Statistical Machine Translation

Yogesh B, EE12B066 R Santhosh Kumar, EE12B101 Sagar J P, CS12B039

INDIAN INSTITUTE OF TECHNOLOGY MADRAS

Abstract. We describe 2 models of Statistical Machine Translation (from French to English) by proposed by IBM, the IBM model 1 and the IBM model 2. The original version of the models were proposed for predicting alignment between two given sentences in the parallel corpus. We propose a generative framework to the problem, which enables us to translate from one language to another. Our approach allows us to reduce the search complexity for the translated sentence from $\mathcal{O}(V^n)$ to $\mathcal{O}(V^2)$ where V is the vocabulary size and n is the length of the given sentence. Also, we experiment with different alignment techniques and provide a comparative study. We measure the translations by the BLEU scores, which are reported for each experiment.

1 IBM Models

1.1 IBM Model 1

IBM model 1 is a generative model with breaks up the translation model into smaller steps. This model does only lexical translation. It is a word alignment model that is widely used in working with parallel bilingual corpora. It uses a EM algorithm to estimate the translation probabilities. The model gives us the conditional probability Pr(f|e), which we call the likelihood of the translation (f, e). We first choose a length for the French string, assuming all reasonable lengths to be equally likely. Here we assume that all the connections for the translations are equally likely. (f, e). This has a simple mathematical model so as to calculate EM iterations exactly.

The algorithm goes as follows:
1. Choose initial values for t(f|e).
2. For every pair of sentence (F(s), E(s)), 1 <s <S, and find the counts c(f|e, F(s), E(s)).

$$\text{c(f|e, F(s), E(s))} = \frac{t(f|e)}{\sum_{i=0}^{l} t(f|e_i)} * (\sum_{j=0}^{m} \delta(f, F_j)) * (\sum_{k=0}^{l} \delta(e, E_k))$$

where $\delta(a, B)$ gives a output as 1 if word a is present in the sentence B otherwise it is 0.
3. For each e that appears at least one of the sentence E.

– Compute λ_e according to the equation:

$$\lambda_e = \sum_f \sum_{s=1}^{S} \text{c(f|e, F(s), E(s))}$$

– For each f that appears in at least one F(s)

$$\text{t(f|e)} = \lambda_e^{-1} * \sum_{s=1}^{S} \text{c(f|e, F(s), E(s))}$$

4. We have to repeat the steps 2 and 3 until the values of t(f|e) have converged to the desired degree.

Now using these t(f|e) we will calculate the probabilities as follows:

$$\text{Pr(F|E)} = \frac{\varepsilon}{(l+1)^m} \prod_{j=1}^{m} \sum_{i=0}^{l} t(f_j|e_i)$$

1.2 IBM Model 2

IBM model 2 is an direct extension of IBM model 1. Here we assume that the probability of a connection depends on the positions it connects and on the lengths of the two strings. The orders matters in this case. The likelihood function of this model do not have a local maximum. In order to account for the dependence of the alignment on the position, an alignment probability is introduced given by

$$a(a_j|j, m, l) = P(a_j|a_1^{j-1}, f_1^{j-1}, m, l)$$

These must satisfy the constraint that the alignment probabilities for a French word to each English word must sum to 1. This is given by

$$\Sigma_{i=0}^{l} a(i|j, m, l) = 1$$

for each French word. So, the likelihood term effectively becomes

$$P(\mathbf{f}|\mathbf{e}) = \epsilon \sum_{a_1=0}^{l} ... \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(f_j|e_{a_j}) a(a_j|j, m, l)$$

As a result, the following are the update equations for IBM model 2:

$$\text{t(f|e)} = \lambda_e^{-1} * \sum_{s=1}^{S} \text{c(f|e, F(s), E(s))}$$

$$\text{Pr(F|E)} = \varepsilon \prod_{j=1}^{m} \sum_{i=0}^{l} t(f_j|e_i)$$

$$\text{a(i|j, m, l)} = u_{j,m,l}^{-1} \sum_{s=1}^{S} a(i|j, m, l, F(s), E(s))$$

$$\text{c(f|e, F(s), E(s))} = \frac{t(f|e)}{\sum_{i=0}^{l} t(f|e_i)} * \left(\sum_{j=0}^{m} \delta(f, F_j)\right) * \left(\sum_{k=0}^{l} \delta(e, E_k)\right)$$

$$\lambda_e = \sum_f \sum_{s=1}^{S} \text{c(f|e, F(s), E(s))}$$

$$u_{j,m,l} = \sum_i \sum_{s=1}^{S} \text{c(i|j,m, l, F(s), E(s))}$$

$$\text{c(i|j,m, l, F(s), E(s))} = \frac{t(f_j|e_i) * a(i|j,m,l)}{\sum t(f_j|e_i) * a(i|j,m,l)}$$

1.3 Approximated IBM Model 2

The alignment probabilities are stored in an alignment matrix for IBM Model 2. The alignment matrix should contain information for every (i, j) for each pair of (m, l). Since our training data is small in size, it is certain that most of the elements of the matrix will not be updated properly. To overcome this, we group together sets of (m, l) into bins. For example, $1 \leq m \leq 5$ and $1 \leq l \leq 5$ fall into one bin, $6 \leq m \leq 10$ and $6 \leq l \leq 10$, and so on. This will ensure that the alignment matrix for each group will get sufficient data to be trained. It also reduces the space complexity for our algorithm.

