EXPERT

Statistical Machine Translation

Part I: Khalil Sima'an
Data and Models

Universiteit van Amsterdam

Part II: Trevor Cohn
Decoding and efficiency

University of Sheffield

Statistical Machine Translation: PART I

Dr. Khalil Sima'an
Statistical Language Processing and Learning
Institute for Logic, Language and Computation
Universiteit van Amsterdam

Some slides use figures from Philipp Koehn, Barry Haddow and Sophie Arnoult

Data and Models: Structure of lecture

- General statistical framework
- Word-based models: word alignments
- Phrase-based models: phrase-alignments
- Tree-based models: tree-alignments

Statistical Approach: Parallel Corpora

Task: Translate a source sentence **f** to a target sentence **e**.
Data: Parallel corpus (source-target sentence pairs).

Source-Channel Approach: IBM Models (1990's)

Parallel Corpus Example

Parallel corpus **C** = a collection of text-chunks and their translations.
Parallel corpora are the by-product of *human translation*.
Every source chunk is paired with a target chunk.

Dutch	English
De prijs van het huis is gestegen.	The price of the house has risen.
Het huis kan worden verkocht.	The house can be sold.
Als het de marktprijs daalt zullen sommige gezinnen een zware tijd doormaken.	If the market price goes down, some families will go through difficult times.
⋮ ⋮ ⋮	⋮ ⋮ ⋮
⋮ ⋮ ⋮	⋮ ⋮ ⋮

- Hansards Canadian Parliament Proc. (English-French).
- European Parliament Proc. (23 languages).
- United Nations documents.
- Newspapers: Chinese-English; Arabic-English; Urdu-English.
- TAUS corpora.

Generative Source-Channel Framework

Given source sentence **f**, select target sentence **e**

$$\arg\max_{\mathbf{e}\in E(\mathbf{f})}\{\ P(\mathbf{e}\mid\mathbf{f})\ \} = \arg\max_{\mathbf{e}\in E(\mathbf{f})}\{\ \overbrace{P(\mathbf{e})}^{L.M.}\times\overbrace{P(\mathbf{f}\mid\mathbf{e})}^{T.M.}\ \}$$

Set $E(\mathbf{f})$ is the set of hypothesized translations of **f**.

$P(\mathbf{f}\mid\mathbf{e})$: accounts for <u>divergence</u> in ...

- word order
- morphology
- syntactic relations
- idiomatic ways of expression
- ⋮

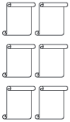
How to estimate $P(\mathbf{e}\mid\mathbf{f})$? Sparse-data problem!

Inducing The Structure of Translation Data

e = Mary did not slap the green witch .

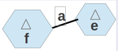
? ? ? ?

f = Maria no dio una bofetada a la bruja verde .



The latent structure of translation equivalence

- Graphical representations $\Delta_\mathbf{f}$ and $\Delta_\mathbf{e}$ for **f** and **e**
- Relation **a** between $\Delta_\mathbf{f}$ and $\Delta_\mathbf{e}$



$$\arg\max_{\mathbf{e} \in E(\mathbf{f})} \{\ P(\mathbf{e} \mid \mathbf{f})\ \} =$$

$$\arg\max_{\mathbf{e} \in E(\mathbf{f})} \{\ \sum_{\langle \Delta_\mathbf{f}, \mathbf{a}, \Delta_\mathbf{e} \rangle} P(\mathbf{e}, \Delta_\mathbf{f}, \Delta_\mathbf{e}, \mathbf{a} \mid \mathbf{f})\ \}$$

The difficult question: Which $\Delta_{\mathbf{f}/\mathbf{e}}$ and **a** fit data best?

Structure in current models

$$\Delta_{\mathbf{f}} \xrightarrow{\mathbf{a}} \Delta_{\mathbf{e}}$$

In most current models structure of reordering:

- $\Delta_{\mathbf{f}/\mathbf{e}}$ are structures over word positions.
- **a** is an **alignment** between groups of word positions in $\Delta_{\mathbf{f}}$ and $\Delta_{\mathbf{e}}$.

Challenge: Number of permutations of n words is *n*!

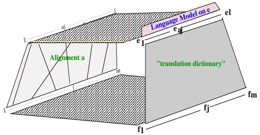
Structure shows translation units composing together

- What are the atomic translation units?
- How these compose together efficiently?
- How to put probs. on these structures?

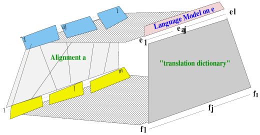
Structure helps combat sparsity and complexity

Structure in Existing Models: Sketch

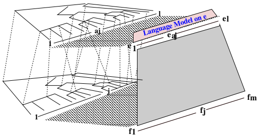
Word-based



Phrase-based



Tree-based



Problem: No sufficient stats to estimate $P(\mathbf{e} \mid \mathbf{f})$ from data

Word-Based Models: Word Alignments

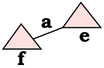
Some History and References

Statistical models with word-alignments:

- Brown, Cocke, Della Pietra, Della Pietra, Jelinek, Lafferty, Mercer and Roossin. A statistical approach to machine translation. Computational Linguistics, 1990.
- Brown, Della Pietra, Della Pietra and Mercer. The mathematics of statistical machine translation: parameter estimation., Computational Linguistics, 1993.
- Och and Ney: A Systematic Comparison of Various Statistical Alignment Models. Computational Linguistics, 2003.

Word-Based Models and Word-Alignment

a is a mapping between word positions.



