ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Machine translation

Lê Thanh Hương
School of Information and Communication Technology
Email: huonglt@soict.hust.edu.vn

An example

- Au sortir de la saison 97/98 et surtout au debut de cette saison 98/99…

- With leaving season 97/98 and especially at the beginning of this season 98/99…

Challenges

1. Capture variation and similarities amongst languages

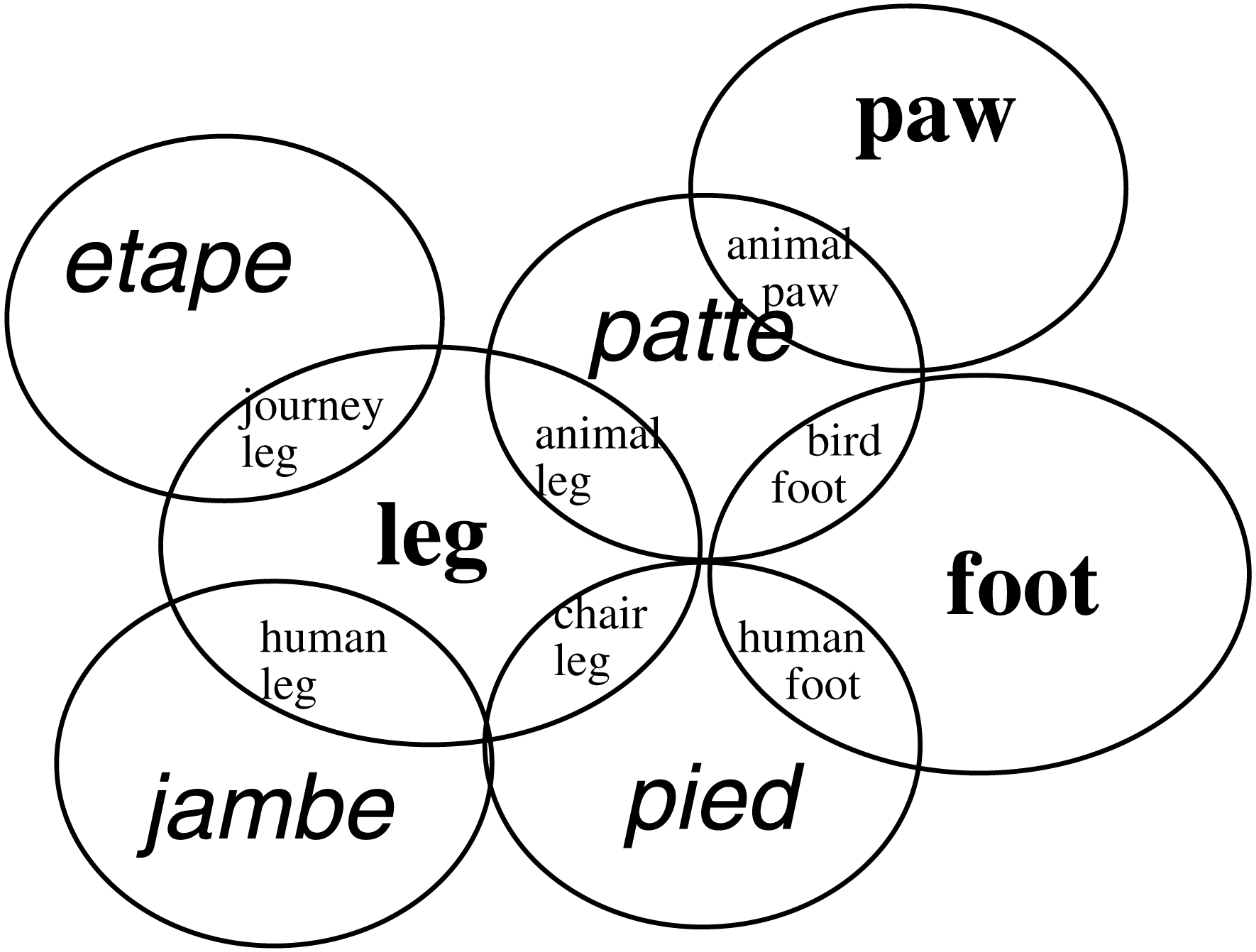
- Morphologically: # morphemes/word:
  - *Isolating* languages (Vietnamese, Cantonese) – 1 word/ 1 morpheme
  - *Polysynthetic languages* (Siberian Yupik), 1 word = a whole sentence
- Degree to which morphemes are segmentable

Challenges

2. Syntax: order of words in a sentence
    - *To Yukio; Yukio ne*
- English vs. Vietnamese:
    - *The* (affix1) *red* (affix2) *flag* (head)
    - *Lá cờ* (head) *đỏ* (affix2) *ấy* (affix1)
3. Differences in specificity

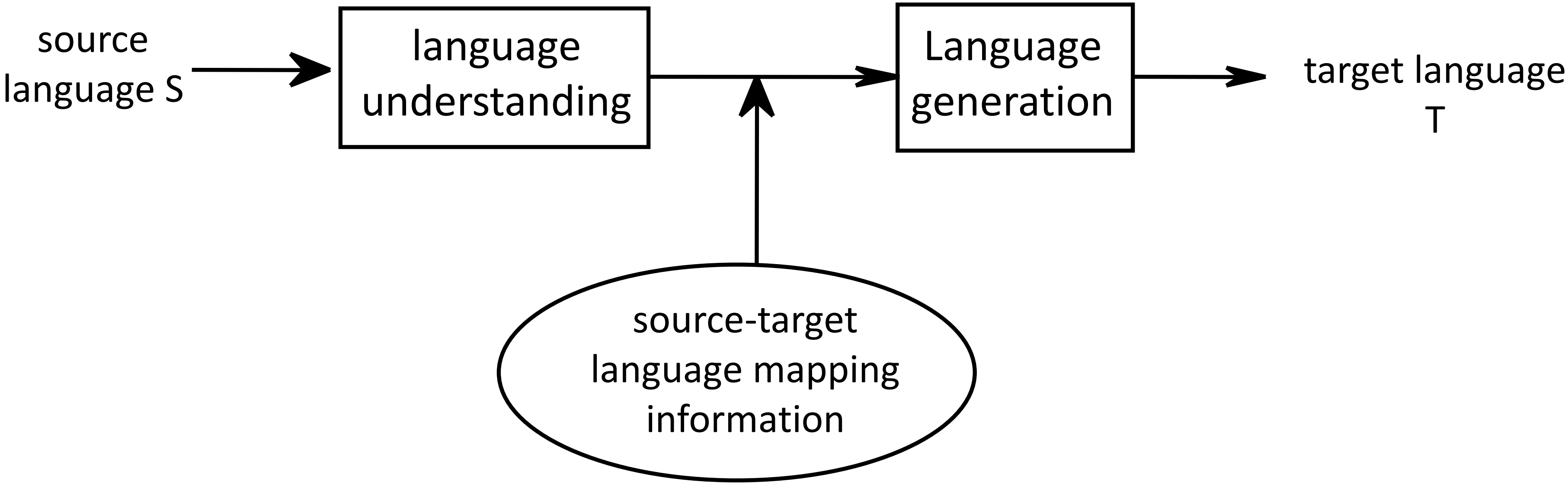
English	brother	Vietnamese	anh em
English	wall	German	wand (inside) mauer(outside)
German	berg	English	hill mountain

Conceptual space



Lexical gap: Jp, no word for *privacy;* Eng: no word for *oyakoko* (filial piety)

