Machine Translation: Phrase-Based Models

Jordan Boyd-Graber
University of Colorado Boulder

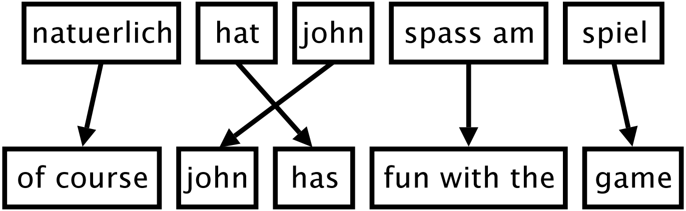
12. NOVEMBER 2014

Adapted from material by Philipp Koehn

Motivation

- Word-Based Models translate *words* as atomic units
- Phrase-Based Models translate *phrases* as atomic units
- Advantages:
  - many-to-many translation can handle non-compositional phrases
  - use of local context in translation
  - the more data, the longer phrases can be learned
- "Standard Model", used by Google Translate and others

Phrase-Based Model



- Foreign input is segmented in phrases
- Each phrase is translated into English
- Phrases are reordered

Phrase Translation Table

- Main knowledge source: table with phrase translations and their probabilities
- Example: phrase translations for **natuerlich**

of course	0.5
naturally	0.3
of course ,	0.15
, of course ,	0.05

Real Example

- Phrase translations for den Vorschlag learned from the Europarl corpus:

English	$\phi(\bar{e}\|\bar{f})$	English	$\phi(\bar{e}\|\bar{f})$
the proposal	0.6227	the suggestions	0.0114
's proposal	0.1068	the proposed	0.0114
a proposal	0.0341	the motion	0.0091
the idea	0.0250	the idea of	0.0091
this proposal	0.0227	the proposal ,	0.0068
proposal	0.0205	its proposal	0.0068
of the proposal	0.0159	it	0.0068
the proposals	0.0159	...	...

  - lexical variation (proposal vs suggestions)
  - morphological variation (proposal vs proposals)
  - included function words (the, a, ...)
  - noise (it)

Linguistic Phrases?

- Model is not limited to linguistic phrases
  (noun phrases, verb phrases, prepositional phrases, ...)
- Example non-linguistic phrase pair

  spass am $\rightarrow$ fun with the

- Prior noun often helps with translation of preposition
- Experiments show that limitation to linguistic phrases hurts quality

Probabilistic Model

- Bayes rule

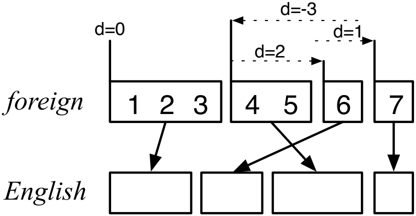
$$\mathbf{e}_{\text{best}} = \text{argmax}_{\mathbf{e}}\; p(\mathbf{e}|\mathbf{f}) = \text{argmax}_{\mathbf{e}}\; p(\mathbf{f}|\mathbf{e})\; p_{\mathrm{LM}}(\mathbf{e})$$

  - translation model $p(\mathbf{e}|\mathbf{f})$
  - language model $p_{\mathrm{LM}}(\mathbf{e})$
- Decomposition of the translation model

$$p(\bar{f}_1^I|\bar{e}_1^I) = \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i)\; d(start_i - end_{i-1} - 1)$$

  - phrase translation probability ϕ
  - reordering probability d

Distance-Based Reordering

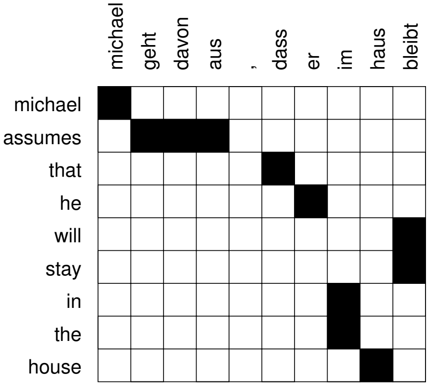


phrase	translates	movement	distance
1	1–3	start at beginning	0
2	6	skip over 4–5	+2
3	4–5	move back over 4–6	-3
4	7	skip over 6	+1

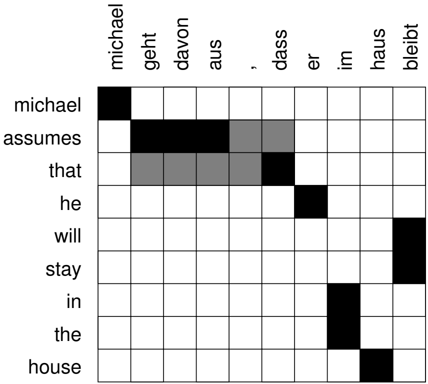
Learning a Phrase Translation Table

- Task: learn the model from a parallel corpus
- Three stages:
  - word alignment: using IBM models or other method
  - extraction of phrase pairs
  - scoring phrase pairs

