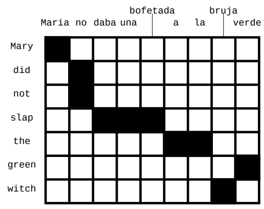
Data Intensive Linguistics — Lecture 17
Machine translation (IV): Phrase-Based Models

Philipp Koehn

9 March 2006

Word alignment

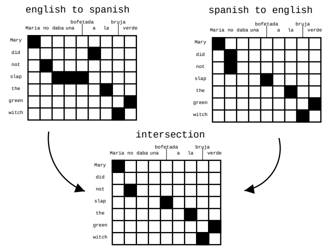
- Notion of **word alignment** valuable
- Shared task at NAACL 2003 and ACL 2005 workshops



Word alignment with IBM models

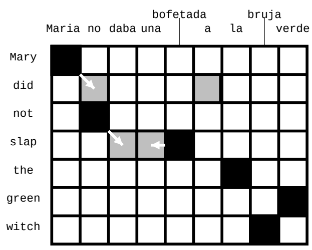
- IBM Models create a *many-to-one* mapping
  - words are aligned using an **alignment function**
  - a function may return the same value for different input (one-to-many mapping)
  - a function can not return multiple values for one input (*no many-to-one* mapping)
- But we need *many-to-many* mappings

Symmetrizing word alignments



- *Intersection* of GIZA++ bidirectional alignments

Symmetrizing word alignments



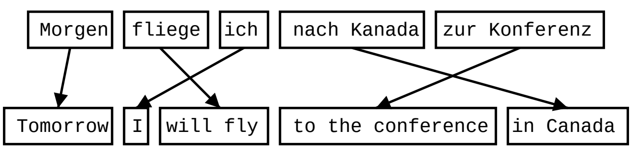
- *Grow* additional alignment points [Och and Ney, CompLing2003]

Growing heuristic

```
GROW-DIAG-FINAL(e2f,f2e):
  neighboring = ((-1,0),(0,-1),(1,0),(0,1),(-1,-1),(-1,1),(1,-1),(1,1))
  alignment = intersect(e2f,f2e);
  GROW-DIAG(); FINAL(e2f); FINAL(f2e);

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 and 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 alignment a )
        add alignment point ( e-new, f-new )
```


Phrase-based translation



- Foreign input is segmented in phrases
  - any sequence of words, not necessarily linguistically motivated
- Each phrase is translated into English
- Phrases are reordered

Phrase-based translation model

- Major components of phrase-based model
  - **phrase translation model** $\phi(\mathbf{f}|\mathbf{e})$
  - **reordering model** $\omega^{\text{length}(\mathbf{e})}$
  - **language model** $p_{\text{LM}}(\mathbf{e})$
- Bayes rule

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

- Sentence $\mathbf{f}$ is decomposed into I phrases $\bar{f}_1^I = \bar{f}_1, ..., \bar{f}_I$
- Decomposition of $\phi(\mathbf{f}|\mathbf{e})$

$$\phi(\bar{f}_1^I|\bar{e}_1^I) = \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i) d(a_i - b_{i-1})$$

Advantages of phrase-based translation

- *Many-to-many* translation can handle non-compositional phrases
- Use of *local context* in translation
- The more data, the *longer phrases* can be learned

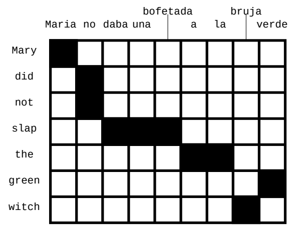
Phrase translation table

- Phrase translations for *den Vorschlag*

English	$\phi(\mathbf{e}\|\mathbf{f})$	English	$\phi(\mathbf{e}\|\mathbf{f})$
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	...	...

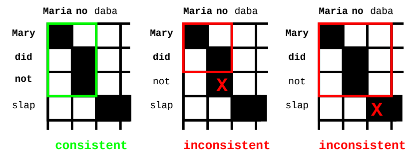
How to learn the phrase translation table?

- Start with the *word alignment*:



- Collect all phrase pairs that are **consistent** with the word alignment

Consistent with word alignment

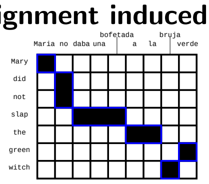


- **Consistent with the word alignment** :=

  phrase alignment has to *contain all alignment points* for all covered words

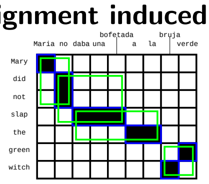
$$(\bar{e},\bar{f}) \in BP \Leftrightarrow \quad \forall e_i \in \bar{e} : (e_i, f_j) \in A \rightarrow f_j \in \bar{f}$$
$$\text{AND} \quad \forall f_j \in \bar{f} : (e_i, f_j) \in A \rightarrow e_i \in \bar{e}$$

Word alignment induced phrases



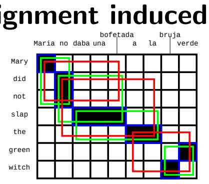
(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green)

Word alignment induced phrases



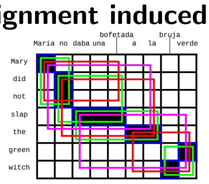
(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green),
(Maria no, Mary did not), (no daba una bofetada, did not slap), (daba una bofetada a la, slap the),
(bruja verde, green witch)

Word alignment induced phrases



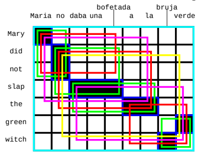
(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green),
(Maria no, Mary did not), (no daba una bofetada, did not slap), (daba una bofetada a la, slap the),
(bruja verde, green witch), (Maria no daba una bofetada, Mary did not slap),
(no daba una bofetada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

