Partially modelling word reordering as a sequence labelling problem

Anoop KUNCHUKUTTAN[1] *Pushpak BHATTACHARYYA*[1]
(1) IIT BOMBAY, Mumbai, India
anoopk@cse.iitb.ac.in, pb@cse.iitb.ac.in

ABSTRACT
Source side reordering has been shown to improve the performance of phrase based machine translation systems. In this work, we explore the learning of source side reordering given a training corpus of word aligned data. Given the large number of re-orderings this problem is NP-hard. We explore the possibility of representing the problem as a reordering of word sequences, instead of words. To this end, we propose a sequence labelling framework to identify work sequences. We also model the reversal of word sequences as a sequence labelling problem. These transformations reduce the problem to a phrase reordering problem, which has a smaller search space.

KEYWORDS: Statistical machine translation; Source reordering; Sequence labelling; Word alignment.

Proceedings of the Workshop on Reordering for Statistical Machine Translation, pages 47–54,
COLING 2012, Mumbai, December 2012.

1 Introduction

Phrase based machine translation is one of the most successful SMT paradigms in recent times. However, one of its major weaknesses has been the lack of a good distortion model, due to which reordering could not be handled correctly. Distance based penalty models would work only for language pairs where the word order is very similar. Practical constraints like the decoder's large search space also limit the possible reorderings that can be searched during decoding. Lexical binary reordering model (Koehn, 2008) proposes a limited reordering model conditioned on the phrases. An alternative approach which has been proposed is to reorder the source language sentence to conform to the target language word order before decoding. The search space for the decoder is thus simplified, and thus translation can be performed effectively with a simple distortion model. Many solutions for manipulating source side parse trees, with manual (Collins et al., 2005; Ananthakrishnan et al., 2008) or automatic rules (Xia and McCord, 2004), have shown improvement in the performance of PBSMT. However, these solutions are language pair specific, cannot be easily scaled to new language pairs and may require linguistic resources like parsers on the source/target sides.

Therefore, recently, approaches have been explored to learn word reorderings on the source side in a language independent way. Visweswariah et al. (2011) model the word reordering as a Travelling salesperson problem whereas Tromble and Eisner (2009) model it as a linear ordering problem. Given the large number of re-orderings this problem is NP-hard. We explore the possibility of representing the problem as a reordering of word sequences, instead of words. To this end, we propose a sequence labelling framework to identify work sequences. We also model the reversal of word sequences as a sequence labelling problem. These transformations reduce the problem to a phrase reordering problem, which has a smaller search space.

In Section 2, we discuss our word sequence based reordering model, and how it has been partially cast as a sequence labelling problem. Section 3 discusses our experiments. Section 4 describes the results of our experiments and analyses the results.

2 Reordering Model

2.1 Motivation

Visweswariah et al. (2011) and Tromble and Eisner (2009) have considered the source reordering problem to be a problem of learning word reordering from word-aligned data. Finding the right reordering is exponential in the number of words in the sentence and hence intractable. Use of heuristics to overcome this bottleneck will result in suboptimal solutions. However, a key observation that can be made is that word sequences, as opposed to individual words, are displaced from their original position. Another common transformation is that a sequence of words get reversed. As an example, in 1, we can see that the that are two word sequences. The second word sequence also undergoes reversal. Source side reordering can thus be considered to be a composition of the following operations on a sentence:

- Breaking the sentence into word sequences
- Reverse some of the word sequences
- Reorder the word sequences

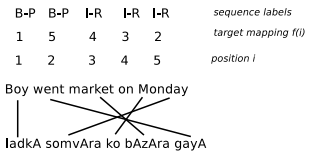

Figure 1: An example of word sequences and their sequence labelling

2.2 Sequence Labelling Model

We model the first two steps as a sequence labelling problem. For this purpose, we use a labelling scheme similar to the one used for sentence chunking. Our label set consists of two types of labels at the top level: beginning-of-word-sequence (B) and inside-word-sequence (I). There is only one type of B label (B-P), whereas there are two types of I labels:

- I-R: This indicates that the word is part of a word sequence which has been reversed.
- I-S: This indicates that the word is part of a word sequence which has not been reversed.

Figure 1 gives you an example of the sequence labels.

2.2.1 Generating training data

Here we describe a method to generate the label sequences from the word alignment information. Given the word alignment information, for every position i in a training data sentence, we can obtain its position in the reordered sentence $f(i)$. The beginning of a word sequence is indicated by the mismatch between i and $f(i)$, whereas within a word sequence we can label a word at position i as 'I-S' or 'I-R' depending on the whether $f(i)$ precedes or follows $f(i-1)$. Algorithm 1 describes our method to determine the label sequences given $f(i)$ for the training data.

