Markov Chain Monte Carlo and Coverage Vector Models for Decoding

Akshay Srivatsan, Ian Mukherjee, Nathan Smith

March 10, 2016

1 Introduction

Our group incrementally implemented three translation decoder algorithms beginning with an extension of the beam decoder described in the assignment guidelines. Firstly, we extended the beam-search decoder to swap adjacent phrases. From there, we further extended the stack decoding method to utilized coverage vectors, yielding significantly improved model scores. Finally, we built a Markov Chain Monte Carlo model using Gibbs sampling, a combination of retranslations, merges, and reorderings to iteratively determine the optimal English translation. A detailed mathematical and theoretical discussion of each translation decoding algorithm, as well as their decoding scores, are as follows:

2 Phrase Swapping Beam Model:

Our implementation of the phrase-swapping beam model builds upon the given baseline decoder. We added adjacent phrase swapping by integrating the notion of lexicalized reordering mentioned in class (Koehn et. 2016). Lexicalized reordering specifies three different phrase pair orientations: monotone, swap, and discontinuous. Consider a phrase translation table between a source and target language. In the best case, our model finds each word in a phrase on the diagonal, i.e., there is no difference in placement of the two words in the source or target phrase. However, in our case, we deal with phrases as opposed to words. We consider an alignment as "monotone" if, given a negative-slope diagonal, a phrase alignment point to the top left exists. A "swap" alignment occurs when that phrase alignment point exists to the top right. If neither occurs, we consider the alignment "discontinuous."

For our purposes, we consider only the first two cases. The code implementation of this model is fairly simple when building from the given project file. We can consider the original process a "monotone" translation, i.e. the phrase operator acts upon each phrase individually without consideration of any phrases in front or behind it. Our added process can be considered a "swap" translation, in that it takes into account the current and next phrase. By checking the hypotheses for each phrase, and in turn their log-probabilities, we can decide whether the two phrases must be switch places within the sentence. On a lower level, we can describe the operation as follows:

Starting from the first phrase (p_1), we iterate through every possible adjacent phrase, extending from the end of p_1 until the end of our translation model, TM. We calculate the log-probability of the adjacent phrase, p_a, and assign it p_1's language model. We also set a new hypothesis for p_a. Similarly, we summate the log-probabilities for words within p_1. If p_a extends to the end of the sentence, add the ending language model state. Lastly, we consider a case of recombination, as done previously.

This new implementation provided the following results:

Accuracy Test for the Phrase-Swapping Beam Model:

Total TM Log-Probability	-1402.061933

3 Vector Coverage Model:

After completing the phrase swapping beam model, we looked into other implementations of stack-based models that could yield an improved accuracy. One implementation that was touched on in class is the use of coverage vectors, in which a set of source words is indicated to have already been translated. Much like the original implementation, we initialize stacks for each of the phrases in our sentence and intstantiate the initial hypothesis (a "blank" hypothesis without any translation information) in the zeroth stack. The iterations through each stack are more or less identical to the prior process, with an added "cover" tuple to store the current set of translated source words. In this instance, we also split our cases into monotone and swap, beginning our operation with the monotone probabilities and following with our nested for loop of swap probabilities. For each of these phrases, we create a new hypothesis not by simply creating a new hypothesis tuple, but by calling our $coverage_hypothesis$ function and computing new coverage vectors based on the currently trans-

lated phrases and extending to a new set of possible phrase translations from our current sentence index i to j. We continue this until all possibilities have been covered and each hypothesis has a sufficient coverage vector. This new Vector Coverage Model provided the following results: Accuracy Test for the Vector Coverage Model:

Total TM Log-Probability	-1259.093581

4 Markov Chain Monte Carlo:

Similar to our alignment markov model implementation, we developed a markov chain monte carlo decoder , using the Gibbs sampler to determine the optimal english translation. The model generates sample derivations from the complete search space, conditioned on previous samples. The Gibbs sampler applies a set of Gibbs operators to the previous sample; the retrans operators varies translations of a single French phrase and compares their resultant probabilities, the merge-split operator attempts splits and merges of multiple phrases to produce new target phrases, and the reorder operator varies the order of English phrases. Ideally the reorder operator would use a reordering limit to improve results, which we are currently in the process of implementing. The model runs each of these operators in succession for a specified number of iterations, with the most frequently produced English translation being selected as the finalized translation. In our implementation, the retrans step iterates through each sentence's French phrases and their corresponding English translations provided in the translation model, and replaces the english phrase in the event that a more probable translation is found. This may be represented by the following equation:

$$P(but|c'est, C) = \frac{P_{tm}(but|c'est) * P_{lm}(\text{S but some})}{Z}$$

where Z is the summation of the translation probabilities for e and f multiplied by the language model probabilities of the full English sentence with this phrase translation. For the merge-split operation, we iterate through each word, testing if splitting its corresponding phrase and selecting the maximum probability translation of the new phrases improves our model. When at the last word of any phrase, we instead merge the phrases and determine if the optimal English translation surpasses our current probability. For the reordering operation, we initially tested every possible word reordering, however an ordering limit may help to improve operation runtime and accuracy.

Accuracy Test for the Markov Chain Monte Carlo:

Total TM Log-Probability	Pending

5 Bibliography:

References

[1] Abhishek Arun, Chris Dyer, Barry Haddow, Phil Blunsom, Adam Lopez, Philipp Koehn Monte Carlo inference and maximization for phrase-based translation

[2] Nadir Durrani, Helmut Schmid, Alexander Fraser, Philipp Koehn, Hinrich Schutze The operation Sequence Model

[3] Daniel Ortiz Martinez, Ismael Garcia Varea, Francisco Casacuberta Nolla Generalized Stack Decoding Algorithms for Statistics Machine Translation