Machine Translation
Phrase-based Models 1 (Training)

Miles Osborne
(slides by Philipp Koehn and Barry Haddow)

13 February 2012

School of informatics

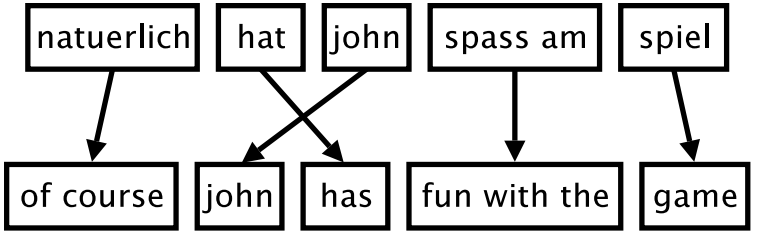
Outline

- From word-based to phrase-based
- Creating the phrase table
- The log-linear model and the "standard" phrase-based MT features

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

Translation	Probability $\phi(\bar{e}\|\bar{f})$
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	...	...

Real Example: Observations

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

Linguistic Phrases?

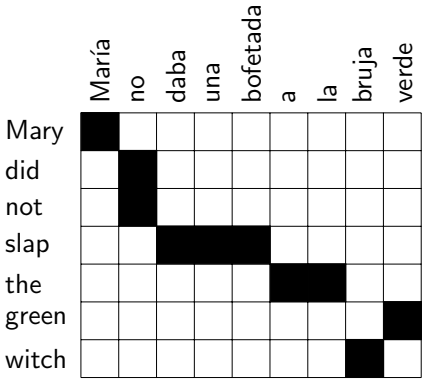
- Model is not limited to linguistic phrases
  (noun phrases, verb phrases, prepositional phrases, ...)
- Example non-linguistic phrase pair

  spass am → fun with the
- Prior noun often helps with translation of preposition
- Experiments show that limitation to linguistic phrases hurts quality

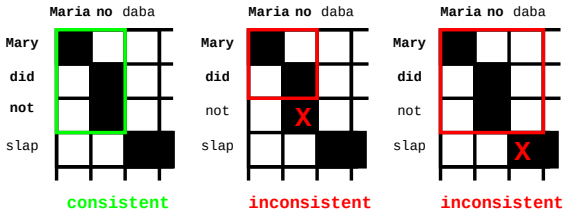
Learning a Phrase Translation Table

- 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



Phrase Extraction Criteria



- Phrase alignment has to *contain all alignment points* for all covered words
- Phrase alignment has to *contain at least one alignment point*

Phrase Extraction Criteria, Formalised

A phrase pair $(\overline{e}, \overline{f})$ is *consistent* with an alignment A if and only if:

1. No English words in the phrase pair are aligned to words outside it.

$$\forall e_i \in \overline{e}, (e_i, f_j) \in A \Rightarrow f_j \in \overline{f}$$

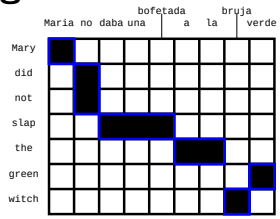
2. No Foreign words in the phrase pair are aligned to words outside it.

$$\forall f_j \in \overline{f}, (e_i, f_j) \in A \Rightarrow e_i \in \overline{e}$$

3. The phrase pair contains at least one alignment point.

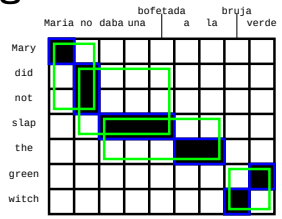
$$\exists e_i \in \overline{e}, f_j \in \overline{f} \; s.t. \; (e_i, f_j) \in A$$

Word alignment induced phrases



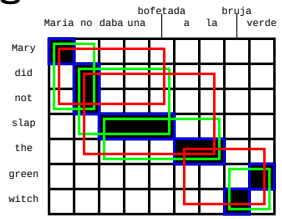
(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green)

Word alignment induced phrases



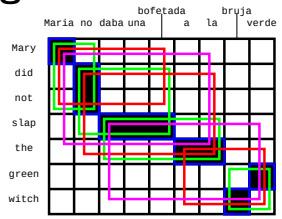
(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green),
(Maria no, Mary did not), (no daba una bofetada, did not slap), (daba una bofetada a la, slap the),
(bruja verde, green witch)

