Alignment with Minimum Translation Units

October 4, 2013

1 Task definition

A sentence is a sequence of words $S = w_1 w_2 ... w_n$, where w_i is the i-th word of the sentence. A span is a continuous ordered set $\{i | k \leqslant i \leqslant l\}$ (or equivalently represented as $[i, j]$). Given a source sentence S with length n and a target sentence T with length m, a phrasal alignment A is a set of span pairs $A = \{(s, t) \mid \bigcup_s = [1, n], \bigcap_s = \emptyset, \bigcup_t = [1, m], \bigcap_t = \emptyset\}$. Our task is to

- define a model $f : \{S, T, A\} \rightarrow \mathbf{R}$
- estimate the parameters in f
- given S, T, use some algorithm to find $A = \mathrm{argmax}_A\, f(S, T, A)$.

2 Notation related to phrase

- Phrase: Given a span $s = [i, j]$, the phrase corresponding to that span is $S_s = w_i w_{i+1} ... w_j$
- Span boundary: Given a span $s = [i, j]$, we define its left boundary position as $left(s) = i$, right boundary position as $right(s) = j$.
- Span pair: We define span pair $spair = (s, t)$, source span $src(spair) = s$, target span $tgt(spair) = t$.
- Segmentation of span: $seg(s) = \{s' | \bigcup s' = s, \bigcap s' = \emptyset\}$

3 Model f

We can define various model f. Here we give some examples.

- Model 1:

1. generate a group of phrase pairs $\{(ps, pt)\}$
2. order the phrase pairs into sentence pairs

The probability of tuple (S, T, A) is

$$p(S, T, A) = \prod_{(s,t) \in A} p(S_s, T_t) \tag{1}$$

- Model 1C: same as Model 1, except that we adopt conditional probability for phrase pair. $p(S_s, T_t) = p(T_t|S_s) * p(S_s)$ and $p(S_s) = \prod p(w|w \in S_s)$.
- Model 2:

1. generate begin of sentence pair ("$<s>''$, "$<s>''$)
2. generate phrase pair (ps, pt), append ps to the end of generated partial source sentence
3. repeat step 2 step until end of sentence pairs ("$</s>''$, "$</s>''$) was generated.
4. order the target phrases generated in step 2 into target sentence

Let's assume the reordering follow a uniform distribution. The probability of tuple (S, T, A) is

$$p(S, T, A)) = \prod_{(s_i,t_i) \in A} p((S_{s_i}, T_{t_i})|(S_{s_{i-1}}, T_{t_{i-1}})...(S_{s_1}, T_{t_1})) \tag{2}$$

where $left(s_i) - right(s_{i-1}) = 1$

- Model 3: Same as model 2, but here we assume the reordering follow a slightly more complicated distribution than uniform.

$$f(S, T, A)) = \prod_{(s_i,t_i) \in A} p((S_{s_i}, T_{t_i})|(S_{s_{i-1}}, T_{t_{i-1}})...(S_{s_1}, T_{t_1}))p(d(i)|d(i-1)...d(1))) \tag{3}$$

where $left(s_i) - right(s_{i-1}) = 1, \quad d(i) = left(t_i) - right(t_{i-1})$

4 Parameter estimation

4.1 Basic Approach

Given a sentence pair (S, T), $p(S, T) = \sum_A p(S, T, A)$.
Given a parallel corpus C where A is the latent data,

$$p(C) = \prod_{(S,T) \in C} p(S, T) = \prod_{(S,T) \in C} \sum_A p(S, T, A) \tag{4}$$

We estimate the parameters by optimizing the objective function $p(C)$. EM is one of the ways to do the optimization with latent variables.

4.2 Complexity

Given a source sentence S with length n, then we have number of phrases $\frac{n^2+n}{2}$ and number of segmentations $\sum\limits_{i=1}^{n} \binom{n}{i} = 2^n$.

Given a source sentence S with length n and a target sentence T with length m. Let $n \leqslant m$, the number of possible phrasal alignments

$$|\{A\}| = \sum_{i=1}^{n} \binom{n}{i} \binom{m}{i} i! > \sum_{i=1}^{n} \binom{n}{i} \binom{n}{i} = \binom{2n}{n} \sim \frac{4^n}{n^{1/2}\sqrt{\pi}}. \tag{5}$$

Search for the best phrasal alignment is NP-Hard, sum over all alignments is PSPACE-hard [2].

4.3 Pruning

It's impractical to do parameter estimation and search in the original model space. Below are some heuristic ways to do pruning.

