Lecture-31 Statistical Machine Translation (EM Algorithm)

Statistical Machine Translation (SMT) is a translation technique where **translation is framed as a probability problem**. Given a source sentence in one language (e.g., English), SMT searches for the most probable sentence in the target language (e.g., Hindi) based on a trained statistical model.

Key Concepts in SMT

1. **Bayes' Rule**:

   SMT uses Bayes' theorem to find the best translation:

$$\hat{e} = \arg\max_e P(e|f) = \arg\max_e P(f|e) \cdot P(e)$$

   - f = foreign/source sentence (e.g., Hindi)
   - e = English/target sentence
   - $P(e)$: **Language model** (fluency of English sentence)
   - $P(f|e)$: **Translation model** (how likely Hindi words come from English words)

We want the computer to learn how to translate English words into Hindi words using a parallel corpus (same meaning, different language).

We're using IBM Model 1, which:

- Ignores word order (just learns word-word pairs).
- Learns translation probabilities.
- Uses an algorithm called Expectation-Maximization (EM).

Example

English	Hindi
the cat runs	बिल्ली दौड़ती है
a dog jumps	कुत्ता कूदता है
the dog runs	कुत्ता दौड़ता है
a cat jumps	बिल्ली कूदती है

Step 1: Vocabulary Setup

- **English Words:** `["the", "cat", "runs", "a", "dog", "jumps"]`
  Hindi Words: `["बिल्ली", "दौड़ती", "है", "कुत्ता", "कूदता", "दौड़ता", "कूदती"]`

We include all Hindi words that appear in any sentence. So every English word could translate to any Hindi word.

Step 2: Initialize Uniform Probabilities

We don't know anything yet — so we assume each English word can translate to each Hindi word equally.

For example:

- `P(बिल्ली | the) = 1 / 7 = 0.143`
- `P(है | cat) = 1 / 7 = 0.143`
  And so on for all combinations.

Step 3: Expectation Step (E-step)

We go sentence by sentence. For each English-Hindi pair:

- **For every Hindi word h in that sentence, and every English word e:**
    - **We say: “Let’s give e a fair share of contributing to h, based on how likely it is.”**
    - **We calculate expected counts of which word likely created which.**

total_s = P(बिल्ली | cat) + P(दौड़ती | cat) + P(है | cat)

= 0.143 + 0.143 + 0.143 = 0.429

Now we calculate the fractional count assigned to each Hindi word for "cat":

- **Count(cat, बिल्ली) = 0.143 / 0.429 = 0.333**
- **Count(cat, दौड़ती) = 0.143 / 0.429 = 0.333**
- **Count(cat, है) = 0.143 / 0.429 = 0.333**

Same goes for "the" and "runs", because their initial probabilities are also equal.

So for each English word in the sentence, the fractional count for each Hindi word becomes:

Let’s say total 6 English words and 8 Hindi words

Step 2: Initialize — Start with equal guessing

Since we don’t know any translations yet, we guess equally:

Each English word can be translated to any Hindi word with equal chance.

So:

P(हिंदी|English) = 1 / 8 = 0.125 (or 12.5%)

Step 3: E-Step (Expectation Step)

Let’s take one sentence pair and calculate how much “credit” each English word gets for producing each Hindi word.

Let’s take Sentence 1:

English: the cat runs
Hindi: बिल्ली दौड़ती है

- 3 English words → `the`, `cat`, `runs`
- 3 Hindi words → बिल्ली, दौड़ती, है

We now compute for each Hindi word:

Step 3.1: For each Hindi word, divide the probability among English words:

- For "बिल्ली":
    - Candidates: the, cat, runs
    - All have equal initial chance (0.125)

      = P(बिल्ली|the) + P(बिल्ली|cat) + P(बिल्ली|runs)

      = `0.125 + 0.125 + 0.125 = 0.375

Now divide each by total to get their share of credit:

- **the: 0.125 / 0.375 = 0.333**
- **cat: 0.333**
- **runs: 0.333**

So we say:

"the", "cat", and "runs" each get 1/3 credit for producing "बिल्ली".

Repeat the same for "दौड़ती" and "है".

Do the same process for all 4 sentences.

Step 1: Vocabulary Extraction

- **English words (E):** `{"the", "a", "boy", "girl", "eats", "reads", "runs"}`
- **Hindi words (H):** `{"है", "लड़का", "खाता", "लड़की", "खाती", "पढ़ती", "दौड़ता"}`

Step 2: Initialize Translation Probabilities

Assume uniform probabilities:

- **Every English word can translate to any Hindi word equally.**
- **Total Hindi words = 7, so each** `P(h|e)` **= 1/7 ≈ 0.143**

Hindi word लड़का **aligning to each English word in sentence:** `the`, `boy`, `eats`