least squares fit to observed $(\log\ c, \log\ n_c)$ values
  - Now we can estimate a non-zero n_c value for any c
  - Often, the observed counts are used for small c (e.g., $c \leq 10$)

Good-Turing Smoothing

- Problem: For large c it is almost always the case that $n_c = 1$



Good-Turing Smoothing

- Recompute n_c based on its density
  - Let $l = \text{argmax}_{c' < c} : n_{c'} > 0$
  - Let $r = \text{argmin}_{c' > c} : n_{c'} > 0$
- $n'_c = \frac{n_c}{0.5(r-l)}$
- For small c typically: $n'_c = n_c$
- The larger c gets the lower the density of non-zero count-of-counts
- What about the largest value of c?

Good-Turing Smoothing

- The Good-Turing smoothed phrase translation probability can now be computed as:
  - $p_{gt}(\bar{f}|\bar{e}) = \frac{gt(c(\bar{f},\bar{e}))}{\sum_f gt(c(\bar{f},\bar{e})) + p(\bar{e})n_1}$
  - where $p(\bar{e}) = \frac{c(\bar{e})}{\sum_{\bar{e}} c(\bar{e})}$
- Note that the count mass assigned to unseen phrase pairs is $gt(c(0)n_0 = n_1$

Kneser-Ney Smoothing

- Kneser-Ney smoothing subtracts a fixed discount from all observed non-zero counts
  - Then redistribute the count mass
- $p_{kn}(\bar{f}|\bar{e}) = \frac{c(\bar{f},\bar{e})-D}{\sum_{\bar{f}} c(\bar{f},\bar{e})} + \alpha(\bar{e})\ p_b(\bar{f}|\bar{e})$
  - D is some discount constant
  - $\alpha(\bar{e}) = \frac{D n_{1+}(\bullet,\bar{e})}{\sum_{\bar{f}} c(\bar{f},\bar{e})}$
  - $n_{1+}(\bullet,\bar{e})$ is the number of phrases $\bar{f}$, s.t. $c(\bar{f},\bar{e}) > 0$
  - $p_b(\bar{f}|\bar{e}) = p_b(\bar{f}) = \frac{n_{1+}(\bar{f},\bullet)}{\sum_{\bar{f}} n_{1+}(\bar{f},\bullet)}$
- $\alpha(\bar{e})$ is the average number of unique translations of $\bar{e}$ multiplied by discount D
- $p_b(\bar{f})$ is the proportion of target phrases that have $\bar{f}$ as a translation

Kneser-Ney Smoothing

- How to set the discounts?
- Typically three separate discounts are used:
  - D_1: If $c(\bar{f},\bar{e}) = 1$, $D_1 = 1 - 2(\frac{n_1}{n_1+2n_2})\ \frac{n_2}{n_1}$
  - D_2: If $c(\bar{f},\bar{e}) = 2$, $D_2 = 1 - 3(\frac{n_1}{n_1+2n_2})\ \frac{n_3}{n_2}$
  - D_{3+}: If $c(\bar{f},\bar{e}) \geq 3$, $D_{3+} = 1 - 4(\frac{n_1}{n_1+2n_2})\ \frac{n_4}{n_3}$
- Values for D can also be estimated directly for MT (no significant differences)

Lower-Order Smoothing

- Smoothing for language models has a natural interpretation of lower-order events
  - Drop the last (i.e., oldest) word in the n-gram history
- What should be the lower order event of a phrase pair?
- $p_{lo}(\bar{f}|\bar{e}) = \sum_{i=1}^{I} \frac{c_i^*(\bar{f},\bar{e})}{\sum_{\bar{f}} c_i^*(\bar{f},\bar{e})} \frac{1}{I}$
  - where $c_i^*(\bar{f},\bar{e}) = \sum_{e_i} c(\bar{f}, e_1 \ldots e_i \ldots e_I)$
  - introduce a wildcard for all target positions i
- can be refined by weighting the contribution of different values for e_i to $\bar{e}$
  - general informativeness: *idf* value
  - paraphrase probability: $p(e_i'|e_i)$
- Not much experimental research as estimation of unseen phrase pairs is not an issue in standard SMT

Interpolation Smoothing

- Standard approach is to use counts from entire data set
  - $p(\bar{f}|\bar{e}) = \frac{c(\bar{f},\bar{e})}{\sum_{\bar{f}} c(\bar{f},\bar{e})}$
- Parallel data can be partitioned
  - randomly (overlapping or disjunct)
  - according to source (newswire vs. parliamentary proceedings)
  - similarity to test data (perplexity, vocabulary overlap, etc.)
  - quality
- Compute translation probabilities for each partition d separately and then combine
  - Log-linear combination: $p(\bar{f}|\bar{e}) = \prod_d p_d(\bar{f}|\bar{e})^{\lambda_d}$
  - Linear interpolation: $p(\bar{f}|\bar{e}) = \sum_d \lambda_d p_d(\bar{f}|\bar{e})$
- Weights for log-linear combination can be estimated during SMT parameter tuning
- Linear interpolation weights have to be pre-computed

Interpolation Smoothing

- Surprisingly, log-linear combination does not work!
- Even simple linear interpolation with uniform weights ($\lambda_d = \frac{1}{|D|}$) yields improvements
- Linear interpolation weights can be estimated by maximizing the average log-likelihood
- Use held-out data set with phrase counts
  - Ideally based on human alignments
  - Ideally in the same domain and genre as the test data
- Estimate the optimal $(\lambda_1^*, \ldots, \lambda_{|D|}^*)$ by expectation maximization (EM)