Three main blocks in machine translation



Language understanding

1. Lexical ambiguity:

   English: *book* - Spanish *libro, reservar*

$\Rightarrow$ Use syntactic context

2. Syntactic ambiguity:

   I saw the guy on the hill with the telescope

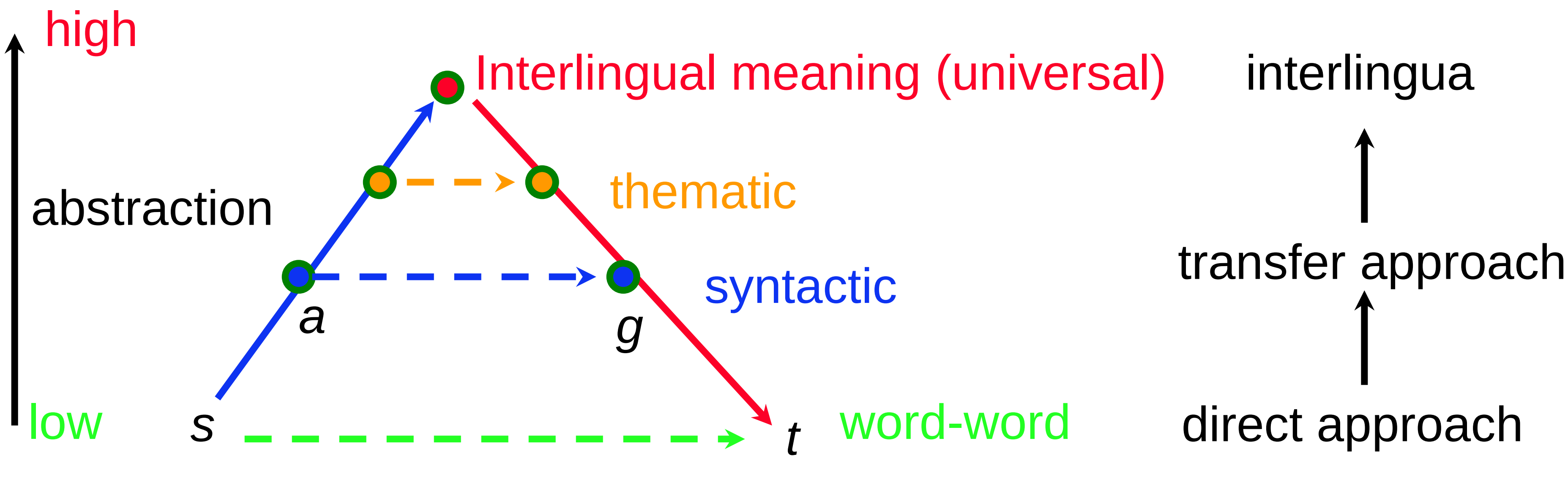
3. Semantic ambiguity

- *E: While driving, John swerved & hit a tree*

  John's car

- *S: Minetras que John estaba manejando, se desvio y golpeop con un arbo*

Methods of machine translation

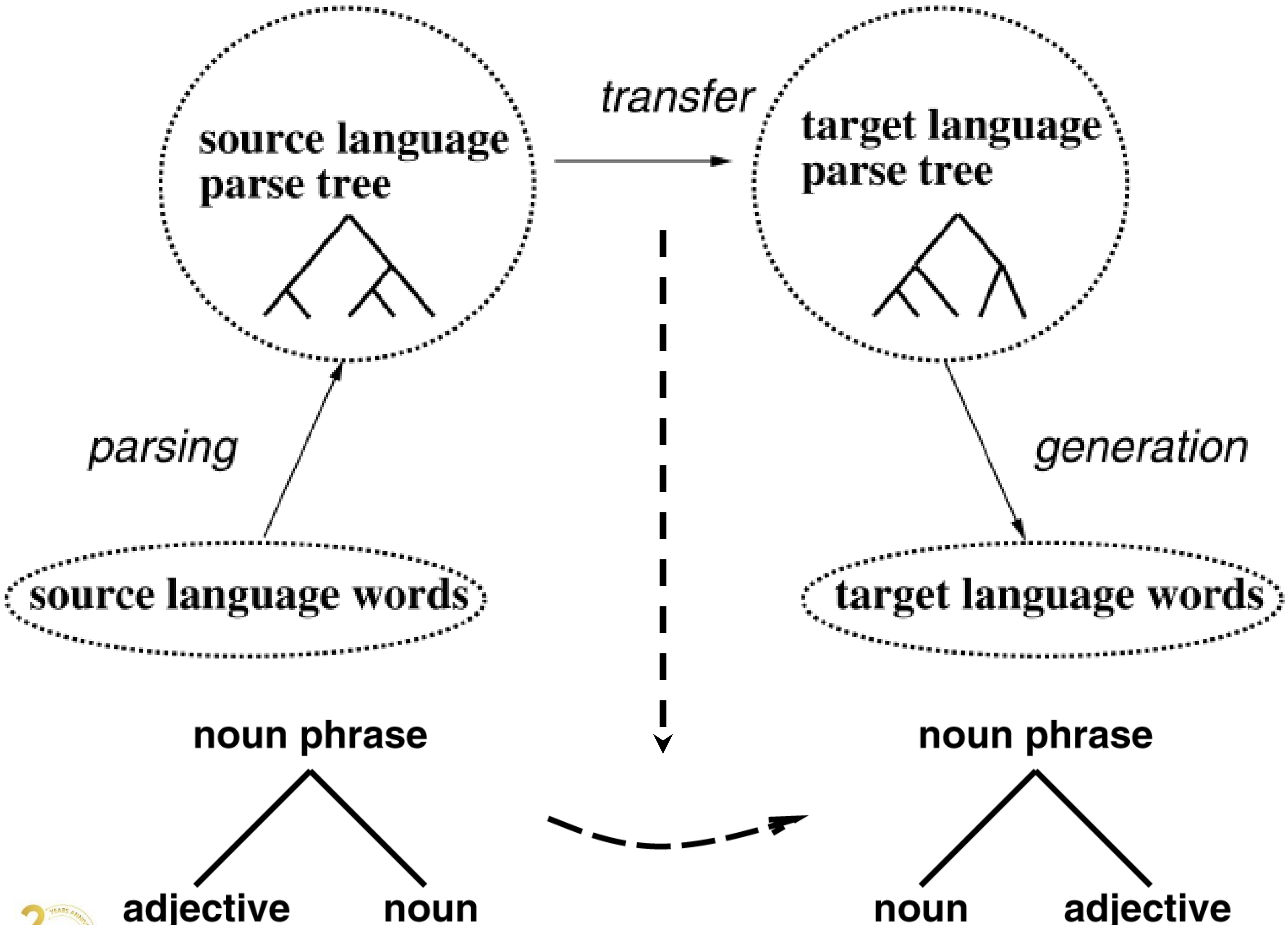


a = *a(s)*

g = *f(a(s)); f* - transfer function

t=g(f(a(s)))

The triangle/transfer diagram



List of transforms

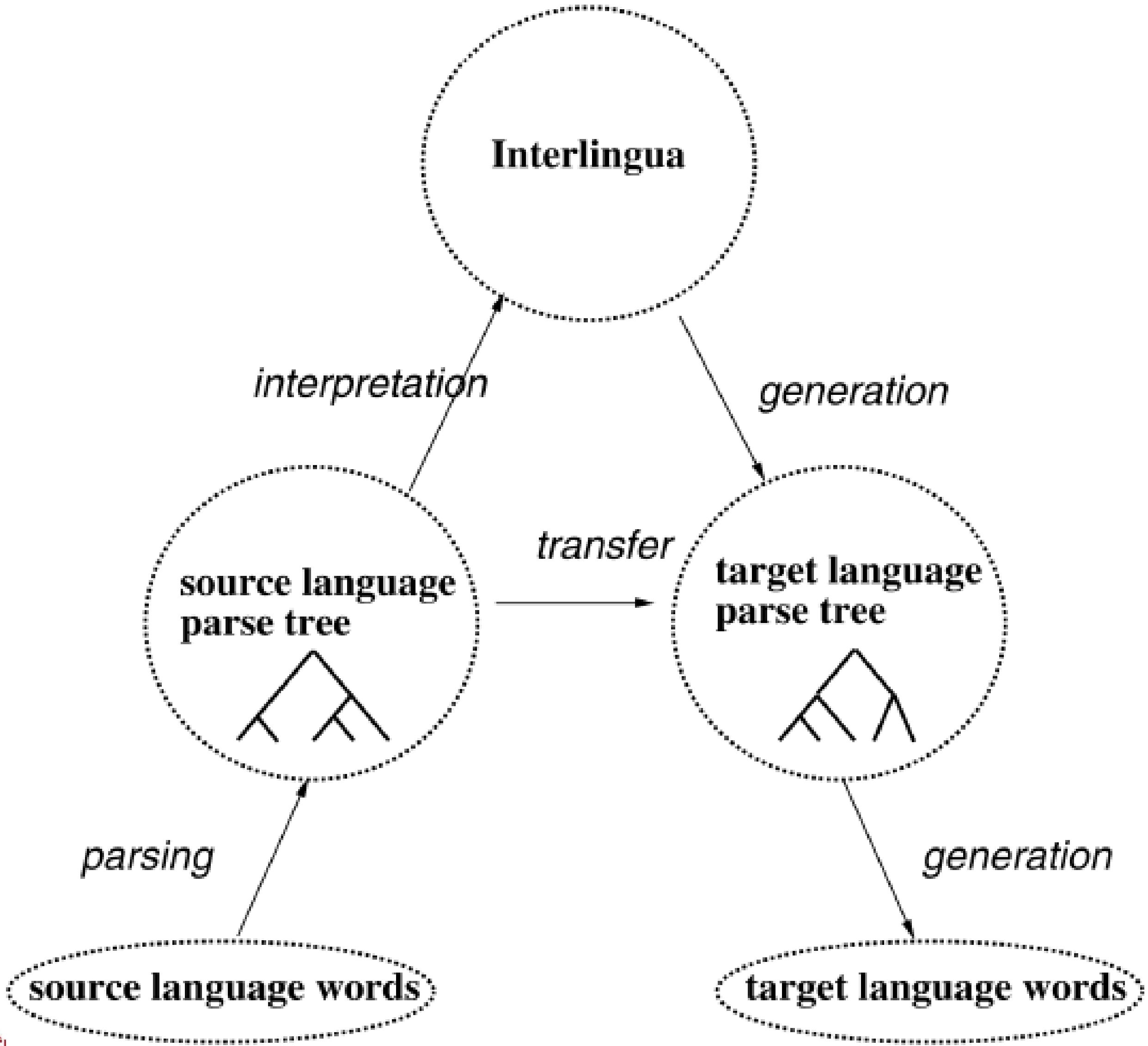
English to French:

1. $NP \rightarrow Adjective_1\ Noun_2$
   $\Rightarrow$
   $NP \rightarrow Noun_2\ Adjective_1$

Japanese to English:

2. $Existential\text{-}There\text{-}Sentence \rightarrow There_1\ Verb_2\ NP_3\ Postnominal_4$
   $\Rightarrow$
   $Sentence \rightarrow (NP \rightarrow NP_3\ Relative\text{-}Clause_4)\ Verb_2$
3. $NP \rightarrow NP_1\ Relative\ Clause_2$
   $\Rightarrow$
   $NP \rightarrow Relative\text{-}Clause_2\ NP_1$

The triangle/transfer diagram



Interlingua approach: using meaning

- Transfer: one pair of rules per language pair
- Objects/events (ontology)

$$
\begin{bmatrix}
\text{event} & \text{gardening} \\
\text{agent} & \begin{bmatrix} \text{man} & \\ \text{number} & \text{sg} \\ \text{definiteness} & \text{indef} \end{bmatrix} \\
\text{aspect} & \text{progressive} \\
\text{tense} & \text{past}
\end{bmatrix}
$$

Statistical machine translation

The Main Idea

- Treat translation as a noisy channel problem

Input (Source) → **The channel (adds "noise")** → "Noisy" Output (target)

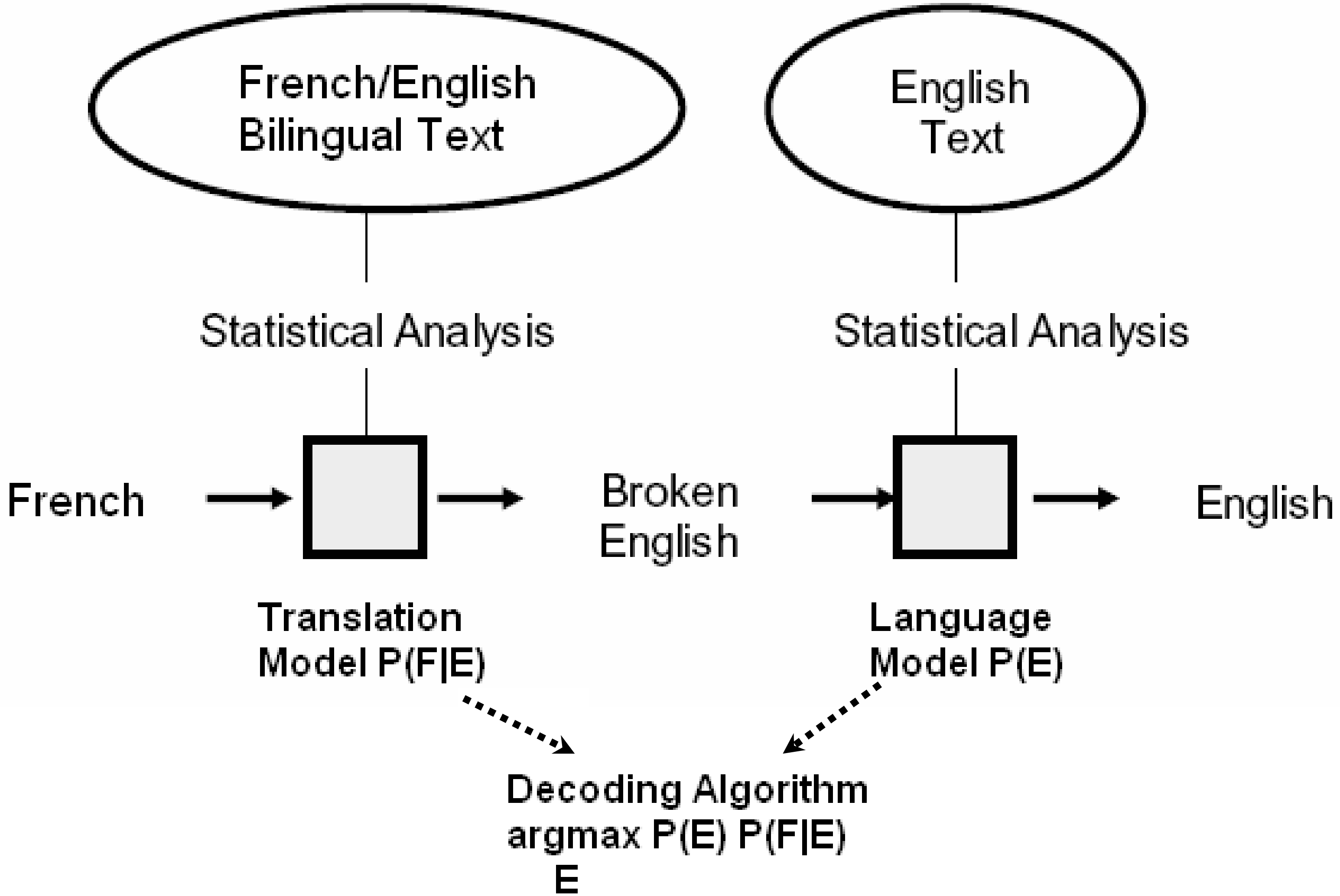
E: English words... → The channel (adds "noise") → F: Les mots Anglais...

