MT Evaluation

Illustrative translation results

• *la politique de la haine .*	*(Foreign Original)*
• politics of hate .	(Reference Translation)
• the policy of the hatred .	(IBM4+N-grams+Stack)
• *nous avons signé le protocole .*	*(Foreign Original)*
• we did sign the memorandum of agreement .	(Reference Translation)
• we have signed the protocol .	(IBM4+N-grams+Stack)
• *où était le plan solide ?*	*(Foreign Original)*
• but where was the solid plan ?	(Reference Translation)
• where was the economic base ?	(IBM4+N-grams+Stack)

对外经济贸易合作部今天提供的数据表明，今年至十一月中国实际利用外资四百六十九点五九亿美元，其中包括外商直接投资四百点零七亿美元。

the Ministry of Foreign Trade and Economic Cooperation, including foreign
direct investment 40.007 billion US dollars today provide data include
that year to November china actually using foreign 46.959 billion US dollars and

MT Evaluation

- Manual (the best!?):
  - SSER (subjective sentence error rate)
  - Correct/Incorrect
  - **Adequacy and Fluency** (5 or 7 point scales)
  - Error categorization
  - **Comparative ranking of translations**
- Testing in an application that uses MT as one sub-component
  - Question answering from foreign language documents
- Automatic metric:
  - WER (word error rate) – why problematic?
  - **BLEU (Bilingual Evaluation Understudy)**

BLEU Evaluation Metric

(Papineni et al, ACL-2002)

Reference (human) translation:
The U.S. island of Guam is maintaining a high state of alert after the Guam airport and its offices both received an e-mail from someone calling himself the Saudi Arabian Osama bin Laden and threatening a biological/ chemical attack against public places such as the airport .

Machine translation:
The American [?] international airport and its the office all receives one calls self the sand Arab rich business [?] and so on electronic mail , which sends out ; The threat will be able after public place and so on the airport to start the biochemistry attack , [?] highly alerts after the maintenance.

- N-gram precision (score is between 0 & 1)
  - What percentage of machine n-grams can be found in the reference translation?
    - An n-gram is an sequence of n words
  - Not allowed to match same portion of reference translation twice at a certain n-gram level (two MT words *airport* are only correct if two reference words *airport;* can't cheat by typing out "the the the the the")
  - Do count unigrams also in a bigram for unigram precision, etc.
- Brevity Penalty
  - Can't just type out single word "the" (precision 1.0!)
- It was thought quite hard to "game" the system (i.e., to find a way to change machine output so that BLEU goes up, but quality doesn't)

BLEU Evaluation Metric

(Papineni et al, ACL-2002)

Reference (human) translation:
The U.S. island of Guam is maintaining a high state of alert after the Guam airport and its offices both received an e-mail from someone calling himself the Saudi Arabian Osama bin Laden and threatening a biological/ chemical attack against public places such as the airport .

Machine translation:
The American [?] international airport and its the office all receives one calls self the sand Arab rich business [?] and so on electronic mail , which sends out ; The threat will be able after public place and so on the airport to start the biochemistry attack , [?] highly alerts after the maintenance.

- BLEU is a weighted geometric mean, with a brevity penalty factor added.
  - Note that it's precision-oriented
- BLEU4 formula
  (counts n-grams up to length 4)

exp (1.0 * log p1 +
0.5 * log p2 +
0.25 * log p3 +
0.125 * log p4 –
max(words-in-reference / words-in-machine – 1, 0)

p1 = 1-gram precision
P2 = 2-gram precision
P3 = 3-gram precision
P4 = 4-gram precision

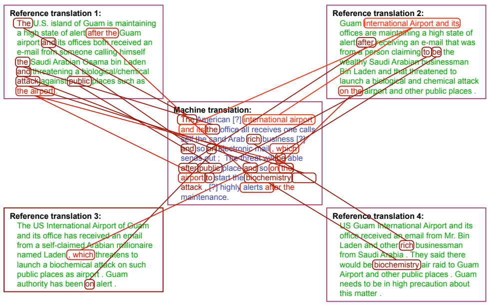
Note: only works at corpus level (zeroes kill it);
there's a smoothed variant for sentence-level

BLEU in Action

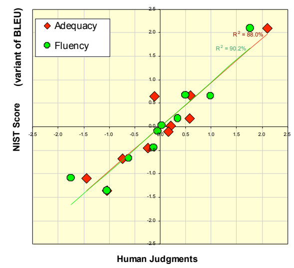
枪手被警方击毙。	**(Foreign Original)**
the gunman was shot to death by the police .	**(Reference Translation)**
the gunman was police kill .	**#1**
wounded police jaya of	**#2**
the gunman was shot dead by the police .	**#3**
the gunman arrested by police kill .	**#4**
the gunmen were killed .	**#5**
the gunman was shot to death by the police .	**#6**
gunmen were killed by police ?SUB>0 ?SUB>0	**#7**
al by the police .	**#8**
the ringer is killed by the police .	**#9**
police killed the gunman .	**#10**

green = 4-gram match (good!)
red = word not matched (bad!)

