Machine Translation Systems

Gerald Penn
CS224N / Ling 284
[Based on slides by Kevin Knight, Dan Klein, Dan Jurafsky and Chris Manning]

A complete translation system

Decoding for IBM Models

- Of all conceivable English word strings, find the one maximizing P(e) x P(f | 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*.

Search for Best Translation

voulez – vous vous taire !

Search for Best Translation

voulez – vous vous taire !

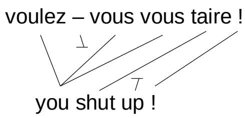
you – you you quiet !

Search for Best Translation

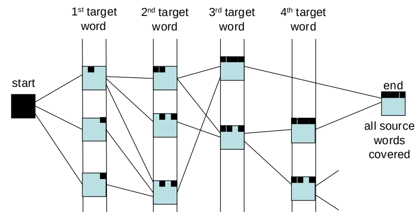
voulez – vous vous taire !

quiet you – you you !

Search for Best Translation



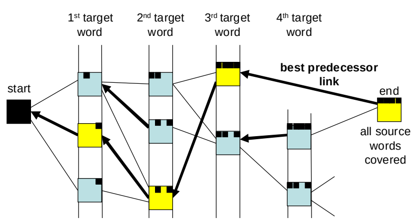
Dynamic Programming Beam Search



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)

$$\hat{e} = \operatorname*{argmax}_e \; P(e \mid f)$$

$$= \operatorname*{argmax}_e \; P(e) \times P(f \mid e) \,/\, P(f)$$

$$= \operatorname*{argmax}_e \; P(e) \times P(f \mid e)$$

What StatMT people do in the privacy of their own homes

$$\operatorname*{argmax}_e \; P(e \mid f) =$$

$$\operatorname*{argmax}_e \; P(e) \times P(f \mid e) \,/\, P(f) \neq$$

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

Which model are you now paying more attention to?

What StatMT people do in the privacy of their own homes

$$\operatorname*{argmax}_e \; P(e \mid f) =$$

$$\operatorname*{argmax}_e \; P(e) \times P(f \mid e) \,/\, P(f)$$

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

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

What StatMT people do in the privacy of their own homes

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

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").

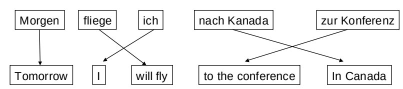
Problem: How to set the weights?
(We look at one way later: maxent models.)

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

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.

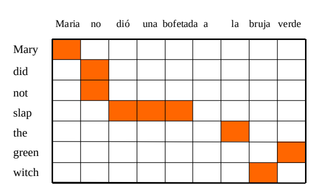
This is still pretty much the state-of-the-art!

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?

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

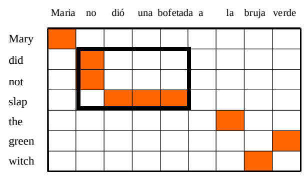


This word-to-word alignment is a by-product of training a translation model like IBM-Model-3.

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

How to Learn the Phrase Translation Table?

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

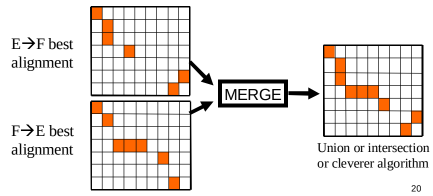


This word-to-word alignment is a by-product of training a translation model like IBM-Model-3.

This is the best (or “Viterbi”) alignment.

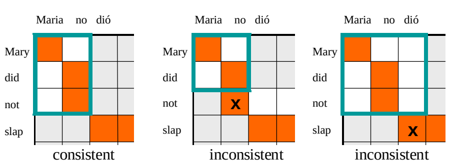
IBM Models are 1-to-Many

- 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*

Phrase Pair Probabilities

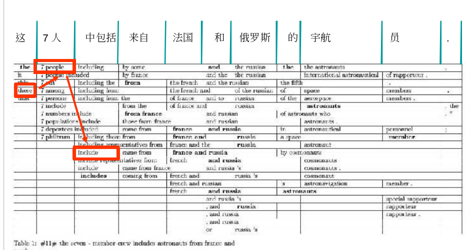
- A certain phrase pair (f-f-f, e-e-e) may appear many times across the bilingual corpus.
- No EM training
- Just relative frequency:

$$P(\text{f-f-f} \mid \text{e-e-e}) = \frac{\text{count(f-f-f, e-e-e)}}{\text{count(e-e-e)}}$$

Phrase-Based Translation

Scoring: Try to use phrase pairs that have been frequently observed.
Try to output a sentence with frequent English word sequences.