- The translation model: $P(E|F) = P(F|E)\, P(E) / P(F)$
- Interested in rediscovering <u>E</u> given <u>F</u>:

  After the usual simplification (P(F) fixed):

$$\boxed{\operatorname{argmax}_E P(E|F) = \operatorname{argmax}_E P(F|E)\, P(E)}$$

A Statistical MT System



The necessities

- Language Model (LM): our *expectation* of seeing a particular *English* (E) sentence, *a priori*:

  P(E)
- Translation Model (TM): Target sentence in *French* (F) *given* source sentence in English:

  P(F|E)
- Search procedure
  - Given F, find best E using the LM and TM distributions.
- Usual problem: sparse data!
  - We cannot create a “sentence dictionary” E ↔ F
  - Typically, we do not see a sentence even twice!

Alignment Idea

- TM doesn't care about correct strings of English words
- Use the "tagging" approach:
  - 1 English word ("tag") ~ 1 French word ("word")

  $\rightarrow$ not realistic: even #words in two sentences isn't equal

  $\rightarrow$ use "Alignment".

- Sentence alignment: find some group of sentences in one language corresponds to some other group of sentences in another language.

Sentence Alignment

The old man is happy. He has fished many times. His wife talks to him. The fish are jumping. The sharks await.

El viejo está feliz porque ha pescado muchos veces. Su mujer habla con él. Los tiburones esperan.

Sentence Alignment

1. The old man is happy.
2. He has fished many times.
3. His wife talks to him.
4. The fish are jumping.
5. The sharks await.

1. El viejo está feliz porque ha pescado muchos veces.
2. Su mujer habla con él.
3. Los tiburones esperan.

Sentence Alignment

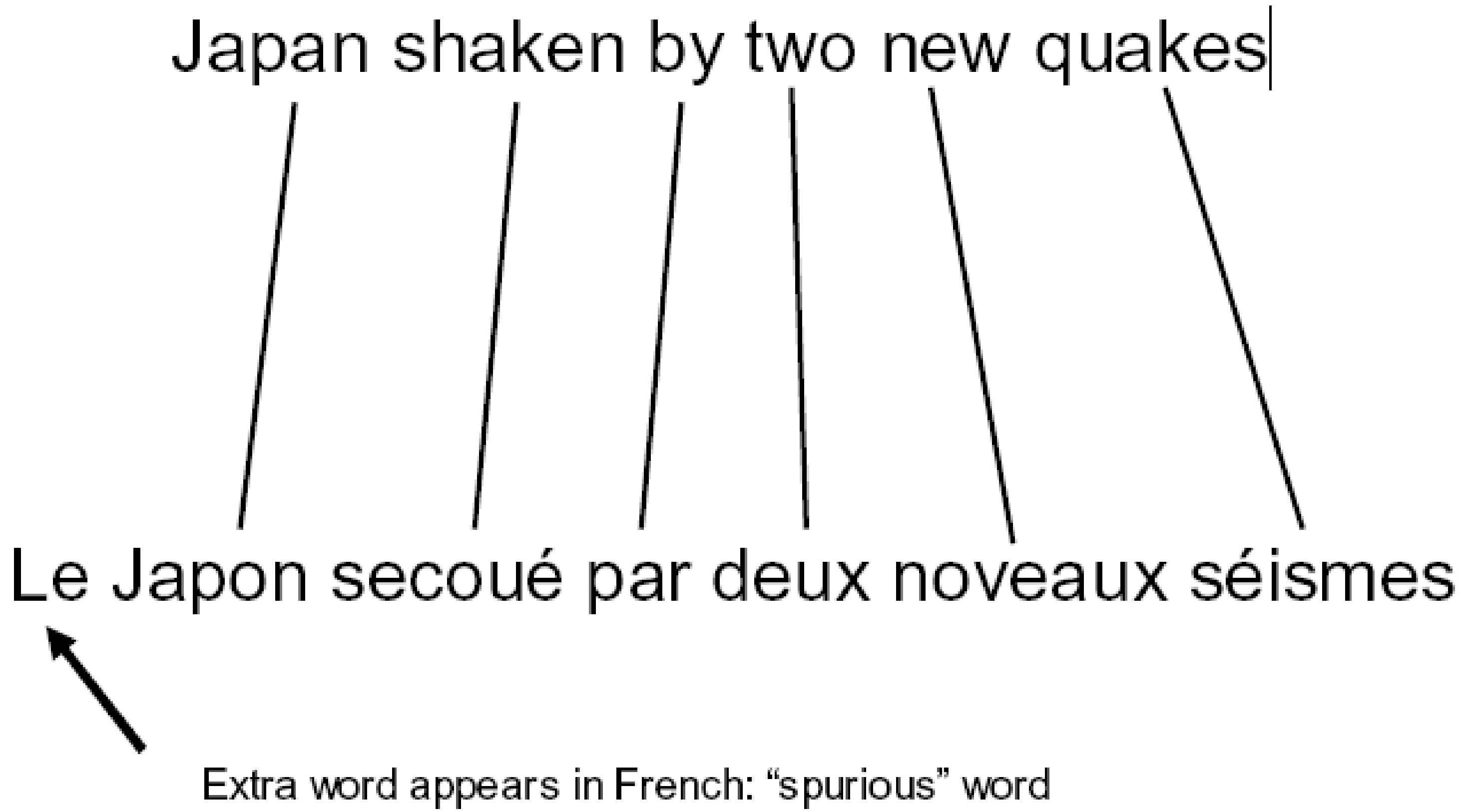
1. The old man is happy.
2. He has fished many times.
3. His wife talks to him.
4. The fish are jumping.
5. The sharks await.

1. El viejo está feliz porque ha pescado muchos veces.
2. Su mujer habla con él.
3. Los tiburones esperan.

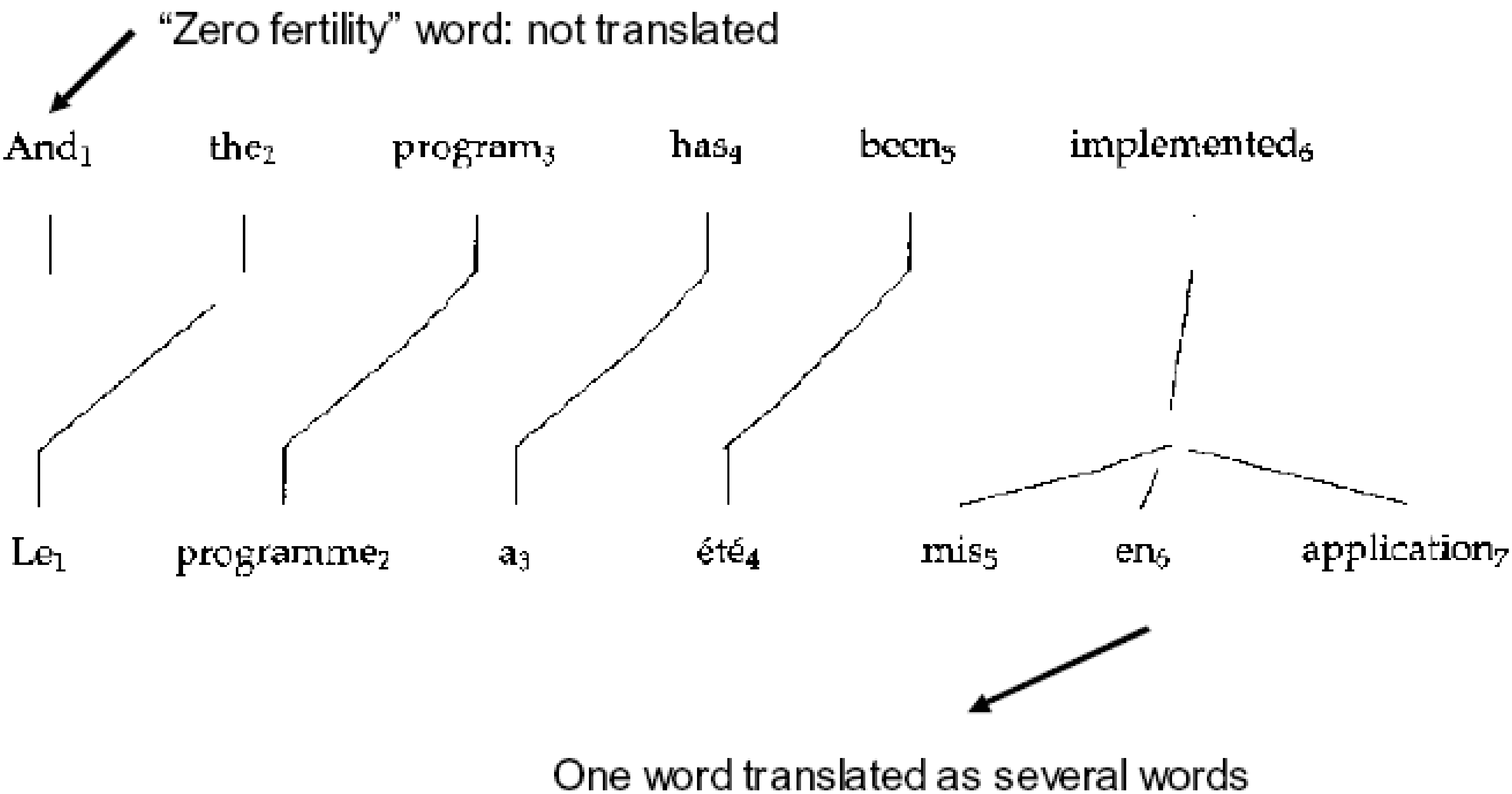
Difficulties:

- Crossing dependencies: the order of sentences are changed in the translation.