Multiple Reference Translations



Initial results showed that BLEU predicts human judgments well



slide from G. Doddington (NIST)

Quiz question!

MT Hypothesis: ***the gunman was shot dead by police .***

- Ref 1: The gunman was shot to death by the police .
- Ref 2: The cops shot the gunman dead .

- What is the:
  - Unigram precision?
  - Trigram precision?

Note: punctuation tokens *are* counted in calculation

Automatic evaluation of MT

- People started optimizing their systems to maximize BLEU score
  - BLEU scores improved rapidly
  - The correlation between BLEU and human judgments of quality went way, way down
  - StatMT BLEU scores now approach those of human translations but their true quality remains far below human translations
- Coming up with automatic MT evaluations has become its own research field
  - There are many proposals: TER, METEOR, MaxSim, SEPIA, our own RTE-MT
  - METEOR is a representative good one that handles some word choice variation.
- MT research really requires *some* automatic metric to allow a rapid development and evaluation cycle.

MT: The early history (1950s)

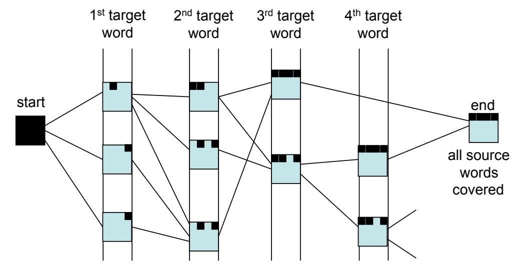
- Earl
  less
- Fou
  lang
- First
- MT
  word
- Little
  sem
- Prob

A complete translation system

Decoding for IBM Models

- Of all conceivable English word strings, find the one maximizing $P(e) \times P(f \mid e)$
- Decoding is NP hard
  - (Knight, 1999)
- Several search strategies are available
  - Usually a beam search where we keep multiple stacks for candidates covering the same number of source words
- Each potential English output is called a *hypothesis*.

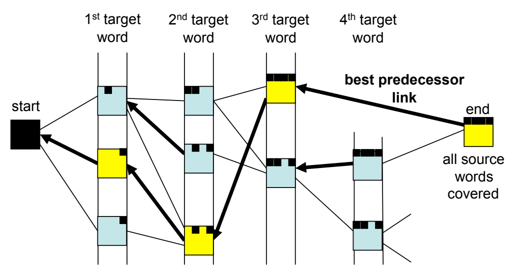
Dynamic Programming Beam Search



Each partial translation hypothesis contains:
- Last English word chosen + source words covered by it
- Next-to-last English word chosen
- Entire coverage vector (so far) of source sentence
- Language model and translation model scores (so far)

[Jelinek, 1969; Brown et al, 1996 US Patent; (Och, Ueffing, and Ney, 2001]

Dynamic Programming Beam Search



Each partial translation hypothesis contains:
- Last English word chosen + source words covered by it
- Next-to-last English word chosen
- Entire coverage vector (so far) of source sentence
- Language model and translation model scores (so far)

[Jelinek, 1969; Brown et al, 1996 US Patent; (Och, Ueffing, and Ney, 2001]

The “Fundamental Equation of Machine Translation” (Brown et al. 1993)

$$\hat{e} = \underset{e}{\operatorname{argmax}}\ P(e \mid f)$$

$$= \underset{e}{\operatorname{argmax}}\ P(e) \times P(f \mid e) / P(f)$$

$$= \underset{e}{\operatorname{argmax}}\ P(e) \times P(f \mid e)$$

What StatMT people do in the privacy of their own homes

$$\underset{e}{\operatorname{argmax}}\ P(e \mid f) =$$

$$\underset{e}{\operatorname{argmax}}\ P(e) \times P(f \mid e) / P(f) =$$

$$\underset{e}{\operatorname{argmax}}\ P(e)^{1.9} \times P(f \mid e)$$ … works better!

Which model are you now paying more attention to?

What StatMT people do in the privacy of their own homes

$$\underset{e}{\operatorname{argmax}}\ P(e \mid f) =$$

$$\underset{e}{\operatorname{argmax}}\ P(e) \times P(f \mid e) / P(f)$$

$$\underset{e}{\operatorname{argmax}}\ P(e)^{1.9} \times P(f \mid e) \times 1.1^{\text{length}(e)}$$

Rewards longer hypotheses, since these are ‘unfairly’ punished by P(e)

What StatMT people do in the privacy of their own homes

$$\operatorname*{argmax}_{e} \; \underbrace{P(e)^{1.9} \times P(f \mid e) \times 1.1^{\text{length}(e)} \times KS^{3.7} \ldots}$$

Lots of knowledge sources vote on any given hypothesis.

