Comparitive Study of Word Alignment Models

Chirag Nagpal
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania
chiragn@cs.cmu.edu

Abstract

Word Alignment is an intermediate step in the process of Word Based Machine Translation and aims to find words in the source and there corresponding translation in the target language. Early Word Alignment models, utilised simple heuristics, which were then replaced by Statistical Approaches. In this study we compare the performance of such statisitcal alignment models.

1 Introduction

Early approaches towards Machine Translation aimed at looking at individual words, and finding there corresponding translations, in order to perform translation. However, a significant challenge in this process is that the order of the translated words in the target language may not correspond exactly with the order in the source language. Thus it is impossible to directly estimate the probability of a source word translating into a foreign word.

This essentially leads to a 'chicken and egg' problem, where translation probabilites are dependent on the alignments, while the alignments are dependent on the translation probabilities, making the estimation of both simultaneously, challenging.

Early statiscal approaches aimed to solve this issue using Expectation Maximisation. Such approaches, basically assign a prior value (equal proabability)to each source word aligning to each foreign word. They then estimate perform multiple rounds of EM, before the solution converges. Perhaps the most well know example of this approach is IBM Model 1. In principle, IBM Model 1 is convex, and hence after multiple iterations one can reach a global maximum.

However, there are certain challenges associated with Model 1. The major one being that the model has equal probabilites of aligning with any word in the target sentence however, in practice we observe that the alignment follows an order, and does not make arbitary jumps, rather prefers alignment to the next word in the target sentence.

Hidden Markov Model based alignment model (Vogel et al., 1996) exactly this phenomenon by penalising the model for aligning with words other than the immediately next word. Thus this model, can be thought of an HMM where the emission probabilites are the source word to target word translation probability and the transmission probability is the jump probability.

In this assignment, we implement three word alignment models,

- A Heurestic Aligner
- The IBM Model 1 Aligner
- HMM Based Model

We then compare these models on the basis of there Precision, Recall and AER on the test set, the amount of training data used. We then go onto to describe a techniques inorder to further improve these models.

2 Heuristic Aligner

The first model that we implement is a herestic aligner. The Heurisitc Aligner takes a simple function that aims to model the co occurence of words in order to determine the translation probability.

In our implementation we use the following function to define the co occurence.

$c(e, f)/(c(e), c(f))$

Here $c(e, f)$ is the count of co occurence in the training set and $c(e)$, $c(f)$ are the total count of words e, f in the corpus.

Tr. Size	0	10	100	1000	10k
Pre.	0.152	0.151	0.165	0.241	0.303
Re.	0.071	0.06	0.107	0.234	0.340
AER	0.873	0.875	0.854	0.761	0.685

Table 1: Heurestic Aligner

3 IBM Model 1

we then proceed to train an IBM Model 1 Aligner that uses EM inorder to estimate the model parameters. Inference is carried out by finding the word with the highest translation probability. For the given model we use 15 iterations of EM

$$e^* = \arg\max_e p(e|f)$$

Tr. Size	0	10	100	1000	10k
Pre.	0.300	0.313	0.344	0.427	0.52
Re.	0.414	0.414	0.464	0.624	0.769
AER	0.662	0.654	0.617	0.510	0.400

Table 2: IBM Model 1

4 Intersection

We notice that while our model has high recalls, we in general have low precision. Inorder to treat this, we use an intersected model, which involves the computation of both the source to target and target to source alignment and then the intersection of the two resulting alignment.

This increases the precision sharply, with very little fall in recall, which leads to an overall higher AER. Table below describes our intersected MODEL 1 aligner.

Tr. Size	0	10	100	1000	10k
Pre.	0.804	0.770	0.777	0.842	0.52
Re.	0.248	0.248	0.301	0.440	0.769
AER	0.599	0.602	0.542	0.405	0.400

Table 3: IBM Model 1 Intersected

5 Hidden Markov Model

We then proceed to implement the hidden markov model based aligner as described in (Vogel et al., 1996). However, we do not learn the 'jump' probabilites, rather use a heuristical function to determine the same.

