Stack Decoder

Yating Jing

March 5, 2015

1 Summary

I implemented a Phrase-based Decoder to search for the best English translation given a French sentence. The model encodes Beam Search Algorithm and Reordering of sentence segments of all possible lengths. To reduce the search space, risk-free recombination is adopted. Setting both the translation-per-phrase and stack size to 280, the total score is -1284.19.

2 Model Description

The main components of the model is Phrase Model, Language Model and Reordering Model. The phrase translation score $\phi(\mathbf{f}_i \mid \mathbf{e}_i)$ can be directly looked up using the given translation model. The language score $p_{LM}(w_i \mid w_{i-(n-1),\ldots,w_{i-1}})$ is obtained from the given n-gram language model. For reordering, here let spans of all possible lengths open to be reordered given a sentence. The probabilistic model is:

$$\mathbf{e}_{best} = arg\ max_{\mathbf{e}} \prod_{i=1}^{I} \phi(\mathbf{f}_i \mid \mathbf{e}_i) d(start_i - end_{i-1} - 1) p_{LM}(\mathbf{e})$$

2.1 Implementation

The data structures below are used to keep track of all the information needed:

1. **Stacks**: put the hypotheses with the same lengths of untranslated spans into the same stack.
2. **Coverage vector**: a boolean vector to indicate whether each French word in a given sentence has already been translated.
3. **Future costs**: the estimated probability of translating the remaining words, which is the negative log-probability of the untranslated words.
4. **Back pointer**: keep track of the predecessor of the current hypothesis.

In particular, for each given French sentence, precompute the costs (negative log probabilities) of spans of all possible lengths before exploring the hypothesis space. Consider all segmentations of the a given span by going through all possible phrase boundaries, but only keep track of the lowest cost. Then during the exploration of hypotheses, use the coverage vector to compute the future costs in real time.

The **score** for each **partial** (not fully translated) hypothesis then becomes:

$$score = logprob - future\_cost$$

2.2 Reordering

This model allows reordering of sentence segments of all possible lengths by using a translation **coverage vector** for each sentence to memorize which segments of the given sentence have already been translated.

2.3 Recombination

Given two hypotheses that cover the same number of translated foreign words, the inferior one would be pruned out. This is realized by putting the hypotheses that have translated the same number of French words into the same stack and drop the hypothesis with worse score given the same English output.

3 Results

With different translation-per-phrase **k** and the stack size **s**, the model would produce different scores, as showed in the table below.

Translation-per-phrase **k**	Stack Size **s**	Total Score (LM+TM)
50	50	-1305.27
80	160	-1295.36
250	250	-1287.94
280	280	**-1284.19**
285	285	**-1284.19**
290	290	**-1284.19**
300	300	-1286.01
500	500	-1288.29

Table 1: Model Score

References

[1] Philipp Koehn. *Statistical machine translation.* 2009. Cambridge University Press.

[2] Chris Dyer, Alon Lavie. *Slide: Phrase-Based MT: Decoding.* 2013.