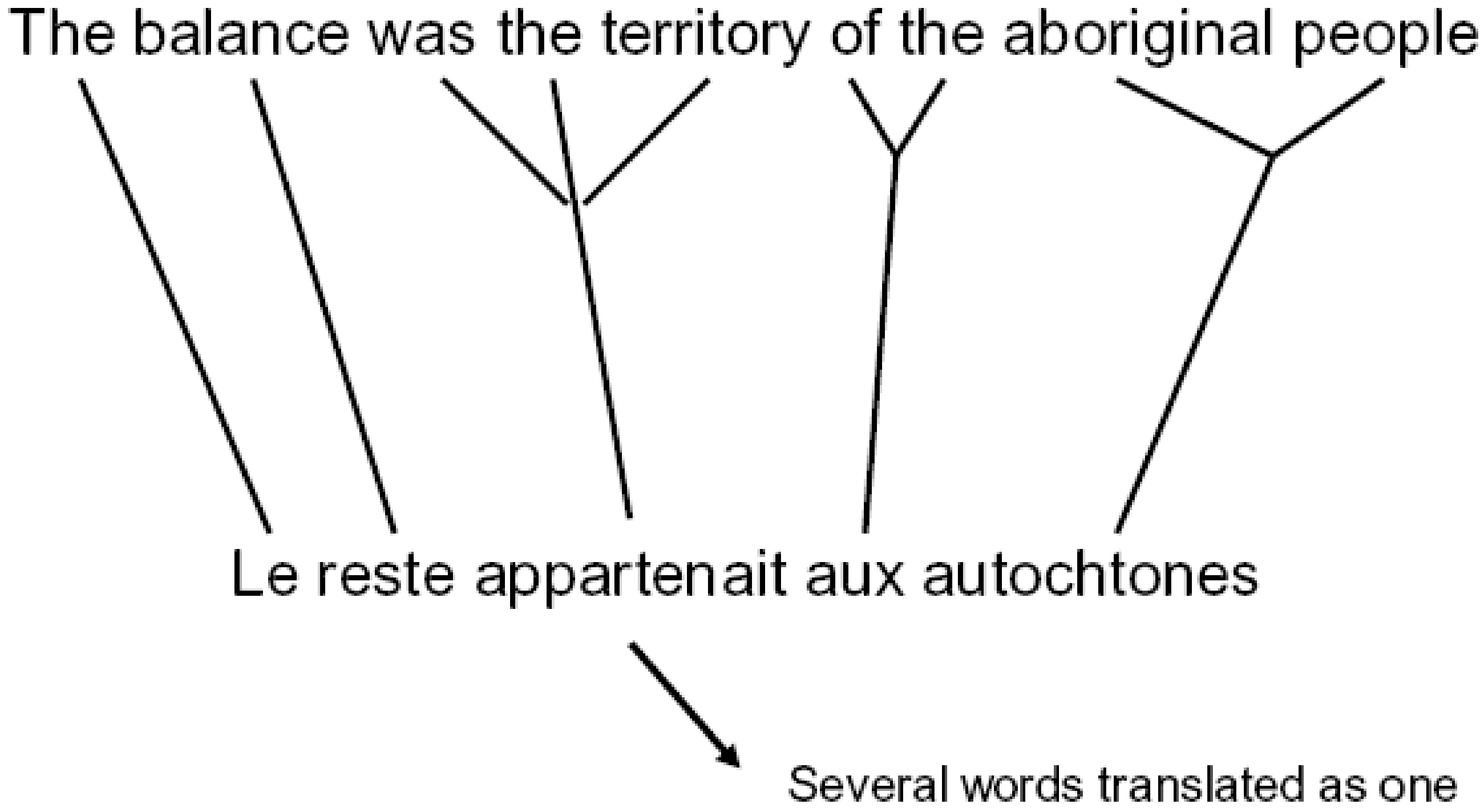
Word alignment - easy



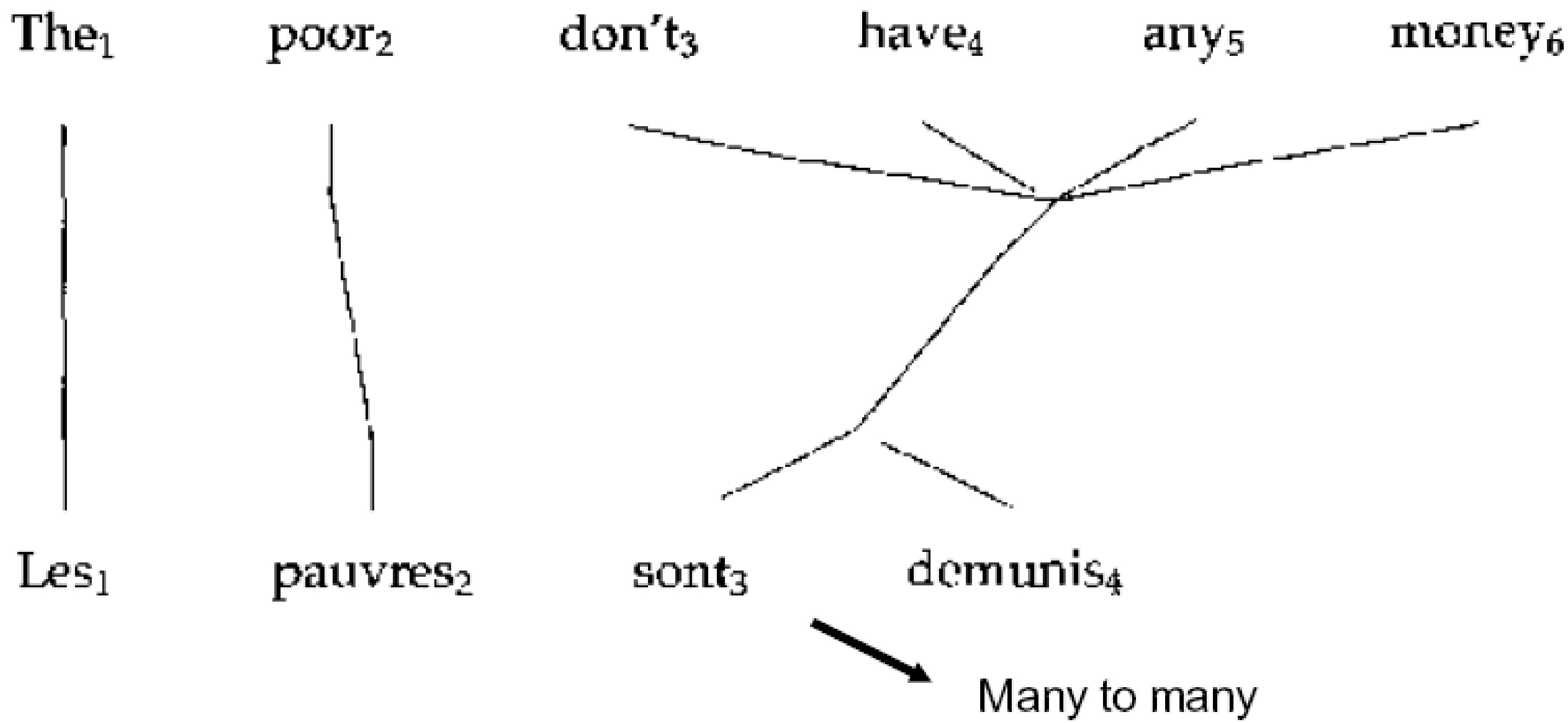
Word alignment - harder



Word alignment - harder



Word alignment - hard



- A line group linking a minimal subset of words is called a ‘cept’ in the IBM work

Word alignment - encoding

0 1 2 3 4 5 6

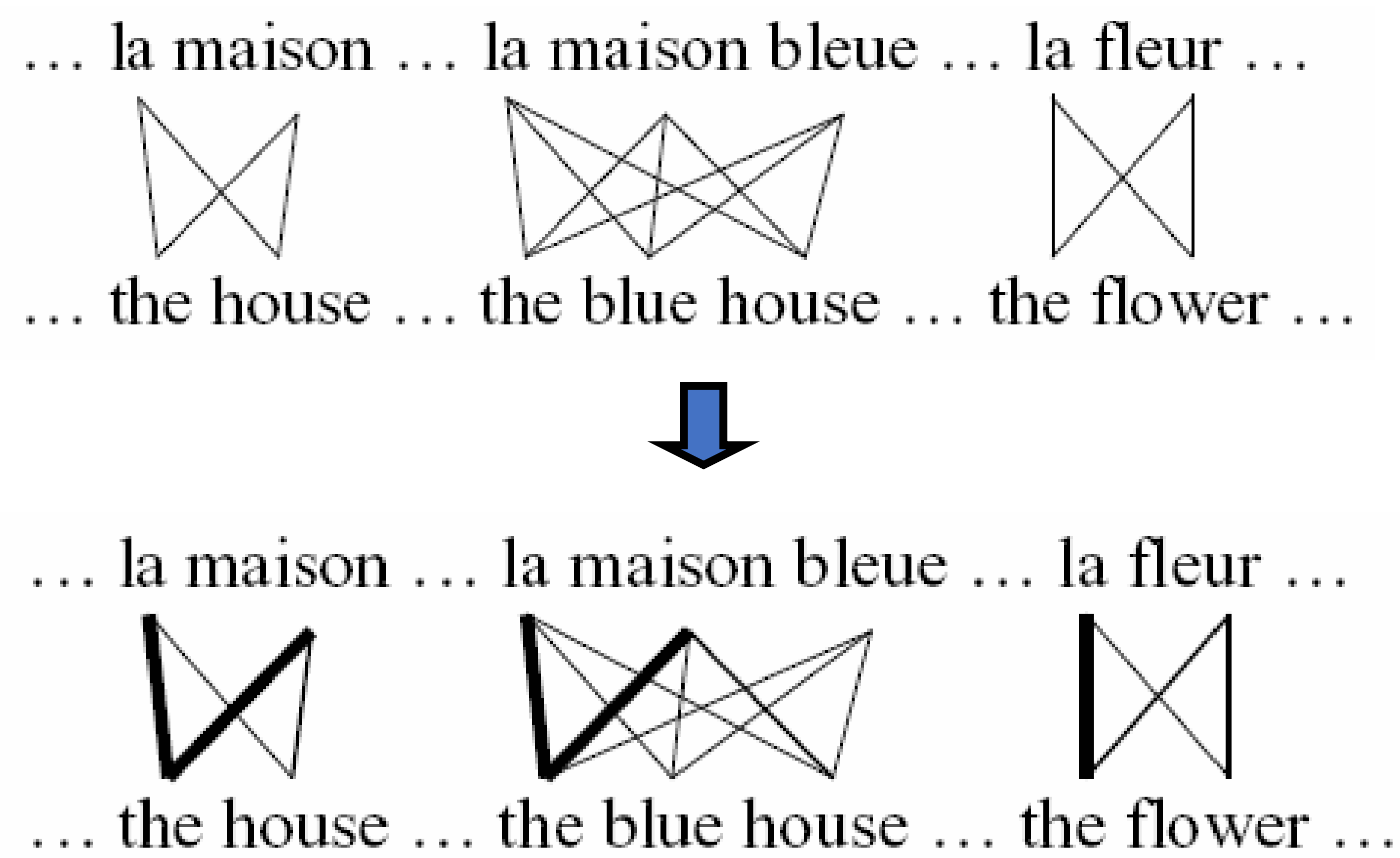
- e_0 And the program has been implemented