$p(d) = e^{(|d|-u)}$, here d is the jump size, $i-j$ and u is an arbitarily set value, for our purpose we use 1. This function is a special case of the Laplace distribution over the jump size.

Inorder to train the model, we use the classical Forward-Backward or Baum-Welch Algorithm and inference is carried out using Viterbi decoding. We also train the alignments in both side and use the intersection of the two in the model.

Tr. Size	0	10	100	1000	10k
Pre.	0.754	0.752	0.761	0.810	0.862
Re.	0.396	0.390	0.431	0.579	0.582
AER	0.453	0.458	0.422	0.308	0.293

Table 4: HMM Model - Intersected

6 Pretraining

We finally use a combination of MODEL1 and HMM inorder to come up with a final best model. The final model uses the MODEL1 initially to compute the translation (emission) probabilities and then uses these learnt values as a prior to train the HMM Model.

Since the HMM model is not convex, using pretraining is a method of ensuring a better generalisable model, as compared to one that uses a uniform prior.

Tr. Size	0	10	100	1000	10k
Pre.	0.77	0.782	0.804	0.841	0.856
Re.	0.443	0.449	0.491	0.606	0.727
AER	0.411	0.403	0.369	0.280	0.202

Table 5: HMM Model - Intersected with Pretraining

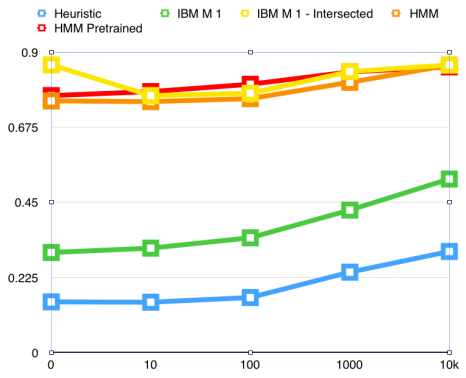

Figure 1: Precision

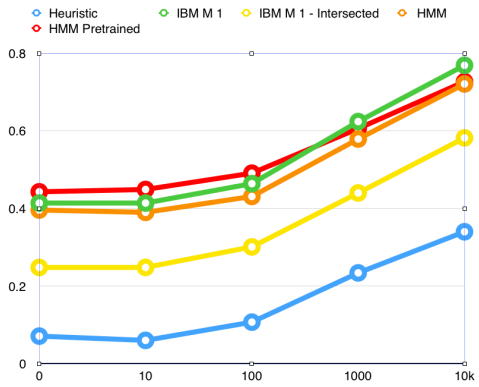

Figure 2: Recall

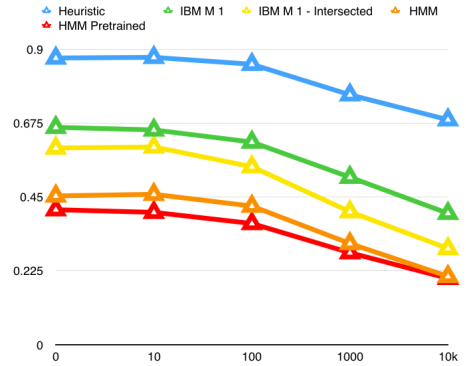

Figure 3: AER

7 Conclusion

In this study, we implemented 3 different word alignment models inorder to use for a word based statistical Machine Translation system. We found that the HMM model performs the best, we also implemented certain tricks to give an overall improved aligner, which included computing the alignments in both sides and taking the intersection and also pretraining with MODEL1 and using the outputs as priors for HMM.

Our final model had an AER of

Acknowledgments

We would like to convey our gratitude to Professor Taylor Berg-Kirkpatrick, Wanli Ma and Kartik Goyal for all the help for this assignment.

References

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. Hmm-based word alignment in statistical translation. In *Proceedings of the 16th conference on Computational linguistics-Volume 2*, pages 836–841. Association for Computational Linguistics.