Word alignment induced phrases



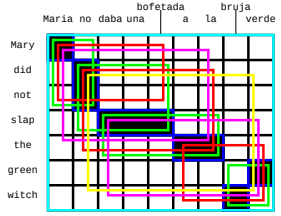
(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green),
(Maria no, Mary did not), (no daba una bofetada, did not slap), (daba una bofetada a la, slap the),
(bruja verde, green witch), (Maria no daba una bofetada, Mary did not slap),
(no daba una bofetada a la, did not slap the), (a la bruja verde, the green witch)

Word alignment induced phrases



(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green),
(Maria no, Mary did not), (no daba una bofetada, did not slap), (daba una bofetada a la, slap the),
(bruja verde, green witch), (Maria no daba una bofetada, Mary did not slap),
(no daba una bofetada a la, did not slap the), (a la bruja verde, the green witch),
(Maria no daba una bofetada a la, Mary did not slap the),
(daba una bofetada a la bruja verde, slap the green witch)

Word alignment induced phrases (5)



(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green),
(Maria no, Mary did not), (no daba una bofetada, did not slap), (daba una bofetada a la, slap the),
(bruja verde, green witch), (Maria no daba una bofetada, Mary did not slap),
(no daba una bofetada a la, did not slap the), (a la bruja verde, the green witch),
(Maria no daba una bofetada a la, Mary did not slap the), (daba una bofetada a la bruja verde,
slap the green witch), (no daba una bofetada a la bruja verde, did not slap the green witch),
(Maria no daba una bofetada a la bruja verde, Mary did not slap the green witch)

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:

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

Scoring Translations

- Bayes rule

$$\begin{aligned}\mathbf{e}_{\text{best}} &= \text{argmax}_{\mathbf{e}}\ p(\mathbf{e}|\mathbf{f}) \\ &= \text{argmax}_{\mathbf{e}}\ p(\mathbf{f}|\mathbf{e})\ p_{\text{LM}}(\mathbf{e})\end{aligned}$$

  - translation model $p(\mathbf{f}|\mathbf{e})$
  - language model $p_{\text{LM}}(\mathbf{e})$

- Decomposition of the translation model

$$p(\mathbf{f}|\mathbf{e}) = p(\bar{f}_1^I|\bar{e}_1^I) = \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i)$$

Linear Model

We always work in log space, so translations are scored as follows:

$$score(\mathbf{e})\ (\equiv logp(\mathbf{e}|\mathbf{f})) = \log p_{\text{TM}}(\mathbf{f}|\mathbf{e}) + \log p_{\text{LM}}(\mathbf{e})$$

This suggests that we could weight the models differently

$$score(\mathbf{e}) = \lambda_{\text{TM}} \log p_{\text{TM}}(\mathbf{f}|\mathbf{e}) + \lambda_{\text{LM}} \log p_{\text{LM}}(\mathbf{e})$$

So why not just go the whole way and use a linear (log-linear?) model

$$score(\mathbf{e}) = \sum_{k=1}^{n} \lambda_k h_k(\mathbf{e}, \mathbf{f})$$

But where do the weights come from? (Lecture 10!)

Expanding the Model

- Now we can add more feature functions
  - Reverse phrase translation probability (i.e. $p_{\mathrm{TM}}(\mathbf{e}|\mathbf{f})$)
  - Lexical translation probabilities (from IBM models)
  - Phrase count
  - Word count
  - Distortion cost (Lecture 11)
  - Reordering score (Lecture 11)
- Why might these be useful?

Using the Model

To translate a foreign sentence $\mathbf{f}$, we have to solve

$$\mathbf{e}_{\text{best}} = \text{argmax}_{\mathbf{e}}\, p(\mathbf{e}|\mathbf{f})$$

In reality what we do is (Viterbi approximation)

$$(a, \mathbf{e})_{\text{best}} = \text{argmax}_{(a,\mathbf{e})}\, p(a, \mathbf{e}|\mathbf{f})$$

- This is known as *decoding*
- It's a search problem
- The search space is huge

Size of the Phrase Table

- Phrase translation table typically bigger than corpus
  ... even with limits on phrase lengths (e.g., max 7 words)

→ 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

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
- EM training of the phrase model