- f_0 Le programme a été mis en application

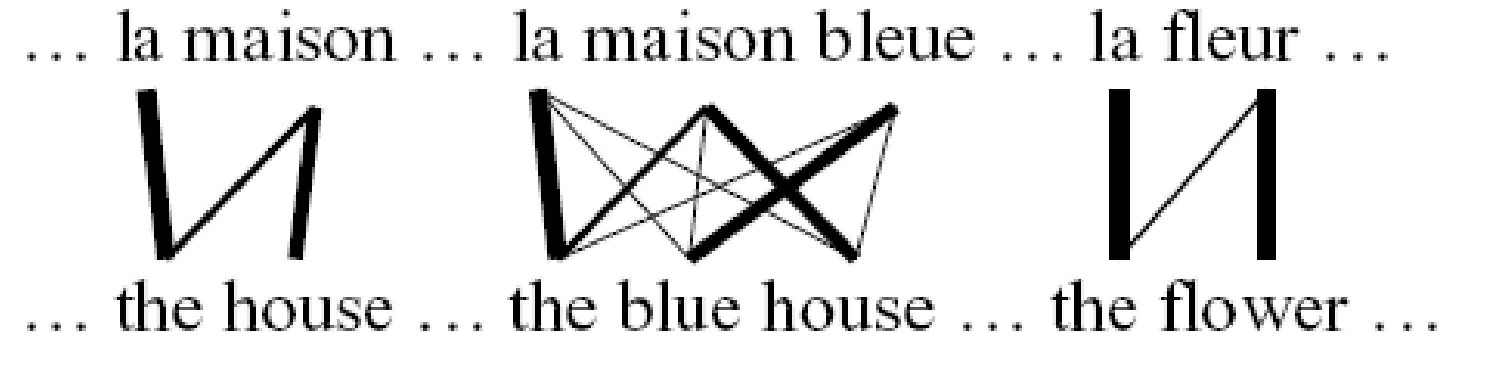
0 1 2 3 4 5 6 7

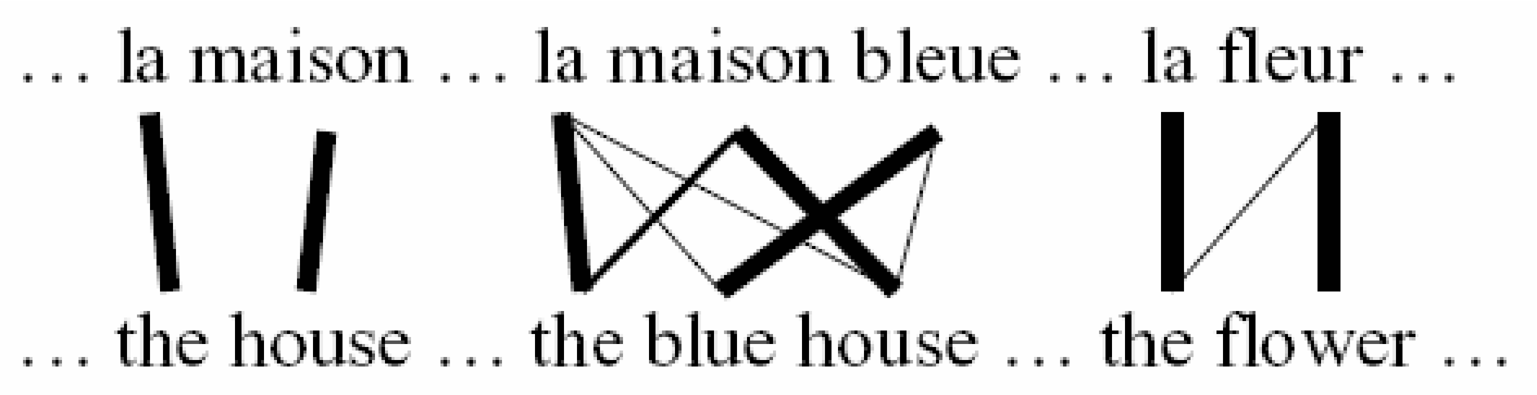
- Linear notation:
  - f_0(1) Le(2) programme(3) a(4) été(5) mis(6) en(6) application(6)
  - e_0 And(0) the(1) program(2) has(3) been(4) implemented(5,6,7)

Word alignment learning with EM

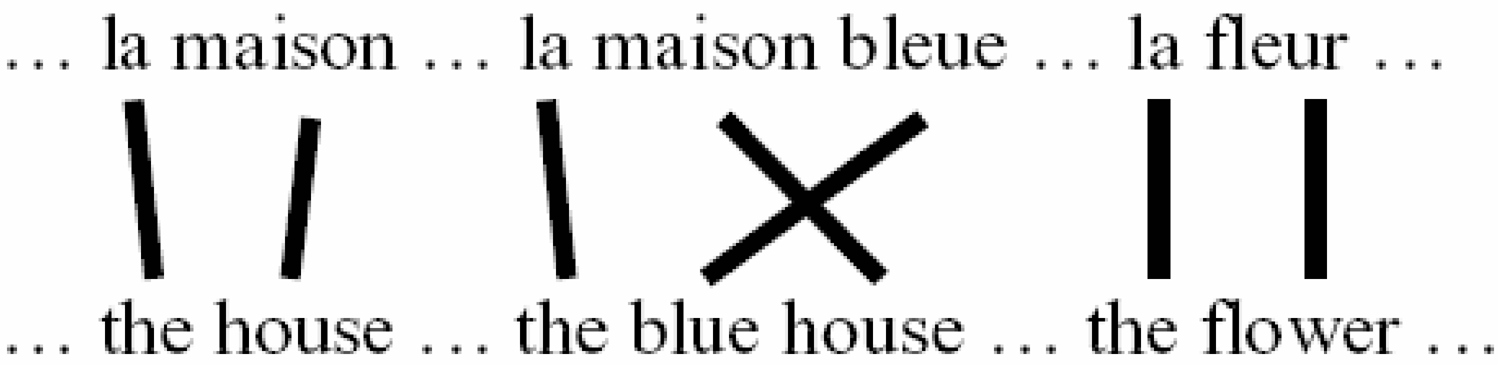


Word alignment learning with EM

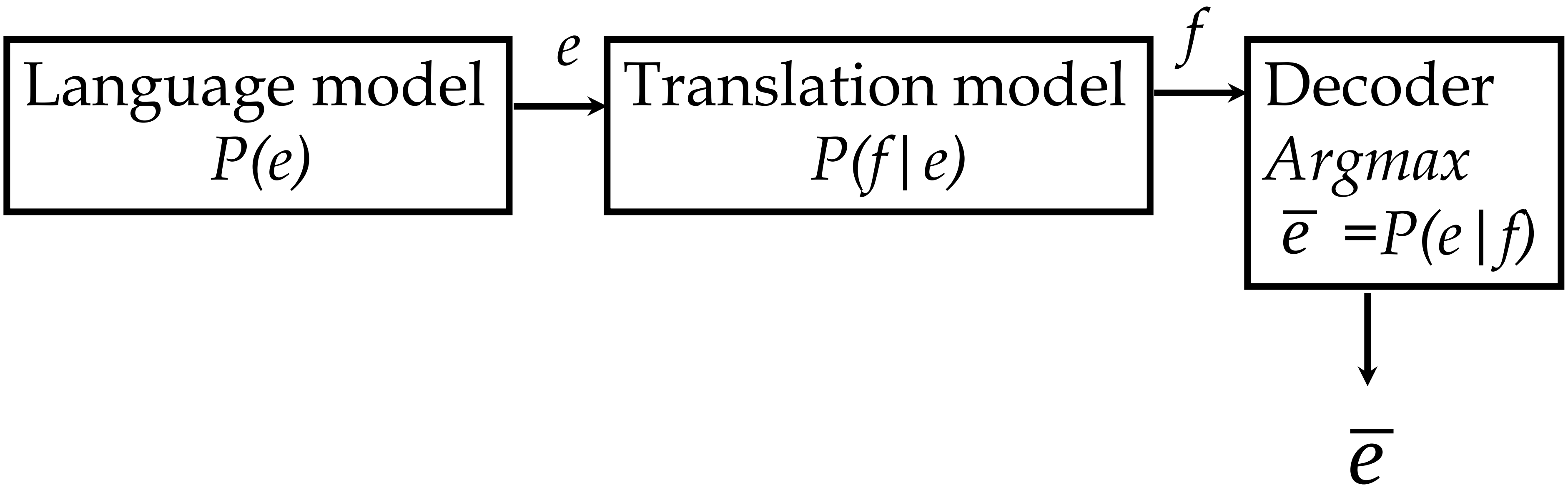




Word alignment learning with EM



Noisy channel again



Another Alignment System

- Available corpus assumed:
  - parallel text (translation E $\leftrightarrow$ F)
- Sentence alignment
  - sentence detection
  - sentence alignment
- Word alignment
  - tokenization
  - word alignment (with restrictions)

Sentence Boundary Detection

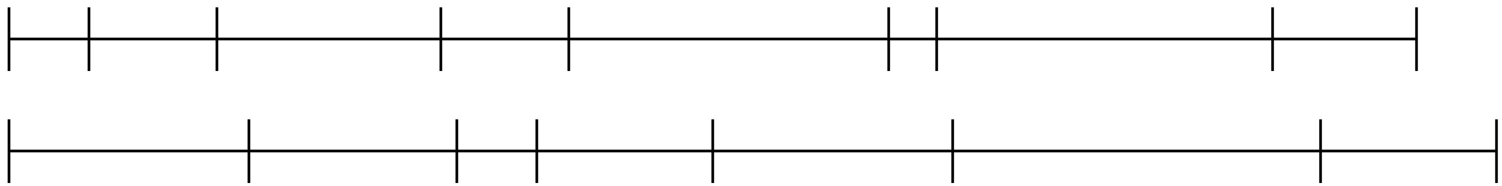
- Rules, lists:
  - Sentence breaks:
    - paragraphs (if marked)
    - certain characters: ?, !, ; (...almost sure)
    - Problem: period .
      - end of sentence (... left yesterday. He was heading to...)
      - decimal point: 3.6 (three-point-six)
      - thousand segment separator: 3.200
      - abbreviation: cf., e.g., Calif., Mt., Mr.
      - ellipsis: ...
      - other languages: ordinal number indication (2nd ~ 2.)
      - initials: A. B. Smith
- Statistical methods: e.g., Maximum Entropy

Sentence Alignment

- Problem: sentences detected only:

E:

F:



- **<u>Desired output</u>**: Segmentation with equal number of segments, spanning continuously the whole text.
- Original sentence boundaries kept:

E:

F:

- Alignments obtained: 2-1, 1-1, 1-1, 2-2, 2-1, 0-1

Alignment Methods

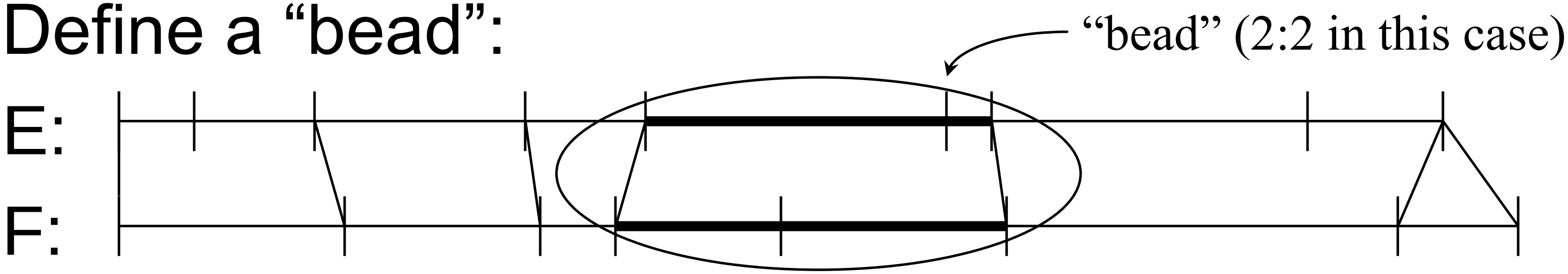
- Several methods (probabilistic and not prob.)
  - character-length based
  - word-length based
  - “cognates” (word identity used)
    - using an existing dictionary (F: prendre ~ E: make, take)
    - using word “distance” (similarity): names, numbers, borrowed words, Latin origin words, ...
- Best performing:
  - statistical, word- or character- length based (with some words perhaps)

Length-based Alignment

- First, define the problem probabilistically:

  $\arg\max_A P(A|E,F) = \arg\max_A P(A,E,F)$ (E,F fixed)

- Define a "bead":



- Approximate:

  $P(A,E,F) \cong \Pi_{i=1..n} P(B_i)$,

  where B_i is a bead; $P(B_i)$ does not depend on the rest of E,F.