- $\Delta_{\mathbf{f}}$ and $\Delta_{\mathbf{e}}$ are sequences of word positions.
  $\mathbf{e} = e_1^l = e_1 \ldots e_l$ and $\mathbf{f} = f_1^m = f_1 \ldots f_m$
- A hidden word-alignment **a**:

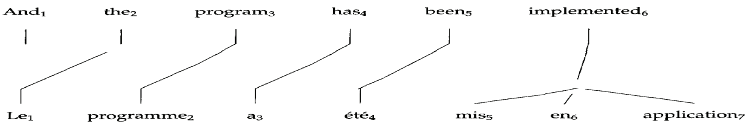
$$P(\mathbf{f} \mid \mathbf{e}) = \sum_{\mathbf{a}} P(\mathbf{a}, \mathbf{f} \mid \mathbf{e})$$

- Assume that a target word-position e_i translates into zero or more source word-positions

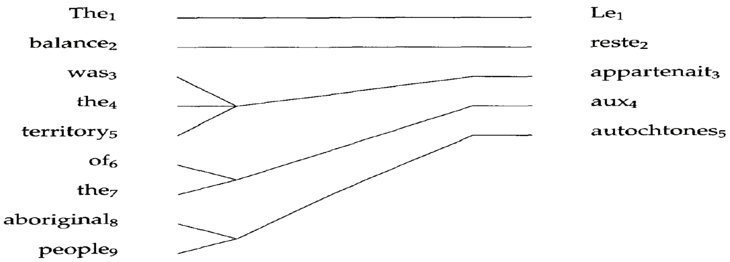
$$\mathbf{a} : \{pos_{\mathbf{f}}\} \to (\{pos_{\mathbf{e}}\} \cup \{0\})$$

- $\mathbf{a}_i$ or $\mathbf{a}(i)$, i.e., word position in **e** with which $\mathbf{f}_i$ is aligned.

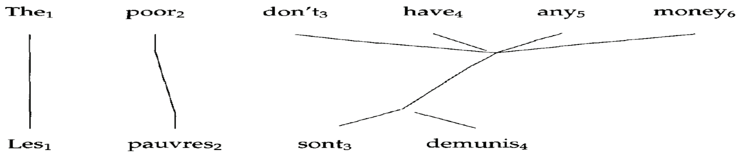
Word Alignment Example



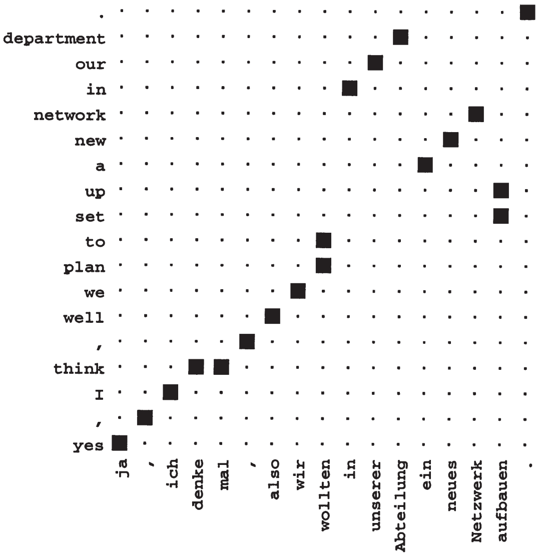
Word Alignment Example



Word Alignment Example: Not covered in this setting



Word Alignment Matrix Example



Translation model with word alignment

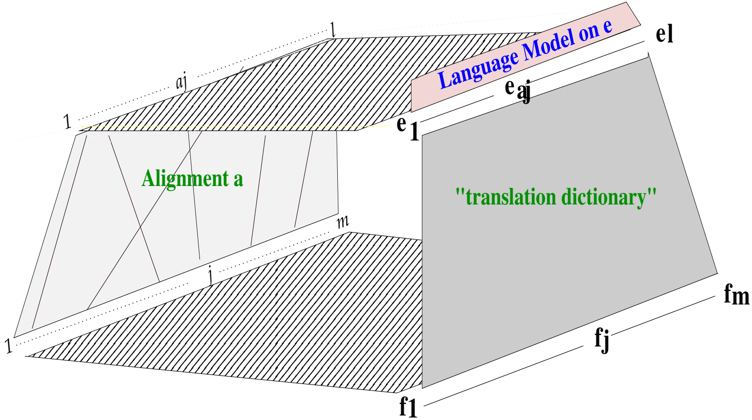
$$\arg\max_{\mathbf{e}} P(\mathbf{e} \mid \mathbf{f}) = \arg\max_{\mathbf{e}} P(\mathbf{e}) \times P(\mathbf{f} \mid \mathbf{e})$$

$$P(\mathbf{f} \mid \mathbf{e}) = \sum_{\mathbf{a}} P(\mathbf{a}, \mathbf{f} \mid \mathbf{e}) = \sum_{\mathbf{a}} P(\mathbf{a} \mid \mathbf{e}) \times P(\mathbf{f} \mid \mathbf{a}, \mathbf{e})$$

Questions

- How to parametrize the model?
  How are **e**, **f** and **a** composed from basic units?
- How to train the model?
  How to acquire word alignment?
- How to translate with this model?
  Decoding and computational issues (for second part)

Word-Alignment As Hidden Structure



We need to decompose

- The alignment **a** and the length m: $P(\mathbf{a} \mid \mathbf{e})$
- "Translation dictionary" $P(\mathbf{f} \mid \mathbf{e}, \mathbf{a})$

Word Alignment Models: General Scheme

Alignment of positions in **f** with positions in **e**:
$\mathbf{a} = a_1^m = a_1 \ldots a_m$

Markov process over **a**

$$P(a_1^m, f_1^m \mid e_1^l) \quad = \quad P(m \mid \mathbf{e}) \; \times$$
$$\prod_{j=1}^{m} P(a_j \mid a_1^{j-1}, f_1^{j-1}, m, \mathbf{e}) \times P(f_j \mid a_1^j, f_1^{j-1}, m, \mathbf{e})$$

In words: to generate alignment **a** and foreign sentence **f**

1. Choose a length m for **f**
2. Generate alignment a_j given the preceding alignments, words in **f**, m, and **e**
3. Generate word f_j conditioned on structure so far and **e**.