Phrase-Based Translation

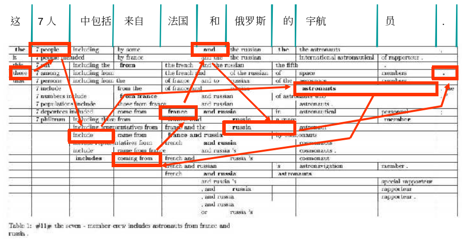


Scoring: Try to use phrase pairs that have been frequently observed.
Try to output a sentence with frequent English word sequences.

Phrase-Based Translation

Scoring: Try to use phrase pairs that have been frequently observed.
Try to output a sentence with frequent English word sequences.

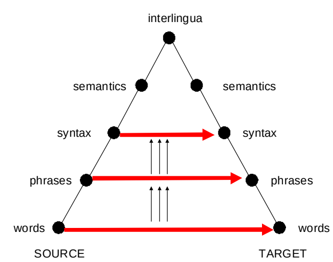
Phrase-Based Translation



Scoring: Try to use phrase pairs that have been frequently observed.
Try to output a sentence with frequent English word sequences.

Syntax and Semantics in Statistical MT

Vauquois Triangle

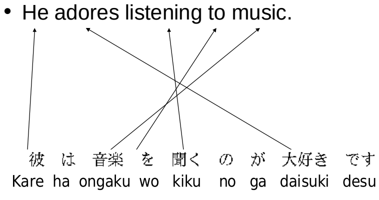


Why Syntax?

- 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.



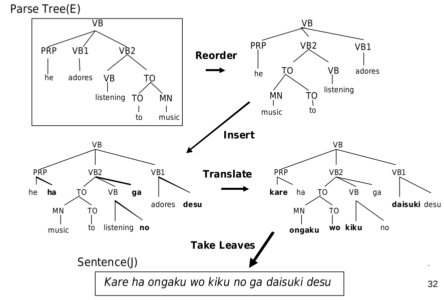
Syntax-based Model

- 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)



Experiment

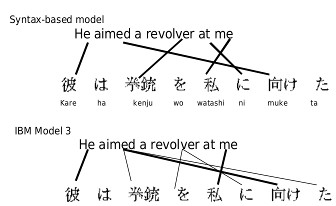
- Training Corpus: J-E 2K sentence pairs
- J: Tokenized by Chasen [Matsumoto, et al., 1999]
- E: Parsed by Collins Parser [Collins, 1999]
  --- Trained: 40K Treebank, Accuracy: ~90%
- E: Flatten parse tree
  --- To Capture word-order difference (SVO->SOV)
- EM Training: 20 Iterations
  --- 50 min/iter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

Result: Alignments

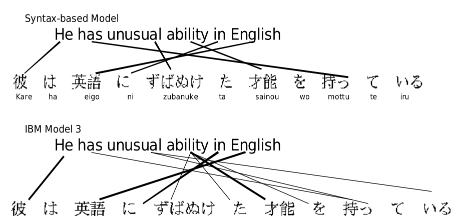
	Ave. Score	# perf sent
Y/K Model	0.582	10
IBM Model 5	0.431	0

- 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



MT Applications

Gerald Penn

CS 224N

2011

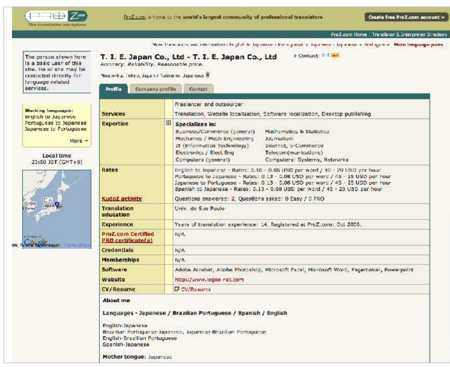
[Based on slides by Chris Manning]

MT: The early history (1950s)

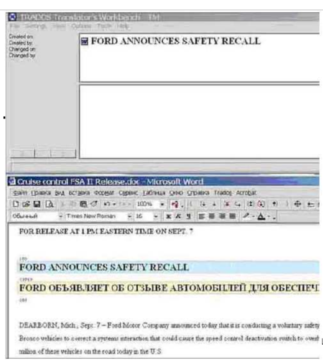
- Earl
  less
- Fou
  lang
- First
- MT
  word
- Little
  sem
- Prob