"Knowledge source" = "feature function" = "score component".

Feature function simply scores a hypothesis with a real value.

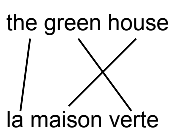
(May be binary, as in "e has a verb").

Problem: How to set the weights?
(We look at one way later: maxent models.)

Flaws of Word-Based MT

- Multiple English words for one French word
  - IBM models can do one-to-many (fertility) but not many-to-one
- Phrasal Translation
  - "real estate", "note that", "interested in"
- Syntactic Transformations
  - Verb at the beginning in Arabic
  - Translation model penalizes any proposed re-ordering
  - Language model not strong enough to force the verb to move to the right place

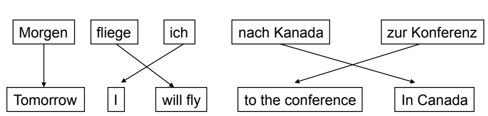
Alignments: linguistics



- There isn't enough linguistics to explain this in the translation model … have to depend on the language model … that may be unrealistic … and may be harming our translation model

Phrase-Based Statistical MT

Phrase-Based Statistical MT



- Foreign input segmented into phrases
  - "phrase" is any sequence of words
- Each phrase is probabilistically translated into English
  - P(to the conference | zur Konferenz)
  - P(into the meeting | zur Konferenz)
- Phrases are probabilistically re-ordered

See J&M or Lopez 2008 for an intro.

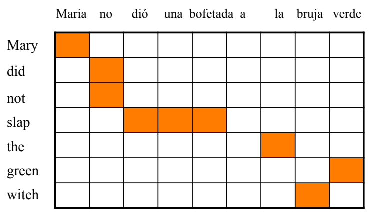
This is still pretty much the state-of-the-art!

Advantages of Phrase-Based

- Many-to-many mappings can handle non-compositional phrases
- Local context is very useful for disambiguating
  - "interest rate" → …
  - "interest in" → …
- The more data, the longer the learned phrases
  - Sometimes whole sentences

How to Learn the Phrase Translation Table?

- One method: "alignment templates" (Och et al, 1999)
- Start with word alignment, build phrases from that.

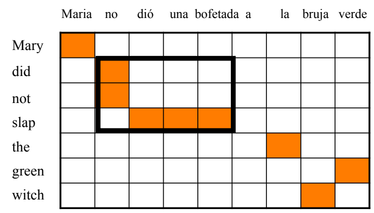


This word-to-word alignment is a by-product of training a translation model like IBM-Model-3.

This is the best (or "Viterbi") alignment.

How to Learn the Phrase Translation Table?

- One method: "alignment templates" (Och et al, 1999)
- Start with word alignment, build phrases from that.

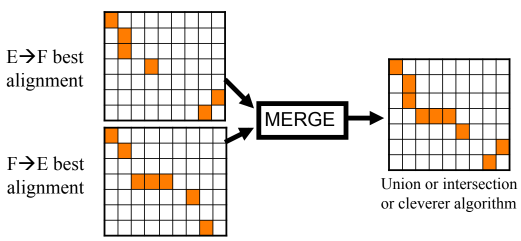


This word-to-word alignment is a by-product of training a translation model like IBM-Model-3.

This is the best (or "Viterbi") alignment.

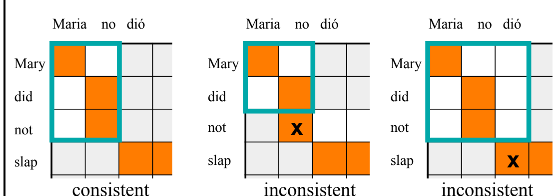
IBM Models are 1-to-Many

- Run IBM-style aligner both directions, then merge:



How to Learn the Phrase Translation Table?

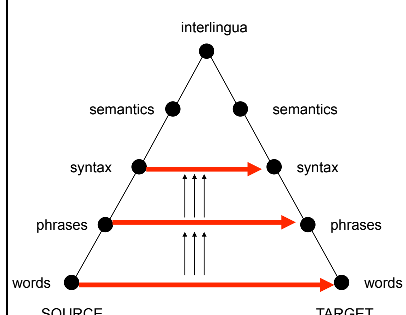
- Collect all phrase pairs *that are consistent with the word alignment*



- Phrase alignment must contain all alignment points for all the words in both phrases!
- These phrase alignments are sometimes called *beads*

Syntax and Semantics in Statistical MT

MT Pyramid

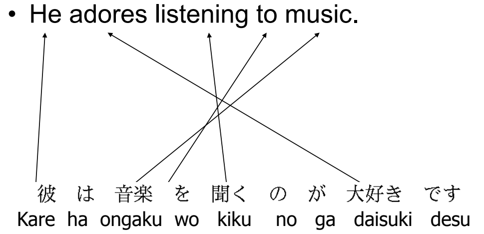


Why Syntax?