IBM models are obtained by simplifications of this formula.

IBM Model I

$$P(a_1^m, f_1^m \mid e_1 \ldots e_l) = P(m \mid \mathbf{e}) \times \prod_{j=1}^{m} P(a_j \mid a_1^{j-1}, f_1^{j-1}, m, \mathbf{e}) \times P(f_j \mid a_1^j, f_1^{j-1}, m, \mathbf{e})$$

IBM Model I:

Length: $P(m \mid \mathbf{e}) =\approx P(m \mid l) \approx = \epsilon$ A fixed probability ϵ.

Align with uniform probability j with any a_j in $\mathbf{e}_1^l$ or NULL: $P(a_j \mid a_1^{j-1}, f_1^{j-1}, m, \mathbf{e}) \approx (l+1)^{-1}$

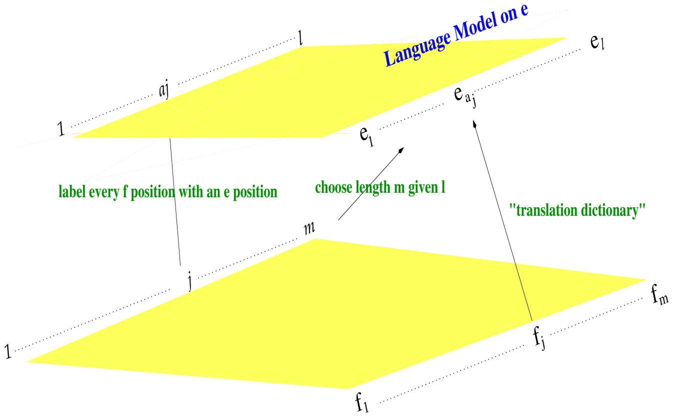
Note that a_j can be linked with l positions in $\mathbf{e}$ or with NULL.

Lexicon: lexicon parameters $\pi_t(f \mid e)$

$$P(f_j \mid a_1^j, f_1^{j-1}, m, \mathbf{e}) \approx P(f_j \mid e_{a_j}) = \pi_t(f_j \mid e_{a_j})$$

Parameters: ϵ and $\{\pi_t(f \mid e) \mid \langle f, e \rangle \in \mathbf{C}\}$.

Sketch IBM Model I



IBM Model I Parameters and Data Likelihood

Data Likelihood:

$$P(\mathbf{f} \mid \mathbf{e}) = \sum_{a_1^m} P(a_1^m, f_1^m \mid e_1 \ldots e_l)$$
$$= \frac{\epsilon}{(l+1)^m} \times \sum_{a_1=0}^{l} \ldots \sum_{a_m=0}^{l} \prod_{j=1}^{m} \pi_t(f_j \mid e_{a_j})$$

Parameters: ϵ and $\{\pi_t(f \mid e) \mid \langle f, e\rangle \in \mathbf{C}\}$.

Fix ϵ, i.e., in practice put a uniform probability over a range $[1..m]$, for some natural number m.

Dilemma

To estimate these parameters we need word-alignment
To get word-alignment we need these parameters.

IBM Model II

Extends IBM Model I at alignment probs:

$$P(a_1^m, f_1^m \mid e_1 \dots e_l) \approx \epsilon \times \prod_{j=1}^{m} \underline{P(a_j \mid a_1^{j-1}, f_1^{j-1}, m, \mathbf{e})} \times \pi_t(f_j \mid e_{a_j})$$

IBM Model II: changes only one element in IBM Model I:

- IBM Model I does not take into account the position of words in both strings

$$P(a_j \mid a_1^{j-1}, f_1^{j-1}, m, \mathbf{e}) = P(a_j \mid j, l, m) := \pi_A(a_j \mid j, l, m)$$

Where $\pi_A(.|.)$ are parameters to be learned from data.

IBM Models III, IV and V concentrate on more complex alignments allowing, e.g., $1 - to - n$ (fertility)

IBM Model II Parameters

$$P(a_1^m, f_1^m \mid e_1 \dots e_l) \approx \epsilon \times \prod_{j=1}^{m} \pi_A(a_j \mid j, l, m) \times \pi_t(f_j \mid e_{a_j})$$

Parameters: $\{\pi_A(a_j \mid j, l, m)\}$ and $\{\pi_t(f_j \mid e_{a_j})\}$

Dilemma

To estimate these parameters we need word-alignment
To get word-alignment we need these parameters.

Estimating Model Parameters

Maximum-Likelihood Estimation of model M on parallel corpus **C**

$$\arg\max_{m \in M} P(\mathbf{C} \mid m) = \arg\max_{m \in M} \prod_{\langle \mathbf{e},\mathbf{f} \rangle \, in \, \mathbf{C}} P_m(\mathbf{e} \mid \mathbf{f})$$

Example IBM Model I:

- Model M is defined by model parameters.
- Data is incomplete: no closed form solution.
- Expectation-Maximization (EM) sketch

  Init: Set the parameters at some m_0 and let $i = 0$

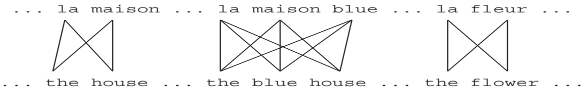
  Repeat until convergence (in perplexity)

  $EM_i(\mathbf{C})$ = **C** completed using estimate m_i

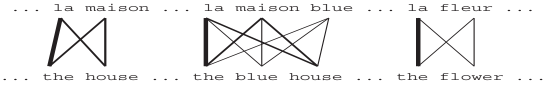
  $EM_i(\mathbf{C})$ contains m_i-expectations over $\langle \mathbf{e}, \mathbf{f}, \mathbf{a} \rangle$: $P(\mathbf{a} \mid \mathbf{f}, \mathbf{e})$

  m_{i+1} = Relative Frequency Estimates from $EM_i(\mathbf{C})$.