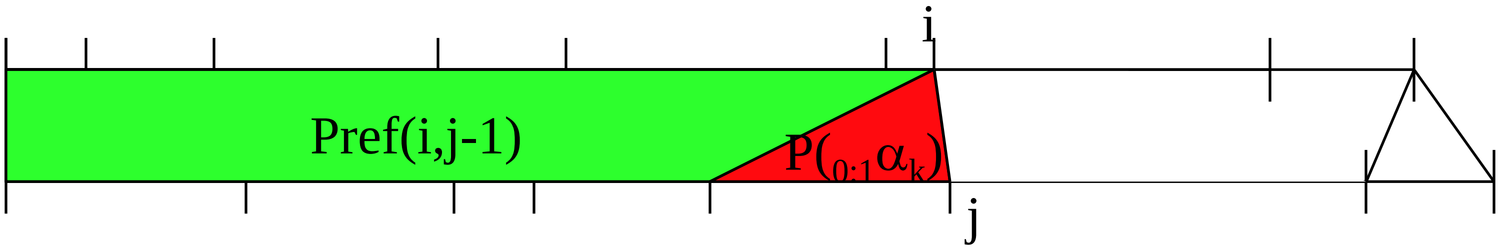
The Alignment Task

Some definitions:

- Given $P(A,E,F) \cong \Pi_{i=1..n} P(B_i)$,
  find the partitioning of (E,F) into n beads $B_{i=1..n}$, that maximizes P(A,E,F) over training data.
- $B_i = {}_{p:q}\alpha_i$, where $p:q \in \{0:1,1:0,1:1,1:2,2:1,2:2\}$ describes the type of alignment
- Pref(i,j) - probability of the best alignment from the start of (E,F) data (1,1) up to (i,j)

Recursive Definition

- Initialize: Pref(0,0) = 0.
- Pref(i,j) = max (
  Pref(i,j-1) $P(_{0:1}\alpha_k)$, Pref(i-1,j) $P(_{1:0}\alpha_k)$, Pref(i-1,j-1) $P(_{1:1}\alpha_k)$,
  Pref(i-1,j-2) $P(_{1:2}\alpha_k)$, Pref(i-2,j-1) $P(_{2:1}\alpha_k)$, Pref(i-2,j-2) $P(_{2:2}\alpha_k)$)
- This is enough for a Viterbi-like search.
- E:
- F:



Probability of a Bead

- Remains to define $P({}_{p:q}\alpha_k)$:
  - <u>k</u> refers to the "next" bead, with segments of <u>p</u> and <u>q</u> sentences, lengths $l_{k,e}$ and $l_{k,f}$.
- Use normal distribution for length variation:

  $P({}_{p:q}\alpha_k) = P(\delta(l_{k,e},l_{k,f},\mu,\sigma^2),p:q) \cong P(\delta(l_{k,e},l_{k,f},\mu,\sigma^2))P(p:q)$

  $\delta(l_{k,e},l_{k,f},\mu,\sigma^2) = (l_{k,f} - \mu l_{k,e})/\sqrt{l_{k,e}\sigma^2}$
- Estimate P(p:q) from small amount of data, or even guess and re-estimate after aligning some data.
- Words etc. might be used as better clues in $P({}_{p:q}a_k)$ def.

Word Alignment

- Length alone does not help:
  - words can be swapped, and mutual translations have often vastly different length.

- Idea:
  - Assume some (simple) translation model.
  - Find its parameters by considering virtually all alignments.
  - After we have the parameters, find the best alignment given those parameters.

Given the following bilingual dataset:
Mike yêu Jane. Mike loves Jane.
Jane yêu hoa. Jane loves flowers.
Jane thích đọc sách. Jane likes reading.
Compute the following translation probabilities:
P(“Mike likes reading| Mike thích đọc sách”)
P(“Mike likes flowers| Mike yêu hoa”)

- Start with sentence-aligned corpus.
- Let (E,F) be a pair of sentences (actually, a bead).

1. Initialize p(f|e) randomly, $f \in F$, $e \in E$.
2. Compute expected counts over the corpus:

$$c(f,e) = \Sigma_{(E,F);e \in E, f \in F}\ p(f|e)$$

$\forall$ aligned pair (E,F), find if e in E and f in F; if yes, add p(f|e).

3. Reestimate:

$$p(f|e) = c(f,e) / c(e) \qquad [c(e) = \Sigma_f\ c(f,e)]$$

4. Iterate until change of p(f|e) is small.

Best Alignment

Select, for each (E,F),

$A = \text{argmax}_A P(A|F,E) = \text{argmax}_A P(F,A|E)/P(F) =$

$\text{argmax}_A P(F,A|E) = \text{argmax}_A (\varepsilon / (l+1)^m \prod_{j=1..m} p(f_j|e_{a_j})) = \text{argmax}_A \prod_{j=1..m} p(f_j|e_{a_j})$

- Use dynamic programming, Viterbi-like algorithm.
- Recompute p(f|e)

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Exercise

- Given the following bilingual dataset:
  - Mike yêu Jane. Mike loves Jane.
  - Jane yêu hoa. Jane loves flowers.
  - Jane thích đọc sách. Jane likes reading.
- Compute the following translation probabilities:
  - P(“Mike thích đọc sách| Mike likes reading”)
  - P(“Mike yêu hoa| Mike likes flowers”)

Evaluate

Evaluation based on Hansard corpus:

- 48% of French sentences are translated correctly
- 2 types of errors:
  - Mistranslation:
    - Permettez que je donne un example à chambre
    - Let me give an example in the House (incorrect decoding)
    - (Let me give the House an example)
  - Grammatical translation:
    - Vous avez besoin de toute l'aide disponsible
    - You need all of the benefits available (ungrammatical decoding)
    - (You need all the help you can get)

Reason

- Distortion: English words at the beginning of a sentence are aligned with French words at the end of a sentence – this reduces the probability of alignment

- Fertility : the correspondence between English and French words (1-to-1, 1-to-2, 1-to-0, ...),
  - For example, fertility(farmers) in the corpus = 2, because this word when translated into English usually consists of 2 words: les argiculteurs
  - To go

Reason

- Independent Assumptions: Short sentences are preferred because there are fewer probabilities (when multiplying)

$\Rightarrow$ multiply the result by a constant proportional to the sentence length

- Training data dependence: a small change in the training data causes a large change in the parameter estimates

For example, *P(le|the)* changed from 0.610 to 0.497

- Efficiency. Remove sentences > 30 words, because the search space increases exponentially
- Lack of language knowledge

Lack of language knowledge

- Can't save information about terms: example can't be aligned “ to go ” and “ aller ”
- No local binding :

Eg, is she a mathematician

- Phonemic. Words made up of different phonemes are considered separate symbols
- Sparse data. Ratings for uncommon words are incorrect

Open sources

- GIZA++: statistical machine translation tool to train IBM 1-5 model for word alignment
- MOSES: statistical machine translation tool
- Moses has two types of translation: **phrase-based** and **tree-based**

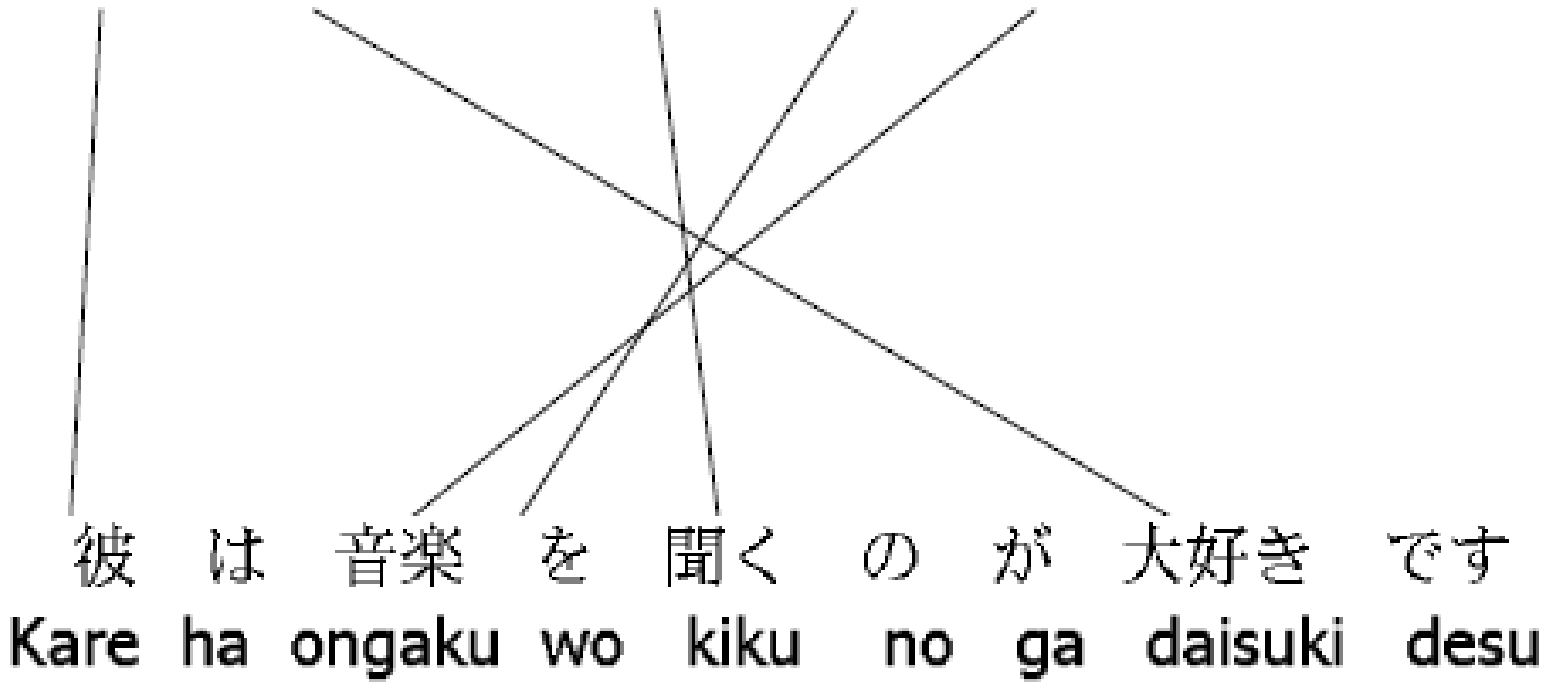
Machine translation using the syntax

Why Syntax?