MT Applications: 1. Traditional

- Traditional scenario:
  - Documents had to be translated for your company/organization. Document production for organization
  - Generally, the quality/accuracy demands are high
  - High cost
    - Though most of it is now done as outsourced piecework
- MT tends to be ineffective: The cost of post-translation error correction is too high
- Main technology in the game: translation memory/translation workbench/terminology management
  - E.g., TRADOS.
    - Very slowly, MT technology is starting to be incorporated, but most of the action is in terminology lexicon management



Bad TRADOS Screenshot…



Trados is relatively pricey (high hundreds for PC versions, thousands for server version); seen as necessary productivity tool (Photoshop for translators)

MT Applications: 2. Web

- Web applications:
  - Dominant scenario: User-initiated translation
    - Crucial difference: The quality doesn't have to be great. The user is usually okay with just understanding the gist of what is going on
  - Second scenario
    - Somehow on the web people will accept medium quality results. Accessible information is better than no information
- MT is saved!!! "It's the web, stupid."
  - (But is there money in it?)

AltaVista BabelFish

1997:

Free, automatic translation for the masses.

Revolutionary.

But, what was the underlying technology?

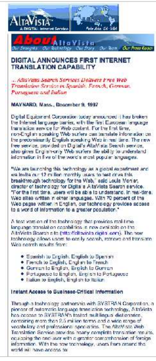
SYSTRAN.

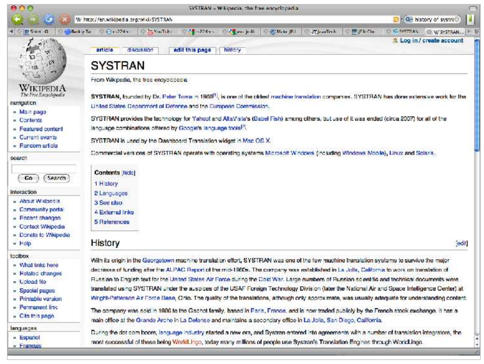
MacOS Dashboard?

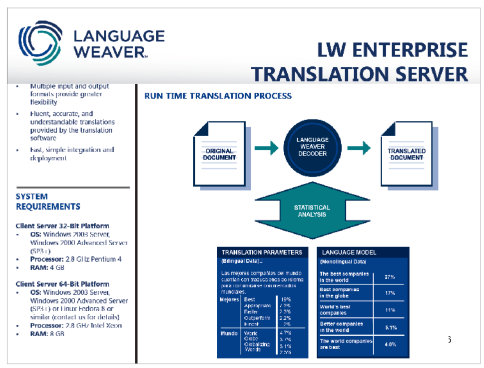
SYSTRAN

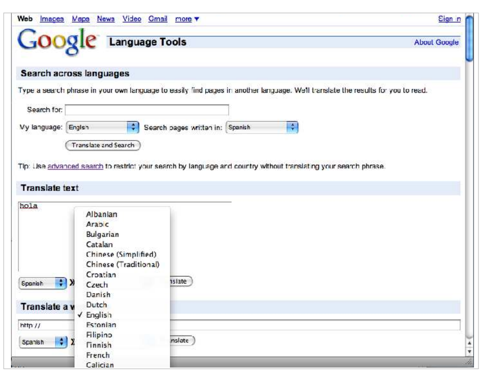
Google until 2006?

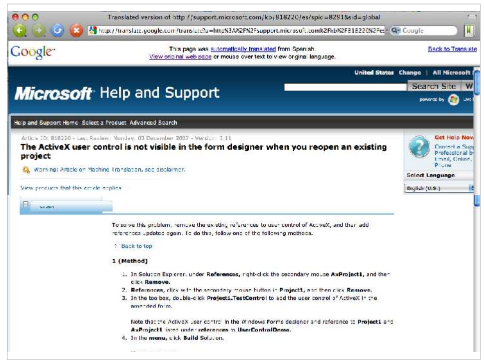
SYSTRAN

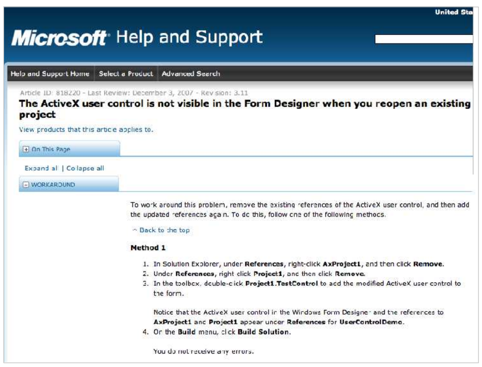












Machine Translation Summary

- 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?)