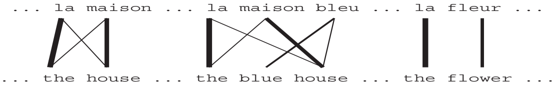
EM for Lexicon and Word Alignment Probs



EM for Lexicon and Word Alignment Probs



EM for Lexicon and Word Alignment Probs

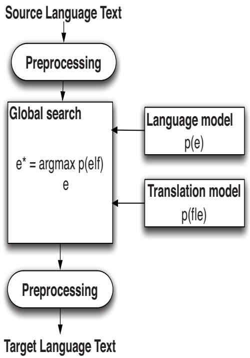


EM for Lexicon and Word Alignment Probs

Translation Using EM Estimates

- Lexicon probability estimates: $\{\hat{\pi}_t(f_j \mid e_{a_j})\}$
- Alignment probabilities: $\{\hat{\pi}_A(a_j \mid j, m, l)\}$
- Translation Model + Language Model + Decoder

$$\arg\max_{\mathbf{e}} P(\mathbf{e} \mid \mathbf{f}) = \arg\max_{\mathbf{e}} P(\mathbf{e}) \times \sum_{\mathbf{a}} P(\mathbf{a}, \mathbf{f} \mid \mathbf{e})$$



Viterbi Word-Alignment using EM estimates

After EM has stabilized on estimates

$$\{\hat{\pi}_t(f_j \mid e_{a_j})\} \quad \textit{and} \quad \{\hat{\pi}_A(a_j \mid j, m, l)\}$$

For every $\langle \mathbf{f}, \mathbf{e} \rangle$ in $\mathbf{C}$ apply the following

$$\arg\max_{a_1^m} P(a_1^m, f_1^m \mid e_1 \ldots e_l) \approx$$

$$\arg\max_{a_1^m} \epsilon \times \prod_{j=1}^{m} \hat{\pi}_A(a_j \mid j, m, l) \times \hat{\pi}_t(f_j \mid e_{a_j})$$

HMM Alignment Model: General Form

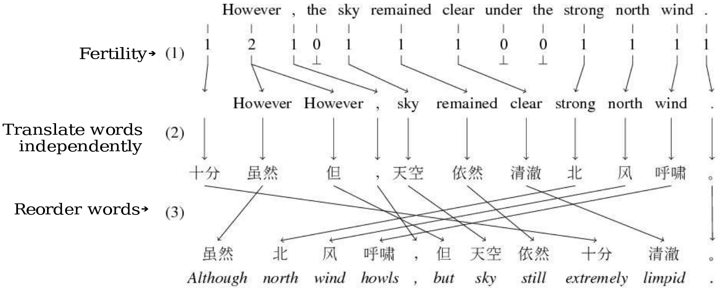
$$P(a_1^m, f_1^m \mid e_1 \dots e_l) \approx \epsilon \times \prod_{j=1}^{m} \underline{P(a_j \mid a_1^{j-1}, f_1^{j-1}, m, \mathbf{e})} \times \pi_t(f_j \mid e_{a_j})$$

- Words do not move independently of each other: condition word movement on previous word movement

$$P(a_j \mid a_1^{j-1}, f_1^{j-1}, m, \mathbf{e}) \approx P(a_j \mid a_{j-1}, m)$$

IBM Model III (and IV): Example

- A hidden word-alignment **a**: $P(\mathbf{f} \mid \mathbf{e}) = \sum_{\mathbf{a}} P(\mathbf{a}, \mathbf{f} \mid \mathbf{e})$



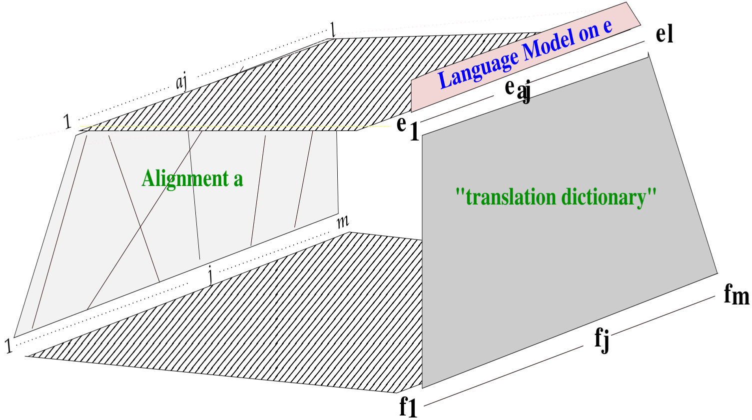
Estimate alignment + lexicon + reordering + fertility parameters.

Word-based Models (Och & Ney 2003)

Table 1
Overview of the alignment models.

Model	Alignment model	Fertility model	E-step	Deficient
Model 1	uniform	no	exact	no
Model 2	zero-order	no	exact	no
HMM	first-order	no	exact	no
Model 3	zero-order	yes	approximative	yes
Model 4	first-order	yes	approximative	yes
Model 5	first-order	yes	approximative	no
Model 6	first-order	yes	approximative	yes

Word-Alignment As Hidden Structure: Sufficient?