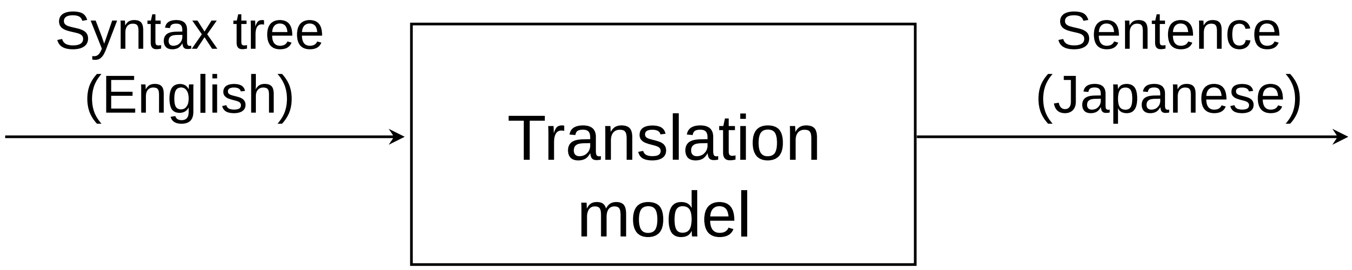
- Need much more grammatical output
- Need accurate control over re-ordering
- Need accurate insertion of function words
- Word translations need to depend on grammatically-related word

Yamada and Knight (2001): The need for phrasal syntax

- He adores listening to music.

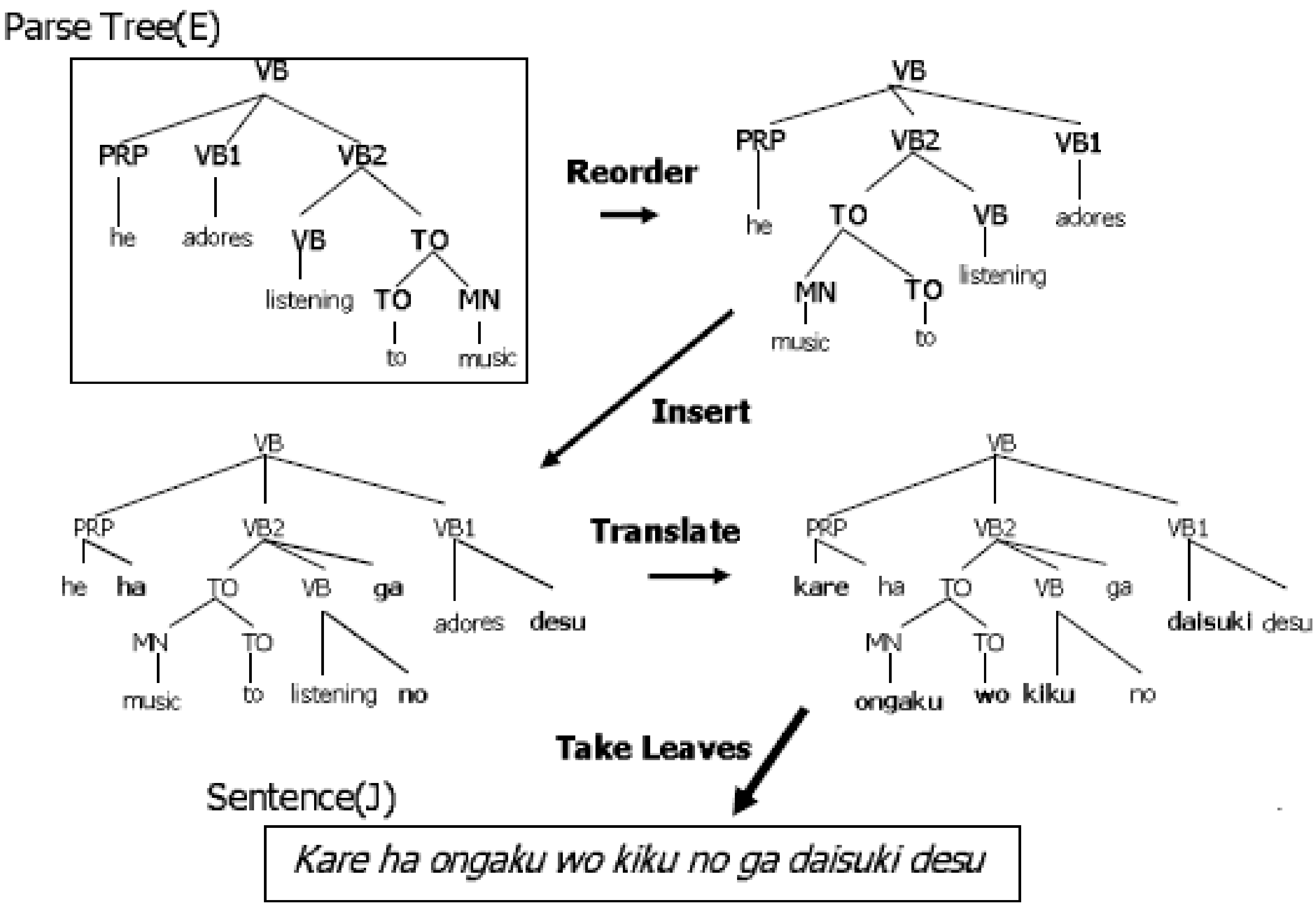


Syntax-based model

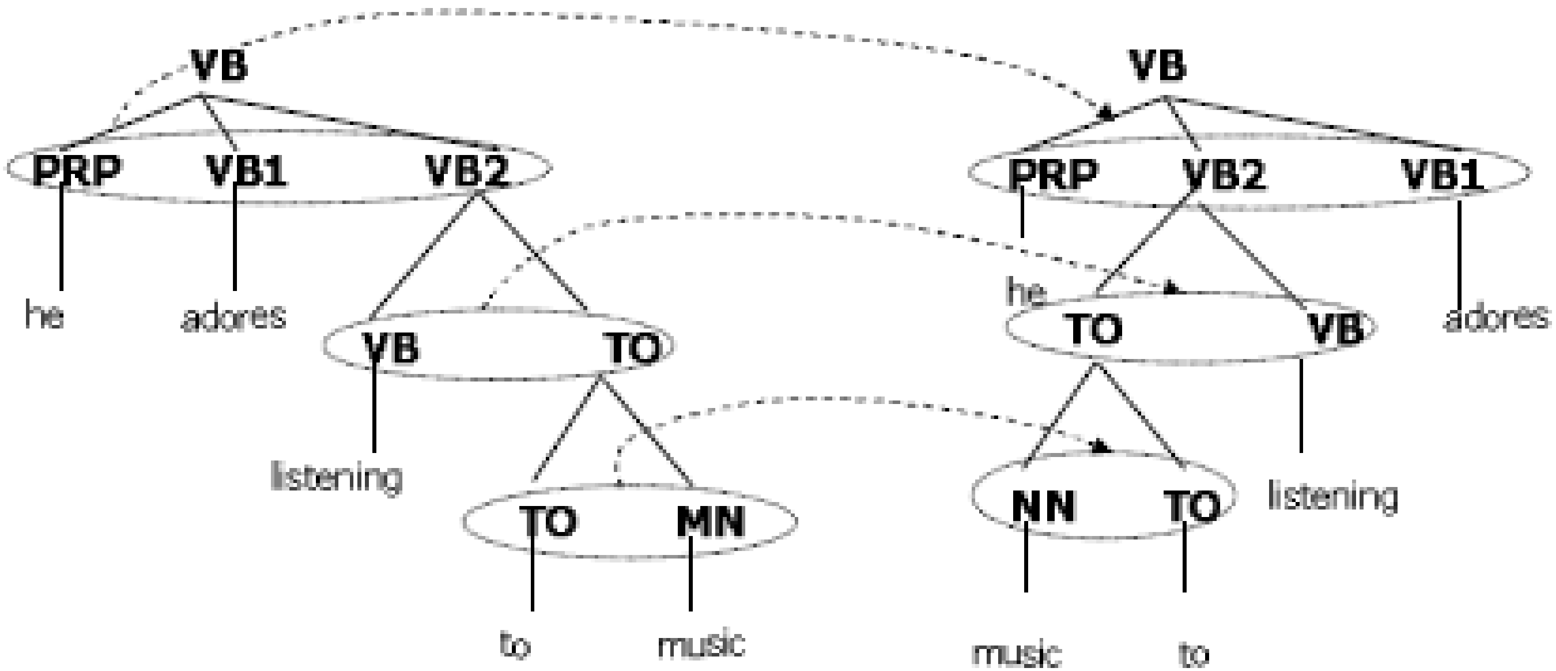


- Preprocess English by a parser
- Probabilistic operations on a parse-tree
  1. Reorder child nodes
  2. Insert extra nodes
  3. Translate leaf words

Parse Tree(E) → Sentence (J)



1. Reorder



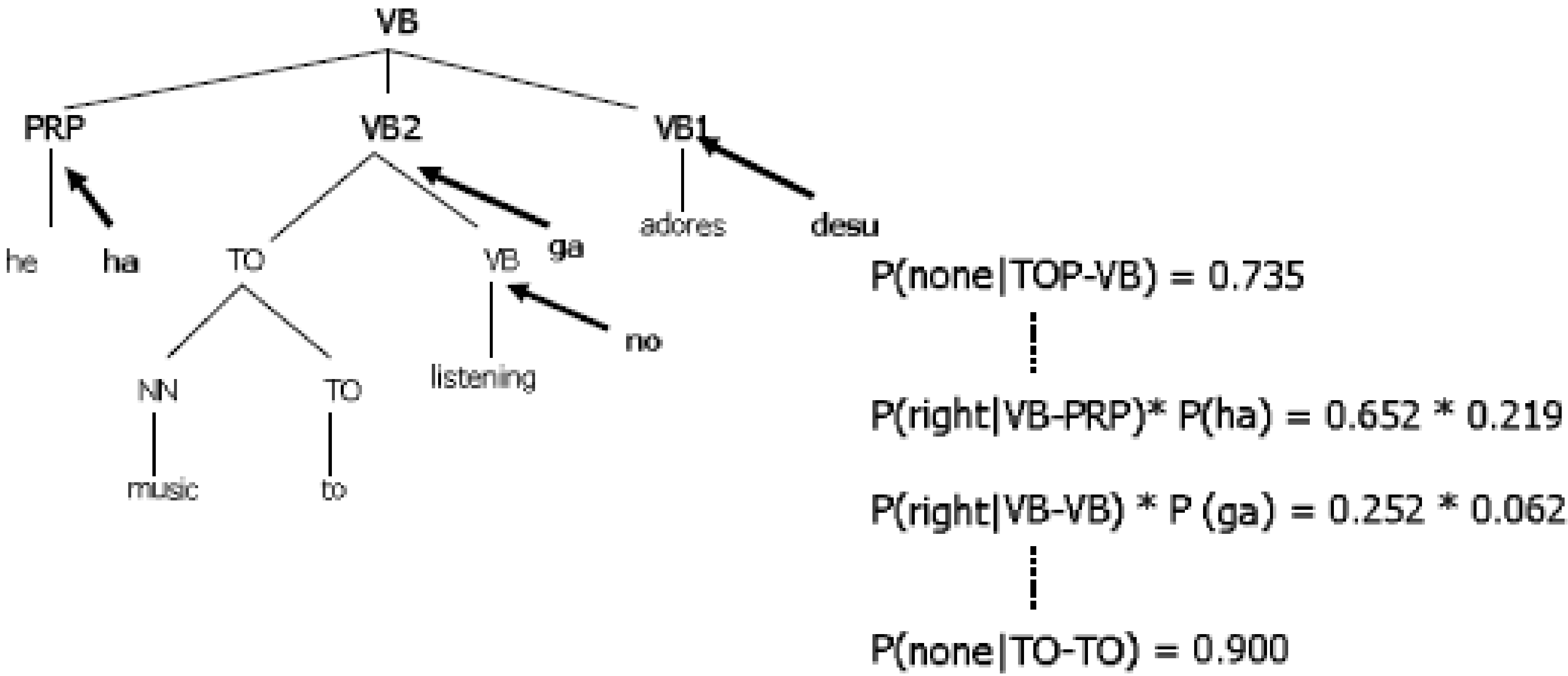
P(PRP VB1 VB2 → PRP VB2 VB1) = 0.723
P(VB TO → TO VB) = 0.749
P(TO NN → NN TO) = 0.893

