A Study of Word-Classing for MT Reordering

Ananthakrishnan Ramanathan, Karthik Visweswariah

IBM Research India
{aramana2,v-karthik}@in.ibm.com

Abstract

MT systems typically use parsers to help reorder constituents. However most languages do not have adequate treebank data to learn good parsers, and such training data is extremely time-consuming to annotate. Our earlier work has shown that a reordering model learnt from word-alignments using POS tags as features can improve MT performance (Visweswariah et al., 2011). In this paper, we investigate the effect of word-classing on reordering performance using this model. We show that unsupervised word clusters perform somewhat worse but still reasonably well, compared to a part-of-speech (POS) tagger built with a small amount of annotated data; while a richer tagset including case and gender-number-person further improves reordering performance by around 1.2 monolingual BLEU points. While annotating this richer tagset is more complicated than annotating the base tagset, it is much easier than annotating treebank data.

Keywords: SMT, Reordering, POS-Tagging

1. Introduction

Word-reordering is one of the crucial problems in machine translation. Early statistical machine translation (SMT) systems depended on statistical distortion models combined with a target-side language model to come up with the correct target-language order (Brown et al., 1993). The simplest of these distortion models just penalize long jumps in the source sentence when producing the target sentence. These have also been generalized (Koehn et al., 2005; Al-Onaizan and Papineni, 2006; Tillmann, 2004) into lexicalized models. While these models are simple, and can be integrated with the decoder, they have been found wanting for language pairs that have divergent constituent orders (Birch et al., 2009; Al-Onaizan and Papineni, 2006) (like Hindi [1]-English).

Phrase-based systems can also capture short range reorderings via the phrase table. Though, even this short-range reordering performance is constrained by the amount of training data available. For example, if adjectives precede nouns in the source language and follow nouns in the target language we still need to see a particular adjective noun pair in the parallel corpus to handle the reordering via the phrase table.

One common method of achieving high-quality reordering for MT involves using a parser. In SMT, this can be done either by incorporating parsing information within the translation model (e.g., syntax-based models (Yamada and Knight, 2001; Zollmann and Venugopal, 2006)), or through pre-ordering – where hand-made (Collins et al., 2005) or automatically learnt rules (Visweswariah et al., 2010) over parse trees are used to transform the source language sentence into target order. This latter technique of pre-ordering reduces the burden of reordering from the core SMT system [2].

One problem with these methods is that they require high-quality parsers, which are not available for most languages. Treebank data for learning parsers are expensive to acquire. Consequently, there has been a lot of emphasis lately on learning parsers from data that can be annotated easily (for example, alignment data (Mannem and Dara, 2011)) and on performing reordering without requiring a parser, typically using word-alignment data, obtaining which is fairly easy compared to obtaining treebank data.

Our previous work on word-reordering (Visweswariah et al., 2011) is an example of the latter approach of reordering without parsing. We cast reordering as a Travelling Salesman Problem (TSP) (Tillmann and Ney, 2003; Zaslavskiy et al., 2009), where words are cities, and the problem essentially is to learn the costs (distances) of moving from one word to another, so that the shortest tour corresponds to the ordering of the words in the source sentence in the target language. The TSP distances for reordering are learnt from a small amount of high-quality word alignment data by means of pairwise word comparisons and an informative feature set involving words and part-of-speech (POS) tags adapted and extended from prior work on dependency parsing (McDonald et al., 2005). Closely related to our work is (Tromble and Eisner, 2009), which formulates word reordering as a Linear Ordering Problem (LOP), and learns LOP model weights capable of assigning a score to every possible permutation of the source language sentence from an aligned corpus by using a averaged perceptron learning model. The key difference between our model and the model in (Tromble and Eisner, 2009) is that while they learn costs of a word w_i appearing *anywhere* before w_j, we learn costs of w_i *immediately preceding* w_j. This results in more compact and better models (Visweswariah et al., 2011).