Word Alignment

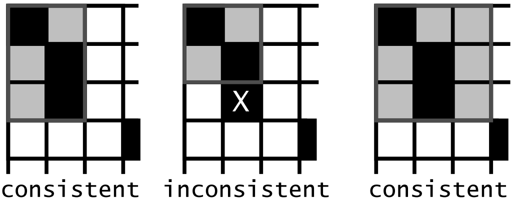


Extracting Phrase Pairs



extract phrase pair consistent with word alignment:
assumes that / geht davon aus , dass

Consistent

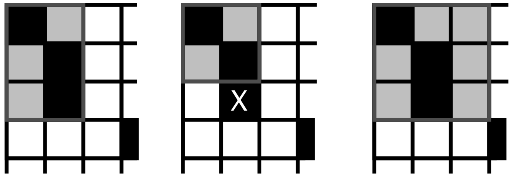


ok	violated	ok
	one alignment point outside	unaligned word is fine

Bottom line:

All words of the phrase pair have to align to each other.

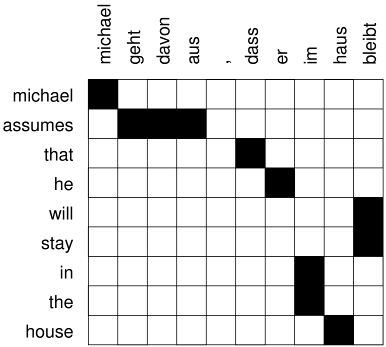
Consistent



Phrase pair $(\bar{e}, \bar{f})$ consistent with an alignment A, if all words $f_1, ..., f_n$ in $\bar{f}$ that have alignment points in A have these with words $e_1, ..., e_n$ in $\bar{e}$ and vice versa:

$$
\begin{aligned}
(\bar{e}, \bar{f}) \text{ consistent with } A \Leftrightarrow & \\
& \forall e_i \in \bar{e} : (e_i, f_j) \in A \rightarrow f_j \in \bar{f} \\
& \text{AND } \forall f_j \in \bar{f} : (e_i, f_j) \in A \rightarrow e_i \in \bar{e} \\
& \text{AND } \exists e_i \in \bar{e}, f_j \in \bar{f} : (e_i, f_j) \in A
\end{aligned}
$$

Phrase Pair Extraction



Smallest phrase pairs:

michael — michael

assumes — geht davon aus / geht davon aus ,

that — dass / , dass

he — er

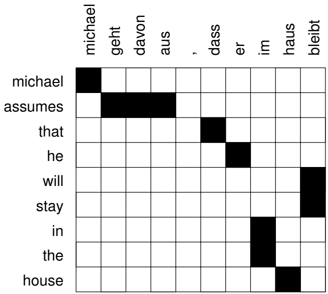
will stay — bleibt

in the — im

house — haus

unaligned words (here: German comma) lead to multiple translations

Larger Phrase Pairs



michael assumes — michael geht davon aus / michael geht davon aus , assumes that — geht davon aus , dass ; assumes that he — geht davon aus , dass er that he — dass er / , dass er ; in the house — im haus michael assumes that — michael geht davon aus , dass michael assumes that he — michael geht davon aus , dass er michael assumes that he will stay in the house — michael geht davon aus , dass er im haus bleibt assumes that he will stay in the house — geht davon aus , dass er im haus bleibt that he will stay in the house — dass er im haus bleibt ; dass er im haus bleibt , he will stay in the house — er im haus bleibt ; will stay in the house — im haus bleibt

Scoring Phrase Translations

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

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

Size of the Phrase Table

- Phrase translation table typically bigger than corpus

  ... even with limits on phrase lengths (e.g., max 7 words)

$\rightarrow$ Too big to store in memory?

- Solution for training
  - extract to disk, sort, construct for one source phrase at a time
- Solutions for decoding
  - on-disk data structures with index for quick look-ups
  - suffix arrays to create phrase pairs on demand

Weighted Model

- Described standard model consists of three sub-models
  - phrase translation model $\phi(\bar{f}|\bar{e})$
  - reordering model d
  - language model $p_{LM}(e)$

$$e_{\text{best}} = \text{argmax}_e \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i)\; d(start_i - end_{i-1} - 1) \prod_{i=1}^{|\mathbf{e}|} p_{LM}(e_i|e_1 \ldots e_{i-1})$$

- Some sub-models may be more important than others
- Add weights λ_ϕ, λ_d, λ_{LM}

$$e_{\text{best}} = \text{argmax}_e \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i)^{\lambda_\phi}\; d(start_i - end_{i-1} - 1)^{\lambda_d} \prod_{i=1}^{|\mathbf{e}|} p_{LM}(e_i|e_1 \ldots e_{i-1})^{\lambda_{LM}}$$

Log-Linear Model

- Such a weighted model is a log-linear model:

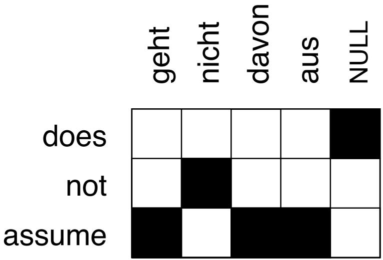
$$p(x) = \exp \sum_{i=1}^{n} \lambda_i h_i(x)$$

- Our feature functions
  - number of feature function $n = 3$
  - random variable $x = (e, f, start, end)$
  - feature function $h_1 = \log \phi$
  - feature function $h_2 = \log d$
  - feature function $h_3 = \log p_{\text{LM}}$

Weighted Model as Log-Linear Model

$$
\begin{aligned}
p(e, a|f) = \exp(\lambda_\phi \sum_{i=1}^{I} & \log \phi(\bar{f}_i|\bar{e}_i) + \\
\lambda_d \sum_{i=1}^{I} & \log d(a_i - b_{i-1} - 1) + \\
\lambda_{LM} \sum_{i=1}^{|\mathbf{e}|} & \log p_{LM}(e_i|e_1 \ldots e_{i-1}))
\end{aligned}
$$

More Feature Functions



- Bidirectional alignment probabilities: $\phi(\bar{e}|\bar{f})$ and $\phi(\bar{f}|\bar{e})$
- Rare phrase pairs have unreliable phrase translation probability estimates

$\rightarrow$ lexical weighting with word translation probabilities

$$\text{lex}(\bar{e}|\bar{f}, a) = \prod_{i=1}^{\text{length}(\bar{e})} \frac{1}{|\{j|(i,j) \in a\}|} \sum_{\forall (i,j) \in a} w(e_i|f_j)$$

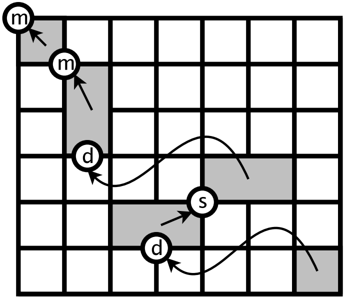
More Feature Functions

- Language model has a bias towards short translations

  → word count: $\mathrm{wc}(e) = \log |\mathbf{e}|^{\omega}$
- We may prefer finer or coarser segmentation

  → phrase count $\mathrm{pc}(e) = \log |I|^{\rho}$
- Multiple language models
- Multiple translation models
- Other knowledge sources

Lexicalized Reordering



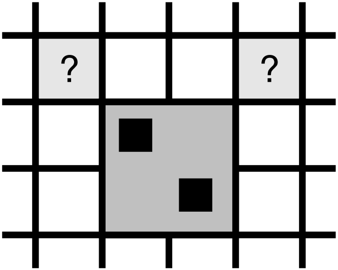
- Distance-based reordering model is weak

  $\rightarrow$ learn reordering preference for each phrase pair
- Three orientations types: (m) monotone, (s) swap, (d) discontinuous

$$\text{orientation} \in \{m, s, d\}$$

$$p_o(\text{orientation}|\bar{f}, \bar{e})$$

Learning Lexicalized Reordering



- Collect orientation information during phrase pair extraction
  - if word alignment point to the top left exists → **monotone**
  - if a word alignment point to the top right exists→ **swap**
  - if neither a word alignment point to top left nor to the top right exists
    → neither monotone nor swap
    → **discontinuous**

Learning Lexicalized Reordering

- Estimation by relative frequency

$$p_o(\text{orientation}) = \frac{\sum_{\bar{f}} \sum_{\bar{e}} count(\text{orientation}, \bar{e}, \bar{f})}{\sum_o \sum_{\bar{f}} \sum_{\bar{e}} count(o, \bar{e}, \bar{f})}$$

- Smoothing with unlexicalized orientation model $p(\text{orientation})$ to avoid zero probabilities for unseen orientations

$$p_o(\text{orientation}|\bar{f}, \bar{e}) = \frac{\sigma\ p(\text{orientation}) + count(\text{orientation}, \bar{e}, \bar{f})}{\sigma + \sum_o count(o, \bar{e}, \bar{f})}$$

EM Training of the Phrase Model

- We presented a heuristic set-up to build phrase translation table (word alignment, phrase extraction, phrase scoring)
- Alternative: align phrase pairs directly with EM algorithm
  - initialization: uniform model, all $\phi(\bar{e}, \bar{f})$ are the same
  - expectation step:
    - estimate likelihood of all possible phrase alignments for all sentence pairs
  - maximization step:
    - collect counts for phrase pairs $(\bar{e}, \bar{f})$, weighted by alignment probability
    - update phrase translation probabilties $p(\bar{e}, \bar{f})$
- However: method easily overfits
  (learns very large phrase pairs, spanning entire sentences)

Summary

- Phrase Model
- Training the model
  - word alignment
  - phrase pair extraction
  - phrase pair scoring
- Log linear model
  - sub-models as feature functions
  - lexical weighting
  - word and phrase count features
- Lexicalized reordering model
- EM training of the phrase model