- filter out phrases appearing less than 5 times in the corpus, but keep all unigrams [4]
- filter out phrases which is not compatible with word alignments. [1]

4.4 Search Algorithm

- Greedily hill climbing, with some heuristic actions : breaking and merging concepts, swapping words between concepts, and moving words across concept. [4]
- Exponential-time dynamic programming with word alignment constraint. [1]

5 Analysis

Why both of them underperform heuristic phrase extraction from word alignment?

[4] why?

[1] did not generate new phrases.

My guess is for MT, the decoding algorithm and LM are so powerful that the probability of phrase pair doesn't matter a lot. And the content of phrase pairs matter more. The phrasal alignment approaches don't necessarily generate more correct phrase pairs than the heuristic way.

5.1 Pilot experiment

If the guess above is correct, then the contribution from EM will not be significant. Initialization will be good enough. To test it, we just use the phrase table estimated from the initialization step for decoding. And compare the performance with state-of-the-art phrase table extraction.

5.1.1 Setting

- Training Data: corev5
- Dev Data: corev5.6 tune
- Test Data: corev5.6 test
- Language model: trigram language model trained on GIGAWORD3.
- System: Moses [3]

5.1.2 Phrase pairs extraction approaches

We try and compare following different approaches.

1. Approach in [4]. Filter out phrases appearing less than 5 times in the corpus, but keep all unigrams, attach t-distribution score to them.
2. Use Model 1C with constraint 0 < phrase length < X, attach IBM model 1 score for each phrase pair. How to choose the X? State-of-the-art MT system Moses [3] sets the maximal phrase length to be 7. We can try different value from 5 to 7.
3. State-of-the-art heuristic phrase extraction: run GIZA++($1^5 2^5 H^5 3^3 4^3$), compute the viterbi alignment for source-to-target and target-to-source directions, use heuristic grow-diag-final to combine the two alignments. Extract the phrases according to the alignment constraint. For each phrase pair, compute phrasal and lexical conditional probabilities for source-to-target and target-to-source directions.

5.1.3 Results

For the Approach in [4], Questions: 1. how many phrases will appear more than 5/X times? And what's the average length of such phrases?

For State-of-the-art heuristic phrase extraction,

1. try blocking the phrase conditional probability. blocked vs unblocked BLEU
2. try different maximal lengths of phrases and compare their performance. BLEU size

BLEU size

	Chinese		**English**	
cutoff	**#phrases**	**avg. length**	**#phrases**	**avg. length**
1	5.07e+07	6.40	6.57e+07	6.69
2	5.42e+06	5.07	5.78e+06	5.02
3	1.86e+06	4.11	2.31e+06	4.30
4	1.04e+06	3.58	1.36e+06	3.86
5	0.72e+06	3.31	0.97e+06	3.65