We assumed alignment between words and dictionary:

- Alignment **a** and the length m: $P(\mathbf{a} \mid \mathbf{e})$
- Dictionary $P(\mathbf{f} \mid \mathbf{e}, \mathbf{a})$

Limitations of Word-based Models

Limitations of word-based translation:

- Many-to-one and many-to-many is common: "Makes more difficult"/bemoeilijkt "Dat richtte (hen) ten gronde"/"That destroyed (them)"
- Reordering takes place (often) by whole blocks. Reordering individual words increases *ambiguity.* "The (big heavy) cow/la vaca (pesada grande)"
- Translation works by "fixed expressions" (idiomatic). Concatenating word-translations increases *ambiguity.*

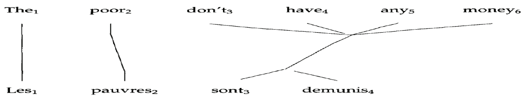
Estimates of $P(\mathbf{f} \mid \mathbf{e})$ by word-based models are inaccurate.

Instead of words as basic events: multi-word events in corpus.

Obtaining Symmetrized Word Alignments

Asymetric Alignments: Limitations

- Word-based models presented so far are based on asymmetric word alignment.
  Each position i in $\mathbf{f}$ is aligned with at most one position in $\mathbf{e}$: $\mathbf{a}_i$
- What about such word alignments?

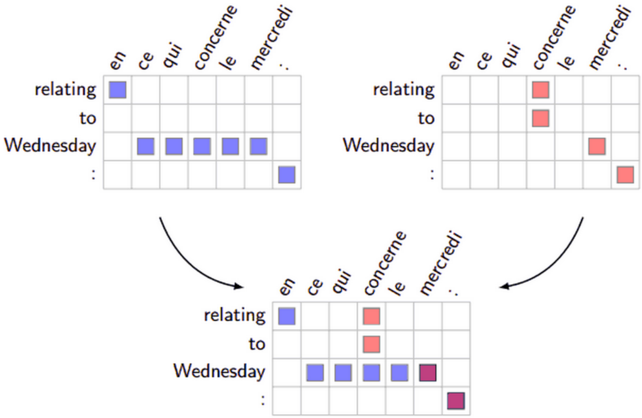


- Or when a word in $\mathbf{f}$ translated into two or more in $\mathbf{e}$?

Symmetrization Heursitics

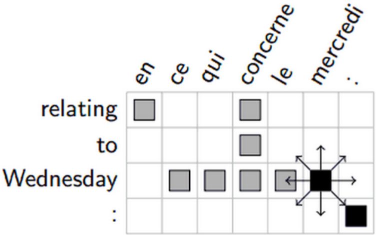
Obtain $A_{\mathbf{f}\to\mathbf{e}}$ and $A_{\mathbf{e}\to\mathbf{f}}$

From Intersection $A_{\mathbf{f}\to\mathbf{e}} \cap A_{\mathbf{e}\to\mathbf{f}}$ to Union $A_{\mathbf{f}\to\mathbf{e}} \cup A_{\mathbf{e}\to\mathbf{f}}$



Symmetrization Heursitics Algoritm

Obtain $A_{\mathbf{f}\rightarrow\mathbf{e}}$ and $A_{\mathbf{e}\rightarrow\mathbf{f}}$
From Intersection $A_{\mathbf{f}\rightarrow\mathbf{e}} \cap A_{\mathbf{e}\rightarrow\mathbf{f}}$ to Union $A_{\mathbf{f}\rightarrow\mathbf{e}} \cup A_{\mathbf{e}\rightarrow\mathbf{f}}$



- from intersection $A = A_{f\rightarrow e} \cap A_{f\rightarrow e}$ to union $A_{f\rightarrow e} \cup A_{f\rightarrow e}$
- step 1 (diagonal): add neighbouring points (f, e) in union s.t. $\nexists(f, e') \in A$ or $\nexists(f', e) \in A$
- step 2 (finalize): add remaining points in union s.t. $\nexists(f, e') \in A$ and $\nexists(f', e) \in A$

Phrase-based Models: Alignment between Phrases

Statistical "Memory-based" Translation

Store arbitrary length source-target translation units from training parallel corpus.

Translate new input by "covering" it with translation units replayed from memory.

Idiomatic = Tiling: Phrase-Based SMT

- Assume word-alignment **a** is given in parallel corpus.
- Phrase-pair = contiguous source-target $\langle n, m \rangle$-grams that are *translational equivalents* under **a**.
- Estimate phrase-pair probabilities $\Theta(\bar{f}_i \mid \bar{e}_i)$
- Translate **f** by "tiling it with phrases with order permutation"

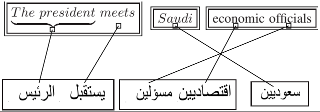
PBSMT some references

- Alignment-template Approach to Stat. Machine Translation (RWTH Aachen 1999)
- Phrase-based statistical machine translation (Zens, Och and Ney 2002)
- Phrase based SMT (Koehn, Och and Marcu 2003)
- Joint Phrase-based SMT (Wang and Marcu 2005)
- Statistical Machine Translation. (Ph. Koehn, Cambridge University Press 2010)

Relation to EBMT 1984; Data-Oriented Translation (2000).