As mentioned, in our reordering model, apart from the words themselves, POS tags have proven to be important features for learning TSP distances. In this paper, we focus on the use of different kinds of word-classes within our reordering model, ranging from completely unsupervised word clusters, to a POS tagger learnt from a small amount of annotated data, to the use of richer tags that we believe may be more pertinent to MT reordering. Our results for Hindi-to-English MT show that:

- A POS tagger trained on a small amount of data brings

[1] Hindi is the principal official language of India with over 300 million speakers

[2] This also speeds up the SMT system by reducing the search space of the decoder

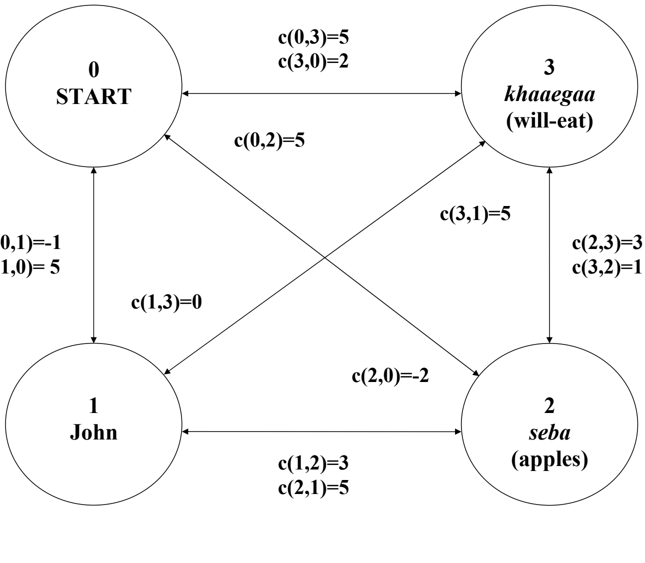

Figure 1: Example of an ATSP for reordering "John seba khaaegaa"

substantial improvements

- In the absence of any annotated data, unsupervised word clusters are a good substitute for POS tags
- Richer tags for grammatical relations (case) and gender-number-person are beneficial

The rest of the paper is organized as follows: Section 2. describes our reordering model briefly; section 3. lists the key word-order differences between Hindi and English; section 4. discusses word-classing features relevant to reordering, and also describes the techniques used for word-classing, including the POS tagset, annotation, tagger training, and unsupervised clustering; Section 5. contains the experimental results and section 6. concludes the paper with some discussion of the results and directions for possible future work.

2. Reordering Model

The details of our reordering model and its implementation can be found in (Visweswariah et al., 2011). This section provides a brief summary.

Consider a source sentence $\mathbf{w}$ consisting of a sequence of n words w_1, w_2, ... w_n that we would like to reorder. Given a permutation π of the indices $1..n$, let the candidate reordering be w_{π_1}, w_{π_2}, ..., w_{π_n}. Thus, π_i denotes the index of the word in the source sentence that maps to position i in the candidate reordering. There are $n!$ such permutations. Our reordering model assigns costs to candidate permutations as:

$$C(\pi|\mathbf{w}) = \sum_i c(\pi_{i-1}, \pi_i).$$

The cost $c(m, n)$ can be thought of as the cost of the word at index m immediately preceding the word with index n in the candidate reordering. In our model, we parametrize the costs as:

$$c(m, n) = \theta^T \Phi(\mathbf{w}, m, n),$$

where θ is a learnt vector of weights and Φ is a vector of feature functions.

Given a source sentence $\mathbf{w}$ we reorder it according to the permutation π that minimizes the cost $C(\pi|\mathbf{w})$. Thus, we would like our cost function $C(\pi|\mathbf{w})$ to be such that the correct reordering π^* has the lowest cost of all possible reorderings π.

The minimization problem that we need to solve with this model structure is identical to solving an Asymmetric Traveling Salesman Problem (ATSP) with each word corresponding to a city, and the costs $c(m, n)$ representing the pairwise distances between the cities.

Consider the following example:

Hindi: John seba (apples) khaaegaa (will eat)
English: John will eat apples
Desired reordering: John khaaegaa (will eat) seba (apples)

The ATSP that we need to solve is represented pictorially in Figure 1 with sample costs. Note that we have an extra "START" node numbered 0, where we start and end the tour.

In this example the minimum cost tour is:

START $\rightarrow$ *John* $\rightarrow$ *khaaegaa* $\rightarrow$ *seba*,

which is the desired reordering for translation into English.

To solve the ATSP, we first convert the ATSP to a symmetric TSP and then use the Lin-Kernighan heuristic as implemented in *Concorde*, a state-of-the-art TSP solver [3].

To be able to generalize from relatively small amounts of data, we use features that in addition to depending on the words in the input sentence $\mathbf{w}$, depend on the part-of-speech (POS) tags of the words. All features $\Phi(\mathbf{w}, i, j)$ we use are binary features, that fire based on the identities of the words and POS tags at or surrounding positions i and j in the source sentence. There are two categories of features

[3] http://www.tsp.gatech.edu/

we use: Bigram features and Context features. Bigram features depend only on the identities of the word and POS tag of the two positions i and j. Context features look at surrounding positions ($i-1$, $i+1$, $j-1$, $j+1$, and one position in between), in addition to positions i and j. Each of these is conjoined with the signed distance (quantized) between i and j. The results in (Visweswariah et al., 2011) show that all these features contribute to the reordering performance of the model.