2.3 Reordering the phrases

For reordering word sequences, we adapt the method of Visweswariah et al. (2011) to word sequences - where reordering is modeled as a Travelling Salesperson (TSP) problem. We consider word sequences, instead of words, to be the cities, and define the cost of travelling from one city to another. Since the cost is asymmetric, we convert the problem into a symmetric one by using the method suggested by Hornik and Hahsler (2009) of doubling the number of edges and setting weights appropriately . Finally, we use the Lin-Kernighan heuristic to solve the symmetric TSP problem tractably.

2.3.1 Generating training data

We model the cost between two word sequences, $c(w_i, w_j)$, as follows:

$$c(w_i, w_j) = |g(i) - g(j) - 1| \text{ if } g(i) < g(j) \quad (1)$$

$$= 2 * |g(i) - g(j) - 1| + 1 \text{ otherwise} \quad (2)$$

where,
i, j are source word sequence positions and $i < j$

Algorithm 1 Generating sequence labels from word alignments for training

```
                                          ▷ i: position i in source sentence
                  ▷ f(i): position in target sentence of word at position i in source sentence
state ← 'start'
for i = 1 → len(sentence) do
    if state = 'in' then
        if f(i) = f(i−1) + 1 then
            label[i] ← 'I-S'
        else
            if f(i) = f(i−1) − 1 then
                label[i] ← 'I-R'
            else
                state ← 'start'
            end if
        end if
    end if
    if state = 'start' then
        label[i] ← 'B-P'
        state ← 'in'
    end if
end for
```

$i = 0$ represents a beginning-of-sentence marker
$g(k)$ is the position of the phrase sequence at position k after source reordering
The cost for the case $i > j$ can be obtained by symmetry from the above equation.

The intuition behind the cost assignment is that the cost is less if the words are closer to each other in the reordered sentence. In addition, if the positions of word sequences are reversed with respect to each other, it incurs an additional cost which is modelled by the multiplicative factor.

To illustrate this consider the following example with word sequences and their reorderings

I	walk	fast office	*sentence with word sequences*
1	2	3	*i - index of word sequence*
1	3	2	$g(i)$ *- index of word sequence after reordering*

The cost assignment between word sequences is shown in Figure 2:

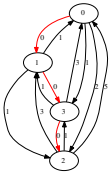

Figure 2: An example of cost assignment to phrase sequence orderings

2.4 Estimating cost between cities

The cost is defined as a function of the lexical and syntactic features as follows:

$$c(i, j) = \phi(s_i, s_j, (s))$$

where,
i, j are positions of the source side word sequences as identified by the sequence labeller (s) is the set of word sequences in the sentence

Our conjecture is that the reordering and the cost depends only the words at the ends of the word sequences under consideration. Hence for features we use information from the words at ends of the word sequences only.

ϕ is assumed to be a linear function of the features and the parameters are learnt using SVM regression.

However, for each sentence containing n word sequences, there will be $n(n+1)$ such cost instances. This results in a large number of training instances. Hence we train the regression model from the cost instance of a subset of the training sentences. However, we assume that this would not affect the accuracy of the regression model, since the same syntactic effects would be affecting the cost model across all the sentences in the same language pair.

3 Experiments

This section describes the dataset used and the sequence labelling, cost modelling and TSP experiments.

3.1 Data

Language Pair	Training	Dev	Test
en-fa	5000	500	500
en-ur	5000	500	500
en-it	4000	500	500

Table 1: Data set size

We experimented with word aligned data in three languages - English, Persian and Italian. The details of the dataset are summarized in Table 1. In addition to the word alignment information, coarse POS tag, fine POS tag and chunk information were used. The chunk information was obtained by flattening complete parse tree information available from a statistical parser.

3.2 Sequence Labelling

Sequence labelling was done using the *CRF++* toolkit. The sequence labels for training the CRF were obtained using the procedure mentioned in Algorithm 1 and all the training data was used for learning the CRF model. We experimented with binary features involving only the current label (unigram features) as well as current and previous label (bigram features). The entire list of features used is listed below:

Unigram features - $f(l_i, x)$, where x stands for one of

- Tokens at positions 0, 1, 2, -1 and -2 from current token.
- POS tags at positions 0, 1, 2, -1 and -2 from current token.
- Chunk tags at positions 0, 1, 2, -1 and -2 from current token.