Phrase-Based Models: Conceptual

Segment foreign sentence **f** into I phrases $\bar{f}_1^I$



$$\arg\max_{\mathbf{e}} P(\mathbf{e} \mid \mathbf{f}) = \arg\max_{\mathbf{e}} P(\mathbf{e}) \times P(\mathbf{f} \mid \mathbf{e})$$

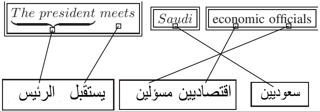
$$P(\mathbf{f} \mid \mathbf{e}) = \sum_{\langle \bar{f}_1^I, \bar{e}_1^I \rangle} P(\bar{f}_1^I, \bar{e}_1^I \mid \mathbf{e}) \quad \prod_{i=1}^{I} P(\bar{f}_i \mid \bar{e}_i) \times d(start_i - end_{i-1} - 1)$$

$$\arg\max_{\mathbf{e}} P(\mathbf{f} \mid \mathbf{e}) \approx \arg\max_{\langle \bar{f}_1^I, \bar{e}_1^I \rangle} \overbrace{\prod_{i=1}^{I} \Theta(\bar{f}_i \mid \bar{e}_i)}^{ph.table} \times \overbrace{d(start_i - end_{i-1} - 1)}^{Dist.reord.}$$

$start_i/end_i$ are positions of first/last words of $\bar{f}_i$ (translateing to $\bar{e}_i$).
$d(x) = \alpha^x$ exponentially decaying in words skipped ($\alpha \in (0, 1]$).

Phrase-Based Models: Linear-interpolation

Segment foreign sentence **f** into I phrases $\bar{f}_1^I$



Log-linear interpolation of factors:

$$score(\mathbf{e}|\mathbf{f}) = \sum_{\mathbf{f} \in F} \lambda_{\mathbf{f}} \times \log H_{\mathbf{f}}(\mathbf{e}, \mathbf{f})$$

Where set F consists of:

- Bag of phrases translation = $\prod_{i=1}^{I} \Theta(\bar{f}_i \mid \bar{e}_i)$

d() Phrases reordered with reordering model $d(.)$

lm Language model (5-grams or even 7-grams).

other Smoothing + length penalty terms.

Topics to discuss

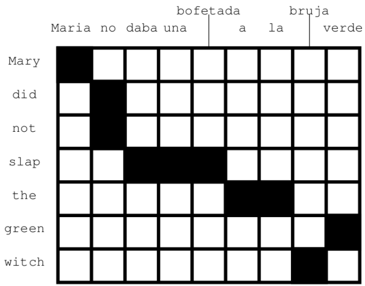
- Phrase table extraction
- Estimating $\{\Theta(\bar{f}_i \mid \bar{e}_i)\}$ and $\{\Theta(\bar{e}_i \mid \bar{f}_i)\}$
- Lexicalized and hierarchical phrase reordering models
- Other: phrase, length penalty . . .
- Log-linear interpolation and minimum error-rate training
- Decoding and optimization

Extracting phrase pairs

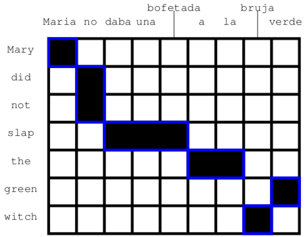
A phrase pair $\langle \bar{f}, \bar{e} \rangle$ is consistent with alignment **a** iff

- Non-empty: at least one alignment pair from **a** is in $\langle \bar{f}, \bar{e} \rangle$
- No foreign positions inside $\langle \bar{f}, \bar{e} \rangle$ aligned to positions outside it
- No english positions inside $\langle \bar{f}, \bar{e} \rangle$ aligned to positions outside it

Extracting phrase pairs

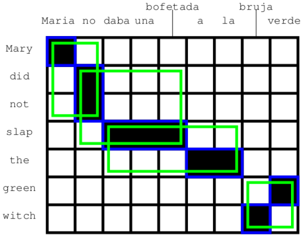


Word Alignment Induced Phrases (2)



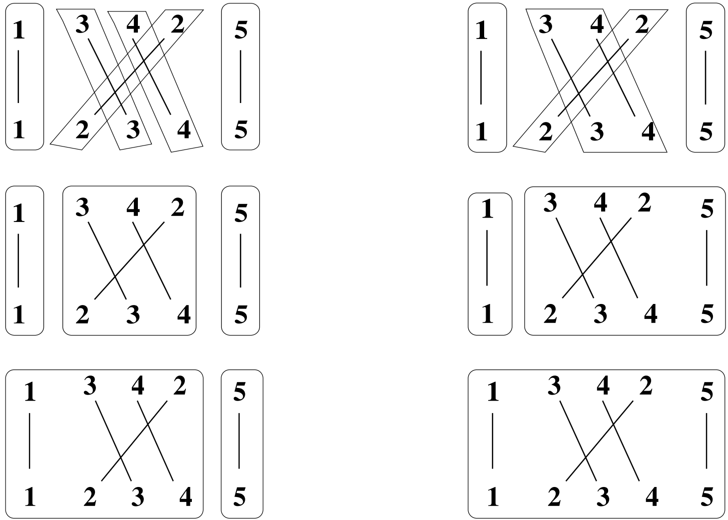
(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green)

Word Alignment Induced Phrases (3)



(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)