Table 1: cutoff frequency vs phrase table size

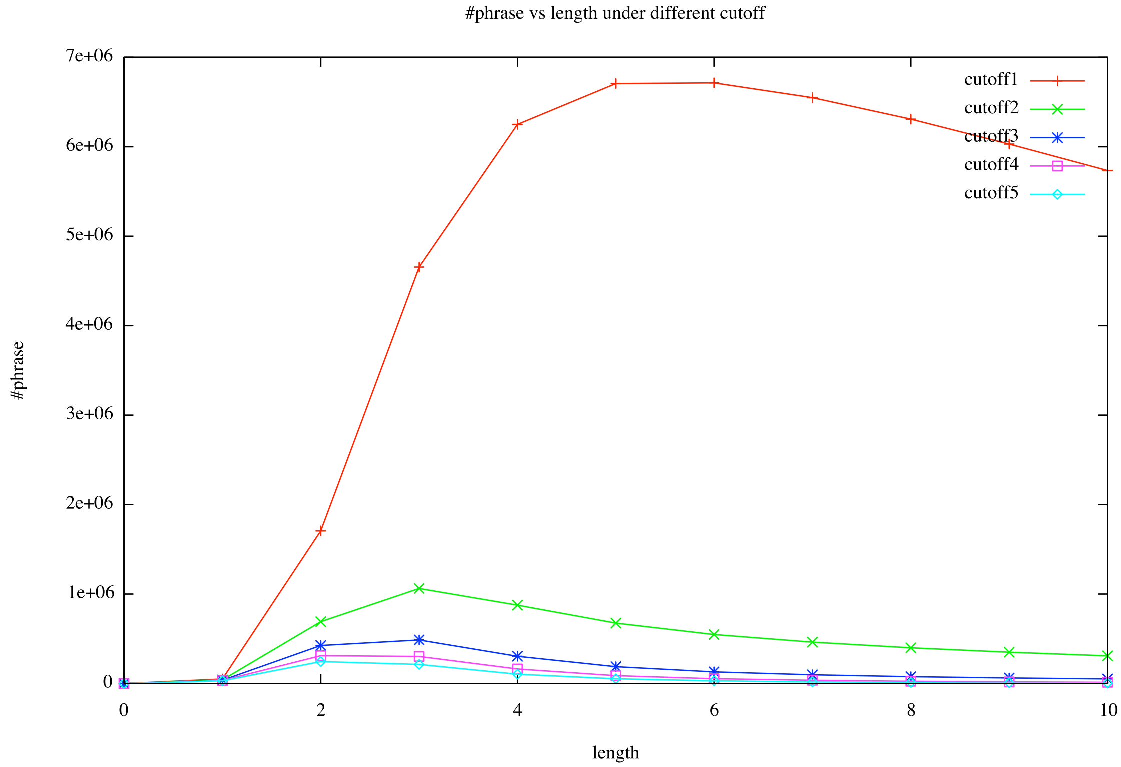

Figure 1: phrase count vs cutoff: Chinese

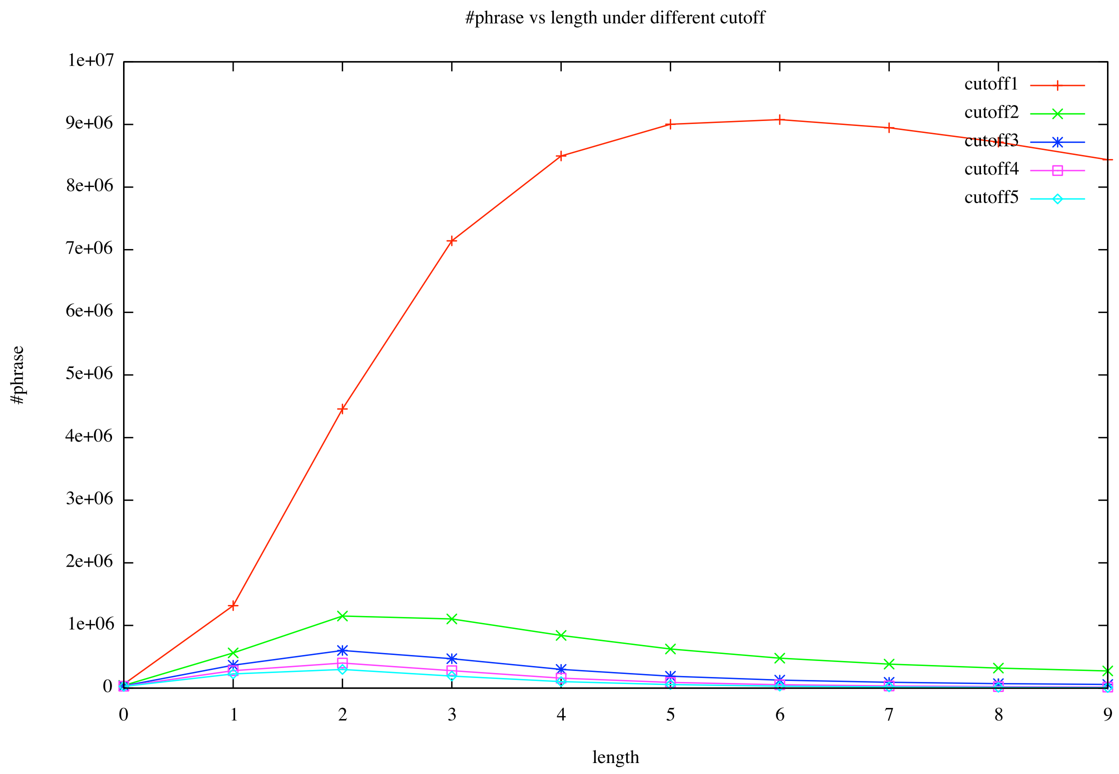

Figure 2: phrase count vs cutoff: English

References

[1] John DeNero, Dan Gillick, James Zhang, and Dan Klein. Why generative phrase models underperform surface heuristics. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 31–38, New York City, June 2006. Association for Computational Linguistics.

[2] John DeNero and Dan Klein. The complexity of phrase alignment problems. In *Proceedings of ACL-08: HLT, Short Papers*, pages 25–28, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[3] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[4] Daniel Marcu and Daniel Wong. A phrase-based,joint probability model for statistical machine translation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 133–139. Association for Computational Linguistics, July 2002.