Bigram features - $f(l_i, l_{i-1}, x)$, where x stands for one of

- Token at position 0 from current token.
- POS tag at position 0 from current token.
- Chunk tag at position 0 from current token.

Adjacent labels - $f(l_i, l_{i-1})$

where, l_k is label of position k in the sentence.

We experimented with different configurations of the above features and the best result was obtained when all the above mentioned features were used.

3.3 Cost Regression Model

The regression models were trained using the SVMLight (Joachims, 1999) implementation of SVM regression. An RBF kernel with default values for the parameters c and γ was used. Three cost regression models were build using subsets of the entire training set containing 500, 1000 and 1500 sentences respectively.

The following features were used for both the word sequences in each training instance :

- If the word sequence is reversed
- Token, POS and Chunk from positions 0 and 1 from the left end of the word sequence
- Token, POS and Chunk from positions 0 and 1 from the right end of the word sequence

4 Results and Analysis

Sequence Label	en-fa			en-ur			en-it		
	P	R	F-1	P	R	F-1	P	R	F-1
B-P	79.75	70.22	74.69	83.71	80.47	82.06	77.92	61.55	68.78
I-R	65.64	70.12	67.81	45.88	52.57	49.00	64.78	62.24	63.48
I-S	39.55	35.16	37.23	56.47	54.90	55.67	36.47	27.70	31.48

Table 2: Best sequence labelling results

Table 2 shows the per label accuracies of sequence labelling, for the best performing feature configuration (which is all the features listed in Section 3.2). Table 4 shows the monolingual BLEU scores for reordering. The scores are reported for three training configurations corresponding to the number of sentences used for training the cost estimation models. In all these cases, the best performing feature configuration for sequence labelling was used. The baseline

comparison was the BLEU score on the original, unreordered sentences, which is shown in Table 3.

It is clear that the accuracy of sequence labelling is not good enough to predict the word sequence or sequence reversals. Especially, the accuracy in the prediction of sequence reversal is very low. This low level of accuracy of sequence labelling obviously makes it difficult to get good results with word sequence reordering. This is reflected in the low BLEU scores - which are near the baseline scores.

The use of a small subset for training the cost regression model does not result in significant deterioration of the BLEU score.

The low monolingual BLEU score could be attributed to the following reasons:

- The features used for sequence labelling are not sufficient to capture the phenomena of word sequence displacement and reversal.
- The choice of the cost model and features for regression
- About 10% of the TSP problems (with number of nodes < 8) were not solved by the Concorde solver.

Language Pair	BLEU
en-fa	51.36
en-ur	39.54
en-it	68.98

Table 3: Baseline Results (dev set)

Language Pair	500 sent	1000 sent	1500 sent
en-fa	49.14	49.2	49.1
en-ur	39.14	39.35	39.3
en-it	68.82	68.64	NA

Table 4: Evaluation results for different sizes of cost regression training data

Conclusion and perspectives

Although word sequence labelling is intuitively appealing, and the success of parser output based reordering rules suggests that phrase reordering is useful, the results presented have not been encouraging. However, we believe more work could be done in the following areas to improve the system:

- Choice of better features for the sequence labelling
- Better modeling of the cost between two word sequences
- Choice of better features for the cost regression model

References

Ananthakrishnan, R., Hegde, J., Bhattacharyya, P., and Sasikumar, M. (2008). Simple Syntactic and Morphological Processing Can Help English-Hindi Statistical Machine Translation. In *IJCNLP*.

Collins, M., Koehn, P., and Kučerová, I. (2005). Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics - ACL '05*.

Hornik, K. and Hahsler, M. (2009). Tsp-infrastructure for the traveling salesperson problem. *Journal of Statistical Software, 23 (i02)*.

Joachims, T. (1999). Making large-Scale SVM Learning Practical. In *Advances in Kernel Methods - Support Vector Learning*.

Koehn, P. (2008). *Statistical Machine Translation*. Cambridge University Press.

Tromble, R. and Eisner, J. (2009). Learning linear ordering problems for better translation. In *Conference on Empirical Methods in Natural Language Processing*.

Visweswariah, K., Rajkumar, R., Gandhe, A., Ramanathan, A., and Navratil, J. (2011). A Word Reordering Model for Improved Machine Translation. In *Empirical Methods in Natural Language Processing*.

Xia, F. and McCord, M. (2004). Improving a statistical MT system with automatically learned rewrite patterns. In *Proceedings of the 20th International Conference on Computational Linguistics - COLING '04*.