MT HW3

Arijit Nukala, Claire Jung, and Bhavik Agarwal

October 7, 2023

1 Introduction

Decoding is an essential piece of Machine Translation programs that enables programs to find the best translation of a given input by searching through translation probabilities. Practical translation requires programs to parse through possible translations of a sentence given translation probabilities of words and a language model that helps search algorithms determine if the chosen translation choices are probable. The decoding process is computationally intensive due to the enormous search space generated when creating translation hypothesis of a sentence using a phrase table which gives translation probabilities. In this assignment, we explore decoding algorithms that aim to reduce this search space while yielding a perfomant translation of an input sentence.

2 Decoding Algorithms

2.a Heuristic Beam Search

Heuristic beam search is a dynamic programming solution to decoding that keeps the top n translations of a sentence as it decodes word by word. Beam search uses a translation model, which contains log probabilities of word or phrase translations between two languages, and a language model, which contain the n-gram log probabilities of phrase occurrences in the target language. Starting at the beginning of the sentence, the algorithm takes a word from the sentence and looks at the next words to determine if a translation for the word or a phrase containing that word exists in the translation model. If a phrase exists, the algorithm adds language model scores for phrase and the translation model scores for the words that make up the phrase, generating a hypothesis for the translation. The algorithm chooses the top n hypothesis and continues the search until it reaches the end of the sentence. Finally, the highest scoring translation is used as the translation for the input sentence.

2.b Reordering

When translating between different languages, models have to take differences in grammatical structure into account. For example, in Hindi, the verbs are placed at the end of the sentence. If one were to say, मै ठीक हू, which translates to ”I am okay” or ”I am fine”, the translation without reordering would be ”I okay am”. Reordering requires permuting the words of an input sentence to generate possible alignments of a translation. However, as the sentence length increases, the search space with permutations increases factorially. We can reduce this space by enforcing a distance limit on the number of positions a word can move when permuted. Reordering enables the sensible translation of grammatically different languages and must be considered when implementing decoding.

2.c Future Cost Estimation

As an extension to beam search, we can also incorporate future cost estimation, which determines how expensive the rest of the sentence will be to translate, to improve our search for the best translation. We use an optimistic approach, finding the cheapest phrase probabilities from the translation model and the cheapest probability of generating the result without context from the language model to test the performance of the entire sentence from a hypothesis before we complete the translation. This helps guide the decoding system in chosing the best overall translation at each step in the translation process

3 Implementation

3.a Heuristic Beam Search

Algorithm 1: Heuristic Beam Search Algorithm

```
while s in fs do
    winner ← (0,BOS)
    perms ← permutations(s)
    while p in perms do
        while i, stack in S do
            h ← sort(stack)[0 : n]
            while j in j + 1 to len(s) do
                winner_cur ← NULL
                lm_state ← h_state
                while phrase in TM(s[i : j]) do
                    prob ← TM_prob_words + LM_prob
                end
                hyp ← (prob, h_state)
                S[j][lm_state] ← hyp
            end
        end
        winner_cur = max(S_end)
        if winner_cur > winner then
            winner = winner_cur
        end
    end
end
```

Where S is the DP 2d array that stores previous hypotheses and fs is the foreign sentence.

3.b Future Costs Decoding Algorithm

Algorithm 2: Future Costs Decoding Algorithm

```
while there are sentences f in the input do
    Load translation and language models for f;
    Initialize an empty matrix costs to estimate future translation costs;
    for each column col in the costs matrix do
        for each row row in the costs matrix do
            Calculate the start and end positions based on row and col;
            Extract the French phrase french_phrase from f;
            Calculate the best estimate best_estimate based on available phrases in the translation model;
            for each i from 0 to col do
                Calculate candidate_cost and update costs matrix with the maximum value;
            end
        end
    end
    Define a function getFutureCostForHypothesis to compute future costs for a hypothesis;
    Define a function getHypothesis to retrieve a hypothesis at a given depth;
    Define a function create_hypothesis to construct a new hypothesis with phrases;
    Initialize an empty stack for hypotheses;
    Initialize an initial hypothesis initial_hypothesis with zero log probability;
    Push initial_hypothesis onto the stack;
    for each word start in f do
        for each word end in f do
            if the span is not covered then
                if the span exists in the translation model then
                    Create new hypotheses based on the translation model and language model;
                    Update hypotheses with the best ones;
                end
            end
        end
    end
    Find the winning hypothesis based on the maximum log probability;
    Extract the English translation from the winning hypothesis;
    if in verbose mode then
        Compute and display language model and translation model scores;
    end
end
```