Conditional feature = child label sequence

Parameter Table: Reorder

Original order	Coming back	P(Reorder|Original order)
PRP VB1 VB2	PRP VB1 VB2	0.074
	PRP VB2 VB1	**0.723**
	VB1 PRP VB2	0.061
	VB1 VB2 PRP	0.037
	VB2 PRP VB1	0.083
	VB2 VB1 PRP	0.021
VB TO	VB TO	0.107
	TO VB	**0.893**
TO NN	TO NN	0.251
	NN TO	**0.749**

2. Insert



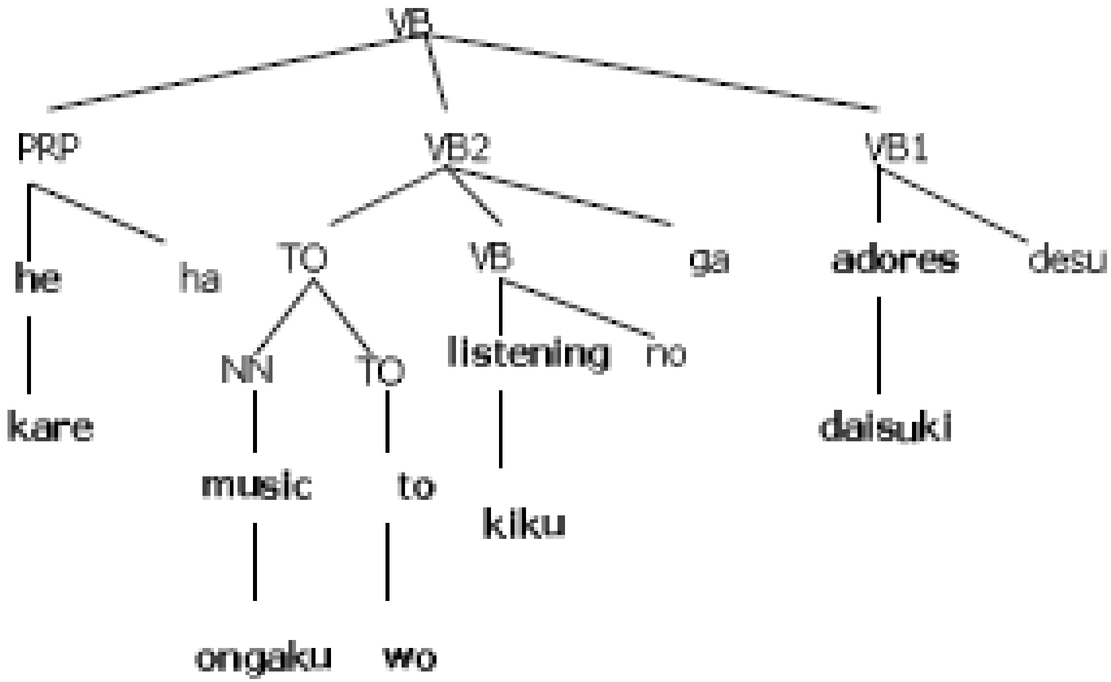
Conditional Feature = parent label & node label (position) & none (word selection)

Parameter table: insert

Parent label node level	TOP VB	VB VB	VB TO	TO TO	TO NN	TO NN
P (none)	0.735	0.687	0.344	0.700	0.900	0.800
P (left)	0.004	0.061	0.004	0.030	0.003	0.096
P (right)	0.260	0.252	0.652	0.261	0.097	0.104

W	P (insert-w)
ha	0.219
ta	0.131
wo	0.099
no	0.094
ni	0.080
te	0.078
ga	0.062
|	|
desu	0.0007
|	|

3. Translate



P (he —— kare) = 0.952
P (music —— ongaku) = 0.900
P (to —— wo) = 0.038
P (listening —— kiku) = 0.333
P (adore —— daisuki) = 1.000

Conditional feature = word identity (English)

Parameter table: Translate

E	adores	he	listening	music	to
J	daisuki 1.000	kare 0.952 NULL 0.016 nani 0.005 da 0.003 shi 0.003 | |	kiku 0.333 kii 0.333 mi 0.333	ongaku 0.900 naru 0.100	ni 0.216 NULL 0.204 to 0.133 no 0.046 wo 0.038 | |

Note: Translation to NULL = deletion

Experiment

- Training Corpus: J-E 2K sentence pairs
- J: Tokenized by Chasen [Matsumoto, et al., 1999]
- E: Parsed by Collins Parser [Collins, 1999]
  - Trained: 40K Treebank, Accuracy: ~90%
- E: Flatten parse tree
  - To Capture word-order difference (SVO->SOV)
- EM Training: 20 Iterations
  - 50 min/iter (Sparc 200Mhz 1-CPU) or
  - 30 sec/iter (Pentium3 700Mhz 30-CPU)

Result

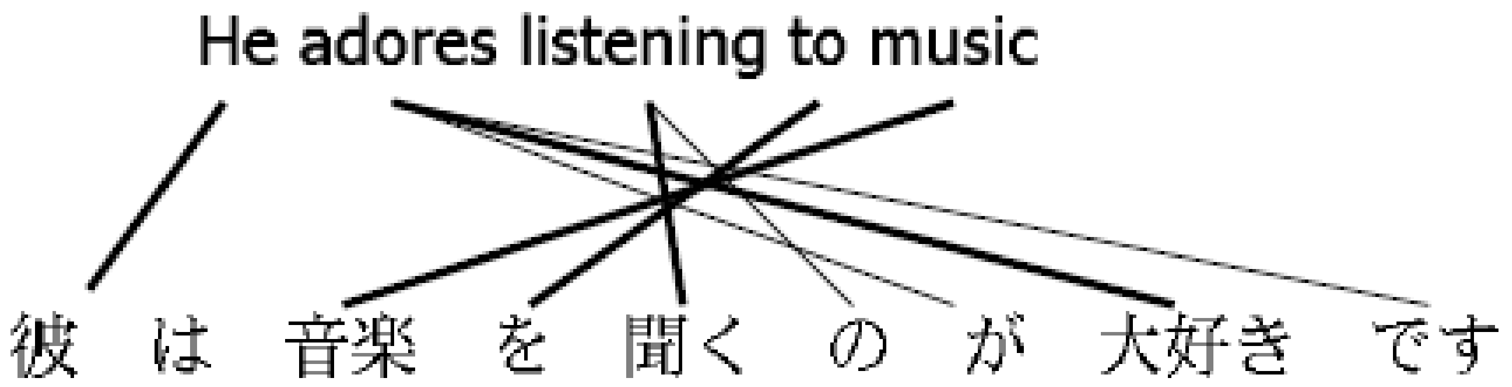
	Average	#perf sent
Y/K models	0.582	10
IBM model 5	0.431	0

- Average by 3 humans for 50 sentences
- ok(1.0), not sure (0.5), wrong (0.0)
- Precision only

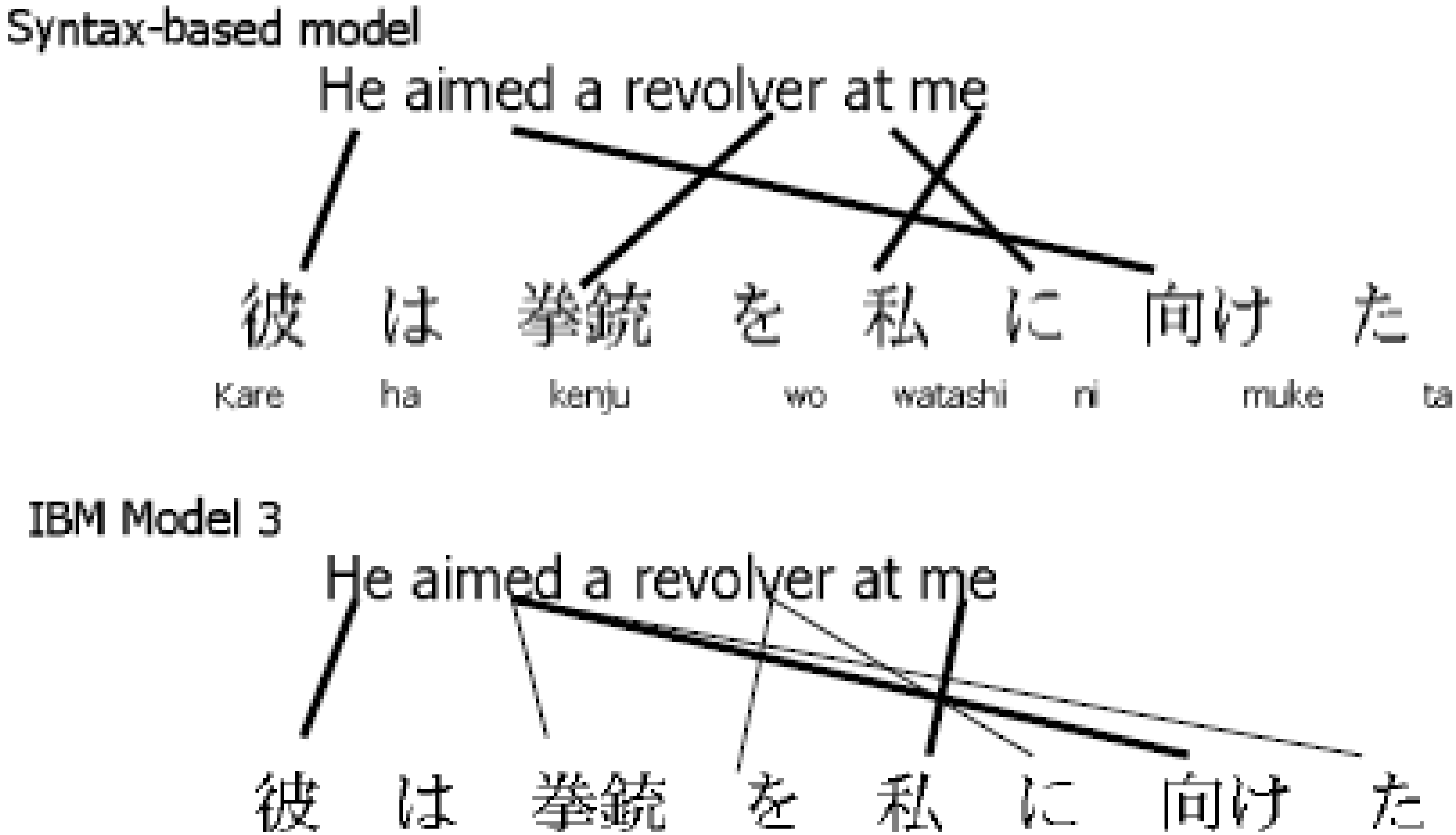
Result: Alignment 1

Syntax-based Model

IBM Model 3



Result: Alignment 2



Some open sources

- See http://fosmt.org/
  - Moses
  - Giza++

Some MT systems on the web

- http://www.google.com/language_tools?hl=en
- http://www.systransoft.com/index.html
- http://babelfish.altavista.digital.com/