Extracting phrases from permutations



Phrase pair weights

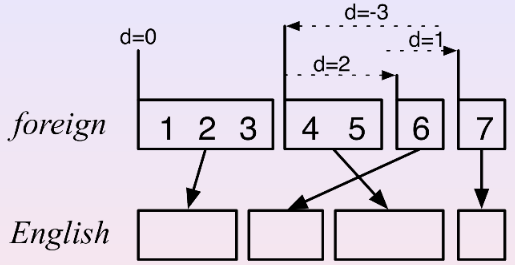
Extract phrase pairs from corpus into multiset

$$Tab = \{\langle \bar{f}, \bar{e} \rangle, freq(\bar{f}, \bar{e})\}$$

Weights for $\langle \bar{f}, \bar{e} \rangle$

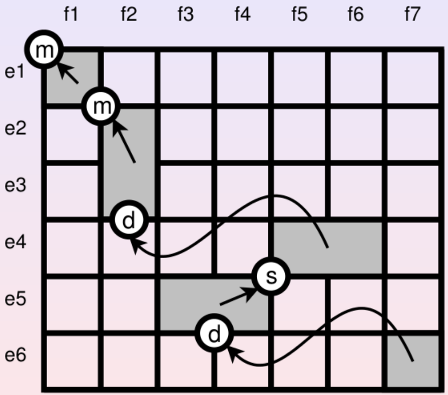
- $\Theta(\bar{f} \mid \bar{e}) = \dfrac{freq(\bar{f}, \bar{e})}{\sum_{\langle \bar{f}', \bar{e} \rangle \in Tab} freq(\bar{f}', \bar{e})}$
- $\Theta(\bar{e} \mid \bar{f}) = \dfrac{freq(\bar{e}, \bar{f})}{\sum_{\langle \bar{f}, \bar{e}' \rangle \in Tab} freq(\bar{f}, \bar{e}')}$
- Smoothing with lexical word alignment estimates from IBM models

Distance-Based Reordering Sketch



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
		Total	6

Lexicalized Reordering Sketch (Tillmann 2004)



Three types: **monotone**, **swap**, **discontinuous**

Condition on phrase pair : $p(o|e, f)$

Gives six features (3 orientations, current and next phrase pair).

Limited generalization over parallel data (1)

Non-productive Phrase Table: Phrase Variants?

Morphological e.g., changing inflection, agreement

Al$areka**t** Alhindiyya the−companies the−Indian	$areka hindiyya company Indian
the Indian companies	(an) Indian company

Syntactic e.g. adding adjective/proposition/adverbials

the fish in the deep sea swims	the fish swims

Reordering minor reordering of same words not allowed
In Arabic V-S-O and S-V-O are allowed.

Semantic e.g. synonyms, paraphrases

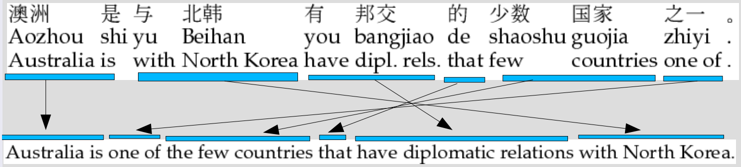
Non-productive Phrase Table = Data Sparseness

Limited generalization over parallel data (2)

Reordering

Local, monotone, almost non-lexicalized reordering.

What about long range reordering?



Five phrases need to be reversed: see Chiang 2007 (J. CL).

Reordering target phrases with a coarse "source road map"?

Limitations: Data-Sparseness

Non-productive phrase table + Local, *Uncharted* reordering

$\Downarrow$

Data-sparseness: Shorter phrases will apply down to word level.

$\Downarrow$

Shorter phrases combined assuming independence.

$\Downarrow$

Target phrase selection hard due to large hypotheses lattice

Target Language Model = Only "GLUE" over target phrases.

The Shorter the Phrases, the Greater the Risk

Idiomatic Approach: GOOD, BAD and UGLY

Phrases as atomic units

Good: Less ambiguity in lexical choice and reordering.
Match-Retrieve exactly is largely safe.

Bad: Weak generalization over data.
No phrase variants, weak reordering

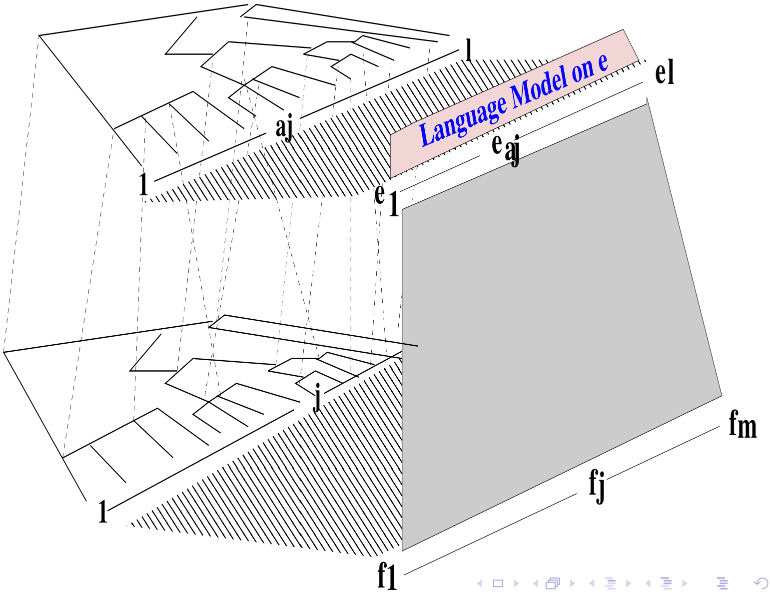
Ugly: Fall-back on shorter phrases downto word-to-word
LM as "glue" is insufficient.

Idiomatic approach does not alleviate data-sparseness

How Should We Translate Novel Phrases?

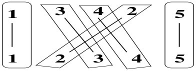
Towards the land of bi-trees
Alignments between Tree Pairs, ITG, Hierarchical Models and Syntax

Hidden Structure of Translation: Tree Pairs



Reordering n words

- Permutations of n words: $n!$

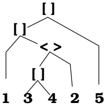


- Surely not all permutations are needed! (Wu 1995)
- Use trees and allow permutations on the nodes?