EM for Linear Interpolation

- Initialize $\lambda_d^{(0)}$, such that $\forall d : 0 < \lambda_d^{(0)}$ and $\sum_{d=1}^{|D|} \lambda_d^{(0)} = 1$
  Iteration counter $t = 0$
- Update: $\forall d : \lambda_d^{(t+1)} = \frac{1}{N} \sum_{i=1}^{N} \frac{\lambda_d^{(t)} p_d(\bar{f}|\bar{e})}{\sum_{d=1}^{|D|} \lambda_d^{(t)} p_d(\bar{f}|\bar{e})}$
  - where N is the number of phrase pair tokens in the held-out set
- Convergence check: $\ell(\lambda^{(t)}) = \frac{1}{N} \sum_{i=1}^{N} \log \sum_{d=1}^{|D|} \lambda_d^{(t)} p_d(\bar{f}|\bar{e})$
  and $\ell(\lambda^{(t+1)}) = \frac{1}{N} \sum_{i=1}^{N} \log \sum_{d=1}^{|D|} \lambda_d^{(t+1)} p_d(\bar{f}|\bar{e})$
- Stop if for some small value ε: $\frac{\ell(\lambda^{(t+1)}) - \ell(\lambda^{(t)})}{|\ell(\lambda^{(t+1)})|} \leq \varepsilon$
  Else $t = t + 1$
- $(\lambda_1^*, \ldots, \lambda_{|D|}^*) = (\lambda_1^{(t)}, \ldots, \lambda_{|D|}^{(t)})$

Translation Model Smoothing

- Lexical Weighting:
  - All methods, i.e., KMO, IBM-1, and Noisy-OR, yield improvements
  - No clear 'winner'
  - Best approach is to use them all, but puts burden on parameter optimization
- Phrase table smoothing:
  - Both Good-Turing and Kneser-Ney yield improvements
  - No clear 'winner'
- Phrase table interpolation
  - Optimization λ_d performs best, but requires knowledge about test data
  - Uniform λ_d values still outperform baseline
  - Can be used in combination with phrase table smoothing

Translation Models in Practice

- Translation models can be huge: several gigabytes (gzipped)
  - During decoding entire translation model has to be kept in memory
- Under research settings:
  - Filter translation model wrt to test data
  - This is not realistic for actual online translation systems
- Engineering solutions:
  - Keep most of translation model on disk (slow)
  - Reorganize data such that it is possible to read-in translation probabilities in one go (cheaper than random seeks)
  - Solid-state drives??
- Prune translation model to remove low-quality phrase pairs

Translation Model Pruning

- Number of simple strategies:
  - Ignore phrase pairs where $p(\bar{f}|\bar{e}) < \theta$ or $p(\bar{e}|\bar{f}) < \theta$
  - Ignore phrase pairs with low counts
  - Ignore phrase pairs where number of un-aligned (source or target) words $> n$
- Pruning based on significance testing (Fisher's exact test)
  - What is the probability that a phrase pair is extracted by chance
  - Ignore word alignments
  - $C(\bar{f},\bar{e})$ is the number of sentence pairs in which $\bar{f}$ and $\bar{e}$ co-occur
  - It can be that $C(\bar{f},\bar{e}) \neq c(\bar{f},\bar{e})$
    - If $(\bar{f},\bar{e})$ was not extracted because of alignment constraints
    - If multiple occurrences of $(\bar{f},\bar{e})$ were extracted from the same sentence pair
  - $C(\bar{f})$ and $C(\bar{e})$ are defined analogously

Translation Model Pruning

- Each phrase pair can be assigned a 2x2 contingency table $CT(\bar{f},\bar{e}) =$

$C(\bar{f},\bar{e})$	$C(\bar{f}) - C(\bar{f},\bar{e})$	$C(\bar{f})$
$C(\bar{e}) - C(\bar{f},\bar{e})$	$N - C(\bar{f}) - C(\bar{e}) + C(\bar{f},\bar{e})$	$N - C(\bar{f})$
$C(\bar{e})$	$N - C(\bar{e})$	N

- p-value$(CT(\bar{f},\bar{e})) = \sum_{k=C(\bar{f},\bar{e})}^{\min(C(\bar{f}),C(\bar{e}))} p_h(k)$
- where $p_h(k) = \frac{\binom{C(\bar{f})}{k}\binom{N-C(\bar{f})}{C(\bar{e})-k}}{\binom{N}{C(\bar{e})}}$

Translation Model Pruning

- Remove phrase pairs where $\text{p-value}(CT(\bar{f},\bar{e})) > \theta$, i.e. they could be due to chance
- Empirically, most 1-1-1 phrase pairs are removed: $c(\bar{f},\bar{e}) = c(\bar{f}) = c(\bar{e}) = 1$
- Significantly reduces the size of the translation model
  - Reductions of down to 10-20% only lead to minor changes in translation quality (± 1 BLEU)
  - Not commonly used in research settings (every BLEU fraction counts!)
  - Makes online MT feasible
- More recently other pruning methods have been proposed
  - Entropy-based pruning (Zens 2012)
  - Conditional significance pruning (Johnson 2012)

Recap

- From word alignment to refined alignment
- Phrase extraction:
  - Continuous
  - Discontinuous
- Phrase translation probabilities
  - Low count issues
  - Lexical weighting strategies
  - Phrase translation smoothing
    - Good-Turing
    - Kneser-Ney
    - Interpolation
- Translation Model Pruning
  - Significance based pruning