The model is trained using word-aligned data, from which we obtain reference reorderings. The Margin Infused Relaxed Algorithm (MIRA) (McDonald et al., 2005) is used for learning the weights θ in the model.

This reordering model is used to preorder the training and test data before the main SMT system kicks in. In this setting, the model results in improved reordering and translation performance for: Hindi → English, English → Hindi, and Urdu → English. For English → Hindi, the results with our reordering model were better than with rules applied to the source side parse.

3. Hindi-English: Word Order Differences

The fundamental differences between Hindi and English in terms of word-order are (Ananthakrishnan et al., 2009):

- English follows subject–verb–object (SVO) order, whereas Hindi follows SOV order
- English uses post-modifiers (and prepositions), whereas Hindi uses pre-modifiers (with post-positions)
- Hindi allows greater freedom in word-order, identifying constituents through case-marking

4. Word-Classing

Based on the differences between Hindi and English, the following word-class features are quite relevant to reordering.

4.1. Features Relevant to Reordering

- Grammatical relations (case): These are the most important features relevant to word-order. For example, identifying the subject, object, modifiers, etc. is the primary step towards translating from a SOV language like Hindi to an SVO language like English.
- POS: These features are primarily relevant to identifying the word-order within phrases, e.g., to know that the post-position within a modifying phrase in Hindi should be moved from the end to the beginning of the phrase when pre-ordering for Hindi-English translation ("*meza* (the table) *para* (on)" should be pre-ordered as "on the table").
- Agreement features: Different constituents of a sentence agree with each other on features such as gender, number, person, etc., signifying certain relations between these constituents (e.g., subject-verb agreement). These features can be useful, for example, in determining the modifiers of a phrase (which we do not mark as part of the case tags). Consider the following two pairs of Hindi-English translations (gender is marked as *mas* and *fem*):

  Hindi: *ladakaa/mas (boy) daudataa/mas huaa/mas (running) nadii/fem (river) ke paas (near) aayaa/mas (came)*

  English: The boy came running towards the river.

  Hindi: *ladakaa/mas (boy) bahatii/fem huii/fem (flowing) nadii/fem (river) ke paas (near) aayaa/mas (came)*

  English: The boy came towards the flowing river.

  The gender tags indicate that in the first sentence it is "came" and "boy" that are being modified by "running", while the similarly placed modifier "flowing" in the second sentence modifies "river", which is reflected in the placement of these modifiers in the translations [4]

4.2. POS Tagging

We had around 38K words of text annotated by two annotators. A CRF (Lafferty et al., 2001) tagger was trained on this set and tested on around 500 words.

The tags used and the tagging accuracies on the test set are detailed below:

- Case: Subject, object, indirect object, and passive subject were the grammatical relations that were marked. Accuracy: 85%.
- POS: Our POS tagset was based on the tagging guidelines in (Bharati et al., 2006) for Hindi POS annotation. Accuracy: 84%.
- Agreement Features: Gender (*masculine/feminine*), number (*singular/plural*) person (*first/second/third*), tense (*past/present/future*), and aspect (we mark *perfect*, *progressive*, and *habitual*) are the features marked. Gender-Number-Person (GNP) accuracy: 83%; Tense-Aspect accuracy: 90%.

The tagset has 26 *base* tags, which are decorated with case and agreement features. For example, a common noun (NN), which has features feminine, singular, and third-person, and which is the head word of the subject of the clause is marked as "NN_fs3_subj".

The inter-annotator agreement on various sets of data ranged between 92–93% for the standard POS tags, 94% for case tags, between 90–92% for GNP, and 97–98% for tense-aspect.