- 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 words

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

- He adores listening to music.



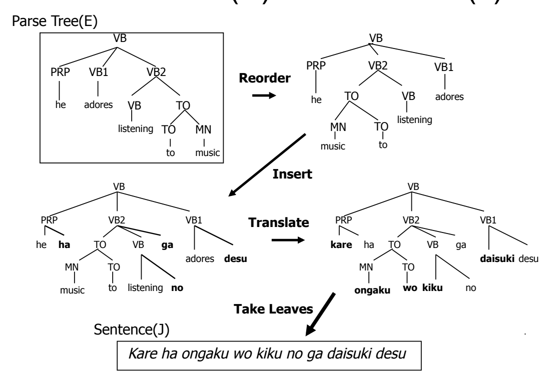
Syntax-based Model

- E→J Translation (Channel) Model

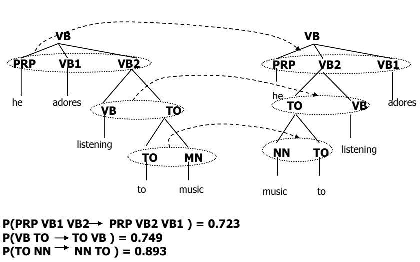
Parse Tree (English) ⇒ Translation model ⇒ Sentence (Japanese)

- 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



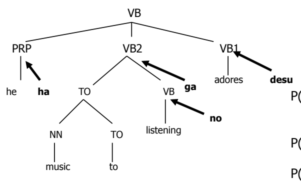
P(PRP VB1 VB2→ PRP VB2 VB1) = 0.723
P(VB TO → TO VB) = 0.749
P(TO NN → NN TO) = 0.893

Conditioning Feature = Child label Sequence

Parameter Table: Reorder

Original Order	Reordering	P(reorder|original)
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



P(none|TOP-VB) = 0.735

P(right|VB-PRP)* P(ha) = 0.652 * 0.219

P(right|VB-VB) * P (ga) = 0.252 * 0.062

P(none|TO-TO) = 0.900

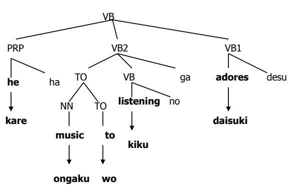
Conditioning Feature = Parent Label & Node Label (position)
none (word selection)

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

Conditioning Feature = word (E) identity

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

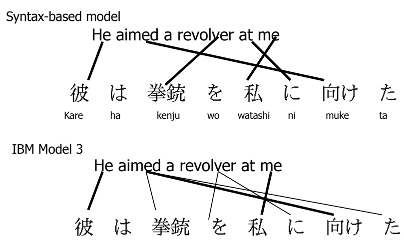
- Training Corpus: J-E 2K sentence pairs
- J: Tokenized by Chasen [Matsumoto, et al., 1999]
- E: Parsed by Collins Parser [Collins, 1999]
  --- Trained: 40K Treebank, Accuracy: ~90%
- E: Flatten parse tree
  --- To Capture word-order difference (SVO->SOV)
- EM Training: 20 Iterations
  --- 50 min/iter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

Result: Alignments

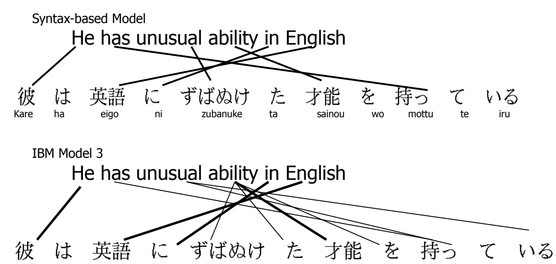
	Ave. Score	# perf sent
Y/K Model	0.582	10
IBM Model 5	0.431	0

- Ave. by 3 humans for 50 sents
- okay(1.0), not sure(0.5), wrong(0.0)
- precision only

Result: Alignment 2



Result: Alignment 3



Machine Translation Summary

- Usable Technologies
  - "Translation memories" to aid translator
  - Low quality screening/web translators
- Technologies
  - Traditional: Systran (Altavista Babelfish, what you got till mid-2006 on Google) is now seen as a limited success
  - Statistical MT over huge training sets is successful (ISI/LanguageWeaver, Microsoft, Google)
- Key ideas of the present/future
  - Statistical phrase based models
  - Syntax based models
  - Better language models (e.g., bigger, using grammar)
  - Better decoding models (e.g., by restricting model?)