Probability distribution of phrase pairs

- We need a **probability distribution** $\phi(\overline{f}|\overline{e})$ over the collected phrase pairs

⇒ Possible *choices*

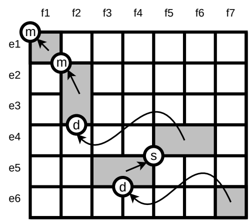
- *relative frequency* of collected phrases: $\phi(\overline{f}|\overline{e}) = \frac{\text{count}(\overline{f},\overline{e})}{\sum_{\overline{f}} \text{count}(\overline{f},\overline{e})}$
- or, conversely $\phi(\overline{e}|\overline{f})$
- use *lexical translation probabilities*

Reordering

- *Monotone* translation
  - do not allow any reordering

  → worse translations
- *Limiting* reordering (to movement over max. number of words) helps
- *Distance-based* reordering cost
  - moving a foreign phrase over n words: cost ω^n
- *Lexicalized* reordering model

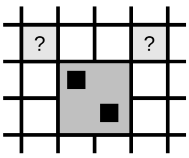
Lexicalized reordering models



[from Koehn et al., 2005, IWSLT]

- Three **orientation** types: **monotone**, **swap**, **discontinuous**
- Probability $p(swap|e,f)$ depends on foreign (and English) *phrase* involved

Learning lexicalized reordering models



[from Koehn et al., 2005, IWSLT]

- Orientation type is *learned during phrase extractions*
- *Alignment point* to the *top left* (monotone) or *top right* (swap)?
- For more, see [Tillmann, 2003] or [Koehn et al., 2005]

Log-linear models

- IBM Models provided mathematical justification for factoring *components* together

  $p_{LM} \times p_{TM} \times p_D$
- These may be *weighted*

  $p_{LM}^{\lambda_{LM}} \times p_{TM}^{\lambda_{TM}} \times p_D^{\lambda_D}$
- *Many components* p_i with weights λ_i

⇒ $\prod_i p_i^{\lambda_i} = exp(\sum_i \lambda_i log(p_i))$

⇒ $log \prod_i p_i^{\lambda_i} = \sum_i \lambda_i log(p_i)$

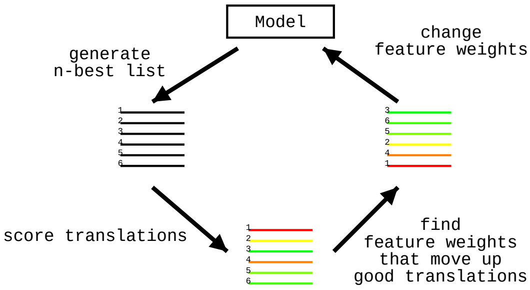
Knowledge sources

- Many different **knowledge sources** useful
  - language model
  - reordering (distortion) model
  - phrase translation model
  - word translation model
  - word count
  - phrase count
  - drop word feature
  - phrase pair frequency
  - additional language models
  - additional features

Set feature weights

- Contribution of components p_i determined by weight λ_i
- Methods
  - *manual setting* of weights: try a few, take best
  - *automate* this process
- Learn weights
  - set aside a **development corpus**
  - set the weights, so that **optimal translation performance** on this development corpus is achieved
  - requires *automatic scoring* method (e.g., BLEU)

Learn feature weights



Discriminative vs. generative models

- Generative models
  - translation process is broken down to *steps*
  - each step is modeled by a *probability distribution*
  - each probability distribution is estimated from the data by *maximum likelihood*
- Discriminative models
  - model consist of a number of *features* (e.g. the language model score)
  - each feature has a *weight*, measuring its value for judging a translation as correct
  - feature weights are *optimized on development data*, so that the system output matches correct translations as close as possible

Discriminative training

- Training set (*development set*)
  - different from original training set
  - small (maybe 1000 sentences)
  - must be different from test set
- Current model *translates* this development set
  - *n-best list* of translations (n=100, 10000)
  - translations in n-best list can be *scored*
- Feature weights are *adjusted*
- N-Best list generation and feature weight adjustment repeated for a number of iterations

Learning task

- Task: *find weights*, so that feature vector of the correct translations *ranked first*

rank	TRANSLATION	LM	TM	WP	SER
1	Mary not give slap witch green .	-17.2	-5.2	-7	1
2	Mary not slap the witch green .	-16.3	-5.7	-7	1
3	Mary not give slap of the green witch .	-18.1	-4.9	-9	1
4	Mary not give of green witch .	-16.5	-5.1	-8	1
5	Mary did not slap the witch green .	-20.1	-4.7	-8	1
6	Mary did not slap green witch .	-15.5	-3.2	-7	1
7	Mary not slap of the witch green .	-19.2	-5.3	-8	1
8	Mary did not give slap of witch green .	-23.2	-5.0	-9	1
9	Mary did not give slap of the green witch .	-21.8	-4.4	-10	1
10	Mary did slap the witch green .	-15.5	-6.9	-7	1
11	**Mary did not slap the green witch .**	**-17.4**	**-5.3**	**-8**	**0**
12	Mary did slap witch green .	-16.9	-6.9	-6	1
13	Mary did slap the green witch .	-14.3	-7.1	-7	1
14	Mary did not slap the of green witch .	-24.2	-5.3	-9	1
15	Mary did not give slap the witch green .	-25.2	-5.5	-9	1

rank translation — feature vector

Methods to adjust feature weights

- **Maximum entropy** [Och and Ney, ACL2002]
  - match *expectation* of feature values of model and data
- **Minimum error rate** training [Och, ACL2003]
  - try to *rank best translations first* in n-best list
  - can be adapted for various error metrics, even BLEU
- **Ordinal regression** [Shen et al., NAACL2004]
  - *separate* k worst from the k best translations