Statistical Machine Translation
The Main Idea

• Treat translation as a noisy channel problem:
  
  Input (Source) \[\rightarrow\] \[\text{The channel} \] \[\rightarrow\] “Noisy” Output (target)

  E: English words... \(\rightarrow\) (adds “noise”) \(\rightarrow\) F: Les mots Anglais...

• The Model: \(P(E|F) = P(F|E) P(E) / P(F)\)

• Interested in rediscovering \(E\) given \(F\):
  
  After the usual simplification (\(P(F)\) fixed):

  \[\arg\max_E P(E|F) = \arg\max_E P(F|E) P(E) \]
The Necessities

- **Language Model (LM)**
  
  \[ P(E) \]
  
- **Translation Model (TM):** Target given source
  
  \[ P(F|E) \]
  
- **Search procedure**
  
  - Given E, find best F 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!
The Language Model

• Any LM will do:
  – 3-gram LM
  – 3-gram class-based LM (cf. HW #2!)
  – decision tree LM with hierarchical classes

• Does not necessarily operates on word forms:
  – cf. later the “analysis” and “generation” procedures
  – for simplicity, imagine now it *does* operate on word forms
The Translation Models

• Do not care about correct strings of English words (that’s the task of the LM)

• Therefore, we can make more independence assumptions:
  – for start, use the “tagging” approach:
    • 1 English word (“tag”) ~ 1 French word (“word”)
  – not realistic: rarely even the number of words is the same in both sentences (let alone there is 1:1 correspondence!)

• ⇒ use “Alignment”.
The Alignment

• $e_0$ And the program has been implemented

• $f_0$ Le programme a été mis en application

• 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)
Alignment Mapping

• In general:
  - $|F| = m$, $|E| = l$ (length of sent.):
    - $lm$ connections (each French word to any English word),
    - $2^{lm}$ different alignments for any pair (E,F) (any subset)

• In practice:
  - From English to French
    - each English word 1-n connections (n - empirical max.)
    - each French word exactly 1 connection
  - therefore, “only” $(1+1)^m$ alignments ($<< 2^{lm}$)
    - $a_j = i$ (link from j-th French word goes to i-th English word)
Elements of Translation Model(s)

• Basic distribution:

• $P(F,A,E)$ - the joint distribution of the English sentence, the Alignment, and the French sentence (length $m$)

• Interested also in marginal distributions:

$$P(F,E) = \sum_A P(F,A,E)$$

$$P(F|E) = \frac{P(F,E)}{P(E)} = \frac{\sum_A P(F,A,E)}{\sum_{A,F} P(F,A,E)} = \sum_A P(F,A|E)$$

• Useful decomposition [one of possible decompositions]:

$$P(F,A|E) = P(m | E) \prod_{j=1..m} P(a_j|a_1^{j-1},f_1^{j-1},m,E) P(f_j|a_1^j,f_1^{j-1},m,E)$$
Decomposition

- Decomposition formula again:

\[ P(F, A|E) = P(m \mid E) \prod_{j=1..m} P(a_j \mid a_1^{j-1}, f_1^{j-1}, m, E) \cdot P(f_j \mid a_1^j, f_1^{j-1}, m, E) \]

- \( m \) - length of French sentence
- \( a_j \) - the alignment (single connection) going from \( j \)-th French word
- \( f_j \) - the \( j \)-th French word from \( F \)
- \( a_1^{j-1} \) - sequence of alignments \( a_i \) up to the word preceding \( f_j \)
- \( a_1^j \) - sequence of alignments \( a_i \) up to and including the word \( f_j \)
- \( f_1^{j-1} \) - sequence of French words up to the word preceding \( f_j \)
Decomposition and the Generative Model

• ...and again:

\[ P(F,A|E) = P(m \mid E) \prod_{j=1..m} P(a_j|a_{1\ldots j-1},f_{1\ldots j-1},m,E) \, P(f_j|a_{1\ldots j},f_{1\ldots j-1},m,E) \]