The 7–8% disagreement on the base tags seems high. However, as the annotation experiments in (Marcus et al., 1993) reveal, when working with unannotated texts (as against texts pre-tagged using an automatic tagger) the disagreement rate is slightly more than 7%. The 3% error rate is presumably because the annotators defer to the automatic

[4] These features are only indirect indicators, which will not work in all sentences (e.g., gender will not be useful when all constituents have the same gender)

Sample words from the cluster	**Cluster description**
dekhtaa (sees), *badaltaa* (changes), *pahunchtii* (reaches) ...	verbs (habitual form)
aarambh (start), *taiyaar* (ready), *spashta* (clear), *aamantrit* (invited) ...	heads of light verbs
se (from), *par* (on), *ko* (to), *men* (in), *ke* (of) ...	post-positions
tabhii (then), *magar* (but), *jabaki* (whereas), *taaki* (so that)	clause-beginning words

Table 1: Unsupervised word clusters: examples

	monolingual BLEU	MT BLEU
baseline (no reordering)	35.9	15.47
word (only word features)	49.6	-
word+unsupervised clusters	52.8	18.58
word+POS	54.2	18.78
word+POS+case	54.5	18.93
word+POS+gnp	54.6	-
word+POS+case+gnp	55.4	19.14

Table 2: Reordering & MT performance using various features

tagger on many of the confusing cases. Thus, we may hope to achieve greater inter-annotator agreement, once the automatic POS tagger itself becomes more accurate.

4.3. Unsupervised Word Clusters

We use the distributional approach described in (Schütze, 1995) to generate unsupervised word clusters. We represent each word w by a vector $v = \{v_{-2}, v_{-1}, v_1, v_2\}$ where v_i represents the count of the 250 most frequent words in our corpus occuring at position i relative to w (e.g v_{-2} is a 250 dimensional vector, the jth element of which represents the number of times that the jth most frequent word in the corpus occured 2 positions to the left of w.) We then cluster these 1000 dimensional vectors using k-means with cosine distance. This clustering method assigns each word to the same cluster regardless of the context the word occurs in. We also experimented with the technique from (Schütze, 1995) to obtain context dependent word clusters, but since reordering performance was not enhanced by this we use the simpler context independent clustering technique. We obtained our clusters using a corpus of roughly 63 million words obtained by crawling several Hindi news sites. Experiments showed 100 clusters to work well. Manual inspection of the clusters showed several clusters that contain information relevant to reordering; e.g clusters of names, verbs, and a cluster containing punctuation and other words that might indicate a boundary dividing the Hindi sentence into parts that should be separate in English as well. See table 1 for examples.

5. Experiments

5.1. Reordering

We used around 6500 hand-aligned Hindi-English parallel sentences to generate the reference reorderings for the Hindi source sentences. These were then used to train the reordering model described in section 2.. Models were trained with various combinations of the features described in section 4.2.. Table 2 shows the monolingual BLEU results on a test set containing 280 reference sentences. Monolingual BLEU is computed in the same way as BLEU, but by comparing the outputs of the reordering model with the reference reorderings rather than the reference translations.

We also tried other features such as tense-aspect, morphological suffixes, and other combinations of these features, which did not work as well as the combinations reported in Table 2.

5.2. Impact on MT

We evaluated the effectiveness of these reordering models for MT using a phrase-based system (Visweswariah et al., 2010). The system was trained on 280K parallel sentences, and tested on around 2K sentences. The results are in Table 2 (column titled "MT BLEU"). For the systems using our reordering model, both the training and the test data were pre-ordered using the model before being fed to the SMT system.

6. Discussion

Accurate POS tagging requires large amounts of annotated data – for example, the Stanford English POS tagger, a state-of-the-art tagger, uses nearly a million words of training data (Toutanova, 2003). Our results show that even a tagger trained on a small amount of data, though not comparable in accuracy to state-of-the-art taggers, say, in English (which are around 97% accurate), can still be useful for MT. This is not to say that higher accuracy is unnecessary. In fact, our experiments for English-to-Hindi reordering show that the use of a high-quality POS tagger improves the monolingual BLEU from around 50 (with only word features) to 59. This is double the gain that we get with the base tags in our experiments (49.6 to 54.2), suggesting that improvements to the tagger should lead to further gains in reordering performance. Figure 2 shows the learning curve for our Hindi POS tagger (going from 600 words to 38K words of training data). Assuming that the same behaviour holds, more or less, as the training data is increased, this suggests that anywhere between half a million to one million words may be needed to get tagging accuracies comparable to English.