  There is an exponential number of trees in n 

- ITG hypothesis (Wu 1995)

  Assume binary trees with two operations

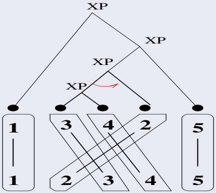


- Phrase-based forms of ITG (Chiang 2005; 2007): Hiero

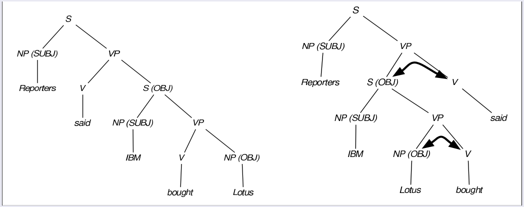
Syntax-Driven Phrase Translation

Syntax-driven Re-Ordering

Hierarchical (ITG)



Linguistic Syntax

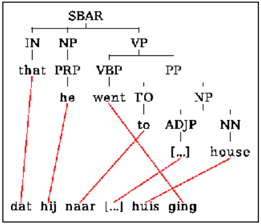


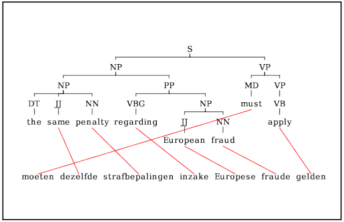
Is translation syntactically cohesive?

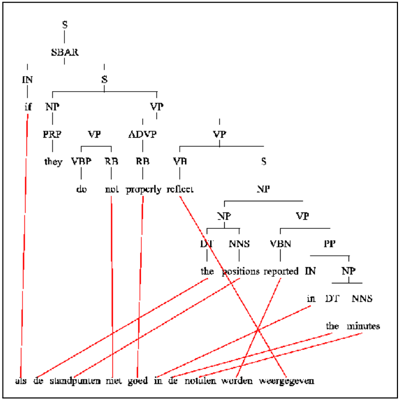
Reordering == Moving children in parse tree?

- Binary: monotone or inverted order at every node.
- Lexical elements can be phrase pairs.
- Covers word-alignments in parallel corpora?

Word order difference and syntax: Impression

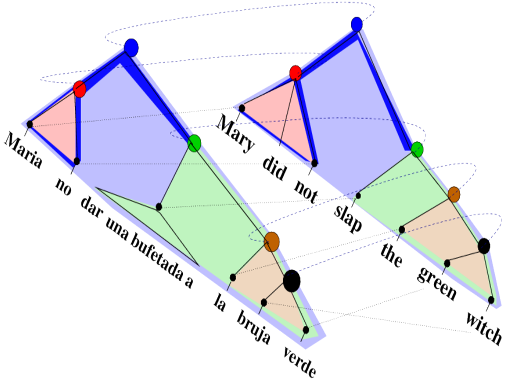






Phrase-based Hierarchical Model: Hiero (Chiang 2005; 2007)

Extracting phrase-pairs with gaps (hierarchical trees):

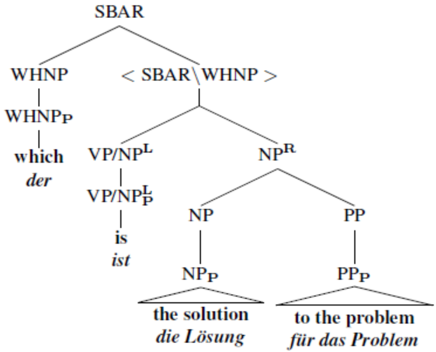


$X \rightarrow X_1$ dar una bufetada a X_2 / X_1 slap X_2

$X \rightarrow$ Maria no X la bruja verde / Mary did not X the green witch

ITG with syntactic labels

$$\mathrm{SBAR} \rightarrow [\mathrm{WHNP}\ \ \mathrm{SBAR}\backslash\mathrm{WHNP}] \qquad \text{(a)}$$
$$\mathrm{SBAR}\backslash\mathrm{WHNP} \rightarrow \langle \mathrm{VP/NP}^{\mathrm{L}}\ \ \mathrm{NP}^{\mathrm{R}} \rangle \qquad \text{(b)}$$
$$\mathrm{NP}^{\mathrm{R}} \rightarrow [\mathrm{NP}\ \ \mathrm{PP}] \qquad \text{(c)}$$
$$\mathrm{WHNP} \rightarrow \mathrm{WHNP}_{\mathrm{P}} \qquad \text{(d)}$$
$$\mathrm{WHNP}_{\mathrm{P}} \rightarrow \text{which / der} \qquad \text{(e)}$$
$$\mathrm{VP/NP}^{\mathrm{L}} \rightarrow \mathrm{VP/NP}^{\mathrm{L}}_{\mathrm{P}} \qquad \text{(f)}$$
$$\mathrm{VP/NP}^{\mathrm{L}}_{\mathrm{P}} \rightarrow \text{is / ist} \qquad \text{(g)}$$
$$\mathrm{NP}^{\mathrm{R}} \rightarrow \mathrm{NP}^{\mathrm{R}}_{\mathrm{P}} \qquad \text{(h)}$$
$$\mathrm{NP}^{\mathrm{R}}_{\mathrm{P}} \rightarrow \text{the solution / die Lösung} \qquad \text{(i)}$$
$$\mathrm{NP} \rightarrow \mathrm{NP}_{\mathrm{P}} \qquad \text{(j)}$$
$$\mathrm{NP}_{\mathrm{P}} \rightarrow \text{the solution / die Lösung} \qquad \text{(k)}$$
$$\mathrm{PP} \rightarrow \mathrm{PP}_{\mathrm{P}} \qquad \text{(l)}$$
$$\mathrm{PP}_{\mathrm{P}} \rightarrow \text{to the problem / für das Problem} \qquad \text{(m)}$$



Part II: Trevor Cohn
Decoding algorithms and efficiency