2 Testing

After the translation matrices are obtained, in the testing phase, we are required to generate the translated sentence in English. A naive approach to this task would be to generate all possible English sentences of length l and picking the best sentence using the translation matrix probability scores. This approach is inefficient, as the search complexity is of the order $O(V^n)$, where V is the size of vocabulary. We propose an efficient approach that reduces the search complexity from $O(V^n)$ to $O(V^2)$.

Let f be the given French sentence and e be the required English sentence. Let $a(x)$ represent the alignment function that tells us the alignment corresponding to each English word. Given these alignments, we are required to find the best possible sentence. Essentially, we are required to maximize $P(e|f)$ for these given alignments.

$$\hat{e} = \underset{e}{\operatorname{argmax}} P(e|f) = \underset{e}{\operatorname{argmax}} P(f_1 f_2 ... f_m | e_1 e_2 ... e_n) P(e_1 e_2 ... e_n)$$

By making conditional independence assumptions and inducing a bigram language model, the above equation reduces to

$$\hat{e} = \underset{e_1 e_2 ... e_n}{\operatorname{argmax}} P(f_{a(1)}|e_1) P(f_{a(2)}|e_2) ... P(f_{a(n)}|e_n) P(e_1) P(e_2|e_1) ... P(e_n|e_{n-1})$$

We solve the above equation efficiently by making Greedy updates at each step. We start by computing and storing $P(f_{a(1)}|e_1)$ for all words e_1 in the vocabulary. In the second step, for each of the word e_1, we compute the best e_2 that maximizes $P(f_{a(2)}|e_2)P(e_2|e_1)$. This demands a running time of $O(V)$. This process is repeated for each e_i, where at i^{th} step, we maximize $P(f_{a(i)}|e_i)P(e_i|e_{i-1})$. Finally, we get V sentences, each with different start word e_1 and we choose the best sentence by picking the sentence with maximum probability. Since we are required to fill nV entries, and computing each entry requires $O(V)$ time, the complexity of the proposed algorithm is $O(V^2)$

3 Experiments

3.1 Dataset

We used the Europarl Parallel Corpus for training and testing our models. In the Europarl Parallel Corpus, we used the French-English parallel corpus. The dataset can be obtained from here.

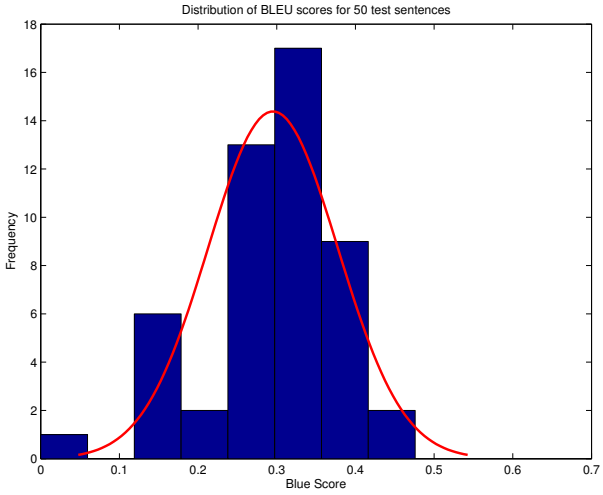
3.2 Model 1: Naive Maximization

We trained IBM Model 1 for 50 iterations. In the testing phase, we didn't use our proposed algorithm, but we tested by using a Naive Maximization approach. Essentially , after assigning the alignments randomly, we pick the English word that maximizes the likelihood with respect to the aligned French word (i.e) for each English word e_i, we maximize

$$\hat{e_i} = \underset{e_i}{\text{argmax}} P(f_{a(i)}|e_i)$$

In simple terms, this method doesn't take language model into account. We tested this model on 50 randomly chosen sentence pairs and stored the BLEU score profile. The histogram of BLEU scores obtained is shown below:

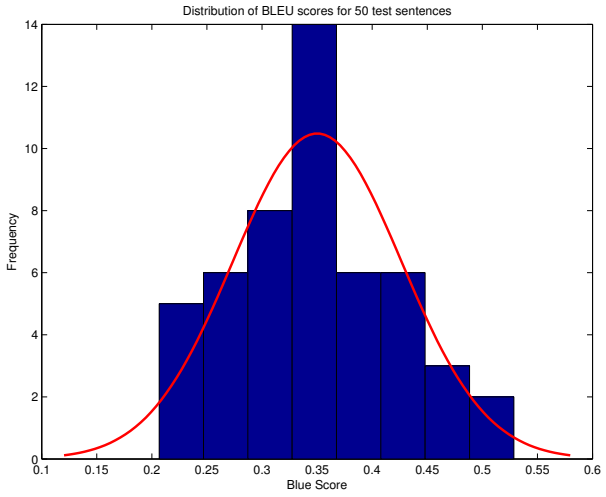
Fig. 1. Model 1 scores Naive implementation



