Machine Translation HW2 Math Description

IBM Model 1 (IBM1):

Motivation:

IBM Model 1 establishes a simple, foundational model for word alignment in machine translation to find the most likely alignment of words between a source and target sentence.

Description:

Let:

F be a sentence in the source language, represented as $f_1, f_2, \cdots, f_n$
E be a sentence in the source language, represented as $e_1, e_2, \cdots, e_m$
a_i represent the alignment of f_i to some e_j denoted as $a_i = j$

IBM makes the following simplifying assumptions:

- Words are conditionally independent: $P(F, E, A) = \Pi_{i=1}^{n} P(a_i) | F, E)$
- Each source word f_i is conditionally dependent on its aligned target word, e_{ai} only: $P(F|E, A) = \Pi_{i=1}^{n} P(f_i \mid e_{ai})$
- All alignments a_i are independent: $P(A|F, E) = \Pi_{i=1}^{n} P(a_i \mid F, E)$

The parameters in IBM1 are:

- $t(e|f)$: The conditional probability of translating target word e into given source word f

The estimation of $t(e|f)$ is done using the Maximum Likelihood Estimation:

$t(e|f) = \frac{Count(f,e)}{\Sigma_{e'} Count(f,e')}$ where $Count(f, e)$ represents the count of the source-target word pair (f, e) in the training data.

IBM Model 2 (IBM2):

Motivation:

IBM Model 2 is an extension of IBM Model 1 that aims to improve the word alignment by considering fertility. Fertility can be described as the number of target words that a source word can align to. The motivation behind IBM2 is to account for more machine translation cases where a source word could produce multiple target words, thus making it more expressive than IBM1.

Description:

Let the parameters of IBM2 be the same as IBM1 of $t(e|f)$ where $q(j|i, l, m)$ is the probability that a source word f_i generates j target words when there are l source words and m target words.

IBM2 makes the following assumptions:

- Each source word f_i generates a fixed number of target words, which is called fertility
- The probability of fertility l for source word f_i depends on i and m but not on f_i

The alignment probability in IBM2 is extended as:

$P(F, A|E) = P(l_1^n, a_1^n| E, m) \cdot P(F|E, l_1^n, a_1^n)$ where l_1^n represents the set of fertility values for source words f_1^n. a_1^n represents the alignments.

The estimation of $t(e|f)$ is done using the Expectation-Maximization Algorithm.

E-step (Expectation Step):

In the E-step, we compute the expected value of the complete data log-likelihood, denoted as $Q\ (\theta \mid \theta^{(t)})$, given the observed data and the current estimate of the parameters $\theta^{(t)}$. This step involves computing the posterior distribution of the latent variables $P(X, Z \mid \theta^{(t)})$.

Mathematically, the E-step is defined as follows:

$$Q\ (\theta \mid \theta^{(t)}) = \Sigma_{\ Z \mid X, \theta(t)}\ [\log P(X, Z \mid \theta^{(t)})]$$

M-step (Maximization Step):

In the M-step, we maximize the expected complete data log-likelihood Q (θ | θ (t)) obtained in the E-step with respect to the model parameters θ. This step involves finding the parameters that maximize Q (θ | θ (t)) .

$$Q\ (\theta \mid \theta^{(t+1)}) = \arg\max_{\theta} Q\ (\ \theta \mid \theta^{(t+1)})$$

The process then iterates between the E-step and M-step until convergence is achieved, i.e., until the change in the estimated parameters between iterations is sufficiently small.

For specific models, the expressions for Q(θ | θ (t)) and the updates in the M-step can vary. The EM algorithm is a general framework, and the details depend on the particular statistical model being used. Typically, the E-step involves computing posterior probabilities or expectations with respect to the latent variables, while the M-step involves maximizing these expectations with respect to the model parameters.