• Generate:
  – first, the length of the French given the English words E;
  – then, the link from the first position in F (not knowing the actual word yet) ⇒ now we know the English word
  – then, given the link (and thus the English word), generate the French word at the current position
  – then, move to the next position in F until m position filled.
Approximations

• Still too many parameters
  – similar situation as in n-gram model with “unlimited” n
  – impossible to estimate reliably.
• Use 5 models, from the simplest to the most complex
  (i.e. from heavy independence assumptions to light)
• Parameter estimation:
  Estimate parameters of Model 1; use as an initial
  estimate for estimating Model 2 parameters; etc.
Model 1

• **Approximations:**
  - French length $P(m \mid E)$ is constant (small $\varepsilon$)
  - Alignment link distribution $P(a_j \mid a_{1:j-1}, f_{1:j-1}, m, E)$ depends on English length $l$ only ($= 1/(l+1)$)
  - French word distribution depends only on the English and French word connected with link $a_j$.

• $\Rightarrow$ Model 1 distribution:

$$P(F, A \mid E) = \varepsilon / (l+1)^m \prod_{j=1..m} p(f_j \mid e_{a_j})$$
Models 2-5

• Model 2
  – adds more detail into $P(a_j|...)$: more “vertical” links preferred

• Model 3
  – adds “fertility” (number of links for a given English word is explicitly modeled: $P(n|e_i)$
  – “distortion” replaces alignment probabilities from Model 2

• Model 4
  – the notion of “distortion” extended to chunks of words

• Model 5 is Model 4, but not deficient (does not waste probability to non-strings)
The Search Procedure

- “Decoder”:
  - given “output” (French), discover “input” (English)
- Translation model goes in the opposite direction:
  \[ p(f|e) = \ldots \]
- Naive methods do not work.
- Possible solution (roughly):
  - generate English words one-by-one, keep only n-best (variable n) list; also, account for different lengths of the English sentence candidates!
Analysis - Translation - Generation (A-T-G)

- Word forms: too sparse
- Use four basic analysis, generation steps:
  - tagging
  - lemmatization
  - word-sense disambiguation
  - noun-phrase “chunks” (non-compositional translations)

- Translation proper:
  - use chunks as “words”
Training vs. Test with A-T-G

• Training:
  – analyze both languages using all four analysis steps
  – train TM(s) on the result (i.e. on chunks, tags, etc.)
  – train LM on analyzed source (English)

• Runtime/Test:
  – analyze given language sentence (French) using identical tools as in training
  – translate using the trained Translation/Language model(s)
  – generate source (English), reversing the analysis process
Analysis: Tagging and Morphology

• Replace word forms by morphologically processed text:
  – lemmas
  – tags
    • original approach: mix them into the text, call them “words”
    • e.g. She bought two books. ⇒ she buy VBP two book NNS.
• Tagging: yes
  – but reversed order:
    • tag first, then lemmatize [NB: does not work for inflective languages]
    • technically easy
• Hand-written deterministic rules for tag+form ⇒ lemma
Word Sense Disambiguation, Word Chunking

• Sets of senses for each E, F word:
  – e.g. book-1, book-2, ..., book-n
  – prepositions (de-1, de-2, de-3,...), many others

• Senses derived automatically using the TM
  – translation probabilities measured on senses: p(de-3|from-5)

• Result:
  – statistical model for assigning senses monolingually based on context (also MaxEnt model used here for each word)

• Chunks: group words for non-compositional translation
Generation

• Inverse of analysis
• Much simpler:
  – Chunks ⇒ words (lemmas) with senses (trivial)
  – Words (lemmas) with senses ⇒ words (lemmas) (trivial)
  – Words (lemmas) + tags ⇒ word forms
• Additional step:
  – Source-language ambiguity:
    • electric vs. electrical, hath vs. has, you vs. thou: treated as a single unit in translation proper, but must be disambiguated at the end of generation phase; using additional pure LM on word forms.