3.3 Model 1: Proposed approach

The same Model 1 used in the above experiment was used here. But instead of the Naive Maximization approach, we use our proposed approach to generate the translations. The alignments were generated randomly by using a Gaussian distribution around the corresponding indices of the French- English word pair (Rephrase). The histogram of BLEU scores obtained is shown below:

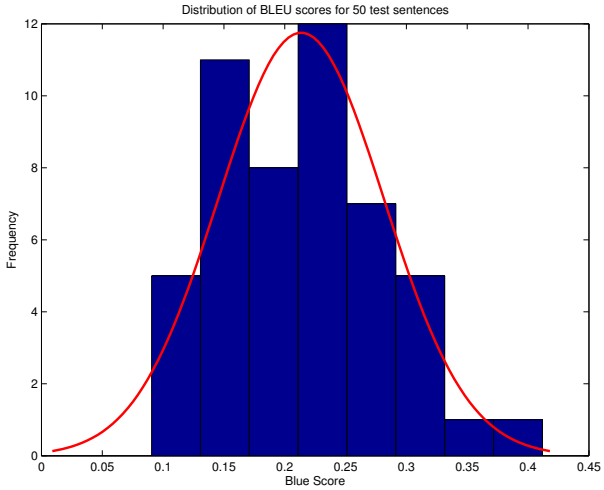
Fig. 2. Model 1 scores DP implementation



3.4 Model 2: Naive Maximization

The two experiments discussed above are carried out for IBM Model 2. The Model 2 was trained for 20 iterations using 1000 English-French sentence pairs. When Naive Maximization was used for generating translations, the following profile of BLEU scores were obtained

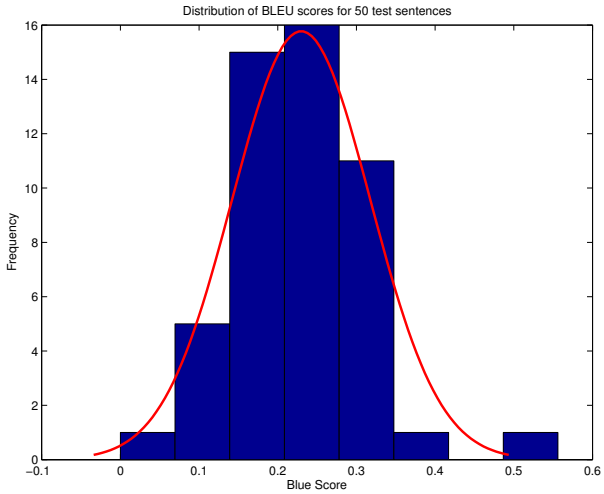
Fig. 3. Model 2 scores Naive implementation



3.5 Model 2: Proposed Approach

In this experiment,the IBM Model 2 used above, with our proposed algorithm was used to generate translations. As in model 1, the alignments were generated randomly by using a Gaussian distribution around the corresponding indices of the French- English word pair. The histogram of BLEU scores were obtained as shown below:

Fig. 4. Model 2 scores DP implementation



4 Conclusions

Based on the experiments performed, we have shown that our improved generation scheme has increased the BLEU scores on an average. Looking at the histogram plots for IBM model 1, we can see that more number of test samples have a higher BLEU score in our proposed approach. Also, the cases where the naive approach failed have a better score in our approach.
We tried IBM model 2, as an improvement to IBM model 1. Most of the errors we observed in IBM model 1, arose from poor alignments. Though our improved algorithm was able to handle the errors locally, the global alignment needed to be improved. IBM model 2 was designed to address this issue. However, our results suggest otherwise. IBM Model 2 under-performs when compared to IBM Model 1. This can be explained based on the number of parameters. IBM Model 2 has a much larger number of parameters when compared to IBM Model 1. Since the training data we have is small, it is likely that IBM Model 2 parameters were not trained properly. As a result, the BLEU scores are less.
To overcome this, we can train IBM Model 2 on a much larger dataset. However, due to computing power constraints, we were unable to do that. But as we can see, our improvised DP approach outperforms the naive approach. This shows that our approach is general and not specific to IBM Model 1.

References

1. Brown, P.; Della Pietra, S.; Della Pietra, V.; and Mercer, R. (1993). The Mathematics of Statistical Machine Translation: Parameter Estimation. Computational Linguistics, 19(2):263–311.
2. Philipp Koehn - Europarl: A Parallel Corpus for Statistical Machine Translation, MT Summit 2005