Our results also show that in the absence of any annotated data, unsupervised word clustering is a viable alternative

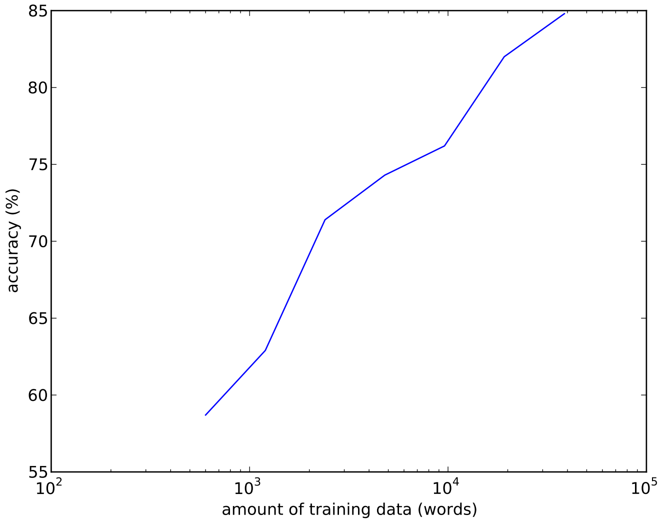

Figure 2: POS tagging: Learning curve

to POS tagging, and can be used to obtain reasonable improvements with no investment in annotation effort.

Finally, we see also that richer tags for case and GNP provide a further boost to reordering performance over usual POS tags. It would be interesting to compare the gains from the use of these tags to the gains from parsing. If these numbers are reasonably close, such tags may turn out to be a low-cost alternative to parsing in applications like MT.

Another interesting direction of future work would be to try unsupervised or semi-supervised clustering of words according to features such as case and agreement.

Our work also aims to highlight the fact that reordering, which is a very important sub-task in MT, can be used effectively for extrinsic evaluation of algorithms for unsupervised word clustering and POS tagging.

References

Al-Onaizan, Y., and Papineni, K. 2006. Distortion models for statistical machine translation. *Proceedings of ACL.*

Ananthakrishnan Ramanathan, Hansraj Choudhary, Avishek Ghosh, and Pushpak Bhattacharyya. 2009. Case markers and morphology: Addressing the crux of the fluency problem in English-Hindi SMT. *Proceedings of ACL-IJCNLP*.

Bharati A., Sangal R., Sharma D. M., and Bai L. 2006. AnnCorra: Annotating Corpora Guidelines for POS and Chunk Annotation for Indian Languages. Technical Report (TR-LTRC-31), LTRC, IIIT-Hyderabad.

Birch, A., Blunsom, P., and Osborne, M. 2009 A quantitative analysis of reordering phenomena. *Proceedings of the Fourth Workshop on Statistical Machine Translation*.

Brown, P., F., Della Pietra, S., A., Della Pietra, V., J., and Mercer, R., L. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2).

Collins, Michael, Philipp Koehn, and Ivona Kucerova. 2005. Clause restructuring for statistical machine translation. *Proceedings of ACL.*

Koehn, P., Axelrod, A., Birch, A., Callison-Burch, C., Osborne, M., and Talbot, D. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. *Proceedings of the International Workshop on Spoken Language Translation*.

Lafferty, J., McCallum, A., and Pereira, F. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proceedings of ICML.*

Mannem, P., and Dara, A. 2011. Partial parsing from bitext projections. *Proceedings of ACL-HLT*.

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. 1993. The Penn treebank: Building a large annotated corpus of English. *Computational Linguistics*, 19(2).

McDonald, R., Pereira, F., Ribarov, K., and Hajiĉ, J. 2005. Non-projective dependency parsing using spanning tree algorithms. *Proceedings of HLT*.

Schütze, Hinrich. 2005. Distributional part-of-speech tagging. *Proceedings of EACL.*

Tillmann, Christoph and Ney, Hermann. 2003. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Computational Linguistics*, 29(1).

Tillmann, Christoph. 2004. A unigram orientation model for statistical machine translation. *Proceedings of HLT-NAACL.*

Toutanova, K., Klein, D., Manning, C., and Singer, Y. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. *Proceedings of HLT-NAACL.*

Tromble, R. and Eisner, J. 2009. Learning linear ordering problems for better translation. *Proceedings of EMNLP*.

Visweswariah, K., Rajkumar, R., Gandhe, A., Ramanathan, A., and Navratil, J. 2011. A Word Reordering Model for Improved Machine Translation. *Proceedings of EMNLP*.

Visweswariah, K., Navratil, J., Sorensen, J., Chenthamarakshan, V., and Kambhatla, N. 2010. Syntax based reordering with automatically derived rules for improved statistical machine translation. *Proceedings of COLING*.

Yamada, K. and Knight, K. 2001. A syntax-based statistical translation model. *Proceedings of ACL.*

Zaslavskiy, M., Dymetman, M. and Cancedda, N. 2009. Phrase-based statistical machine translation as a travelling salesman problem. *Proceedings of ACL-IJCNLP*.

Zollmann, A. and Venugopal, A. 2006. Syntax-augmented machine translation via chart parsing. *Proceedings of the Workshop on Statistical Machine Translation*.